



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

CRISTIANE RIBEIRO DA CUNHA

PROPOSTA DE UTILIZAÇÃO DE ANÁLISE DE PONTOS DE FUNÇÃO EM PROJETOS DESENVOLVIDOS COM FDD

Palmas -TO

2012



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS
ULBRA

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

CRISTIANE RIBEIRO DA CUNHA

**PROPOSTA DE UTILIZAÇÃO DE ANÁLISE DE PONTOS DE FUNÇÃO EM
PROJETOS DESENVOLVIDOS COM FDD**

Trabalho de Conclusão de Curso (TCC) elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA), orientado pela Professora Mestre Cristina D'Ornellas Filipakis Souza.

Palmas – TO

2012



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

CRISTIANE RIBEIRO DA CUNHA

PROPOSTA DE UTILIZAÇÃO DE ANÁLISE DE PONTOS DE FUNÇÃO EM PROJETOS DESENVOLVIDOS COM FDD

Trabalho de Conclusão de Curso (TCC) elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA), orientado pela Professora Mestra Cristina D'Ornellas Filipakis Souza.

Aprovado em: Dezembro de 2012

BANCA EXAMINADORA

Prof. M.Sc. Cristina D'Ornellas Filipakis Souza
Centro Universitário Luterano de Palmas – CEULP

Prof. M.Sc. Jackson Gomes de Souza
Centro Universitário Luterano de Palmas – CEULP

Prof. M.Sc. Edeilson Milhomem da Silva
Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2012



Agradecimentos

Agradeço aos meus pais (Maria Lucilene e Tito Antônio), por terem me criado bem, apesar de todas as dificuldades, por terem se esforçado para me darem a educação e as oportunidades que não tiveram e acima de tudo pelo amor de pai e mãe, pelo carinho, pelos valores passados e por serem minha fonte de força para aguentar todo o resto: a distância, a saudades as dificuldades, a vida. Também agradeço ao meu irmão Otaíres, que sempre cuidou de mim mais do que dele mesmo, eu só posso dizer que serei eternamente grata, pelo que fizestes por mim, irmão.

Agradeço à Família Sistemas de Informação, por tudo que ela foi e ainda é na minha formação, tanto profissional quanto pessoal. Tenho muito a agradecer a todos os professores, tanto pelo conhecimento técnico passado nas disciplinas como pelo conhecimento passado nas conversas extra disciplinares. Primeiro gostaria de dizer obrigado a minha mãe Cristina por ser uma excelente orientadora, amiga, pelos puxões de orelha, quando necessário, e por ter me dito a seguinte frase no TCC I: “Não pensa que não vai dar tempo, continua fazendo e não chora, se não eu vou chorar também”, foi simples, mas me fez pensar diferente. O Jackson, obrigada por todo o incentivo dado e pelas sábias palavras: “Eu sei que você consegue fazer isso”, ele tem o dom de ver as qualidades que a gente desconhece.

Agradeço ao Fernando pela sinceridade dos seus comentários irônicos e por não ter me dado zero na avaliação continuada de BD I (só porque eu coleí errado), ao Fabiano pelos conselhos de vida e por sempre nos animar dizendo que um dia seremos ricos e ganharemos ótimos salários, a Mádia por ser a mãe e isso já é muito coisa, porque ser mãe, tem que ser paciente, ouvir os problemas de todo mundo e ajudar a gente a resolver, a Parcilene, por ser uma excelente profissional e ainda distribuir sorrisos no corredor (não é sempre que ela faz isso), e outra a Parcilene me inspira, e por fim, mas não menos importante, agradeço ao Edeilson (Edi), por ser esse professor chato, mas ser um cara muito legal!

Ah, os amigos, eles são aquelas pessoas que fazem a faculdade parecer algo mais divertido ou que fazem perceber que algo vivido valeu a pena (no caso dos da época da escola), mas amigo é amigo, amigo é irmão e os melhores são aqueles que te chama de chato (mesmo que você não seja) e amigos eu posso não ter muitos, mas tenho os melhores, agradeço a vocês por terem feito essa caminhada mais alegre, seja com seriados para diminuir



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

o tédio, apocalipse zumbi, idas ao cinema, as saídas para comer, as saídas pra beber, os TCCs da vida, os sábados a tarde nos Labins, as caminhadas no Cesamar, são tantas coisas, que daria pra escrever um livro, mas isso é coisa pro George Lucas. Thabata, George, Weltin, Monstro, Douglas Neves, Douglas Ramalho (agradecimentos especiais ao Doug, por nos divertir gratuitamente com doas piadas da própria vida, e por todas as aquelas quedas inexplicáveis), Felipe (Bunto), Charles, Andressa, Naara, Luane, Mayanne, Cleydinha, Diego (Doido), Van Neves, Jan Tarik, o loiro mais inteligente que conheço.

Obrigada aos amigos antigos que mesmo de longe sempre apoiaram muito, Viviane, Sarinha, Carla, Tamires, Rosa Maria, Dayane, Milton, Tia Rosinha, que sempre me incentivou muito. Prima Eliane (Lili) com quem eu sempre pude contar com o apoio.

E de novo agradeço à família, pois ela é a base de tudo, é o que nos define, é quem vai está sempre lá, de braços abertos esperando a gente voltar, é por quem eu faço tudo é quem eu quero sempre dar motivos para orgulhar!



RESUMO

Esse trabalho tem como objetivo o desenvolvimento de uma proposta de utilização conjunta da técnica de Análise de Pontos de Função (APF) com a metodologia de desenvolvimento de *software* Feature Driven Development (FDD). APF é uma técnica de dimensionamento funcional de *software* e FDD é uma metodologia ágil guiada à funcionalidades. Para realização da proposta é apresentado um referencial teórico mostrando as características, abordagens e conceitos dos temas envolvidos. É feita uma análise das abordagens para contagem de pontos de função e por fim é apresentada a proposta de adaptação da metodologia FDD.

PALAVRAS-CHAVE: Metodologia, estimativa, FDD, APF, Ágil.



LISTA DE ILUSTRAÇÕES

FIGURA 1 - PROCESSO DE ESTIMATIVA – AGUIAR (2002, P.6).	13
FIGURA 2 - PROCESSO DE ESTIMATIVAS (HAZAN, 2009)	14
FIGURA 3 - PROCEDIMENTO DE CONTAGEM BASEADO EM HAZAN (2001, APUD AMBR, ONLINE) E VAZQUEZ, SIMÕES E ALBERT (2010)	22
FIGURA 4 - RELAÇÃO ENTRE OS TIPOS DE CONTAGEM (VAZQUEZ, SIMÕES E ALBERT, 2010, P.55).	25
FIGURA 5- APF VISÃO GERAL (HAZAN, 2002).	27
FIGURA 6- DIFERENÇA ENTRE TD E CAMPO DE ARQUIVO. (VAZQUEZ, SIMÕES E ALBERT, 2010, P.75).	32
FIGURA 7 - EXEMPLO DE TIPO DE DADO.....	32
FIGURA 8–EXEMPLO DE CONTAGEM DE TIPO DE DADOS - FK.....	34
FIGURA 9 - EXEMPLO DE EE QUE ATUALIZA DOIS ALI - BASEADO EM LONGSTREET (2002, P.3).....	37
FIGURA 10 - EXEMPLO DE SAÍDA EXTERNA COM 2 ALIS – BASEADO EM LONGSTREET (2002, P.4)	38
FIGURA 11 - EXEMPLO DE CE COM UM ALI BASEADO EM LONGSTREET (ONLINE, P.14)	38
FIGURA 12 - CONTAGEM DE FUNÇÕES TRANSACIONAIS (VAZQUEZ, SIMÕES E ALBERT, 2010, P.107).....	39
FIGURA 13 - PERCENTUAL DE MUDANÇAS EM TDS (NESMA, ONLINE, 2009).	49
FIGURA 14 - VISÃO GERAL DO FDD – HEPTAGON (ONLINE, 2012).....	55
FIGURA 15 - FLUXO DOS PROCESSOS	57
FIGURA 16 – RELAÇÃO DO PROCESSO DPF COM O PROCESSO DMA	66
FIGURA 17 – DIAGRAMA DE COMPONENTES – FONTE: DOCUMENTAÇÃO DO SISTEMA DE ATENDIMENTO PSICOLÓGICO	74
FIGURA 18 – TELA DE CADASTRO DE CLIENTE – SISTEMA DE ATENDIMENTO PSICOLÓGICO	76
FIGURA 19 – DIAGRAMA DE SEQUÊNCIA DE PROJETO – ADICIONAR AGENDA.....	97
FIGURA 20 – PROPOSTA DE ADAPTAÇÃO DO FDD.	149
FIGURA 21 – DETALHAMENTO DO PROCESSO ESTIMAR MODELO ABRANGENTE.	150
FIGURA 22 – DETALHAMENTO DO PROCESSO ESTIMAR POR FUNCIONALIDADE	152



LISTA DE TABELAS

TABELA 1 – LÓGICA DE PROCESSAMENTO EM FUNÇÕES TRANSACIONAIS.	29
TABELA 2 – DEFINIÇÃO DA COMPLEXIDADE DAS FUNÇÕES DO TIPO DADOS.	36
TABELA 3 – CONTRIBUIÇÃO DE ALIS E AIES.	36
TABELA 4 - COMPLEXIDADE PARA EE.	41
TABELA 5 - COMPLEXIDADE PARA SE.	43
TABELA 6 - COMPLEXIDADE PARA CE.	44
TABELA 7 - CONTRIBUIÇÃO DAS FUNÇÕES TRANSACIONAIS.	44
TABELA 8- FATORES DE AJUSTE.	45
TABELA 9 - INFLUÊNCIA DOS PESOS.	45
TABELA 10 - FATOR DE IMPACTO DAS FUNÇÕES DE DADO.	49
TABELA 11 - FATOR DE IMPACTO DAS FUNÇÕES TRANSACIONAIS	50
TABELA 12 – RESULTADO DA CONTAGEM DAS FUNÇÕES DE DADOS.	82
TABELA 13 – RESULTADOS DAS CONTAGENS.	147



LISTA DE ABREVIATURAS E SIGLAS

- ABES – Associação Brasileira de Empresas de *software*
- AIE – Arquivo de Interface Externa
- ALI – Arquivo Lógico Interno
- APF – Análise de Pontos de Função
- AR – Arquivos Referenciados
- BFPUG – *Brazilian Function Points Users Group*
- CASE – Engenharia de *software* Auxiliada por Computador
- CE – Consulta Externa
- CLF – Construir a Lista de Funcionalidades
- CMMI – *Capability Maturity Model Integration*
- CPF – Construir por Funcionalidade
- CPM – Counting Practices Committee
- DMA – Desenvolver um Modelo Abrangente
- DPF – Detalhar por Funcionalidade
- EE – Entrada Externa
- EPF – Estimar por Funcionalidade
- EMA – Estimar Modelo Abrangente
- FDD – Feature Driven Development
- IFPUG. – *International Function Point Users Group*
- ISO - *International Organization for Standardization*
- KPAs – Áreas Chave de Processo
- LOC – Line Of Code
- NESMA - *Netherlands software Metrics Users Association*
- PCU – Pontos por caso de uso
- PF – Pontos de Função
- PPL – Planejar por Funcionalidade
- SE – Saída Externa
- TD – Tipo de Dado
- TI – Tecnologia de Informação



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

TIC – Tecnologias da Informação e Comunicação

TR- Tipo de Registro

VFA – Valor do Fator de Ajuste



SUMÁRIO

1. INTRODUÇÃO	9
2. REFERENCIAL TEÓRICO	12
2.1. Estimativas de <i>software</i> e métricas de <i>software</i>	12
2.1.1. <i>Line Of Code</i>	16
2.1.2. Pontos de Função – UFPUG e BFPUG	18
2.1.3. Pontos por caso de uso	19
2.1.4. Padrão ISO de medição funcional	20
2.2. Análise de Pontos de Função	21
2.2.1. Definir propósito.....	23
2.2.2. Reunir documentação disponível.....	23
2.2.3. Definir tipo da contagem.....	24
2.2.4. Determinar fronteira da aplicação	25
2.2.5. Definir escopo da contagem	26
2.2.6. Processo de contagem	26
2.2.7. Cálculo do fator de ajuste.....	44
2.2.8. Cálculo do tamanho funcional.....	46
2.2.9. Documentar e reportar	52
2.3. NESMA	52
2.3.1. Contagem Indicativa	53
2.3.2. Contagem Estimativa	53
2.3.3. Contagem Detalhada	54
2.4. Feature Driven Development – FDD	54
2.4.1. Papéis	56
2.4.2. Processos	57
2.4.3. Considerações do FDD	68
3. MATERIAIS E MÉTODOS	69
3.1. Local e Período	69
3.2. Materiais	69
3.3. Metodologia.....	69
3.3.1. Projeto	70
3.3.2. Desenvolvimento	70
4. RESULTADOS E DISCUSSÃO.....	72
4.1. Abordagem IFPUG	73
4.2. Contagem Indicativa	142
4.3. Contagem Estimativa	144
4.3.1. Considerações finais das análises do IFPUG e NESMA.	147
4.4. Proposta de Adaptação do FDD.....	148
4.4.1. Processo Estimar Modelo Abrangente.....	150



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

4.4.2. EPF - Processo Estimar por Funcionalidade	152
4.4.3. Artefatos obrigatórios do FDD	154
4.5. Riscos de aplicação da proposta	155
5. CONSIDERAÇÕES FINAIS.....	156
6. REFERÊNCIAS	158

1. INTRODUÇÃO

Estudos realizados e divulgados pelo *The Standishi Group* ao longo dos anos mostram que os projetos de *software* na maioria das vezes apresentam resultados insatisfatórios e os índices de sucesso são em sua maioria menores que os de fracassos totais e fracassos parciais. Em contrapartida os produtores de *software* têm se preocupado cada vez mais em reverter essa situação. Nesse sentido, ao longo dos anos, foram criadas instituições com o objetivo de regulamentar e nortear o processo de desenvolvimento de *software*, tais como modelos de referência, como PMBOK, e modelos de maturidade, como CMMI (*Capability Maturity Model Integration*).

No Brasil o mercado de desenvolvimento de *software* e serviços cresceu 24% em 2010 segundo pesquisas divulgadas pela Associação Brasileira de Empresas de *software* (ABES, 2011). Com o aumento da produção de *software*, a busca pela qualidade dos serviços e produtos entregues aos clientes também aumenta.

Segundo Bartié (2002) 30% dos projetos de *software* fracassam antes de serem concluídos, mais de 70% dos que são finalizados falham nas entregas das funcionalidades esperadas, os custos extrapolam em mais de 180% e os prazos alcançam a marca de 200% de excedência em relação aos cronogramas iniciais. Pesquisas realizadas pelo *The Standish Group* apontam que em 2004 15% dos projetos entregues foram cancelados antes do término, 51% falharam no tempo previsto, custos e funcionalidades esperadas e 34% foram concluídos com sucesso em relação as estimativas iniciais. Em 2006 os números mudaram respectivamente para 19%, 46% e 35%. Já em 2009 o índice de projetos cancelados aumentou em relação aos anos anteriores chegando a 24%. Os produtos entregues com falhas de funcionalidade, mudança de prazo e custo diminuiu em 2%. E a porcentagem de projetos concluídos com sucesso sofreu uma queda de 3% em relação ao ano de 2006.

Comparando os dados divulgados pelo *The Standishi Group* nas várias versões do relatório *Chaos* é possível verificar uma melhoria. Segundo o *The Standishi Group* (1995) apenas 16.2% dos projetos de *software* alcançavam sucesso em 1995, segundo Osiek (online, 2011) o último relatório, divulgado em 2011, informa que o índice de sucesso é de 37% entre dos processos pesquisados. Isso representa uma melhoria de 21% ao longo de 17 anos.

O relatório do *The Standishi Group* (1995) aponta como principais motivos do baixo índice de sucesso, a falta de utilização de técnicas de gerenciamento de projeto, a realização de estimativas não confiáveis e falta de acompanhamento e controle de processo de *software*.

A pesquisa mais recente do *The Standishi Group*, divulgada em 2011, informa que o índice de sucesso em projetos de *software* pode estar diretamente relacionado com a adoção de processos ágeis no desenvolvimento de *software* (OSIEK, online, 2011).

Existem técnicas de dimensionamento de *software* que podem ajudar no processo de estimativas, dentre essas estão: Análise de Pontos de Função, Linha de Código e PCU Pontos de Caso de Uso. O artefato principal para medição através Linhas de Código – *Line Of Code* (LOC) são os artefatos referentes à codificação, o que dificilmente pode ser obtido nas fases iniciais, dificultando, portanto, o processo de estimativas iniciais. Já os PCU dependem da forma de documentação dos requisitos e funcionalidades do sistema, já que recebem como principal artefato de entrada Casos de Uso. Caso os requisitos de um *software* não sejam documentados nessa notação haverá certa dificuldade para a medição e, portanto, para estimativas.

Análise de Pontos de Função é uma técnica de medição de *software* que pode auxiliar no processo de estimativas de *software* desde as fases iniciais do projeto quando ainda não se tem muitos artefatos do mesmo, como código das funcionalidades e ou telas do sistema. Por medir um *software* do ponto de vista funcional a técnica torna-se independente de tecnologia utilizada no desenvolvimento. Além disso, existe mais de uma abordagem para utilização da técnica de APF, como manual do IFPUG (*International Function Point Users Group*) e manual da NESMA (*Netherlands Software Metrics Users Association*), o que permite que ela seja utilizada em diferentes estágios do processo de desenvolvimento.

Das metodologias de desenvolvimento ágil existentes na atualidade, o *Feature Driven Development* se destaca para esse projeto por ser uma metodologia de desenvolvimento e gerenciamento guiado por funcionalidade, ou seja, têm o mesmo princípio básico da técnica de APF, as funcionalidades do sistema, do ponto de vista do usuário. *Feature Driven Development* – Desenvolvimento Orientado a Funcionalidade é uma metodologia de desenvolvimento ágil onde um projeto de *software* é planejado, desenvolvido e testado por funcionalidades.

O FDD define que o prazo máximo para construção de uma funcionalidade deve ser de duas semanas, no entanto, não possui um processo formal de estimativas de *software*, que possa definir de forma mais clara a quantidade de tempo necessário para a conclusão de um projeto.

A utilização de Análise de Pontos de Função para realizar estimativas de *softwares* desenvolvidos com a metodologia de desenvolvimento e gerenciamento de projetos FDD pode vir a facilitar o processo de planejamento de *software*. APF oferece subsídios para

dimensionamento de um *software* na fase inicial, e esse dimensionamento pode ser utilizado por processos do FDD para formação de estimativas do projeto como um todo e também de estimativas mais precisas para cada funcionalidade.

Partindo da premissa que é possível utilizar Análise Pontos de Função para realizar estimativas de *softwares* desenvolvidos com *Feature Driven Development* esse trabalho apresenta uma proposta de utilização conjunta das duas técnicas. Essa junção tem por objetivo auxiliar no processo de desenvolvimento de *software*. Para chegar a tal proposta foram realizados estudos relacionados os temas envolvidos, tais como: estimativas de *software*, Análise de Pontos de Função - APF e *Feature Driven Development* - FDD. No decorrer do trabalho será apresentado um referencial teórico relacionado os temas envolvidos, a metodologia utilizada e os resultados obtidos.

2. REFERENCIAL TEÓRICO

Nessa seção serão apresentados os principais conceitos relacionados ao trabalho, tais como: Análise de Pontos de Função segundo abordagem do IFPUG e Análise de Pontos de Função segundo abordagem da NESMA; *Feature Driven Development*, metodologia ágil de desenvolvimento e gerenciamento de *software* a ser utilizada no projeto.

2.1. Estimativas de *software* e métricas de *software*

Estimativa é avaliação ou cálculo aproximado de algo, é uma parecer sobre algo, que pode ser baseado em evidências existentes (Dicionário, online). Segundo Aguiar (online, 2002) estimativas de *software* são projeções quantitativas dos projetos como: tamanho, esforço, prazo e custo. Realizar estimativas de *software* é um processo que compreende o cálculo de esforço, custo e prazo no desenvolvimento ou manutenção de sistemas de informação.

Existem, segundo Aguiar (online, 2002), várias abordagens para estimativas de projetos de *software* e algumas delas são: os modelos paramétricos, modelos baseados em atividades, analogia e relação simples de estimativa. Em modelos baseados em atividade, as estimativas são obtidas para as atividades de nível mais baixo com base em opinião de especialistas e dados históricos. As estimativas baseadas em analogia são utilizadas com mais frequência em projetos que não são comuns, pois não haverá histórico para esse tipo de projeto. No modelo por analogia, um ou mais projetos que sejam semelhantes ao que será desenvolvido são usados como base para a obtenção da estimativa.

Os modelos paramétricos são baseados na relação matemática entre o esforço, o prazo, o tamanho e a qualidade. Para Andrade (2003, p.9) a entrada mais importante para esse modelo é tamanho, esse tamanho representa a quantidade de funções do *software*, que pode ser dimensionado em pontos de função. Esse modelo deve ser calibrado por fatores mensuráveis, como fatores do ambiente de desenvolvimento e desempenho da equipe de desenvolvimento do *software*. A Figura 1 representa o processo de estimativa de acordo com Aguiar.

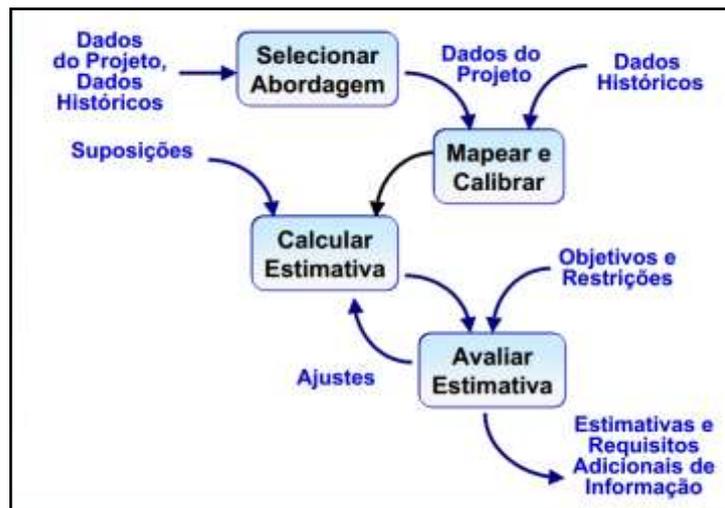


Figura 1 - Processo de Estimativa – Aguiar (2002, p.6).

Como pode ser observado na Figura 1 um processo de estimativa não envolve a simples estimativa de um projeto, mas também envolve a avaliação dessa estimativa. De acordo com a Figura 1 os dados históricos da organização são utilizados para nortear a abordagem de estimativas (modelo a ser utilizado), ou seja, a existência ou não de dados históricos pode determinar ou descartar um tipo de estimativa. Por exemplo: um projeto que não tenha nenhuma base histórica de projetos semelhantes irá nortear para a utilização da estimativa baseada em analogia. Após a seleção da abordagem os dados do projeto e os dados históricos (caso eles existam) são utilizados para o processo de calibração. A fase seguinte é onde as estimativas são finalmente calculadas. O processo deve continuar com a avaliação das estimativas obtidas, os objetivos e as restrições devem ser levados em conta no processo de avaliação das estimativas, após cada avaliação a estimativa passa por reparos necessários e toda estimativa deve ser avaliada após a sua conclusão. A Figura 2 apresenta o processo de estimativa proposto por Hazan (2009).

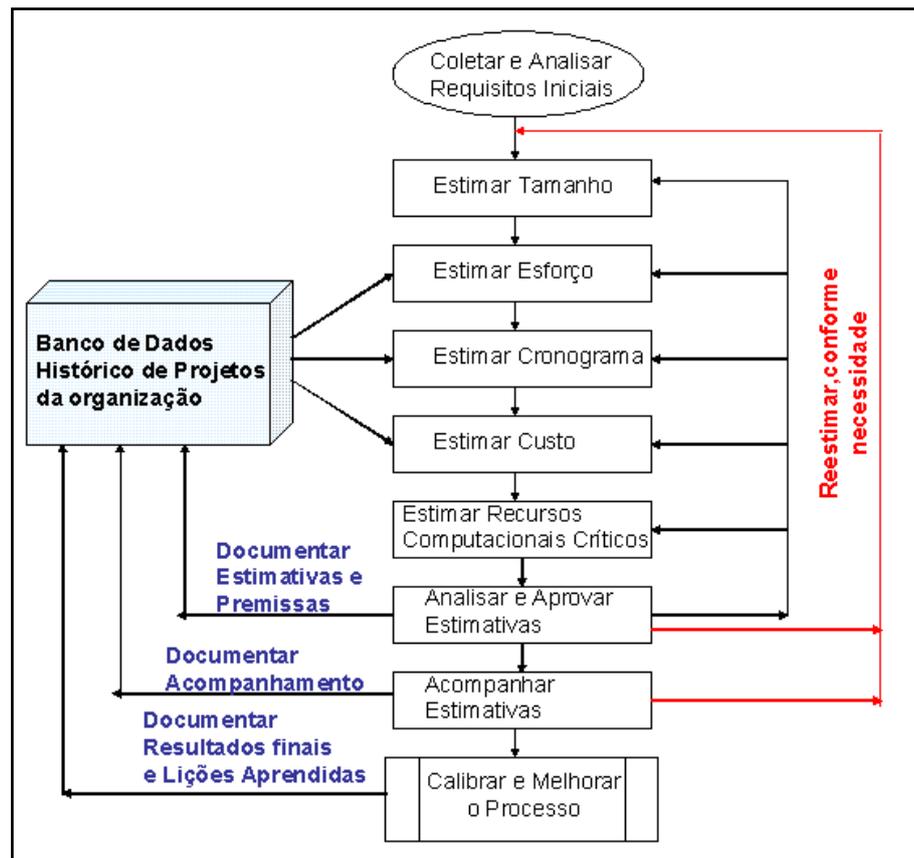


Figura 2 - Processo de Estimativas (Hazan, 2009)

De acordo com o processo de estimativa proposto por Hazan, o maior artefato de entrada para o processo de estimativas de *software* são os requisitos do sistema e esses devem ser coletados, analisados e servem de base para a estimativa de tamanho. Os requisitos que compõem o documento de entrada podem ser: o documento de visão, uma lista de requisitos ou outros artefatos de *software* que estejam disponíveis.

As estimativas de esforço são obtidas a partir dos dados presentes na base histórica de organização e da estimativa de tamanho. As demais estimativas como: custo, prazo e recursos computacionais são obtidas através da estimativa de tamanho e dos dados históricos. Assim como no modelo apresentado por Aguiar (2002) o modelo proposto por Hazan (2009) também prevê avaliação e aprovação das estimativas e a partir dessa avaliação o processo de estimativa pode ser refeito.

O modelo apresentado na Figura 2 indica que as estimativas devem ser monitoradas durante o processo e que esse monitoramento deve ser registrado na base histórica da organização. A calibração e melhoria do processo de estimativas é o último no modelo proposto por Hazan e as lições aprendidas devem ser registradas na base histórica.

Apesar de apresentar algumas diferenças no processo os modelos apresentados por Aguiar (2002) e Hazan (2009) representam de forma satisfatória o processo de estimativas de forma geral, ou seja, mostram que as estimativas são feitas a partir de dados do projeto a ser estimado e de dados históricos e que essas estimativas devem ser acompanhadas e avaliadas durante o desenvolvimento e o *software* deve ser reestimado, caso necessário.

Realizar boas estimativas de *software* é sem sombra de dúvida uma das mais importantes fases da engenharia de *software*, pois as estimativas iniciais são parte da base para planejamento do projeto. A partir da estimativa de tamanho obtêm-se as estimativas de esforço, prazo e custo. Caso o tamanho seja subestimado as outras estimativas também serão subestimadas e conseqüentemente haverá extrapolação de prazo e/ou custo, acarretando prejuízo. Caso haja o tamanho seja superestimado, é provável que as outras estimativas também sejam, acarretando a alocação desnecessária de recursos e pessoal.

Estimativas de *software* podem ser feitas de diversas formas. Podem ser baseadas em métricas históricas, obtidas através de experiências anteriores da equipe, empíricas, obtida através de métricas de dados estatísticos de equipes diferentes ou na combinação das duas. As métricas de *software* fornecem subsídios para os engenheiros de *software* medirem e preverem os processos de um *software*, recursos necessários a um projeto e artefatos que serão relevantes ao desenvolvimento (PETERS e PEDRYCZ, 2001, p. 430).

Segundo Basili e Basili (1989 *apud* Peters e Pedrycz, 2001, p. 430) “uma medição de *software* é uma técnica ou um método que aplica medidas de *software* em uma determinada classe de objetos de forma a atingirem um objetivo predefinido”. Dessa forma, quando se fala em um processo de medição é de suma importância ter as respostas para essas três perguntas: O que medir? Como medir? Para que medir? O que medir ou o objeto a ser medido, pode ser um projeto de *software*, requisitos de *software*, conjunto de testes, processos de manutenção ou um *software* como um todo.

Após a definição do objeto a ser medido deve-se definir como a medição será feita, que implica em definir uma técnica, métrica ou padrão de medida a ser utilizado na medição. Ainda é necessário definir um objetivo, pois a medição de *software* por si só, sem nenhum objetivo final não terá nenhuma importância. Segundo Peters e Pedrycz (2001, p.431) dentre os objetivos pretendidos ao medir um *software* estão: caracterização, avaliação de rendimento, análise e previsão.

Para ajudar no processo de estimativa existem métricas formais para medição do tamanho de um *software* que podem ser utilizadas ainda no início do desenvolvimento. As medidas e estimativas obtidas no início de um projeto devem ser monitoradas ao longo do seu

desenvolvimento. De acordo com Peters e Pedrycz (2001, p. 432 e 480) As medidas referem-se ao tamanho do *software*, seu volume ou tamanho funcional, já as estimativas são calculadas a partir desse tamanho e podem ser de prazo, custo e esforço.

Segundo Peters e Pedrycz (2001, p. 432 e 480) ao se subestimar ou superestimar o tamanho de um *software* o gerente tende a, conseqüentemente, cometer erros nas estimativas de, e custos, erro em estimativas de custo (subestimar ou superestimar) pode custar muito no que tange a recursos alocados ao projeto. Como as estimativas são feitas no início do processo de desenvolvimento, caso elas sejam feitas de maneira errônea as fases subsequentes do projeto irão carregar o mesmo erro.

Se houver erros nas estimativas e esses erros não forem detectados ao longo do processo de desenvolvimento, somente poderão ser evidenciados futuramente, seja com a falta de recursos disponíveis, com custos acima do previsto ou com prazos que excedem os cronogramas iniciais.

Dessa forma foram desenvolvidas técnicas para auxiliarem na medição de *software*, o que ajudam na realização de estimativas. Dentre as mais conhecidas estão:

- *Line of Code* – Linhas de Código;
- Pontos por Caso de Uso;
- Pontos de Função.

Alguns tipos de estimativa de *software* necessitam de medidas de tamanho do *software*, ou projeto e a ser estimado. Dentre as várias medidas de tamanho existentes existem dois grupos que se destacam: as medidas de tamanho e métricas funcionais.

As medidas de tamanho fazem parte do conjunto das técnicas mais comumente utilizadas para medição de *software*, por ser de fácil compreensão e utilização. São conceitualmente mais diretas e se concentram em medir o volume de código (PETERS e PEDRYCZ, 2001).Linha de Código ou LOC é um tipo de medida de tamanho não funcional e será apresentada a seguir.

2.1.1. *Line Of Code*

As medidas de linha de código (LOC) são, segundo BRAGA (1996, p.2) uma das primeiras tentativas de se medir *software*, esse método leva em conta a quantidade de linhas de código de um *software*. O volume de código representa apenas a quantidade de linhas de código em si, mas sabe-se que a quantidade de linhas de código não representa necessariamente complexidade ou facilidade, assim sendo LOC não pode ser utilizada como

indicativo único de volume, tamanho ou complexidade de *software*. A Listagem 1 apresenta um exemplo de diferença na quantidade de linha de código em decorrência da organização de código.

```

Código 1
1 public int somar(int a, int b){return a + b;}

Código 2
1 public int somar(int a, int b)
2 {
3     int result = a + b;
4     return result;
5 }
  
```

Listagem 1 – Diferença de quantidade de LOC

Como demonstrado na Listagem 1 um mesmo código em uma mesma linguagem organizado de formas diferentes pode apresentar uma quantidade de linhas de código diferente, ainda que isso não representa complexidade de *software*.

Segundo Hazan(2009, p. 26) dentre os problemas encontrados na utilização dessa técnica estão:

- O significado de uma linha de código é subjetivo;
- A métrica não é adequada para ser um indicador de produtividade, por exemplo, o desenvolvedor que escrever mais linhas de código será mais produtivo do que o desenvolvedor que escrever um algoritmo com um número menor de linhas e mais eficiente.
- Por ser a contagem de linhas de código, não pode ser realizada no início do projeto, não servindo, portanto, para fins de estimativas de prazos e custos do mesmo, ainda que possa servir de base para estimativas de projetos futuros.
- Medida não pode ser padronizada, em razão da variedade de linguagens existentes.

Um das vantagens da utilização de LOC é que esse tipo de contagem não é feita manualmente, existem *software* que realizam a contagem de linhas de código dentre eles o Cod Counter Pro, *software* gratuito disponível para download em: <http://code-counter-pro.en.softonic.com/>.

2.1.2. Pontos de Função – UFPUG e BFPUG

Segundo Vazquez, Simões e Albert (2010, p.35), análise de Pontos de Função teve seu início em 1970 enquanto Allan J. Albrecht fazia análises em sistemas da IBM com o objetivo de encontrar algum fator crítico que determinasse um tamanho de um sistema de informação. Após analisar centenas de programas, Albrecht apresentou os fundamentos da técnica de Análise de Pontos de Função (Function Points Analysis), que tem o objetivo de medir uma aplicação através das funcionalidades oferecidas ao usuário final, ou seja, essa análise é feita com base na visão do usuário.

Segundo Vazquez, Simões e Albert (2010, p.33) depois que a técnica de APF foi apresentada ao público o seu número de usuários cresceu de forma bastante acelerada, ainda hoje APF é a técnica mais utilizada para dimensionamento de *software*. Em 1986 foi fundando o *International Function Point Users Group* – IFPUG. A partir daí o IFPUG se tornou a entidade que lançou e é responsável pelo *Counting Practices Manual* - CPM, que objetiva a padronização da técnica (VAZQUEZ, SIMÕES e ALBERT 2010 p.35). A partir do manual lançado inicialmente pelo IFPUG e do crescente número desta técnica em todo o mundo, foram lançadas várias outras versões desse manual, sendo que a primeira considerada madura foi a versão 3.0, lançada em 1990. Atualmente a versão mais recente do CPM é a release 4.3.1.

O IFPUG é uma entidade sem fins lucrativos, formada por pessoas e empresas que têm como objetivo melhorar o processo de desenvolvimento e manutenção de *software* através da técnica de APF. Entre as ações promovidas pelo IFPUG está:

- Conferência anual;
- Seminários e Workshops educacionais;
- Certificação profissional;
- Comitês e grupos de trabalho.

O BFPUG (*Brazilian Function Points Users Group*) é uma representação do UFPUG no Brasil. Foi desenvolvido a partir do grupo de usuários de APF do Rio de Janeiro – FPUG-Rio. O BFPUG foi criado em 1998 e hoje possui centenas de associados, entre estudantes, desenvolvedores, consultores, gerentes de sistemas. O BFPUG, entre outras atividades é responsável pelo programa de certificação em APF no Brasil (VAZQUEZ, SIMÕES e ALBERT, 2010, p. 37).

2.1.3. Pontos por caso de uso

Segundo Hazan (2009, p. 26), a técnica de medição Pontos por Caso de Uso - PCU tem como propósito de estimar recursos para projetos de *software* orientados a objeto, que foram modelados exclusivamente por meio da especificação de Casos de Uso. Essa técnica foi desenvolvida em 1993 por Gustav Karner e tem como base a técnica de Pontos de Função, Pontos por Caso de Uso seria uma adaptação específica de Pontos de Função – PF (AGUIAR 2003, p.1).

O processo de contagem de PCU está dividido em seis fases distintas. A primeira delas consiste em determinar identificar os atores do sistema e determinar sua complexidade. Um ator pode ter a complexidade baixa, média e alta essas complexidades possuem respectivamente os pesos um, dois e três.

A segunda fase é contar os casos de uso e determinar a complexidade dos mesmos, a complexidade dos casos de uso é determinada com base no número de transações que ele possui. Segundo Vieira (2007, p.24) “Uma transação é um evento que ocorre entre um ator e o sistema, o qual é executado completamente ou não”. Um caso de uso com até cinco transações é de complexidade simples e tem peso 1. Um caso de uso que possui de 6 a 10 transações é de complexidade média e tem peso 2 e um caso de uso com mais de 10 transações tem complexidade alta e peso 3.

As fases seguintes consistem: na contagem dos Casos de Uso Não Ajustados, essa contagem é feita através de equações matemáticas; Na contagem dos fatores complexidade técnica, que compreendem 13 fatores que são inerentes a características técnicas; determinação e contagem dos fatores ambientais que compreendem 8 fatores inerentes à características do ambiente; e a contagem dos PCUs ajustados que é a unidade final de medida da métrica de Pontos por Caso de Uso.

Segundo Hazan (2005, p.11), após o resultado dessa medição de pontos de caso de uso e tendo informações sobre a produtividade média para produzir um PCU, pode-se estimar o esforço total para o projeto. Sendo assim, essa seria uma técnica apta para a realização de estimativas, mas o insumo necessário para a medição do *software* acaba fazendo dessa uma técnica limitada em relação à Análise de Pontos de Função. Essa limitação vem da necessidade da expressão dos requisitos em forma de Casos de Uso e ainda da falta de padronização dos mesmos. Na prática PCU não pode ser utilizada para medir aplicações prontas que não possuem documentação especificada com Casos de Uso. A solução para isso

seria criar os Casos de Uso a partir da aplicação pronta e então realizar a medição, o que demandaria mais tempo e recursos, tornando isso inviável.

Segundo Aguiar (2003, p.2) em 2001 um relato da Universidade de Oslo revelou a utilização de PCUs em processos de estimativas de esforço para alguns projetos, mas esse mesmo relato diz que a variedade na descrição dos casos de uso pode influenciar no resultado da medição. Casos de usos descritos de maneiras diferentes podem ter valores de PCUs diferentes. Sendo assim, a utilização de Pontos por Casos de Uso para a obtenção de estimativas confiáveis exigiria uma padronização dos estilos de caso de uso, gerando assim um imenso trabalho de calibração do modelo de estimativas de baseado em Pontos de Caso de Uso.

As métricas funcionais dimensionam um *software* ou projeto de *software* do ponto de vista funcional. Seu objetivo é medir o que o *software* oferece para o usuário, o que o *software* é capaz de fazer e não como isso será feito. Por esse motivo as métricas funcionais podem ser mais facilmente utilizadas para estimativas de projetos de *software*. Podem ser obtidas estimativas de um *software* que ainda será desenvolvido. Segundo Vazquez, Simões e Albert (2010 p.37) dentre as métricas funcionais destaca-se a Análise de Pontos de Função, sendo a técnica mais difundida atualmente e que possui maior número de usuário e de material disponível para consulta.

2.1.4. Padrão ISO de medição funcional

Segundo Vazquez, Simões e Albert (2010, p.38-39) “medição funcional” é um termo geral para métodos de dimensionamento de *software* com base nas funções requeridas pelo usuário. Depois do lançamento da técnica de Análise de Pontos de Função proposta por Albrecht foram criadas várias outras técnicas derivadas dela.

Com a criação de várias técnicas de medida do tamanho funcional, também foram criadas inconsistências entre os resultados obtidos, com a utilização dos vários métodos e técnicas para medição do tamanho funcional. Assim, os grupos de usuários de pontos de função de vários países, formaram o WG12 (*Working Group 12*), um grupo de trabalho subordinado ao SC7 (*Sub-Committee Seven*) do JTC1 (*Joint Technical Committee One*), que é estabelecido pela ISO (*International Organization for Standardization*) em conjunto com o IEC (*International Engineering Consortium*). O resultado do trabalho do WG12 foi a criação da norma ISO/IEC 14143, que estabelece padrões internacionais para a medição do tamanho funcional de um *software*.

Segundo Vazquez, Simões e Albert (2010, p.39) a norma ISO/IEC 14143 foi desenvolvida com o objetivo de garantir que todos os métodos e todas as medidas de tamanho funcional sejam baseadas em conceitos similares e, portanto, possam chegar a resultados similares. Ao final de 2002, a versão 4.1 do CMP foi aprovada como técnica de medição funcional reconhecida sob a denominação ISO/IEC 2092:2002. Essa norma contempla a Análise de Pontos de Função até a determinação dos pontos de função não ajustados, uma vez que considerou as características gerais do CMP utilizadas para a determinação do fator de ajuste compreende requisitos tecnológicos e de qualidade o que faz com não possam ser aderidas pelo padrão de medição funcional da ISO.

Sendo a técnica de Análise de Pontos de Função é um dos principais conceitos envolvidos nesse projeto a seção 2.2 irá abordar de forma detalhada essa técnica, desde a definição do seu ciclo de vida até fórmulas para definição do tamanho funcional de um *software*, seja ela projeto de desenvolvimento, projeto de melhoria ou aplicação.

2.2. Análise de Pontos de Função

Segundo Vazquez, Simões e Albert (2010, p.33) Análise de Pontos de Função é uma técnica de medição funcional que consiste na contagem das funcionalidades que são solicitadas e entregues ao usuário. A técnica de Análise de Pontos de Função mede o que é sistema é e não como ele será, ou como ele foi desenvolvido. Segundo BRAGA (1996, p.10) “o principal objetivo da técnica de Análise de Pontos de Função é medir a funcionalidade de um *software* ou aplicativo, baseando-se primeiramente no desenho lógico e, de acordo com a perspectiva do usuário”. Vazquez, Simões e Albert (2010, p.50) corroboram com essa definição, acrescentando aos objetivos primários da técnica “medir as funcionalidades que o usuário da aplicação medida solicita e recebe e medir o desenvolvimento e manutenção de *software* de forma independente da tecnologia utilizada para a sua implementação”.

Abaixo estão os benefícios citados por VAZQUEZ, SIMÕES e ALBERT (2010, p.51-52) oferecidos pela utilização da técnica de Análise de Pontos de Função:

- É uma ferramenta para determinar o tamanho de um pacote adquirido pelo contagem de todas as funções incluídas.
- Apoia o gerenciamento de escopo de projetos. Ao realizar estimativas e medições dos pontos de função do projeto em cada fase de seu ciclo de vida, é possível determinar se os requisitos funcionais cresceram ou diminuíram, e se a essa variação

corresponde a novos requisitos ou a requisitos já existentes e que foram apenas mais detalhados.

- Completam o gerenciamento de requisitos ao auxiliar na verificação da solidez e completeza dos requisitos especificados. O processo de contagem de pontos de função favorece uma análise sistemática e estruturada da especificação dos requisitos e traz benefícios semelhantes a uma revisão em pares de processo.
- Um meio para estimar custo e recursos para o desenvolvimento e manutenção de *software*. Por meio da realização de uma contagem ou estimativa de pontos de função no início do ciclo de vida de um projeto, é possível determinar seu tamanho funcional, e essa medida pode ser usada para derivar outras, como custo, prazo e esforço.
- Um fator de normalização para comparação de *software* ou para a comparação da produtividade na utilização de diferentes técnicas, para saber qual delas melhor se adequa a realidade da organização.

O processo de medição de Pontos de Função está dividido em fases e cada uma delas tem a sua importância no processo de contagem. O procedimento de contagem de pontos de função representado por fases é mostrado na Figura 3.

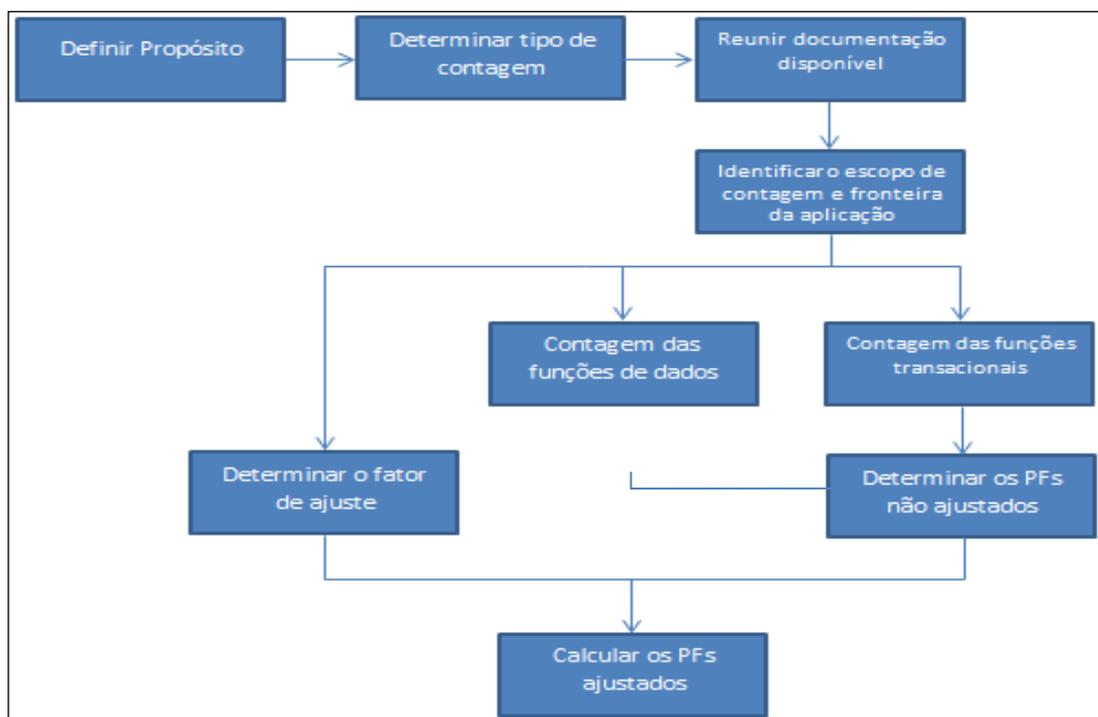


Figura 3 - Procedimento de contagem baseado em Hazan (2001, apud AMBR, online) e Vazquez, Simões e Albert (2010)

Como representado na Figura 3 o processo de contagem de PF (Pontos de Função) é composto por basicamente nove fases. O processo é iniciado com a definição do propósito da contagem, isso irá nortear as fases seguintes, em seguida o tipo da contagem é definido, após definidos o propósito e o tipo deve-se reunir a documentação disponível do *software* ou projeto de *software*. São identificados depois o escopo e a fronteira, os dois devem está de acordo um com o outro. Depois disso, são contadas as funções do tipo dado e as funções do tipo transação, com base nesse calculo é obtido o tamanho funcional. A atividade de calcular o fator de ajuste por ser paralela ao cálculo do tamanho funcional e também pode ser opcional. Após calcular o fator de ajuste é calculado os pontos de função ajustados, com base no fator de ajuste e no tamanho funcional. As seções seguintes irão apresentar cada uma dessas fases de forma mais detalhada.

2.2.1. Definir propósito

A contagem de Pontos de Função por si só não é um fim, não se mede *software* apenas por medir. Para tanto é necessário definir um propósito para a contagem, o motivo que irá levar a contagem de Pontos de Função – PF. O propósito da contagem de pontos de função é fornecer uma resposta a um problema de negócio. Sendo assim, o propósito está atrelado a um problema a ser resolvido. Segundo Vazquez, Simões e Albert (2010, p.53) de forma geral o propósito:

- Determina algumas premissas para o processo de contagem;
- Determina o tipo de contagem.
- Estabelece o escopo da contagem.
- Afeta o posicionamento da fronteira da aplicação.
- Define o nível de detalhe da contagem.

2.2.2. Reunir documentação disponível

Essa fase consiste em reunir toda a documentação disponível para a medição do *software*, a qual pode ser filtrada após a definição do propósito da medição. A filtragem de documentação consiste na seleção dos documentos necessários para a contagem e o descarte dos documentos desnecessários. Os documentos podem ser casos de uso, lista de usuários,

descrição de funcionalidades, diagramas de casos de uso, modelo entidade relacionamento de banco de dados e assim por diante.

Em caso de falta de documentação disponível deve-se buscar outra forma para medir as funcionalidades do *software*, como entrevista com gerentes de projeto e clientes, o que implica necessariamente no aumento do trabalho da medição e conseqüentemente em seu custo.

2.2.3. Definir tipo da contagem

Após a definição do objetivo, pode-se definir, com base no mesmo, o tipo de medição a ser usado, que pode ser:

- Medição de um projeto de desenvolvimento
- Medição de uma aplicação.
- Medição de projeto de melhoria ou manutenção.

O número de pontos de função de um projeto de desenvolvimento representa as funcionalidades que serão entregues ao usuário final do *software* e utilizadas por ele. Segundo Vazquez, Simões e Albert (2010, p.54), essa contagem também deverá incluir eventuais conversões de dados que sejam necessárias. Por exemplo, a empresa vai implantar um novo *software* que irá substituir um antigo, porém os dados do sistema antigo devem estar presentes no sistema novo, sendo assim, a importação dos dados deve ser incluída no escopo da contagem. Em projetos de desenvolvimento as contagens que são realizadas antes do término do projeto são chamadas de contagem estimativa e usam os documentos iniciais como o documento de visão e lista de requisitos (SANTOS, 2011, p.10).

O número de pontos de função de um projeto de melhoria, ou manutenção evolutiva está associado a mudanças nos requisitos funcionais, ou seja, a inclusão, alteração ou exclusão de funcionalidades (SANTOS, 2011, p.19). Após a conclusão da contagem de pontos de função de um projeto de melhoria o número de pontos de função deve ser atualizado, para que assim possa representar a nova realidade da aplicação.

Os pontos de função de uma aplicação pronta ou *baseline* mede um projeto já instalado e devem incluir todas as funcionalidades do mesmo. Segundo Vazquez, Simões e Albert (2010, p.54) esse tipo de medição também é conhecido como medição de pontos de função instalados. Os pontos de função contados representam o tamanho funcional do *software* e não mais uma estimativa do tamanho. A Figura 4 apresenta relação entre os processos de contagem do IFPUG.

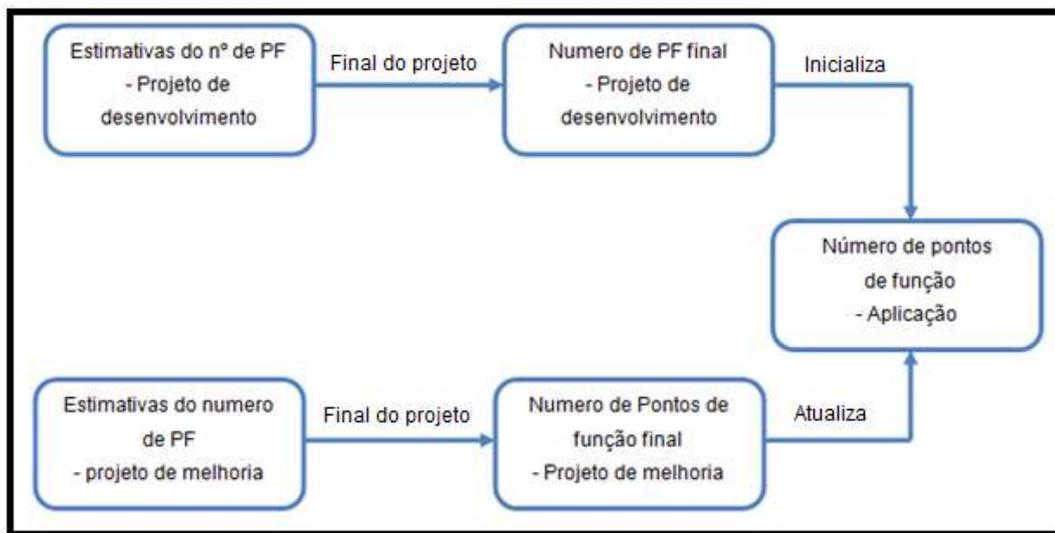


Figura 4 - Relação entre os tipos de contagem (VAZQUEZ, SIMÕES e ALBERT, 2010, p.55).

A Figura 4 mostra a relação entre os tipos de contagem, pode-se perceber através dela que quando se termina a contagem dos pontos de função de um projeto de desenvolvimento é inicializada a contagem dos pontos de função da aplicação. Quando conclui a contagem dos pontos finais de função de um projeto de melhoria deve-se atualizar o número de pontos de função da aplicação.

2.2.4. Determinar fronteira da aplicação

Segundo Vazquez, Simões e Albert (2010, p.56) a fronteira da aplicação é a interface conceitual que separa o *software* que será medido do mundo exterior. Da mesma forma que se precisa saber onde termina um lote em um terreno para que ele possa ser medido é necessário determinar onde termina uma aplicação para que a mesma possa ser medida.

Essa etapa é de fundamental importância para o processo de medição, pois ela serve como premissa para as fases seguintes do processo de contagem. Caso o *software* seja delimitado de forma errônea existe uma grande possibilidade de que os processos posteriores sejam invalidados, inclusive a contagem como um todo.

Segundo Vazquez, Simões e Albert (2010, p.56-57) quando existem mais de uma aplicação no escopo de contagem as regras a seguir definidas pelo IFPUG, devem ser obedecidas para a determinação da fronteira:

- Sua determinação deve ser feita com base no ponto de vista do usuário. O foco deve estar no que ele pode entender e descrever.

- As fronteiras entre as aplicações devem ser estabelecidas com base na separação das funções conforme estabelecido pelo processo de negócio, não em considerações tecnológicas.
- Em projetos de melhoria a fronteira estabelecida no início do projeto deve estar de acordo com a fronteira já estabelecida para aplicação sendo modificada.

Como já foi dito anteriormente a determinação da fronteira da aplicação é um passo muito importante para o processo de contagem de pontos de função e é importante que ela seja feita de acordo com a perspectiva de negócio e não da perspectiva técnica. Segundo Vazquez, Simões e Albert(2010, p.57) quando uma fronteira é definida de acordo com a perspectiva técnica e não de negócio alguns problemas podem surgir como:

- Contagem duplicada da mesma transação por várias “aplicações”, uma vez que cada uma delas contribui com um subprocessamento para a transação do negócio.
- Contagem de funções de transferências de dados entre plataformas e aplicações.

Após a definição da fronteira da aplicação, é necessário definir o escopo da contagem, que será abordado na seção seguinte.

2.2.5. Definir escopo da contagem

Esta fase é inerente à definição do que será medido, ou seja, quais funções serão medidas, se o sistema inteiro será medido, ou apenas uma parte dele. O escopo pode incluir todas as funções do sistema, ou somente aquelas selecionadas, como as que são efetivamente utilizadas pelos usuários, ou aquelas que realizam transações no sistema.

Em caso de um projeto de melhoria, o escopo geralmente abrange somente as novas funções incluídas, ou as que foram modificadas, ou excluídas do sistema. Já em projetos de desenvolvimento o escopo pode envolver todo o sistema, ou simplesmente uma fase, uma parte que será entregue ao cliente. Por exemplo, em sistemas desenvolvidos de forma iterativa o escopo de uma contagem pode envolver apenas as funções que estão contidas em um módulo do sistema.

2.2.6. Processo de contagem

Após definidas as fases anteriores do procedimento geral de contagem, deve-se iniciá-lo. A contagem é dividida em cinco partes que são: funções do tipo dado, função do tipo transação, cálculo do tamanho funcional, que é feito baseado no peso em pontos de função de

cada função do tipo dado e cada função do tipo transação, cálculo do fator de ajuste (opcional) e a documentação da contagem. A Figura 5 apresenta uma visão geral da APF.



Figura 5- APF Visão Geral (Hazan, 2002).

A Figura 5 apresenta os componentes básicos de uma contagem de Pontos de função e como eles interagem com a aplicação medida. São divididos em Funções do Tipo Dado e Funções do Tipo Transação. As funções de dados são divididas entre Arquivos Lógicos Internos que ficam dentro da fronteira da aplicação, e Arquivos de Interface Externa que se situam fora da fronteira da aplicação e estão dentro da fronteira de outra aplicação. As funções de transação são compostas por três tipos de função: Entrada Externa, onde dados entram através da fronteira da aplicação mantendo um ou mais ALIs; Saída Externa, onde dados são enviados para fora da fronteira da aplicação, os dados de uma SE são compostos por dados derivados; Consulta Externa, onde dados atravessam de dentro para fora a fronteira da aplicação, os dados não podem ser dados derivados.

2.2.6.1. Definição de termos utilizados

Para continuar a entender o processo de contagem é necessário apresentar o significado de alguns termos utilizados na contagem de pontos de função.

Informações de controle: São dados que influenciam um processo elementar da aplicação sendo contada. Eles devem especificar o que, quando e como os dados devem ser processados. Em suma, informações de controle são parâmetros (VAZQUEZ, SIMÕES e

ALBERT 2010, p.74). Em funções do tipo dado as informações de controle são armazenadas e mantidas em conjunto com a aplicação. São exemplos de informações de controle: telas que armazenam as preferências dos usuários, como em um navegador de internet as informações de controle por ser: o tempo que os dados ficam armazenados no cache, a página inicial, nível de privacidade. Em uma compra no comércio eletrônico é possível escolher a forma de pagamento (boleto, cartão de crédito, cartão de débito), a forma de pagamento é uma informação de controle.

Processo elementar: Um processo elementar, segundo Vazquez, Simões e Albert (2010, p.74), é a menor unidade de atividade significativa para o usuário final. Um processo elementar deve ser completo em si, e deixar a aplicação em estado consistente. Santos (2011, online, p.11) resume processos elementares como funções transacionais independentes, ou seja, funções sequenciais são parte de uma mesmo processo elementar e funções independentes constituem um processo elementar.

Dado derivado: Um dado derivado consiste em uma informação criada a partir da transformação de dados existentes através de funções matemáticas ou algorítmicas. O dado que é recuperado de um Arquivo Lógico Interno – ALI ou Arquivo de Interface Externa – AIE, mas que não sofre nenhuma alteração não é considerado um dado derivado. Segundo Longstreet Consulting Inc (online, p.4) dados derivados são dados cujo processamento vai além da recuperação simples de dados. Dados derivados são gerados quando um ou mais elementos são combinados através de algoritmos ou fórmulas matemáticas e criam um novo dado. Um dado derivado não deve estar armazenado em um ALI ou AIE.

Modificar comportamento do sistema: Segundo Vazquez, Simões e Albert (2010, p.105) modificar o comportamento do sistema é o mesmo que alterar um parâmetro de negócio, essa alteração ocorre através de uma transação. O efeito causado por essa mudança de parâmetro reflete no comportamento do sistema. Por exemplo, o sistema de compras dá autonomia para que sejam realizadas compras de até 20.000,00 no mês, sem que seja necessária a autorização da chefia. O valor limite de compra é um parâmetro do sistema – uma informação de controle. A alteração de valor vai interferir nas transações de compra, e consequentemente alterar o comportamento do sistema.

Lógica de processamento: A lógica de processamento é definida como qualquer dos seguintes requisitos especificamente solicitados pelo usuário para completar um processo elementar (VAZQUEZ, SIMÕES e ALBERT, 2010, p.106). A Tabela 1 resume as lógicas de processamento que podem ser utilizadas por cada uma das funções do tipo transação – Entrada Externa – EE, Consulta Externa – CE e Saída Externa – SE. Toda função do tipo

transação tem um objetivo principal. A lógica de processamento é a responsável por fazer com que esse objetivo seja alcançado.

Tabela 1 – Lógica de processamento em funções transacionais.

Tipo da lógica de processamento	Função do tipo transação		
	EE	SE	CE
1. Realização de validações	Pode	Pode	Pode
2. Realização de cálculos e fórmulas matemáticas.	Pode	Deve*	Não
3. Conversão de equivalência entre montantes.	Pode	Pode	Pode
4. Seleção e filtragem de dados utilizando critérios para comparar múltiplos conjuntos de dados.	Pode	Pode	Pode
Tipo da lógica de processamento	Funções do tipo Transação		
	EE	SE	CE
5. Condições são analisadas para determinação de qual se aplica	Pode	Pode	Pode
6. Um ou mais ALIs são atualizados	Deve*	Deve*	Não
7. Um ou mais ALI ou AIE são referenciados.	Pode	Pode	Deve
8. Dados ou informações de controle são recuperadas	Pode	Pode	Deve
9. Dados derivados são criados a partir de dados existentes	Pode	Deve*	Não
10. O comportamento do sistema é alterado	Deve*	Deve*	Não
11. Preparar e apresentar informações para fora da fronteira da aplicação	Pode	Deve	Deve
12. Capacidade de aceitar dados ou informações de controle que entra na fronteira da aplicação.	Deve	Pode	Pode
13. Dados são ordenados ou organizados.	Pode	Pode	Pode
Deve – A transação deve obrigatoriamente executar este tipo de lógica de processamento			
Deve* – A transação deve executar pelos menos uma das lógicas de processamento classificadas como deve *			
Pode – A transação pode executar este tipo de lógica de processamento, mas não é obrigatório.			
Não – A transação não pode executar este tipo de lógica de processamento			

Fonte: Vazquez, Simões e Albert (2010).

Agora que as definições dos termos utilizados em um processo de contagem de pontos de função foram apresentadas a seção seguinte irá apresentar o processo de contagem das funções do tipo dado.

2.2.6.2. Funções do tipo dado

Essa seção tem como objetivo apresentar as funções do tipo dado e as regras para a contagem das mesmas. Segundo Vazquez, Simões e Albert (2010, p.71) as funções do tipo dado são as que representam funcionalidades fornecidas ao usuário para atender suas necessidades de dados internos e externos à aplicação, ou seja, são os requisitos responsáveis pelo armazenamento de dados no sistema.

As funções do tipo dado são classificadas em Arquivo Lógico Interno (ALI) e Arquivo de Interface Externa (AIE). Sendo que um ALI representa dados que estão armazenados dentro da fronteira da aplicação, e o AIE representa dados referenciados pela aplicação, ou seja, dados que são utilizados pela aplicação, mas que são mantidos por outra aplicação. No contexto de pontos de função um arquivo não representa o mesmo que um arquivo do sistema operacional, nesse caso, um arquivo é um conjunto de dados relacionados logicamente e reconhecidos pelo usuário.

Um arquivo em PF pode estar relacionado com um arquivo do sistema operacional ou em tabelas no banco de dados. Um exemplo disso é o requisito cadastrar cliente, no cadastro do cliente pode ser que o endereço seja armazenado em uma tabela diferente da tabela cliente, e o telefone em outra tabela, nesse caso, para fins de implementação existem 3 tabelas que são utilizadas no momento para o armazenamento de dados, porém para a contagem de pontos de função essas três tabelas formam um único ALI, pois para a lógica do negócio e para o usuário não há sentido o armazenamento separado desses dados.

O processo de contagem de funções do tipo dado se baseia na identificação de ALIs e AIEs, na definição da complexidade dos mesmo e na atribuição de um peso em PF de acordo com a complexidade. A seguir serão abordadas as definições de ALI e AIE e suas regras de identificação e classificação.

a. Arquivo Lógico Interno – Definições e regras de contagem

Um Arquivo Lógico Interno – ALI é um grupo lógico de dados relacionados, identificável pelo usuário que é mantido inteiramente dentro da fronteira da aplicação é mantido através de Entradas Externas – EE (LONGSTREET, online, p.5). Sua principal função é armazenar dados da aplicação, ou seja, são mantidos por processos elementares que adicionam, alteram ou excluem. Por exemplo, uma tabela que armazena no

banco de dados da aplicação as informações de uma venda através do processo elementar *Vender* é considerada um ALI.

b. Arquivo de Interface Externa

Um Arquivo de Interface Externa é um conjunto de dados ou informações de controle, identificável pelo usuário, logicamente relacionado e mantido fora da fronteira da aplicação que está sendo contada. Um ALI em uma aplicação pode vir a ser AIE em outra aplicação. Um AIE de uma aplicação necessariamente deve ser um ALI em outra aplicação. Um exemplo de Arquivo de Interface Externa é uma tabela *cliente* que é mantida por outra aplicação, e atualizada pela aplicação sendo contada.

A diferença básica entre um ALI e um AIE é que o primeiro é mantido dentro da fronteira da aplicação contada enquanto o segundo é mantido fora da fronteira da aplicação contada.

As funções do tipo dados devem ser classificadas de acordo com a sua complexidade, que pode ser baixa, média ou alta. Essa complexidade é determinada com base no número de Tipo de Dados - TD e no número de Tipos de Registro – TR.

- **Tipo de dados – TD**

Um tipo de dados é um campo único reconhecido pelo usuário, não repetido. É necessário identificar em que faixa de valores eles se enquadram, resumidamente pode-se dizer que um tipo de dado é um campo do arquivo, ainda que essa relação não seja completamente correta. (VAZQUEZ, SIMÕES e ALBERT 2010, p.75). A Figura 6 apresenta a diferença entre um tipo de dado e um campo de um arquivo.

Contatos		3 Campos 3 TD
Telefone Comercial	(63) 32246587	
Telefone Residencial	(63) 32234567	
Telefone Celular	(63) 84567890	
Contatos		3 Campos 1 TD
Telefones	(63) 32246587	
	(63) 32234567	
	(63) 84567890	
Contatos		6 Campos 2 TD e 1 TR
Tipo de telefone	Numero	
Comercial	(63) 32234567	
Residencial	(63) 32246598	
Celular	(63) 84567890	

Figura 6- Diferença entre TD e campo de arquivo. (VAZQUEZ, SIMÕES e ALBERT, 2010, p.75).

Como pode ser visto na Figura 6 não é sempre que um tipo dado é representado por um campo de um arquivo. Na primeira informação de contato têm-se três tipos de telefone, cada um deles armazenado em um campo, logo cada um deles representa um tipo de dado. Já no segundo caso existe um único campo para representar o telefone (um tipo de dado) e esse mesmo tipo de dado guarda três campos. No terceiro caso, existem 6 campos que são as três indicações de telefone e seus respectivos valores, porém existem apenas 2 tipos de dados, que são o tipo do telefone e o número do telefone. Outro exemplo que deixa essa regra mais clara é o representado na Figura 7.

Dia	Mês	Ano
26	04	2012

Figura 7 - Exemplo de tipo de dado

A data armazenada dessa forma tem três campos (dia, mês e ano), porém, na visão do usuário esses três campos representam um tipo de dado que é uma data. Ou seja, nesse caso um campo de um arquivo não representa um tipo de dado, mas um conjunto de campos com significado para o usuário.

Para a contagem dos tipos de dados as regras definidas pelo IFPUG devem ser seguidas. A primeira regra de contagem dos ALIs já foi abordada acima e diz respeito a um ALI ser um conjunto de arquivos ou informações de controle logicamente relacionados com

significado para o usuário. Para que um arquivo seja contado como um tipo de dado, todas as regras que serão apresentadas a seguir devem ser válidas (VAZQUEZ, SMIÕES e ALBERT, 2010 p. 75-77).

1. Conte um tipo de dado para cada campo único reconhecido pelo usuário e não repetido, mantido ou recuperado por um ALI ou AIE por meio da execução de um processo elementar, ressaltando que um processo elementar é a menor unidade de atividade com significado para o usuário, deve ser completo em si mesmo e deixar a aplicação em estado consistente.

Exemplos:

- Em um agendamento do pagamento a data pode ser armazenada em três campos (dia, mês e ano), mas para o usuário esses três campos tem apenas um significado (data), sendo assim seria contado apenas um tipo de dado.
 - Campos calculados e armazenados em um ALI também devem ser contados como tipo de dado. Como a tabela funcionário que armazena além dos dados do funcionário o número total de clientes atendidos por ele, nesse caso, o número total de clientes também deve ser contado como um tipo de dado.
 - Campos de *timestamps*, se reconhecidos pelo usuário também devem ser contados como tipo de dados. Em um processo elementar que altera o cadastro de um cliente a data de alteração é um campo *timestamps*, se o usuário tiver conhecimento desse campo ele deve ser contado como um tipo de dado.
 - Campos que se repetem várias vezes, como os valores do saldo do cliente para cada mês devem ser contados como um tipo de dado para o campo e um tipo de dado para o valor. Sendo assim, em um arquivo que armazena o saldo do cliente a cada mês ao longo de um ano, contará apenas como dois tipos de dados.
2. Quando duas aplicações mantêm ou referenciam o mesmo arquivo ALI/AIE, são contados apenas os campos utilizados pela aplicação que está sendo contada.

Exemplos:

- Uma aplicação mantém ou referencia os campos CPF e nome de um arquivo. Outra aplicação mantém ou referencia os campos nome, logradouro, CEP, cidade e estado. Nesse caso, para a primeira aplicação serão contados dois tipos de dados, e para a segunda aplicação devem ser contados cinco tipos de dados referentes ao

mesmo arquivo. Ou seja, a medição de funções deve levar em conta como um arquivo é visto para a aplicação que está sendo contada.

- O campo endereço do cliente deve ser identificado por uma aplicação por partes (logradouro, cidade, estado, CEP), ou seja, quatro tipos de dado, para uma segunda aplicação é necessário que se tenha um campo endereço como um todo em um único campo, ou seja, apenas um tipo de dado.

3. Conte um tipo de dado para cada campo solicitado pelo usuário para estabelecer um relacionamento com outro arquivo lógico (ALI ou AIE).

Exemplos:

- Em um sistema de controle de pontos as informações de entrada e saída dos funcionários são mantidas no Arquivo Lógico Interno Apontamento. As informações da pessoa são mantidas nesse mesmo ALI Apontamento, os dados da pessoa servem para estabelecer e manter uma relação entre o ALI Apontamento e o AIE Pessoa. Sendo assim, pode-se perceber que os campos conhecidos de um arquivo que representam uma chave-estrangeira são contados como tipos de dados.
- Quando um único arquivo Lógico é composto por mais de uma tabela no banco de dados, uma chave estrangeira é utilizada para estabelecer a relação entre as duas tabelas. Para a contagem dos tipos de dados a chave estrangeira deve ser contada uma única vez, como mostra a Figura 8:

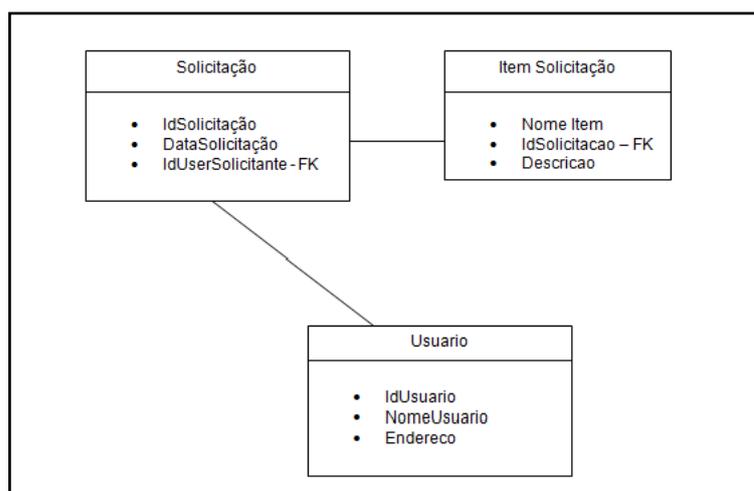


Figura 8—Exemplo de contagem de tipo de dados - FK

No caso desse modelo de dados representado na Figura 8, no qual existe a tabela Usuário, Solicitação e Item Solicitação, o relacionamento seria um usuário faz uma solicitação e essa solicitação pode possuir um ou mais itens. A chave estrangeira presente na

tabela Solicitação deve ser contada para o Arquivo Lógico Interno Solicitação, pois ela estabelece o relacionamento entre a tabela Usuário e a tabela solicitação. No entanto, a chave estrangeira presente na tabela Item Solicitação deverá ser contada apenas uma única vez, pois a tabela Solicitação e a Item Solicitação fazem parte do mesmo Arquivo Lógico.

- **Tipo de Registro - TR**

Segundo ANDRADE (2004, p.14) tipos de Registro também denominados como registros lógicos são subgrupos de dados reconhecidos pelo usuário que fazem parte de um ALI ou AIE. Segundo Vazquez, Simões e Albert (2010, p.77) os Tipos de Registro são divididos em dois tipos, os opcionais e os obrigatórios.

Os Tipos de Registro obrigatórios são aqueles que o usuário solicita que sejam sempre utilizados. São campos de preenchimento obrigatório em um processo elementar. Um exemplo de TR obrigatório é um processo elementar que cadastra um cliente e o CPF do cliente faz parte de sua chave no registro no banco de dados, sendo assim o campo CPF é um Tipo de Registro obrigatório.

Os Tipos de Registro opcionais são aqueles que o usuário tem a opção de preencher ou não em um processo elementar, por exemplo, no processo elementar que cadastra um cliente o usuário tem a opção de não preencher o campo telefone fixo, sendo assim este é um Tipo de Registro Opcional.

Segundo VAZQUEZ, SIMÕES e ALBERT (2010, p.78) a regra de contagem de TR consiste em:

- Contar m TR para cada TD. Cada função do Tipo Dado por default já tem um subgrupo que é uma função do Tipo Registro.
- Contar um TR adicional para cada um dos seguintes subgrupos lógicos de TDs que contem mais de um TD.
 - Entidade associativa com atributos não chave.
 - Subtipo (outro que não seja o primeiro subtipo).
 - Entidade atributiva em um relacionamento que não seja mandatório.

Após os tipos de Dados e tipos de Registro serem identificados e contados no processo de contagem, devem ser classificados de acordo com a Tabela 2 de definição de complexidade.

Tabela 2 – Definição da complexidade das funções do Tipo Dados.

		Tipo de dados		
		Até 19	De 20 a 50	51 ou mais
Tipo de Registros	1	Baixa	Baixa	Média
	De 2 a 5	Baixa	Média	Alta
	6 ou mais	Média	Alta	Alta

Fonte: (ANDRADE, 2005, p.15)

De acordo com a tabela de complexidade uma função do tipo dado que possua 4 TRs e 25 TDs é considerada uma função de complexidade média. A Tabela 3 mostra a contribuição em pontos de função de acordo com a complexidade dos Arquivos de Lógicos Internos e dos Arquivos de Interface Externa.

Tabela 3 – Contribuição de ALIs e AIEs.

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

Fonte: VAZQUEZ, SIMÕES e ALBERT (2010, p.78)

De acordo com a Tabela 3 uma aplicação com ALIs de complexidade baixa contribuiu com 7 PF para a contagem geral, enquanto que AIEs de complexidade também baixa contribuiu com 5 PF para a contagem.

2.2.6.3. Funções do tipo transação

Segundo Andrade (2003, p. 15) as funções do tipo transação são as funções de processamento de dados oferecidos pelo sistema ao usuário. Essas funções podem ser Entrada Externa – EE, Saída Externa – SE e Consulta Externa – CE. São compostas por itens referenciados e de arquivos referenciados.

Uma Entrada Externa é um processo elementar que processa dados ou informações de controle de fora de fronteira da aplicação, sua principal função é manter (incluir processos de inclusão, exclusão e alteração de dados) Arquivo Lógico Interno e/ou modificar o comportamento do sistema que está sendo medido. (VAZQUEZ, SIMÕES e ALBERT, 2010,

p.100-101). Esses dados podem vir de uma tela do sistema que propicia a entrada de dados como uma tela de cadastro, onde o usuário insere dados, ou de outro aplicativo. Esses dados podem ser informações de controle ou informações de negócio. As informações de controle não serão utilizadas para atualizarem arquivos lógicos internos. As informações de negócio são usadas para manter um ou mais Arquivo Lógico Interno (LONGSTREET, online, p.4). A Figura 9 representa uma Entrada Externa.

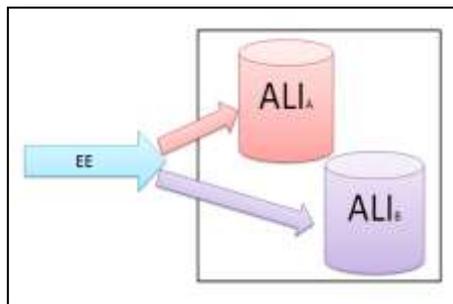


Figura 9 - Exemplo de EE que atualiza dois ALI - Baseado em LONGSTREET (2002, p.3)

O exemplo de EE representado na Figura 9 mostra uma única EE atualizando o ALI_A e o ALI_B.

Uma Saída Externa é um processo elementar em que dados derivados são enviados para fora da fronteira da aplicação (LONGSTREET, online, p.4). Para que um processo elementar seja considerado uma Saída Externa ele deve necessariamente apresentar dados ao usuário através de processamento lógico diferente ou adicional à recuperação de dados como algoritmos e/ou fórmulas matemáticas (ANDRADE, 2004, p.15). Esses dados derivados são enviados para outros aplicativos, caracterizando uma SE. Segundo VAZQUEZ, SIMÕES e ALBERT (2010, p.101) não é necessário que os dados derivados sejam apresentados ao usuário.

Exemplo:

- Em um sistema de estoque os dados dos produtos são armazenados em uma tabela Produtos no modelo de dados. Quando se realiza uma venda a quantidade de produtos vendidos deve ser decrementada da quantidade total de produtos na tabela Produtos.
 - O gerente da loja deseja obter ao final de cada mês um relatório que lhe informe quais são os produtos mais vendidos aos sábados.
 - O gerente também deseja um relatório detalhado do fluxo de produtos em estoque, dados como a quantidade de produto pedido aos fornecedores e por quanto tempo essa quantidade de produtos abasteceu o estoque satisfatoriamente.

Nesse caso apenas resgatar os dados da tabela estoque não será suficiente para a elaboração do relatório pedido pelo gerente. Haverá processamento extra que irá transformar

os dados brutos do Arquivo Lógico Interno em dados derivados, através de algoritmos e fórmulas matemáticas. A Figura 10 representa um exemplo de SE.

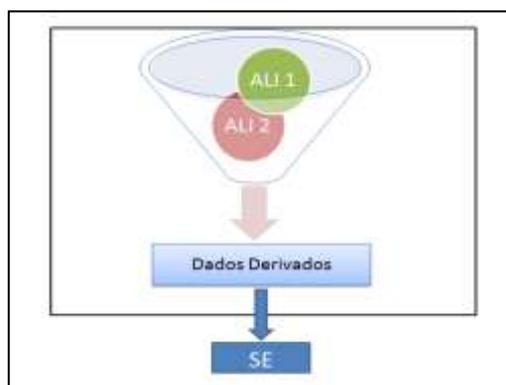


Figura 10 - Exemplo de Saída Externa com 2 ALIs – baseado em Longstreet (2002, p.4)

Uma Consulta Externa é, segundo Andrade (204, p.17), um processo elementar que visa enviar dados para fora da fronteira da aplicação. Uma Consulta Externa consiste em uma combinação entre atividades de entrada e saída de dados, onde o usuário envia uma solicitação para aplicação, a aplicação faz uma recuperação dos dados solicitados e os exibe ao usuário. Um exemplo de Consulta Externa é uma tela em que o usuário informa a identificação de um cliente e o sistema retorna os dados completos do cliente. Nesse caso a aplicação não teve que fazer nenhum cálculo para apresentar o resultado. A Figura 11 representa uma CE.

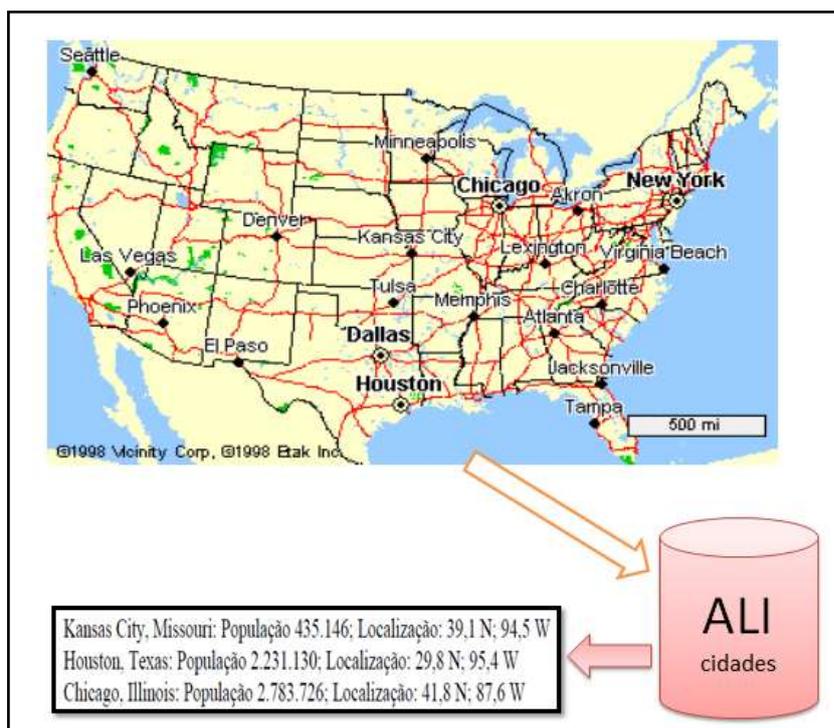


Figura 11 - Exemplo de CE com um ALI baseado em LONGSTREET (Online, p.14)

A Figura 11 mostra um exemplo de mapa quente, nesse tipo de mapa quando o usuário seleciona uma cidade, as informações dessa cidade são apresentadas ao usuário na tela da aplicação, nesse caso as informações são: população e localização. Para que um mapa quente seja considerado uma CE as informações apresentadas ao usuário devem constar em um dos ALIs ou AIEs da aplicação. Nesse caso as informações das cidades estão armazenadas no ALI cidades. No momento em que o usuário seleciona uma cidade no mapa, os dados são recuperados do ALI e apresentados a ele (LONGSTREET, online, p.14).

A Figura 12 a seguir representa o processo de contagem de funções do tipo transação de forma resumida.

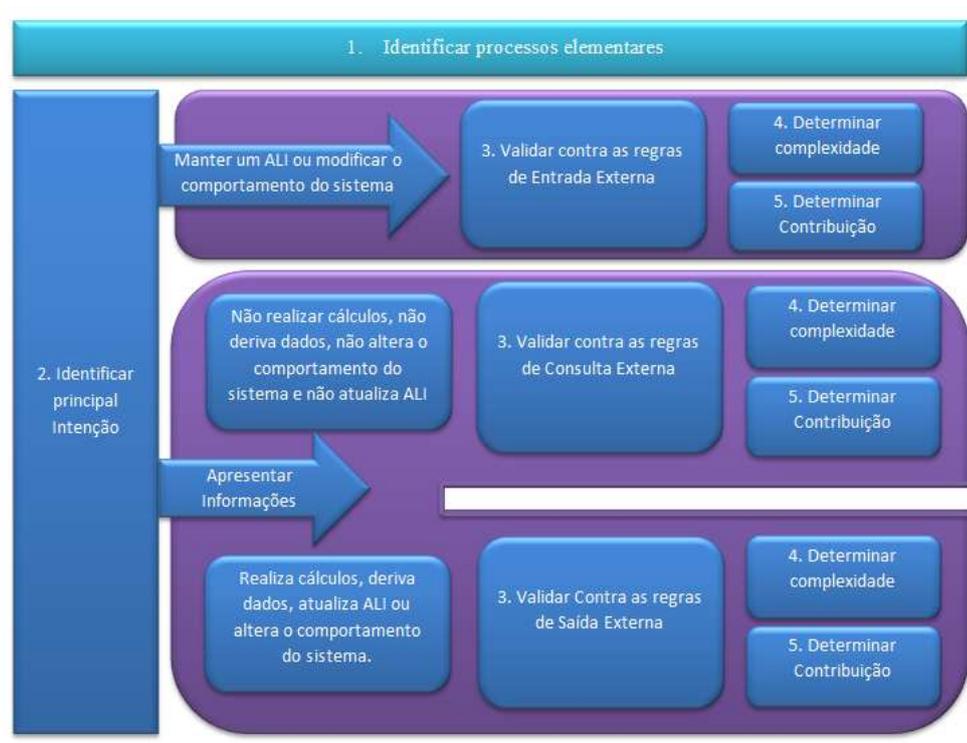


Figura 12 - Contagem de funções transacionais (VAZQUEZ, SIMÕES e ALBERT, 2010, p.107)

A contagem das funções do tipo transação inicia-se com identificação do processo elementar, e essa constitui uma fase muito importante da contagem. Para a identificação dos processos elementares é necessário levar em conta algumas regras, para que um único processo não seja contado como vários processos e para que um conjunto de processos elementares não venha a ser contado como um único processo elementar. Segundo Vazquez, Simões e Albert (2010, p.108) diferenciar as funções apenas com base no nome seria um critério muito frágil, uma vez que as funções podem ter nomenclaturas diferentes e ainda assim realizar exatamente o mesmo processamento no sistema.

Por exemplo: um sistema de Recursos Humanos – RH possui várias consultas de funcionários, que podem ser realizadas em momentos diferentes no uso do sistema e com

funções e ou/parâmetros diferentes. Mas para a APF seria difícil encontrar uma justificativa para diferença entre consultas como “consultar funcionário de férias” e “consultar funcionário em licença médica”. Pode ser fácil perceber que essas duas consultas realizam processamento de lógica diferente, mas em outros casos os nomes podem não dizer tanta coisa sobre a real intenção de um processo elementar.

De acordo com Vazquez, Simões e Albert (2010, P.108), por esse motivo existem regras que definem se um processo elementar é único, que estão descritas a seguir:

- a- Quando comparado a um processo elementar já identificado, conte dois processos elementares similares como um único processo se:
- Eles requerem o mesmo conjunto de tipos de dados;
 - Requerem os mesmos referenciados e
 - Fazem a mesma lógica de processamento para completar o processo elementar.
- b- Não se deve dividir um processo elementar em vários processos elementares quando este possui várias formas de lógica de processamento. Por exemplo: um processo elementar que valida e aceita os dados de um usuário; lê e seleciona os registros de um ALI; ordena e apresenta os resultados em resposta. Nesse caso no processo elementar “Consultar” existe mais de uma lógica de processamento, no entanto o ele continua a ser um único, autocontido e completo deixando o sistema em estado consistente.

Um processo elementar pode ser diferenciado de outros a partir de pequenas variações nos tipos de dados, arquivos referenciados ou lógica de processamento. VAZQUEZ, SIMÕES e ALBERT (2010, p.108).

O passo seguinte para a contagem das funções do tipo transação é identificar a intenção principal do processo elementar. Essa identificação é dividida basicamente em duas partes, um processo elementar deve ter como função principal manter ou atualizar um ALI ou AIE ou apresentar informações. Caso sua função principal seja manter ou atualizar um ALI ou AIE ele será uma função de Entrada Externa.

Se a intenção principal do processo elementar identificado é apresentar informações, ele poderá ser uma Saída Externa ou uma Consulta Externa, sendo que para a definição do tipo de transação, devem-se fazer os seguintes questionamentos:

- O processo elementar realiza cálculos?
- O processo elementar deriva dados?
- O processo elementar mantém ou atualiza um ou mais ALI ou AIE?
- O processo elementar modifica o comportamento do sistema?

Caso todas essas perguntas sejam respondidas de forma negativa então se trata de uma Consulta Externa. Se pelo menos uma das perguntas for respondida de forma positiva então se trata de uma Saída Externa.

Após a definição do tipo de transação ainda se deve submetê-las as regras de definição das funções de EE, SE e CE. Em seguida deve-se determinar a complexidade das funções e a contribuição delas para o processo de contagem. Cada EE, SE e CE deve ser classificada de acordo com sua complexidade que pode ser baixa, média ou alta. Essa complexidade é definida com base no número de Arquivos Referenciados – AR e no número de Tipos de Dados – TD. (VAZQUEZ, SIMÕES e ALBERT, 2010, p.111).

Um arquivo referenciado é um arquivo lógico interno – ALI lido ou mantido por uma função do tipo transação ou um arquivo de interface externa AIE lido por uma função do tipo transação. (VAZQUEZ, SIMÕES e ALBERT, 2010, p.112) Para Andrade (2004, p.16) para que um arquivo seja considerado um arquivo referenciado para uma função de Entrada Externa ele deve obedecer as seguintes regras:

- Os itens de dado de acordo com a visão do usuário;
- Contar os arquivos de dados que aparecem mais de uma vez em virtude da lógica ou técnica de implementação apenas uma vez.
- Considerar um item de dado adicional para as teclas de função ou linhas de comando que especificam ações a serem tomadas.
- Contabilizar os campos não informados pelo usuário, mas que são atualizados em um ALI por uma EE.
- As mensagens de erro, confirmação ou itens de dados adicionais devem ser contabilizados desde que sejam solicitadas pelo usuário.

A complexidade das funções do tipo transação EE é definida de acordo com a Tabela 4:

Tabela 4 - Complexidade para EE.

Arquivos Referenciados – (ARs)	Tipos de Dados – (TDs)		
	Até 4	5 a 15	16 ou mais
0 ou 1	Baixa	Baixa	Média
Igual a 2	Baixa	Média	Alta
3 ou mais	Média	Alta	Alta

Fonte: ANDRADE (2004, p.16).

De acordo com a Tabela 4 uma entrada externa – EE com dezesseis tipos de dados – TDs e dois arquivos referenciados – ARs é de complexidade alta, da mesma forma, uma

entrada externa com três arquivos referenciados e treze tipos de dados é classificada com complexidade alta.

Segundo Andrade (2004, p.16) Para determinar se uma função do tipo transação é uma Saída Externa – SE se devem observar as seguintes regras na lógica de processamento do processo elementar:

- Conter no mínimo um cálculo ou fórmula matemática;
- Criar dados derivados;
- Manter no mínimo um ALI;
- Alterar o comportamento do sistema;

Deve contar por padrão um arquivo referenciado para cada ALI. Para identificação dos tipos de dados e arquivos referenciados em uma SE as seguintes regras devem ser observadas. (Andrade, 2004, p.17):

- Contar um item de dado para cada campo reconhecido pelo usuário que não seja repetido e tenha significado lógico, que entre pela fronteira da aplicação e seja utilizado para especificar quando, qual ou quando um dado será recuperado ou gerado pelo processo elementar. Ou dado que saia da fronteira da aplicação. Caso um dado entre e saia da fronteira da aplicação ele deve ser contabilizado apenas uma única vez.
 - Exemplo: ao realizar uma consulta por clientes que tem contas a vencer em um determinado mês. Os dados dos clientes e o mês devem ser informados pelo usuário e serão usados como parâmetros para realização da consulta que irá gerar o resultado da Saída Externa e, portanto, deverão ser contados como tipo de dado.
- Contar um item de dado quando a aplicação enviar respostas para fora da fronteira de aplicação, desde que sejam solicitadas pelo usuário, essas mensagens podem ser de erro, de sucesso ou solicitação de confirmação para continuar algum procedimento.
 - Exemplo: Ao realizar o cadastro o cliente informou o campo CPF de maneira incorreta o sistema emite uma mensagem informando o erro no preenchimento ao usuário.
- Contar um item de dado para a habilidade de informar uma ação a ser tomada, quando existem múltiplos métodos para chamar um mesmo processo elementar deve-se contar apenas um tipo de dado.
 - Exemplo: para salvar um documento aberto o usuário pode usar o atalho CTRL + S, usar a opção Arquivo – Salvar do menu ou pode acessar o botão Salvar.

São três formas diferentes de salvar um arquivo, no entanto, deve ser contada apenas como um único tipo de dado.

- Não contar os campos que são derivados na lógica de processamento e que são armazenados em um ALI durante o processo elementar, mas que não cruzam a fronteira da aplicação.
 - Exemplo: Quando um relatório é gerado e apresentado ao cliente o sistema pode gerar um código para esse relatório e armazená-lo em um ALI, o cliente não toma conhecimento do código do relatório e ele não deve ser contado como um tipo de dado.
- Não contar variáveis, números de página, data/hora gerados pelo sistema.
 - Exemplo: Ao agendar um recebimento a data e a hora do mesmo são armazenadas em um ALI, mas não devem ser contadas, pois são dados de *timestamp* gerados pelo sistema e não por solicitação do usuário.

Depois de contadas os ARs e o TDs de uma SE deve determinar sua complexidade de acordo com a Tabela 5:

Tabela 5 - Complexidade para SE.

Arquivos Referenciados – (ARs)	Tipos de Dados – (TDs)		
	1 a 5	6 a 19	20 ou mais
0 ou 1	Baixa	Baixa	Média
2 a 3	Baixa	Média	Alta
4 ou mais	Média	Alta	Alta

Fonte: ANDRADE (2004, p.17).

De acordo com a Tabela 5 uma saída externa com um arquivo referenciado e vinte tipos de dado possui complexidade média, da mesma forma que uma saída externa com três arquivos referenciados – TRs e quinze tipos de dados - TDs também é de complexidade média.

Uma Consulta externa é um processo elementar que envia dados para fora da fronteira da aplicação. Os dados devem ser recuperados de um ou mais ALI. Segundo Andrade (2004, p.17) para classificar um processo elementar como uma função do tipo transação CE as seguintes regras devem ser obedecidas:

- Recupera dado ou informações de controle de um ou mais ALI ou AIE;
- Não contem fórmulas matemáticas ou cálculos na lógica de processamento;
- Não cria dados derivados a partir de dados existentes;
- Não altera o comportamento do sistema.

Um exemplo de Consulta Externa é quando o usuário solicita uma lista de clientes, a lista de clientes é recuperada de um ALI e exibida ao usuário. A complexidade de uma CE deve ser determinada de acordo com a Tabela 6.

Tabela 6 - Complexidade para CE.

Arquivos Referenciados – (ARs)	Tipos de Dados – (TDs)			
	0 ou 1	1 a 5	6 a 19	20 ou mais
2 a 3	Baixa	Baixa	Média	Média
4 ou mais	Baixa	Média	Alta	Alta

Fonte: ANDRADE (2004, p.18).

Depois que a complexidade das funções do tipo transação é calculada, então o número de pontos de função não ajustados é obtido através da tabela de contribuição apresentada a seguir (VAZQUEZ, SIMÕES e ALBERT, 2010, p.116).

Tabela 7 - Contribuição das funções transacionais.

Tipo da 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

Fonte: VAZQUEZ, SIMÕES e ALBERT (2010, p.116).

De acordo com a Tabela 7 uma Entrada Externa de complexidade alta contribui com seis pontos de função, uma Consulta Externa com o mesmo nível de complexidade contribui com seis pontos de função, enquanto uma Saída Externa com o mesmo nível de complexidade contribui com sete pontos de função para o contagem de PFs não ajustados.

2.2.7. Cálculo do fator de ajuste

Para se adequar ao padrão ISO o IFPUG tornou o cálculo do fator de ajuste opcional no processo de contagem. O fator de ajuste pode ajustar os pontos de função em até 35% para mais ou para menos. O cálculo do fator de ajuste é feito baseado em 14 características gerais, onde cada uma delas tem seu nível de influência para o processo de contagem. Segundo VAZQUEZ, SIMÕES e ALBERT (2010, p. 60) o objetivo do fator de ajuste é medir requisitos gerais da aplicação, que não são incluídos no processo geral de contagem (requisitos não funcionais).

O valor do fator de ajuste é baseado em 14 características gerais do sistema. Essas 14 características são listadas na Tabela 8.

Tabela 8- Fatores de Ajuste.

1. Comunicação de dados
2. Processamento distribuído
3. Performance/Desempenho
4. Configuração altamente utilizada
5. Volume de transações
6. Entrada de dados on-line
7. Eficiência do usuário final
8. Atualização on-line
9. Complexidade de processamento
10. Reutilização
11. Facilidade de instalação
12. Facilidade de operação
13. Múltiplos locais
14. Facilidade de mudanças

Fonte: KOTLINSKY, CECHELERO e SELZER (online)

Para cada uma das quatorze características é atribuído um peso de 0 a 5 esse fator indica o nível de influência. A influência de cada peso é dada pela Tabela 9.

Tabela 9 - Influência dos pesos.

0	Nenhuma influência
1	Influência mínima
2	Influência moderada
3	Influência média
4	Influência significativa
5	Grande influência

Fonte: KOTLINSKY, CECHELERO e SELZER (online)

Dado o peso de cada uma das características o fator de ajuste deve ser calculado de acordo com a Equação 1

$$VAF = (TDI \times 0,1) + 0,65 \quad (1)$$

Onde TDI representa a soma de total do nível de influência e VAF (*Value Adjustment Factor*) representa o valor do fator de ajuste (ANDRADE, 2004, p.18).

Após o cálculo do fator de ajuste são calculados os Pontos de Função Ajustados. Estes são obtidos através do produto dos Pontos de Função Não Ajustados e do Fator de Ajuste através da Equação 1 apresenta acima.

2.2.8. Cálculo do tamanho funcional

Como foi dito anteriormente, existem 3 tipos de contagem, que são: contagem de projeto de desenvolvimento, contagem de projeto de melhoria e contagem de aplicação. Para obter o tamanho funcional de um projeto deve-se usar fórmulas específicas em cada um dos casos. A seguir será abordado o cálculo do tamanho funcional de um projeto de desenvolvimento.

2.2.8.1. Projeto de desenvolvimento

Segundo Vazquez, Simões e Albert (2010, p.134) o cálculo do tamanho funcional de um projeto de desenvolvimento é composto pelas funcionalidades da aplicação que foram requisitadas pelo usuário e que estão presentes no projeto e pelas funcionalidades de conversão de dados requisitadas pelo usuário. Por exemplo: um sistema A está substituindo o sistema B que deixará de existir, mas os dados atuais do sistema B deverão ser importados pelo sistema A. Nesse caso, o tamanho funcional do sistema A será composto pelas funções do sistema A que foram requisitadas pelo usuário e pelas funções de importação e conversão de dados necessários para importações dos dados do sistema B.

No cálculo, são somados os pontos de função de todas as funções entregues em sua primeira instalação com o total de pontos de função das funções de conversão de dados, caso sejam necessárias de forma que o tamanho funcional de um projeto de desenvolvimento é obtido através da Equação 2:

$$\mathbf{DFP = (ADD + CFP)} \quad (2)$$

Onde DFP é o tamanho funcional do projeto de desenvolvimento, ADD é o tamanho total das funções entregues e CFP é o tamanho total das funções de conversão de dados.

2.2.8.2. Projeto de melhoria – IFPUG

O manual do IFPUG considera para fins de projeto de melhorias apenas a manutenção evolutiva ou adaptativa, que é quando novas funções são incluídas, alteradas ou excluídas para atender aos requisitos de negócio definidos pelo usuário. Manutenções corretivas, preventivas, estéticas não são consideradas como projetos de melhoria no manual do IFPUG. (VAZQUEZ, SIMÕES e ALBERT, 2010, p.136).

Um projeto de melhoria segundo o manual do IFPUG é composto por dois componentes que são: o tamanho em pontos de função das funcionalidades requisitadas entregues ao usuário pelo projeto de melhoria e tamanho em pontos de função das funcionalidades necessárias para conversão de dados.

Estão inclusas no projeto de melhoria novas funções adicionadas por necessidade do negócio, funções que foram alteradas ou excluídas. Também são contabilizadas as funções que foram modificadas em virtude de mudança do tipo de dado de algum ALI ou AIE, desde que a modificação no ALI ou AIE tenha sido efetuada por necessidades do negócio.

Segundo Santos (2011, online, p.20) uma função do tipo dado (ALI ou AIE) é considerada como alterada quando há alteração no item de dados (alteração ou inclusão). A adição de um novo campo de dado em um ALI ou AIE além de afetar as funções do tipo dado irá afetar as funções do tipo transação – processo elementar que mantém ou consulta dados desse ALI.

Caso um item de dado seja inserido em um Arquivo Lógico Interno, mas não seja atualizado e nem referenciado por funções do tipo transação da mesma aplicação, essa inclusão não deve ser considerada na contagem do projeto de melhoria. Por outro lado, caso um item de dados de um ALI ou AIE já existente passa a ser mantido ou referenciado pela aplicação, é considerado que o ALI ou AIE foi alterado.

Santos afirma que uma função do tipo transação (EE, SE, CE) será considerada alterada quando houver:

- Mudança de item de dado de uma função existente;
- Mudança de arquivos referenciados;
- Mudança de lógica de processamento.

Depois de identificadas as funções que foram alteradas no projeto de melhoria a contagem de pontos de função de acordo com o que foi visto na seção 2.2.6. O cálculo do tamanho funcional de um projeto de melhoria é feito com base na Equação 3.

$$EFP = (ADD + CHGA + CFP + DEL) \quad (3)$$

Onde: EFP é o tamanho funcional do projeto de melhoria, ADD representa o tamanho das funções que foram adicionadas pelo projeto de melhoria, CHGA representa o tamanho em PF das funções que foram alteradas pelo projeto de melhoria, CFP representa as funções de conversão e DEL representa o tamanho das funções que foram excluídas pelo projeto de melhoria.

Segundo Vazquez, Simões e Albert (2010, p.138) para o IFPUG as funções incluídas, alteradas e excluídas contribuem da mesma forma para o tamanho do projeto de melhoria, por

exemplo, um novo relatório é avaliado em 6 PF. Se um novo campo for adicionado a um relatório já existente a alteração do relatório é avaliada em 6 PF e se o mesmo relatório existente é excluído ele é avaliado em 6 PF.

Uma avaliação mais concisa faz pensar que as diretrizes do IFPUG para projetos de melhoria dão margem a erro, pois para estimativa de esforço, por exemplo, soa errôneo dizer que as ações de criar uma função nova, alterar uma função existente e excluir uma função geram o mesmo esforço.

No entanto os mesmos autores alertam que o fato das diretrizes do IFPUG darem margem a estimativas que podem não representar a realidade não é uma verdade absoluta, pois segundo eles o método para contagem de projetos de melhoria do IFPUG não possui uma granularidade tão fina a ponto de avaliar e pesar de maneira adequada as alterações feitas em uma determinada função, o modelo do IFPUG apenas leva em conta se uma função foi ou não alterada, a quantidade de alterações feitas e o quanto isso irá influenciar em estimativas de esforço não são tratados pelo CPM.

Vazquez, Simões e Albert (2010, p.139) ainda informam que essa margem de erro nas estimativas tende a ser maior em projetos pequenos, mas que em grandes projetos onde a estimação de esforço consiste em uma análise mais rigorosa de um conjunto de grandes projetos que contemple os vários tipos de demanda por melhoria na organização é possível obter estimativas úteis e bons resultados.

Em razão do manual do IFPUG não tratar questões detalhadas de alteração nas funções e apenas tratá-las de forma genérica a NESMA - *Netherlands Software Metrics Association* lançou em 2009 alternativas técnicas e mais aprofundamento no que tange aos projetos de melhoria do método do IFPUG. A seção 2.3 apresenta a NESMA de forma mais detalhada. A seção a seguir mostra a abordagem da NESMA para projetos de melhoria.

2.2.8.3. Projeto de melhoria – NESMA

O método criado pela NESMA é uma abordagem mais detalhada e refinada do método de APF para projetos de melhoria proposto pelo IFPUG. A abordagem da NESMA traz um fator de impacto que é calculado de acordo com a extensão da manutenção, alterações maiores tendem a ter um fator de impacto maior. A fórmula da NESMA para cálculo do tamanho funcional de projetos de melhoria é a seguinte:

$$EFP = \sum UEFP_{ADD} \times VAFA + \sum UEFP_{CHG} \times VAFA + \sum UEFP_{DEL} \times VAFB \quad (4)$$

Onde:

- **EFP** corresponde ao número de pontos de função do projeto de melhoria da NESMA.
- $\sum \text{UEFP}_{\text{ADD}}$ corresponde ao número de pontos de função não ajustados obtidos pelas funções adicionadas ao projeto de melhoria, incluído as funções de conversão de dados, nesse caso o fator de impacto é sempre 1.00.
- $\sum \text{UEFP}_{\text{CHG}}$ corresponde ao número de pontos de função não ajustados das funções alteradas, o fator de impacto é calculado para cada uma das funções alteradas.
- **Vafa** corresponde ao valor do fator de ajuste da aplicação depois do projeto de melhoria.
- $\sum \text{UEFP}_{\text{DEL}}$ corresponde ao número de pontos de função não ajustados das funções excluídas depois do projeto de melhoria, o fator de impacto é aplicado a cada função.
- **VAFB** corresponde ao fator de ajuste da aplicação antes do projeto de melhoria.

Para cálculo do fator de impacto das funções de dados alteradas é calculado o percentual dos TDs alterados, sendo que o percentual da alteração é definido pela razão entre os TDs alterados e o número original de TDs da função, como mostrado na Figura 13(NESMA).

$$\% \text{Mudança de TDs} = \frac{\text{Número de TDs adicionados/alterados/excluídos}}{\text{Número de TDs originais}} \times 100$$

Figura 13 - Percentual de mudanças em TDs (NESMA, online, 2009).

Depois que se obtiver a porcentagem dos TDs alterados, o fator de impacto é determinado pela Tabela 13:

Tabela 10 - Fator de Impacto das funções de dado.

% de mudança de TDs	$\leq 33\%$	$\leq 67\%$	$\leq 100\%$	$> 100\%$
Fator de Impacto	0,25	0,50	0,75	1,00

(NESMA, online, 2009)

Segundo as diretrizes da NESMA (2009) caso uma função do tipo dado mude de tipo, por exemplo, passando de um ALI a ser um AIE o fator de impacto 0,40 deve ser utilizado. No entanto quando a alteração é no tipo de dado, faz-se necessária a verificação da existência de alterações internas do ALI, ou seja, alteração nos tipos de dado (TDs). Caso o número de TDs venha a ser alterado, o fator de impacto deve ser determinado de acordo com a mudança que ocorreu.

Ainda segundo a NESMA é necessário que se faça uma comparação entre fator de impacto obtido devido a mudança no numero de TDs e o fator de impacto gerado devido a alteração no tipo de dados, maior valor entre os dois deve ser utilizado para cálculo do tamanho funcional de um projeto de melhoria.

Para o cálculo do fator de impacto das funções do tipo transação é feito um levantamento das funções de transação que foram alteradas pelo projeto de melhoria e então é calculada a porcentagem das funções alteradas. Essa porcentagem é obtida pela razão entre o numero de funções do tipo transação alteradas ou incluídas e o número original de Tipos de Dados e Arquivos Referenciados. Após a obtenção da porcentagem o fator de impacto é definido e acordo com a Tabela 11.

Tabela 11 - Fator de impacto das funções transacionais

Mudanças	Percentual de ARs			
	Percentual de TDs	<=67%	<=100	>100
<= 33%		0,25	0,50	0,75
<=67%		0,50	0,75	1,00
<=100		0,75	1,00	1,25
>100		1,00	1,25	1,50

Fonte: NESMA (2009, online)

A NESMA determina que o fator de impacto para as funções excluídas é de 0,40, sendo assim, uma função que conta com 8 PF ao ser excluída contribuirá com 3,2 PF. A NESMA ainda afirma que o fator de impacto não irá influenciar no tamanho da aplicação após a melhoria.

2.2.8.4. Aplicação

Para determinar o tamanho funcional de uma aplicação são utilizadas duas fórmulas a primeira é utilizada quando a aplicação foi instalada pela primeira vez, a segunda fórmula determina o tamanho de uma aplicação instalada após um projeto de melhoria.

Para o cálculo do tamanho funcional de uma aplicação instalada pela primeira vez (quando ainda não houve projetos de melhoria) é utilizada a fórmula descrita da Equação 5

$$AFP = ADD \quad (5)$$

Onde APF representa o tamanho da aplicação e ADD representa o tamanho total em pontos de função das funcionalidades requisitadas pelo usuário da aplicação sendo medida.

Em aplicações que passaram por projetos de melhoria a fórmula para o cálculo do tamanho funcional inclui as funções alteradas, inseridas e excluídas da aplicação, como mostrado na Equação 6

$$AFPA = (AFPB + ADD + CHGA) - (CHGB + DEL) \quad (6)$$

Onde AFPA é o tamanho da aplicação após a melhoria, AFPB é representa o tamanho da aplicação em pontos de função antes da melhoria, ADD representa o total de pontos de função das funcionalidades incluídas pelo projeto de melhoria. CHGA representa o total de pontos de função obtidos das funcionalidades que foram alteradas pelo projeto de melhoria (após o término) e CHGB é o total em pontos de função das funcionalidades alteradas pelo projeto de melhoria antes do término da melhoria e finalmente DEL representa o tamanho total das funcionalidades excluídas pelo projeto de melhoria.

Uma aplicação medida, após passar por melhorias, pode ter seu tamanho aumentado, diminuído ou pode não haver alterações no seu tamanho. As alterações em uma aplicação com relação a funções incluídas, alteradas ou excluídas são tratadas da seguinte forma:

- Uma funcionalidade nova foi inserida e aumenta o tamanho da aplicação;
- Uma funcionalidade alterada aumenta ou diminui ou não influencia o tamanho da aplicação;
- Uma funcionalidade excluída diminui o tamanho da aplicação.

Após a realização de melhorias em uma aplicação o tamanho funcional da aplicação deve ser atualizado, para que possa representar a nova realidade da aplicação. No entanto, uma melhoria não vai obrigatoriamente modificar o tamanho final da aplicação, por exemplo, se uma função excluída medir 6 PF e uma função inserida medir também 6 PF e se não foram realizadas modificações em funções existentes, então de acordo com a fórmula têm-se:

$$AFPA = (X + 6 + 0) - (0 + 6)$$

$$AFPA = X + 6 - 6$$

$$AFPA = X$$

Sendo que X é o valor em pontos de função da aplicação antes do término do projeto de melhoria e AFPA é o valor em pontos de função da aplicação após o término da melhoria, ou seja, não houve modificação no tamanho funcional da aplicação.

Uma das diferenças mais notórias entre as fórmulas para obtenção do tamanho funcional de um projeto de desenvolvimento e uma aplicação em sua primeira instalação é que em uma aplicação os pontos de função das funcionalidades de conversão de dados não são contabilizadas, e em projetos de desenvolvimento fazem parte do somatório final de pontos de função.

2.2.9. Documentar e reportar

Essa fase é realizada após o término da medição tanto em projetos de desenvolvimento, melhoria e aplicação. Os dados da medição devem ser armazenados em um repositório para consultas futuras, por exemplo, para normatizar os processos de comparação de um *software* desenvolvido com tecnologias diferentes. O nível de detalhamento da documentação de uma Análise de Pontos de Função é norteado pelo propósito da mesma.

As técnicas de contagem apresentadas nesta seção referem-se ao manual de práticas de contagem do IFPUG, que prevê uma contagem mais detalhada, e portanto, a necessidade um nível mais detalhado de informações, tais como quantidade de campos armazenados em uma tabela, ou quantidade de transações do *software*. Isso pode dificultar a contagem no início do projeto. A NESMA que é o grupo de usuários métricas de *software* da Holanda possui seu próprio manual de contagem de pontos de função e prevê três tipos de contagens. A utilização de cada tipo depende do nível em que se encontra o projeto. A seção seguinte apresenta abordagem da NESMA para contagem de Pontos de Função.

2.3. NESMA

NESMA – *Netherlands Software Metrics Association* é a fundação holandesa de usuários de Pontos de Função, foi um dos primeiros grupos de usuários da técnica fundada em 1989. Atualmente a NESMA é o maior grupo de usuários de APF da Europa. A NESMA mantém o seu próprio manual de práticas de contagem de pontos de função que está de acordo com o manual do IFPUG e pode ser utilizado com complemento para a utilização da técnica. (NESMA, Online, 2012).

A NESMA possui em seu manual de contagem de pontos de função três abordagens para a contagem de pontos de função que são: contagem estimativa, contagem indicativa e contagem detalhada. A utilização de uma ou outra abordagem depende do nível em que está o projeto e o quanto de informação (artefatos de documentação) se possui sobre ele.

2.3.1. Contagem Indicativa

Segundo NESMA (2012, online) na contagem indicativa é necessária apenas a identificação das funções dos tipos dados, ou seja, os Arquivos Lógicos Internos e Arquivos de Interface Externa, não existe a necessidade do conhecimento das funções do tipo de transação. Também não é necessário o detalhamento das funções do tipo dado com identificação de Tipo de Dados e Tipos de Registro. De acordo com Santos (online, 2009, p.10) após a identificação das funções do tipo dado a quantidade estimada de pontos de função é obtida através da seguinte fórmula.

$$PF = 35 * N^{\circ} \text{ de ALIs} + 15 * N^{\circ} \text{ de AIEs}$$

De acordo com a fórmula o número de PF é calculado a partir do produto entre a quantidade de ALIs e o número 35 somado com o produto entre a quantidade de AIEs e o número 15. Onde os números 35 e 15 representam o peso de cada ALI e AIE respectivamente.

Segundo a NESMA (online, 2012) a contagem indicativa é feita com base na premissa de que existem em média três Entradas Externas (EE), que geralmente são funções de adicionar, alterar e excluir dados do ALI, duas Saídas Externas (SE), e uma Consulta Externa (CE) para cada ALI. E em média uma SE e uma CE para cada AIE.

2.3.2. Contagem Estimativa

A abordagem da contagem estimativa deve ser utilizada quando já se possui um detalhamento maior do projeto que será desenvolvido. A documentação existente deve fornecer subsídios para a identificação de todas as funções do tipo dado e todas as funções do tipo transação.

Segundo NESMA toda função do tipo dado é avaliada com complexidade baixa e toda função do tipo transação é avaliada com complexidade média. Esse método da NESMA se diferencia da contagem de pontos de função do IFPUG por não determinar a complexidade individual de cada função, que no caso do IFPUG é determinada com base nos tipos de dados e nos tipos de registro. Após obter a número de funções do tipo dado e funções do tipo transação calculam-se os pontos de função com base nas Tabelas: 3 – Contribuição dos ALIs e AIEs e 7 - Contribuição das funções transacionais apresentadas na seção 2.2.6 (Processo de contagem). Sendo que, deve ser considerada para as funções de dados a complexidade baixa e para as funções de transação a complexidade média.

2.3.3. Contagem Detalhada

A contagem detalhada proposta pela NESMA é feita com base no número de funções do tipo dado e no número de funções transacionais, porém, diferentemente do modelo de contagem indicativa a complexidade das funções não é determinada de forma geral. Cada função do tipo dado e do tipo transação deve ter sua complexidade definida como: alta, média ou baixa de acordo com as Tabelas: 2 – Definição da complexidade das funções do Tipo Dados, 4 – Complexidade par SE, 5 – Complexidade para SE e 6 – complexidade para CE da seção 2.2.6 (Processo de contagem).

As três abordagens da NESMA apresentadas acima tem praticamente os mesmo princípios de contagem de pontos de função de forma que o resultado da contagem de um projeto através de pontos de função seguindo o manual da NESMA e do IFPUG não representam variações significativas no resultado final da contagem, ou seja, os dois modelos são compatíveis e um pode ser utilizado para complementar o outro.

A contagem de pontos de função oferece o tamanho estimado em PF, mas não fornece a estimativa para outros fatores como: esforço, custo e prazo. No entanto, existem técnicas de estimativas que utilizam como entrada a estimativa de tamanho fornecida pela APF.

2.4. Feature Driven Development – FDD

Feature Driven Development (FDD) ou Desenvolvimento Guiado por Funcionalidade é uma metodologia para gerenciamento e desenvolvimento de *software*. De acordo com descrição presente em Heptagon (online) o FDD combina as melhores práticas do gerenciamento ágil com uma abordagem da Engenharia de *software* orientada por objetos, dessa forma busca conquistar os três principais envolvidos em projetos que *software* que são clientes, gerentes e desenvolvedores.

Apesar de ser uma metodologia ágil de desenvolvimento iterativo e incremental, o FDD busca ainda conservar filosofias tradicionais. Na verdade o FDD busca um equilíbrio entre as práticas mais radicais e as tradicionais. O FDD procura simplificar o desenvolvimento e gerenciamento de *software* trazendo o processo em apenas duas fases que são: concepção e planejamento e construção.

A primeira fase é onde se inicia o projeto. Nessa fase as analistas e participantes da equipe realizam três processos que são: desenvolver um modelo abrangente que represente a base do negocio, criar uma lista de funcionalidades, onde as funcionalidades do sistema são identificadas e listadas e planejar por funcionalidade, onde é feito um planejamento do

projeto, a equipe do projeto classifica as funcionalidades em ordem de prioridade, definindo assim as que serão construídas primeiro. Essa fase é realizada uma única vez durante o processo de desenvolvimento.

A segunda fase do processo consiste na construção das funcionalidades, e é composta por dois processos: detalhar por funcionalidade e construir por funcionalidade. Essa fase é repetida para cada uma das funcionalidades definidas na lista de funcionalidades. No detalhamento é onde são criados artefatos como telas do sistema, casos de uso ou contratos.

A Figura 14 representa todo o processo de gerenciamento de projetos com o FDD.

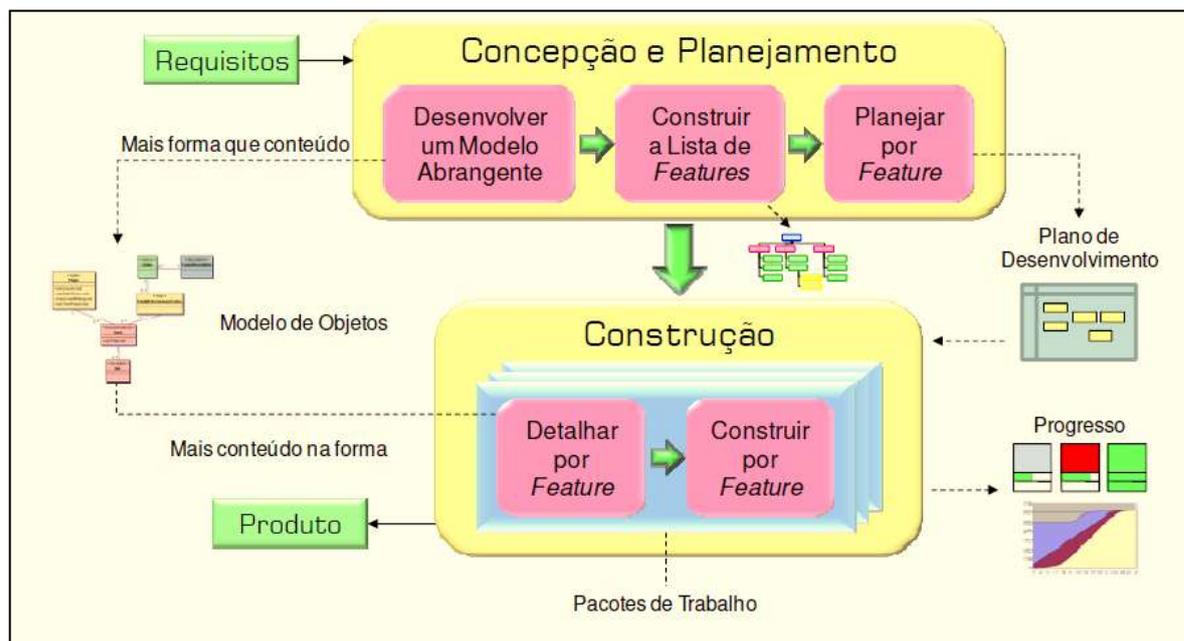


Figura 14 - Visão geral do FDD – HEPTAGON (Online, 2012)

Na Figura 14 está sendo apresentada uma visão geral da metodologia FDD. Pela quantidade de fases e processos pode-se perceber que o FDD busca gerenciar o desenvolvimento de projetos de *software* de forma simples, não tendo o foco somente em código como o XP (*eXtreme Programing*), mas também não tem uma pilha de processos burocráticos como o RUP (*Rational Unified Process*).

As responsabilidades de uma equipe no FDD são distribuídas por papéis, onde cada membro da equipe pode assumir mais de um papel simultaneamente e um papel pode ser assumido por mais de um membro da equipe. A seção 2.4.1 apresenta os tipos de papéis e as funções que devem ser desempenhadas por eles.

2.4.1. Papéis

Segundo Barbosa et al (online) os papéis estão classificados em principais, secundários e adicionais. Os principais papéis no FDD são:

Gerente de Projetos: O membro da equipe que assume esse papel é responsável por assuntos administrativos que estejam ligados ao projeto. Dentre outras funções Purificação (2010) destaca como principal a função de não permitir que nenhum fator externo ao projeto venha interferir na produtividade da equipe. Barbosa acrescenta às funções do Gerente de Projetos as responsabilidades inerentes a pessoal (Recursos Humanos) e calendarização, ou seja, definição de prazos e duração de funcionalidades.

Especialista do Negócio: Segundo Purificação (2010, p.42) O Especialista do Negócio deve apresentar à equipe com base no seu conhecimento do negócio os subsídios necessários para que as necessidades reais do projeto sejam identificadas. Também é função do Especialista do Negócio manter todos os envolvidos no projeto informados em relação às entregas. Esse papel é fixo na equipe e deve estar presente sempre que houver necessidade de detalhamento nas funcionalidades do sistema. Barbosa et. al (online) acrescenta que o Especialista do Negócio está envolvido nas definições de preço do produto e que ele pode ser um desenvolvedor, um cliente, um gerente, patrocinador ou um usuário final do sistema.

Arquiteto: é um arquiteto da informação que deve atuar como consultor de equipe em relação à arquitetura do sistema.

Gerente de Desenvolvimento: Segundo Barbosa et. al (online) o Gerente/Gestor de Desenvolvimento tem a função de acompanhar o dia-a-dia do desenvolvimento norteando a equipe. Seu trabalho é ladeado com o de Gerente de Projetos, inclusive é natural que um mesmo membro da equipe venha a assumir os dois papéis. Os conflitos que venham a ocorrer na equipe desenvolvimento, como entre programadores-chefes, devem ser solucionados pelo Gerente de Desenvolvimento.

Programador-Chefe: A função desse papel é liderar e equipe de programação de uma determinada funcionalidade, ou de um conjunto de funcionalidade, esse papel é um tipo de extensão dos papéis de Gerente de Projetos e Gerente de Desenvolvimento. Segundo Purificação (2010, p.42) esse papel é fundamental nas atividades de absorção de conhecimento do negócio e no planejamento das atividades.

Dono de Classe ou Programador: O membro da equipe designado como *Dono de Classe* é coordenado pelo Programador-Chefe e é responsável pela modelagem, desenvolvimento e testes de uma classe, uma funcionalidade.

Os papéis descritos acima são os papéis definidos como principais em um projeto desenvolvido e gerenciado com a metodologia FDD, mas segundo Barbosa et. al. (online) existem os papéis secundários que são: gerente de liberações, gerente de configurações, administrador de rede, especialista de ferramenta, testador e documentador. Além dos papéis secundários podem surgir os adicionais ou de suporte, que devem atender as necessidades adicionais do projeto.

Após a definição dos principais papéis no FDD e suas funções a seção 2.4.2 irá apresentar os processos que compõem o FDD.

2.4.2. Processos

Como já citado nas seções anteriores o ciclo de vida do desenvolvimento e gerenciamento de projetos de *software* com a metodologia FDD se resume em duas fases, a primeira composta por três processos, que são referentes a planejamento do projeto e a segunda composta por dois processos, referentes a desenvolvimento das funcionalidades. Cada um dos processos é composto por critério de entrada, atividades, verificação e critérios de saída, como representado Figura 15.



Figura 15 - Fluxo dos processos

Como pode ser visto na Figura 15 o fluxo interno de um processo no FDD envolve quatro artefatos. O processo recebe os critérios de entrada e a partir deles são desenvolvidas atividades. Cada processo tem suas próprias atividades. Após a conclusão das atividades são realizadas verificações para que as atividades sejam consideradas válidas. Cada processo gera

uma saída, essa é denominada critério de saída. Por exemplo, o processo DMA gera como saída um modelo de objetos abrangente.

2.4.2.1. Desenvolver um Modelo Abrangente – DMA

Desenvolvimento de um modelo abrangente é segundo Heptagon (online) primeiro processo do ciclo de vida de um projeto desenvolvido com FDD e acontece uma única vez durante o processo de desenvolvimento abrangendo todo o projeto.

Esse processo é iniciado quando acontece a primeira reunião entre o cliente e a equipe de desenvolvimento, nesse ponto o cliente deve apresentar os requisitos do sistema que será desenvolvido do seu ponto de vista, ou seja, do ponto de vista funcional e não técnico.

O processo consiste na “realização de estudos dirigidos sobre o escopo do sistema e seu contexto”, após isso são realizados estudos dirigidos para cada área do domínio de negocio que será modelada. O processo segue com a formação de pequenos grupos, cada grupo irá fazer estudos mais profundos sobre o domínio do negócio e criar um modelo que satisfaça o domínio do negócio.

Ao término do tempo determinado para essa atividade, os modelos criados pelos grupos são apresentados a todos, discussões são iniciadas, e finalmente um modelo é escolhido como modelo de projeto. O modelo escolhido pela equipe pode ser uma junção de dois ou mais modelos.

a) Critérios de entrada

Os critérios de entrada para o DMA são, segundo De Luca (online, 1998): o cliente estar pronto para dar início ao projeto e ter uma ideia do que é desejável no projeto. Além disso, segundo Heptagon (online), devem ser definidos os papéis do especialista do domínio, programador-chefe e arquiteto.

b) Atividades

As atividades são divididas entre opcionais e obrigatórias e sua realização é atribuída a um ou mais papéis e são, segundo Heptagon (online):

Formar equipe de modelagem: Essa atividade é de responsabilidade do gerente do projeto e é obrigatória, nela são definidos os membros da equipe de modelagem que são:

membros permanentes da área de domínio – Especialista do Domínio e Programadores-Chefes.

Fazer estudo dirigido sobre o domínio: Essa atividade é de responsabilidade da equipe de modelagem e é obrigatória. Nessa atividade o Especialista do Negócio apresenta uma visão geral do domínio que será modelado

Estudar a documentação: Nessa fase a equipe de modelagem deve estudar a documentação disponível, tais como: requisitos, casos de uso, modelos de objetos e guias de usuário.

Desenvolver o modelo: O objetivo dessa tarefa é criar um modelo de que suporte a área de domínio. Para que todos possam participar e as várias opiniões serem expostas a equipe de modelagem é dividida em grupos, geralmente com até três membros por grupo. Cada uma das pequenas equipes formadas propõe um modelo que satisfaça o domínio do negócio. É opcional que o arquiteto líder proponha modelos alternativos para servir de base para outros modelos. Um dos membros de cada uma das pequenas equipes apresenta o modelo proposto a todos, ao final um modelo é escolhido ou um novo modelo é criado a partir dos que foram propostos.

Refinar o modelo abrangente: o modelo abrangente é refinado sempre que há novas formas de modelo produzidas por iterações da atividade Detalhar Por Funcionalidade.

c) **Verificação**

A verificação do processo DMA consiste na revisão e avaliação do modelo desenvolvido, com o intuito de verificar se o modelo criado atende as reais necessidades do sistema. É realizada inicialmente pela equipe interna com a presença do especialista do domínio, mas se necessário pode ser feita uma avaliação externa onde o usuário final do sistema atua como consultor.

d) **Critérios de saída**

Os critérios de saída para o processo DMA são (HEPTAGON, online, p.3):

- Diagramas de classes com foco na forma do modelo, isto é, quais classes estão no domínio, como estão conectadas umas às outras e sob quais restrições;
- Métodos e atributos identificados são colocados nas classes;
- Diagrama(s) de sequência se houver;

- Comentários sobre o modelo para registrar o motivo pelo qual uma forma de modelo foi escolhida e/ou quais alternativas foram consideradas.

Os critérios de saída desse processo serão os critérios de entrada para o próximo processo que é Construir uma lista de Funcionalidades – CLF esse processo será abordado na seção 2.4.2.2

2.4.2.2. Construir a Lista de Funcionalidades – CLF

Construir a Lista de Funcionalidades “é uma atividade do projeto realizada na fase inicial é coordenada pelo Arquiteto e tem como função identificar todos os recursos para suportar os requisitos do sistema” (De Luca, online, 1998).

a) Critérios de entrada

Segundo De Luca (online, 1998) os critérios de entrada para essa fase são: definição do especialista do domínio do negócio, programadores-chefes e arquitetos. Para prosseguir com o projeto a equipe inicial deve estar formada e os membros disponíveis.

b) Atividades

Formar Equipe da Lista de Funcionalidades: é uma atividade de responsabilidade do gerente de projeto e classificada como obrigatória. A equipe da lista de funcionalidades é composta pelos especialistas no domínio do negocio, arquiteto e programador-chefe.

Construir a Lista de Funcionalidades: Segundo De Luca a equipe formada deve identificar os recursos necessários por meio de leitura dos casos de uso. “A equipe deve identificar as funcionalidades a partir do conhecimento adquirido no processo nº 1.” (De Luca, online, 1998). Essa identificação é somente uma decomposição funcional nas áreas definidas através da divisão do projeto pelo especialista em cada domínio nos estudos dirigidos realizados no 1º processo. A lista de funcionalidades é decomposta em áreas que englobam atividades de negócio, essas atividades de negócio são descompostas em passos, os passos são por fim funcionalidades. Estas são expressas em termos que possuem valor para o cliente (HEPTAGON, online).

c) **Verificação**

A verificação desse processo consiste em uma avaliação interna, essa avaliação é segundo De Luca feita por membros da equipe que estavam realmente envolvidos com o processo de definição dos requisitos, caso haja a necessidade a avaliação é feita por membros do negócio (usuários).

d) **Crítérios de Saída**

Para esse processo tem-se como critérios de saída “uma lista de áreas de negócio, cada área de negócio possui uma lista atividades de negócio pertencente aquela área. Para cada passo na atividade de negócio existe uma funcionalidade que satisfaça o passo.” (Heptagon, online).

Ao final desse processo uma lista de funcionalidades é gerada, essa lista deve representar as necessidades reais do negócio do ponto de vista do cliente. O processo seguinte é inerente ao planejamento das funcionalidades.

2.4.2.3. **Planejar por funcionalidade – PPF**

Esse processo é o último da primeira fase do ciclo de vida do FDD e como os anteriores a ele é realizado uma única vez e engloba todo o projeto. Essa atividade tem como objetivo produzir um plano de desenvolvimento do projeto e é desenvolvida pela equipe de planejamento composta por: gerente de desenvolvimento e programadores-chefes.

a) **Crítérios de entrada**

Esse processo tem como critério de entrada uma lista completa de funcionalidades, ou seja, o critério de saída do processo anterior.

b) **Atividades**

As atividades que compõem o processo PPF serão apresentadas a seguir:

Formar equipe de planejamento: é uma atividade de responsabilidade do gerente de projetos e tem como objetivo determinar a equipe que será responsável pelo planejamento das

funcionalidades do projeto. E equipe é composta por gerente de desenvolvimento e programadores-chefes.

Determinar sequência de desenvolvimento: essa atividade é de responsabilidade da equipe de planejamento que foi definida na atividade anterior e seu objetivo é determinar a sequência na qual o projeto será desenvolvido. Segundo De Luca (online) a sequência das atividades no desenvolvimento não é definida no formato tradicional “deve” e “desejável”, no processo do FDD essa prioridade é definida por outros fatores como a complexidade. Segundo Nebulon (online) e Heptagon (online) essa definição é feita com base nos seguintes critérios:

- Dependências entre as funcionalidades, no que tange as classes envolvidas;
- Distribuição da carga horária dos proprietários de classes, por exemplo, um proprietário de duas classes não pode construí-las se dedicando em tempo integral no mesmo período de tempo;
- Complexidade das funcionalidades a serem desenvolvidas;
- Antecipação das atividades de negócio de alto risco ou de complexidade alta;
- Considerações externas que estão visíveis, marcos do projeto tais como versões *beta*, *preview*, pontos de verificação de *feedback* e os produtos integrais que satisfaçam esses marcos.

Atribuir Atividades de negócio aos programadores-chefes: atividade de atribuída á equipe de planejamento e seu objetivo é atribuir às classes de negócios aos programadores-chefes. Segundo De Luca (online, 1998) a atribuição é feita baseada nos seguintes critérios:

- Dependência entre atividades no que tange às classes envolvidas;
- Carga de trabalho entre os proprietários das classes;
- Complexidades das classes a serem implementadas.

Atribuir Classes aos Desenvolvedores: essa atividade é de responsabilidade da equipe de planejamento e tem como função a atribuição das classes aos desenvolvedores. Um desenvolvedor pode ser proprietário de várias classes. Segundo Nebulon (online) a atribuição de classes aos desenvolvedores é feita com base nos seguintes critérios:

- Balanceamento da carga de trabalho entre os desenvolvedores;
- Complexidade das classes;
- Utilização das classes (ex. alta utilização);
- Sequência de desenvolvimento das classes.

c) Verificação

A verificação desse processo consiste em uma avaliação interna feita pela própria equipe de planejamento, com a participação dos programadores-chefes, gerentes de projeto e gerente de desenvolvimento.

d) Critérios de Saída

O processo de planejar por funcionalidade gera como saída segundo Heptagon (online, 2008, p.7) os artefatos a seguir:

- Atividades de negócio com datas de término (mês e ano);
- Programadores Líderes atribuídos a atividades de negócio;
- Áreas com datas de término (mês e ano), derivadas da data do último término de suas respectivas atividades de negócio;
- Lista das classes e seus respectivos desenvolvedores proprietários (a lista de proprietários de classes)

Com a conclusão do processo PPL a primeira fase do ciclo de vida do FDD também é concluída. A fase seguinte consiste no processo de detalhamento, construção e testes das funcionalidades, esse processo é iterativo e incremental, é realizado para cada uma das funcionalidades que satisfaça o domínio do negócio.

2.4.2.4. Detalhar por Funcionalidade – DPF

Esse processo é realizado para cada uma das funcionalidades do projeto e é onde uma equipe formada. Um grupo de funcionalidades é agendado para desenvolvimento e é atribuído a um programador-chefe. O programador-chefe seleciona as atividades para desenvolvimento a partir de sua “caixa de funcionalidades”, segundo Heptagon (online, 2008, p.8) o programador-chefe pode escolher várias funcionalidades que estejam ligadas as mesmas classes e, portanto, aos mesmos desenvolvedores/donos de classe. Ainda segundo Heptagon (online, 2008, p.8) opcionalmente o programador-chefe pode escolher um conjunto de funcionalidades para desenvolver de uma vez, esse conjunto de funcionalidades é chamado de Pacote de Trabalho.

a) Critérios de Entrada

O critério de entrada para esse processo é a conclusão do processo anterior, Planejar por Funcionalidade, ou seja para que esse processo seja iniciado é necessário que o processo anterior tenha sido concluído.

b) Atividades

As atividades do processo são divididas em obrigatórias e opcionais, elas serão apresentadas a seguir.

Formar equipe de Funcionalidades: é a primeira atividade do processo, montar a equipe de funcionalidades que é formada por desenvolvedores/donos de classe, é de responsabilidade do programador-chefe e é uma atividade obrigatória. A equipe é selecionada com base nas funcionalidades que são selecionadas para desenvolvimento, pois ao escolher uma funcionalidade, essa pertence a uma classe que foi atribuída a um desenvolvedor/dono de classe no processo Planejar por Funcionalidade. Não necessariamente todos os donos de classe que estão envolvidas irão participar do desenvolvimento de uma funcionalidade, mas apenas aqueles que o programador-chefe julgar necessários.

Estudo Dirigido do domínio: é uma atividade opcional a ser realizada pelo especialista do domínio. O especialista do domínio faz uma apresentação da visão geral de funcionalidade que será projetada. Segundo Heptagon (online, 2008) “essa apresentação também deve incluir informações que estiverem relacionadas a essa funcionalidade, mas que não sejam necessariamente uma parte de sua implementação”. Geralmente a realização ou não dessa atividade depende de sua complexidade, funcionalidades mais complexas necessitam de mais detalhamento.

Estudar a documentação de referência: essa é uma atividade opcional e sua realização irá depender da complexidade da funcionalidade a ser desenvolvida, quando feita é de responsabilidade de toda a equipe de funcionalidades. Segundo Coad, Lefebvre e De Luca (1999, p.194) a equipe estuda a documentação da funcionalidade usando os documentos que foram criados em fases anteriores, o estudo tem como objetivo obter mais detalhamento sobre a funcionalidade a ser desenvolvida.

Desenvolver diagramas de sequência: é uma atividade opcional de responsabilidade da equipe de funcionalidades, nela são desenvolvidos os diagramas de sequência necessários para a funcionalidade que será entregue. Os diagramas produzidos deverão ser armazenados no sistema de controle de versões.

Refinar modelo de objetos: atividade obrigatória e de responsabilidade do programador-chefe. Para essa atividade o programador-chefe cria uma área compartilhada com a equipe de funcionalidades, podendo ser uma pasta no servidor ou mesmo em seus próprios computadores. O propósito dessa área compartilhada é que a equipe de funcionalidades tenha uma visão geral do que está sendo feito, do andamento do projeto. O

programador-chefe refina o modelo para adicionar novas classes, métodos, atributos ou para alterar os existentes, essas inserções ou alterações são feitas baseadas nos diagramas de sequência produzidos na fase anterior. As alterações ou inserções resultam na necessidade de atualização dos arquivos fontes de linguagem de implementação. “O programador-líder cria diagramas do modelo num formato publicável. Esses arquivos devem ser submetidos ao sistema de controle de versões e publicados na intranet do projeto.” (Heptagon, online).

Escrever prefácios das classes: essa atividade é de responsabilidade da equipe de funcionalidade e é obrigatória. Segundo Heptagon (online, 2008, p.9) o proprietário de classe deve escrever os prefácios de classe e métodos para cada item que foi definido pela funcionalidade e pelos diagramas de sequência. O prefácio inclui parâmetros de métodos, tipos de retorno, exceções e mensagens. Essa prefácio é escrito com base nos documentos de arquivos fontes da linguagem de implementação e diagramas de sequência. Após a tarefa ter sido completada por cada desenvolvedor/dono de classe o programador-chefe gera uma documentação e submete para publicação na intranet do projeto.

c) **Verificação**

Segundo Nebulon (online) é uma inspeção é por membros da equipe de funcionalidades e por outros membros do projeto. O programador-chefe é que define se a avaliação será interna (somente a equipe de funcionalidades) ou externa. Após o aceite uma lista de tarefas é adicionada em cada uma das classes afetadas, os membros da equipe incluem suas tarefas à sua agenda de tarefas. O programador-chefe adiciona as tarefas listadas na área compartilhada ao sistema de controle de versões.

d) **Crítérios de Saída**

Segundo Heptagon (online) esse processo gera como saída um Pacote de Projeto inspecionado e aprovado. Esse pacote consiste em:

- Uma capa com comentários, que completa e descreve o pacote de projeto de tal forma a ser suficiente para futuros revisores;
- Os requisitos referenciados (se houver) na forma de documentos e de todos os memorandos de confirmação relacionados e documentação de apoio;
- Os diagramas de sequência;
- O modelo de objetos com classes, métodos e atributos novos/atualizados;
- A saída gerada pela e para os prefácios de classes e métodos, criados ou modificados por esse projeto;

- A lista de tarefas e agendamentos para itens de ação nas classes afetadas para cada membro da equipe.

Assim o 4º processo termina. Vale salientar que esse processo é repetido para cada funcionalidade com valor para o cliente. A seção 2.4.2.5 irá abordar o 5º e último processo do FDD. Esse processo atualiza o modelo de dados abrangente que foi definido no primeiro processo DMA, como a Figura 16 ilustra essa relação.

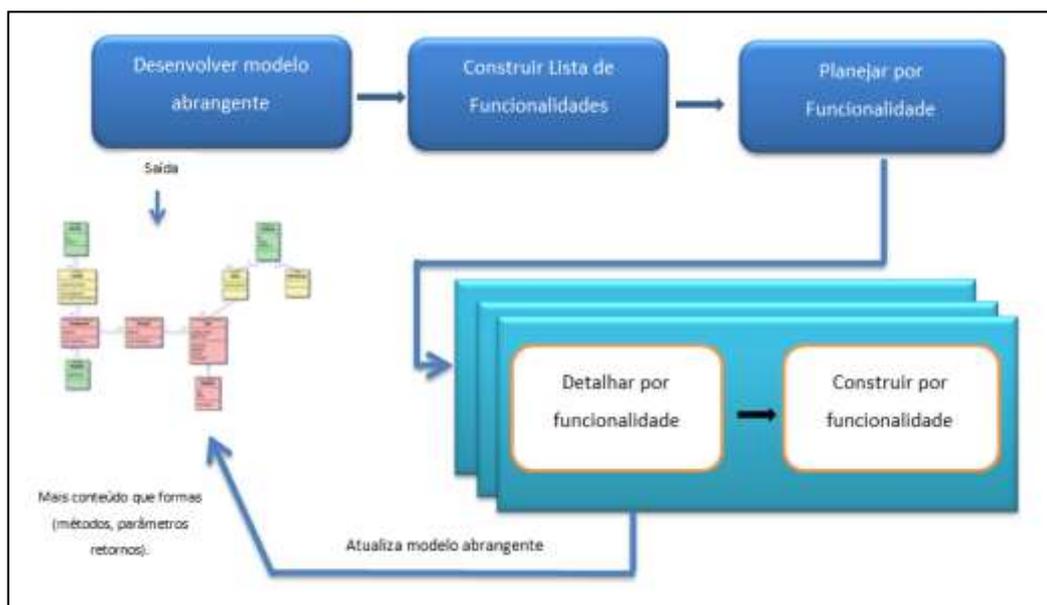


Figura 16 – Relação do processo DPF com o processo DMA

De acordo com a Figura 16 toda vez que o processo Detalhar por Funcionalidade for executado o modelo abrangente deve ser atualizado, essa atualização se dá com a inserção de métodos, parâmetros para métodos e tipos de retorno dos métodos. É possível que para determinadas funcionalidades não seja necessário atualizar o modelo abrangente.

2.4.2.5. Construir por Funcionalidade - CPF

Segundo Coad, Lefebvre e De Luca (1997, p.195) esse processo se inicia com Pacote de Trabalho construído no processo anterior pronto. A partir do pacote de trabalho cada proprietário de classe, desenvolve seus métodos para que suas classes suportem o projeto para a funcionalidade. Os desenvolvedores/donos de classe realizam teste de unidade em suas classes. Uma inspeção é feita pela equipe de funcionalidade ou caso o programador-chefe acredite ser necessário por uma equipe externa, mas participante do projeto. Depois da inspeção de classe o desenvolvedor/dono de classe revisa sua classe, verificando se ela possui todos os métodos para atender as necessidades da funcionalidade.

a) Critérios de entrada

O critério de entrada para esse processo é ter um pacote de trabalho pronto, ou seja, o processo anterior para a funcionalidade deve ter sido concluído.

b) Atividades

As atividades que compõem o processo Construir por Funcionalidade serão descritas a seguir.

Implementar classes e métodos: nessa fase cada desenvolvedor/dono de classe implementa suas classes e métodos necessários para suportar os recursos da funcionalidade. A implementação deve ser feita de acordo com o que foi definido nos diagramas de sequência no processo DPF.

Inspecionar código: uma inspeção no código é realizada pela equipe de funcionalidades e/ou por outros membros da equipe de projeto, sendo a decisão de inspeção externa cabe ao programador-chefe. A inspeção pode ser realizada antes ou depois do teste de unidade, a decisão também cabe ao programador-chefe.

Teste de unidade: é uma atividade de responsabilidade da equipe de funcionalidade, cada proprietário de classe teste seus códigos se certificando que todos os requisitos de suas classes foram satisfeitos. É responsabilidade do programador-chefe determinar quais os testes de unidade serão necessários para as classes.

Promover a versão atual (Build): é uma atividade de responsabilidade do programador-chefe e da equipe de funcionalidade. Essa atividade é realizada quando os códigos de todas as classes foram inspecionados e aprovados na inspeção e nos testes de unidade. O programador-chefe promove as classes para o processo de compilação e atualiza o status do recurso na lista de funcionalidades.

c) Verificação

A verificação compreende os testes de unidade e inspeção, já descritos anteriormente. Segundo Coad, Lefebvre e De Luca(1997, p.195) um escriba da equipe registra os itens de ação para todos os proprietários de classe.

d) Critérios de Saída

Segundo Coad, Lefebvre e De Luca(1999, p.195) para a conclusão desse processo os seguintes resultados devem ser entregues:

- Métodos implementados e inspecionados;
- Testes de unidade para cada método;
- Classes verificadas por proprietários;
- Funcionalidades promovidas para processo de compilação;
- Término de uma função com valor para o cliente.

2.4.3. Considerações do FDD

Um projeto desenvolvido com a metodologia FDD é concluído quando os últimos dois processos são realizados para todas as funcionalidades do sistema e o cliente recebe o produto final, que foi especificado nas fases iniciais.

Como pode ser observado ao longo da descrição dos processos no FDD ele é uma metodologia de desenvolvimento e gerenciamento de projetos de *software* iterativo e incremental, e tem como foco a entrega frequente de valor ao cliente. O FDD também busca ter a presença do cliente no ciclo de vida do desenvolvimento, além da presença constante do especialista do domínio. Em alguns momentos o usuário do sistema também pode participar do processo, para esclarecer dúvidas a respeito das funcionalidades do sistema.

No FDD a construção de uma funcionalidade deve ter duração de uma semana ou menos, caso haja a necessidade de mais tempo essa funcionalidade deve ser dividida em partes menores.

3. MATERIAIS E MÉTODOS

Nessa seção serão apresentados os materiais métodos utilizados na realização desse trabalho.

3.1. Local e Período

Esse trabalho foi desenvolvido no Complexo de Informática do CEULP/ULBRA e na residência da autora, ambos localizados na cidade de Palmas-TO. O desenvolvimento do trabalho compreendeu o período de fevereiro a junho de 2012 – período em que foi definida a proposta e desenvolvida a revisão de literatura – e agosto a dezembro de 2012, período em que foram realizadas análises e desenvolvida a proposta.

3.2. Materiais

Os materiais utilizados para realização desse trabalho foram trabalhos científicos, tais como: livros, artigos, monografias, teses e dissertações. Grande parte do material utilizado foi coletado no meio eletrônico.

3.3. Metodologia

O projeto é desenvolvido na forma de uma pesquisa aplicada e sua finalidade é propor uma maneira de utilização de uma metodologia de desenvolvimento e gerenciamento de *software*, FDD, em conjunto com uma técnica de dimensionamento funcional de *software*, APF. O projeto é abordado de forma qualitativa.

O objetivo metodológico é exploratório, sendo que, o foco foi o entendimento dos problemas existentes no desenvolvimento de *software* e como as técnicas estudadas podem ajudar na mitigação de tais problemas. A pesquisa é caracterizada como bibliográfica, pois foi realizada a partir de material científico já publicado, tais como: livros, artigos publicados em eventos e revistas científicas, teses e dissertações.

A metodologia para realização desse trabalho foi composta por duas principais fases: Projeto e Desenvolvimento. As seções 3.4 e 3.5 apresentam os detalhes de cada uma dessas fases.

3.3.1. Projeto

Nessa fase foi elaborado um Projeto de Pesquisa, para elaboração do projeto as seguintes etapas foram seguidas:

- **Proposta:** uma proposta de realização do trabalho foi elaborada e submetida à banca avaliadora para aprovação.
- **Coleta do material:** foi realizada uma busca por material bibliográfico relevante para construção do referencial teórico, tais como: livros, artigos e trabalhos científicos, teses e dissertações.
- **Estudo do material:** nessa fase foram realizados estudos dos materiais coletados na fase anterior.
- **Construção da revisão de literatura:** foi elaborada uma revisão de literatura abordando os temas estudados, tais como: Estimativas de *software*, Técnicas de Medição de *software*, Análise de Pontos de Função (abordagem IFPUG e as abordagens da NESMA), Feature Driven Development (FDD).

A segunda fase do trabalho refere-se ao desenvolvimento do projeto, a seção 3.5 apresenta um detalhamento do que foi feito nessa fase.

3.3.2. Desenvolvimento

A fase do Desenvolvimento refere-se à elaboração de uma proposta que permita a utilização de APF em projetos desenvolvidos com FDD. Essa fase consta de: análise das abordagens para contagem de pontos de função, elaboração da proposta e avaliação de risco.

Após a construção do referencial teórico, foram realizadas análises das abordagens para contagem de pontos de função: IFPUG e NESMA. As análises foram feitas com o intuito de selecionar uma abordagem que melhor se adapte à metodologia FDD. Para complementação das análises um projeto de *software* foi medido. O projeto medido foi o Sistema de Atendimento Psicológico Online, desenvolvido na disciplina de Estágio Supervisionado em Sistemas de Informação pelo acadêmico Mário Almeida Rodrigues, orientado pela professora Cristina D'Ornellas Filipakis. O sistema foi contado segundo a abordagem do IFPUG e segundo as abordagens Indicativa e Estimativa da NESMA. Os resultados obtidos em cada uma das abordagens foram apresentados e analisados.

A partir dos estudos e da análise realizada foi possível elaborar uma proposta de utilização conjunta de Análise de Pontos de Função e FDD. Essa proposta foi baseada na

inserção de dois novos processos na metodologia de desenvolvimento FDD que são voltados para o dimensionamento de um projeto e *software* e de suas funcionalidades.

Um dos processos foi inserido na fase de Concepção e Planejamento do FDD e foi denominado EMA (Estimar Modelo Abrangente), seu objetivo é obter um tamanho estimado em pontos de função do *software* a ser desenvolvido ainda na fase inicial.

O segundo processo foi inserido na fase de Construção e foi denominado EPF (Estimar por Funcionalidade), esse processo tem o foco no dimensionamento de uma funcionalidade, ou em um conjunto de funcionalidades com significado para o cliente.

Após a elaboração da proposta foram apresentados os principais riscos decorrentes da aplicação da proposta. Foi abordada a possível perda de agilidade da metodologia FDD e, em contrapartida, foram apresentados os benefícios da utilização de uma técnica de medição funcional para ajudar no gerenciamento de projetos.

Essa seção apresentou os passos seguidos para realização do trabalho, a seção seguinte irá apresentar os resultados obtidos na realização do trabalho, mencionados na seção 3.3.2. – Desenvolvimento.

4. RESULTATOS E DISCUSSÃO

Nessa seção serão apresentados os resultados obtidos com a realização do trabalho. Essa seção está subdividida em: Análise das abordagens de Pontos de Função, seção tem que como objetivo apresentar uma análise das abordagens de contagem de PF, mostrando através de exemplos a contagem de PF em cada uma delas; Proposta, seção que apresenta uma proposta de utilização conjunta de APF e FDD e Avaliação de risco, seção que apresenta possíveis riscos decorrentes da aplicação da proposta.

As seções seguintes têm como objetivo apresentar uma análise das abordagens para contagem de pontos de função da NESMA e do IFPUG para, dessa forma, justificar a escolha da abordagem utilizada na proposta elaborada. Parte dessa análise será feita a partir da contagem de Pontos de Função de um Sistema, como já abordado na seção anterior, esse sistema é trabalho da disciplina de Estágio Supervisionado em Sistema de Informação, do aluno Mário Almeida Rodrigues orientado pela professora Cristina Filipakis.

A documentação do Sistema de Atendimento Psicológico Online será utilizada para medição segundo a abordagem do IFPUG e segundo as abordagens Indicativa e Estimativa da NESMA. Os documentos disponíveis do projeto de sistema são:

- Diagrama de caso de uso;
- Diagrama de componentes;
- Diagramas de sequencia (algumas funcionalidades)
- Casos de uso expandido (algumas funcionalidades)
- Telas (algumas funcionalidades)

A próxima seção apresenta a contagem do sistema de Atendimento Psicológico Online segundo a abordagem do IFPUG.

4.1. Abordagem IFPUG

Essa seção apresenta a contagem de PF segundo a abordagem do Manual de Práticas de Contagem do IFPUG.

Para contagem das funções de dados na abordagem do IFPUG, segundo as regras contidas do CPM (Manual de Práticas de Contagem) os artefatos que podem ser utilizados para contagem são: requisitos do usuário, modelo de dados e de processo, janelas, telas, relatórios, diagramas de classes e caso de uso expandido.

Como foi apresentado na Seção de 2.2.6 - Processo de contagem, o primeiro passo para contagem é identificar as funções de dados. Para contagem das funções de dados do Sistema de Atendimento Psicológico Online, utilizando a abordagem do IFPUG serão utilizados o Diagrama de Componente e as Telas do Sistema. Os passos seguidos para contagem das funções de dados são:

1. Identificar as funções de dados;
2. Classificar como ALI ou AIE;
3. Contar os Tipos de Dados;
4. Contar os Tipos de Registro;
5. Determinar Complexidade;
6. Determinar tamanho funcional.

Para identificar as funções de dados, podem ser utilizados: diagramas de base de dados, modelos relacionais, diagrama de fluxo ou diagramas de componentes. Nesse caso será utilizado o diagrama de componentes. A Figura 17 exibe o Diagrama de Componentes do sistema em questão:

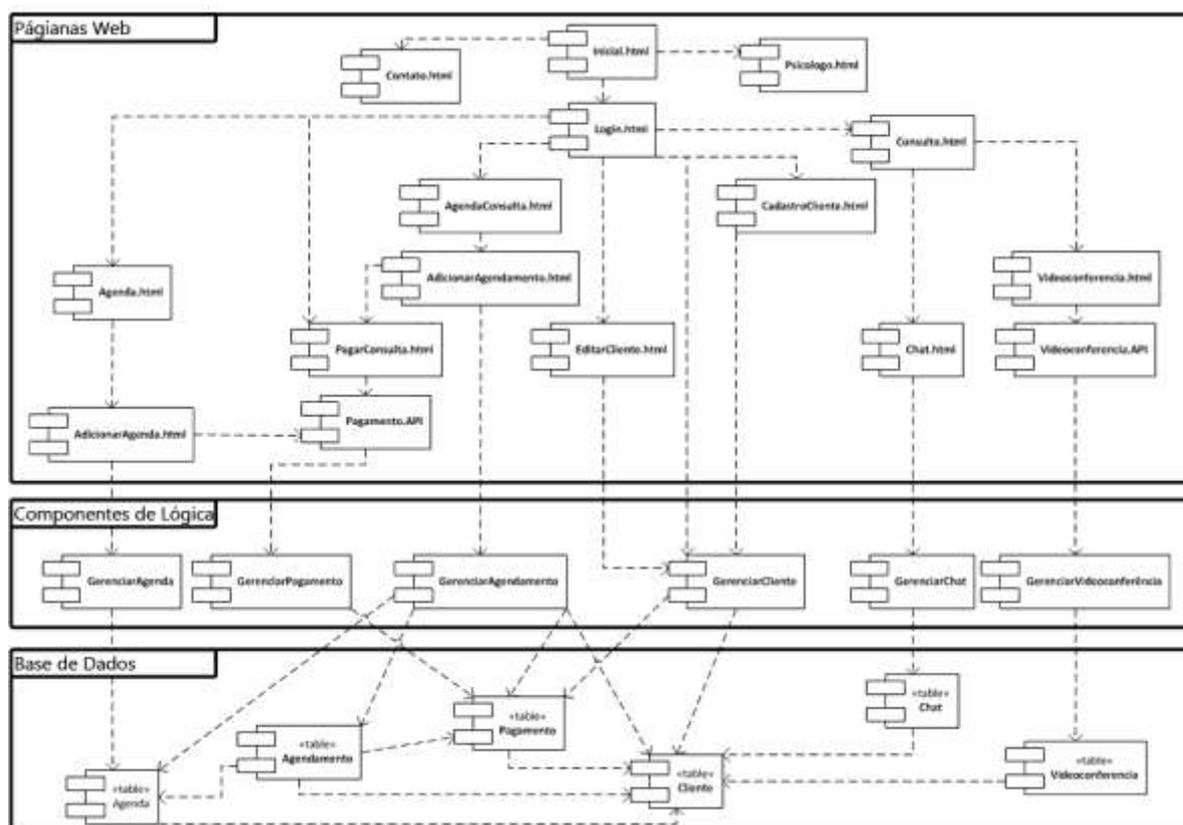


Figura 17 – Diagrama de Componentes – Fonte: Documentação do Sistema de Atendimento Psicológico

Com base nesse diagrama é possível identificar as funções de dados do Sistema de Atendimento Psicológico. É possível ver, que a parte de baixo do Diagrama representa a base de dados do sistema, logo ela será utilizada para identificação das funções de dados. O primeiro passo é identificar as funções do Tipo Dado, o Quadro 1 apresenta as regras para identificação das funções de dados do CPM.

Quadro 1 – Regras para identificação das funções de dados

Regras de Identificação	A regra se aplica?
1. Identifique todos os dados ou informações de controle logicamente relacionados e reconhecidos pelo usuário dentro do escopo da contagem.	<ul style="list-style-type: none"> • Agenda • Agendamento • Pagamento • Cliente • Chat • Videoconferência
2. Exclua entidades que não são mantidas por qualquer aplicação.	<ul style="list-style-type: none"> • Chat • Videoconferência
3. Agrupe entidades relacionadas que são entidades dependentes.	Não existem entidades desse tipo.
4. Exclua as entidades referidas como	Não existem entidades desse tipo.

Dados de Código.	
5. Exclua entidades que não possuem atributos reconhecidos pelo usuário	Não existem entidades desse tipo.
6. Remova as entidades associativas que contém atributos adicionais não requeridos pelo usuário e entidades associativas que contém somente chaves estrangeiras; agrupe os atributos chave estrangeira com as entidades primárias.	Não existem entidades desse tipo.

Após aplicação das regras de identificação das funções de dados, de acordo com o Quadro 1 foram identificados funções de dados: Cliente, Agenda, Pagamento e Agendamento. O segundo passo da contagem é classificar as funções de dados. O Quadro 2 apresenta as regras para classificação das funções de dados.

Quadro 2 – Regras para classificação das funções de dados

Regra de classificação	A regra se aplica?
1. Classificar como ALI se os dados são mantidos pela aplicação sendo contada (Sistema de Atendimento Psicológico <i>online</i>).	As funções Cliente, Agendamento, Agenda e Pagamento são mantidas pela aplicação.
2. Classificar como AIE se: <ul style="list-style-type: none"> • É Referenciado, mas não é mantido pela aplicação sendo contada. • É identificado como ALI em uma ou mais outras aplicações. 	Todas as funções de dados identificadas foram classificadas como ALI, logo não existem AIEs.

A análise mostra que as funções de dados são todas classificadas como ALI, pois são mantidas dentro da fronteira da aplicação, por processos elementares. O próximo passo para a contagem é identificar e contar os Tipos de Dados (TDs) de cada uma das funções de dados identificadas. Nesse passo é necessário avaliar cada atributo relacionado a cada um dos ALIs, e então determinar se o atributo é um Tipo de Dado. Será necessário analisar as telas disponíveis para identificação dos TDs.

O terceiro e o quarto passo será realizado uma vez para cada um dos ALI encontrados. Para o ALI Cliente será utilizada a Tela de Cadastro de Clientes no Sistema de Atendimento Psicológico Online. A Figura 18 apresenta a tela de Cadastro.

Figura 18 – Tela de Cadastro de Cliente – Sistema de Atendimento Psicológico

As regras apresentadas no Quadro 3 devem ser seguidas para identificação dos TDs.

Quadro 3 – Regras para identificação dos Tipos de Dados do ALI Cliente

Regra para identificação de TD	A regra se aplica?
1. Conte um TD para cada atributo único reconhecido pelo usuário, não repetido, mantido ou recuperado de uma função de dados através da execução de todos os processos elementares dentro do escopo da contagem.	<ul style="list-style-type: none"> • Nome • Data de Nascimento • Email • Complemento
2. Conte somente aqueles TD sendo usados pela aplicação sendo medida quando duas aplicações mantem ou referenciam a mesma função de dado.	Não existem entidades desse tipo.
3. Conte um TD para cada atributo requerido pelo usuário para estabelecer relacionamento com outra função de dados	Não existem atributos desse tipo.
4. Revise os atributos relacionados para determinar se eles são agrupados e	Não existem atributos desse tipo.

contados como um único TD ou se eles são contados como múltiplos TD; o agrupamento dependerá de como os processos elementares usam os atributos dentro da aplicação.	
--	--

Após a análise conclui-se que o número de TDs para o ALI Cliente é 4. O quarto passo é a contagem dos tipos de registro para o ALI Cliente.

Quadro 4 – Regras para identificação de Tipos de Registro TR do ALI Cliente.

1. Conte um TR para cada função de dados (isto é, por default, cada função de dados tem um subgrupo de TD para ser contado como um TR).	Conte um TR para o ALI Cliente.
2. Conte um TR adicional para cada subgrupo lógico de TD a seguir (dentro da função de dados) que contém mais do que um TD: <ul style="list-style-type: none"> • Entidade associativa com atributos não chave • Subtipo (subtipo diferente do primeiro tipo) • Entidade associativa, em um relacionamento diferente de 1-1 mandatório. 	<p>Não existem entidades desse tipo.</p> <p>Não existem entidades desse tipo.</p> <p>Não existem entidades desse tipo.</p>

Após essa análise já se sabe o total de TD e TR, logo, se pode definir a complexidade da função de dado e calcular os PFs não ajustados.

Quadro 5 – Quantidade de TDs e TRs do ALI Cliente

TR-Tipo de Registro	TD-Tipo de Dado
<ul style="list-style-type: none"> • Grupo de informações de Cliente 	<ul style="list-style-type: none"> • Nome. • Data de Nascimento. • E-mail. • Complemento.
Total:1 TR	Total:4TDs

De acordo com a Tabela 2 apresentada na seção 2.2.6.2 – Funções de Dados que definem a complexidade funcional, o ALI Cliente é classificado com complexidade baixa. Dando continuidade ao passo 4, agora é necessário contar os TDs e os TR para a o ALI Agenda. O Quadro 6 apresenta a identificação dos TD para o ALI Agenda.

Quadro 6 – Contagem de TD do ALI Agenda

Regra para identificação de TD	A regra se aplica?
1. Conte um TD para cada atributo único reconhecido pelo usuário, não repetido, mantido ou recuperado de uma função de dados através da execução de todos os processos elementares dentro do escopo da contagem.	Não é possível identificar esse tipo de informação, pois não existe documentação disponível para tanto.
2. Conte somente aqueles TD sendo usados pela aplicação sendo medida quando duas aplicações mantem ou referenciam a mesma função de dado.	Não é possível identificar essa informação a partir da documentação disponível.
3. Conte um TD para cada atributo requerido pelo usuário para estabelecer relacionamento com outra função de dados	Não é possível identificar essa informação a partir da documentação disponível.
4. Revise os atributos relacionados para determinar se eles são agrupados e contados como um único TD ou se eles são contados como múltiplos TD; o agrupamento dependerá de como os processos elementares usam os atributos dentro da aplicação.	Não é possível identificar essa informação a partir da documentação disponível.

Não foi possível identificar os tipos de dado do ALI agenda, uma vez que não existe, na documentação disponível, uma tela pra o Cadastro de Agenda no Sistema. Também não existe documentação detalhada da base de dados, logo não é possível identificar os tipos de dados da função.

O Quadro 7 apresenta as regras de contagem de TR aplicadas ao ALI Agenda.

Quadro 7 – Contagem de TRs para o ALI Agenda.

Regras de contagem	A regra se aplica?
1. Conte um TR para cada função de dados (isto é, por default, cada função de dados tem um subgrupo de TD para ser contado como um TR).	Conte um TR para o ALI Agenda.
2. Conte um TR adicional para cada subgrupo lógico de TD a seguir (dentro da função de dados) que contém mais do que um TD: <ul style="list-style-type: none"> • Entidade associativa com atributos não chave • Subtipo (subtipo diferente do 	<p>Não existem entidades desse tipo.</p> <p>Não existem entidades desse tipo.</p>

primeiro tipo) <ul style="list-style-type: none"> Entidade associativa, em um relacionamento diferente de 1-1 mandatório. 	Não existem entidades desse tipo.
---	-----------------------------------

Após essa análise já se sabe o total de TD e TR, logo, se pode definir a complexidade da função de dado e calcular o PF não ajustados. O Quadro 8 apresenta o resultado obtido.

Quadro 8 – Tipos de Dados e Tipos do Registro do ALI Agenda.

TR-Tipo de Registro	TD-Tipo de Dado
<ul style="list-style-type: none"> Grupo de informações de controle de Agenda. 	<ul style="list-style-type: none"> Não pode ser identificado
Total:1 TR	Total:0TDs

Segundo as regras contidas no CPM um ALI que possui um tipo de Registro e até 19 tipos de Dados é de complexidade baixa. Ainda nos passos 3 a 4, agora serão identificados os TRs e TDs dos ALIs Agendamento e Pagamento. O Quadro 9 apresenta as regras de identificação dos tipos de dados aplicadas ao ALI Agendamento.

Quadro 9 – Identificação de TDs para o ALI Agendamento.

Regra para identificação de TD	A regra se aplica?
1. Conte um TD para cada atributo único reconhecido pelo usuário, não repetido, mantido ou recuperado de uma função de dados através da execução de todos os processos elementares dentro do escopo da contagem.	Não é possível identificar esse tipo de informação, pois não existe documentação disponível para tanto.
2. Conte somente aqueles TD sendo usados pela aplicação sendo medida quando duas aplicações mantem ou referenciam a mesma função de dado.	Não é possível identificar essa informação a partir da documentação disponível.
3. Conte um TD para cada atributo requerido pelo usuário para estabelecer relacionamento com outra função de dados	Não é possível identificar essa informação a partir da documentação disponível.
4. Revise os atributos relacionados para determinar se eles são agrupados e contados como um único TD ou se eles são contados como múltiplos TD; o agrupamento dependerá de	Não é possível identificar essa informação a partir da documentação disponível.

como os processos elementares usam os atributos dentro da aplicação.	
--	--

Não foi possível identificar os tipos de dado do ALI Agendamento, uma vez que não existe uma tela para o Cadastro de Agendamento no Sistema. Também não existe documentação detalhada da base de dados, logo não é possível identificar os tipos de dados da função. O Quadro 10 apresenta as regras de contagem de TRs aplicadas ao ALI Agendamento.

Quadro 10 – Regras para Identificação de TRs do ALI Agendamento.

Regras de contagem	A regra se aplica?
1. Conte um TR para cada função de dados (isto é, por default, cada função de dados tem um subgrupo de TD para ser contado como um TR).	Conte um TR para o ALI Agendamento.
2. Conte um TR adicional para cada subgrupo lógico de TD a seguir (dentro da função de dados) que contém mais do que um TD: <ul style="list-style-type: none"> • Entidade associativa com atributos não chave • Subtipo (subtipo diferente do primeiro tipo) • Entidade associativa, em um relacionamento diferente de 1-1 mandatório. 	<p>Não existem entidades desse tipo.</p> <p>Não existem entidades desse tipo.</p> <p>Não existem entidades desse tipo.</p>

Após essa análise já se sabe o total de TD e TR, logo, se pode definir a complexidade da função de dado e calcular o PF não ajustados. O Quadro 11 mostra o resultado da avaliação.

Quadro 11 – Resultado identificação TDs e TRs.

TR-Tipo de Registro	TD-Tipo de Dado
<ul style="list-style-type: none"> • Grupo de informações de controle de Agendamento. 	<ul style="list-style-type: none"> • Não pode ser identificado
Total:1 TR	Total:0TD

Segundo as regras contidas no CPM um ALI que possui um tipo de Registro e até 19 tipos de Dados é de complexidade baixa. Dando sequência a contagem, o Quadro 12 apresenta as regras de identificação de TDs aplicadas ao ALI Pagamento.

Quadro 12 – Regras de identificação

Regra para identificação de TD	A regra se aplica?
1. Conte um TD para cada atributo único reconhecido pelo usuário, não repetido, mantido ou recuperado de uma função de dados através da execução de todos os processos elementares dentro do escopo da contagem.	Não é possível identificar esse tipo de informação, pois não existe documentação disponível para tanto.
2. Conte somente aqueles TD sendo usados pela aplicação sendo medida quando duas aplicações mantem ou referenciam a mesma função de dado.	Não é possível identificar essa informação a partir da documentação disponível.
3. Conte um TD para cada atributo requerido pelo usuário para estabelecer relacionamento com outra função de dados	Não é possível identificar essa informação a partir da documentação disponível.
4. Revise os atributos relacionados para determinar se eles são agrupados e contados como um único TD ou se eles são contados como múltiplos TD; o agrupamento dependerá de como os processos elementares usam os atributos dentro da aplicação.	Não é possível identificar essa informação a partir da documentação disponível.

Não foi possível identificar os tipos de dado do ALI Agendamento, uma vez que não existe uma tela para o Cadastro de Agendamento no Sistema. Também não existe documentação detalhada da base de dados, logo não é possível identificar os tipos de dados da função. O Quadro 13 apresenta as regras de identificação dos Tipos de Registro aplicadas ao ALI Pagamento.

Quadro 13 – Regras pra identificação de TRs aplicada ao ALI Pagamento

Regras de contagem	A regra se aplica?
1. Conte um TR para cada função de dados (isto é, por default, cada função de dados tem um subgrupo de TD para ser contado como um TR).	Conte um TR para o ALI Pagamento.
2. Conte um TR adicional para cada subgrupo lógico de TD a seguir	

(dentro da função de dados) que contém mais do que um TD:	Não existem entidades desse tipo.
<ul style="list-style-type: none"> Entidade associativa com atributos não chave 	Não existem entidades desse tipo.
<ul style="list-style-type: none"> Subtipo (subtipo diferente do primeiro tipo) 	Não existem entidades desse tipo.
<ul style="list-style-type: none"> Entidade associativa, em um relacionamento diferente de 1-1 mandatório. 	Não existem entidades desse tipo.

Após essa análise já se sabe o total de TD e TR, logo, se pode definir a complexidade da função de dado e calcular o PF não ajustados. O resultado da identificação de TDs e TRs é apresentado no Quadro 14.

Quadro 14 – Resultado da identificação de TDs e TRs

TR-Tipo de Registro	TD-Tipo de Dado
<ul style="list-style-type: none"> Grupo de informações de controle de Pagamento. 	<ul style="list-style-type: none"> Não pode ser identificado
Total:1 TR	Total:0TD

De acordo com o manual de práticas de contagem um ALI com 1 tipo de registro e até 19 tipos de dados é classificado com complexidade baixa, como foi não foi possível identificar a quantidade de tipos de dados, então assume-se o valor 0.

Após a identificação das funções de dados de sua classificação como ALI ou AIE e de determinada a complexidade de cada uma delas é possível atribuir um peso em pontos de função para cada um delas. A tabela X apresenta o resultado da contagem das funções de dados.

Tabela 12 – Resultado da Contagem das Funções de Dados.

Função	TDs	TRs	Complexidade	Contribuição
ALI Cliente	4	1	Baixa	7
ALI Agenda	0 até 19	1	Baixa	7
ALI Agendamento	0 até 19	1	Baixa	7
ALI Pagamento	0 até 19	1	Baixa	7
Total				28

O tamanho funcional das funções de dados é dado pela quantidade de Funções de dados de uma determinada complexidade multiplicada pelo peso que a mesma representa em pontos de função. Nesse caso, todos os ALIs identificados foram classificados com

complexidade baixa, ou seja, o tamanho funcional, de acordo com o com, é igual a 28 ($4 * 7$), onde 4 é o número de ALI de complexidade baixa e 7 é o peso em pontos de função desse tipo de função.

Após a contagem das funções de dados, inicia-se a contagem das funções de transação, ou seja, a contagem dos processos elementares. Os passos para contagem de Funções do tipo transação são:

1. Identificar os processos elementares solicitados pelo usuário;
2. Determinar processos elementares únicos;
3. Classifica-los em SE, EE, ou CE;
4. Contar os TDs do processo elementar;
5. Contar os ALRs;
6. Determinar complexidade funcional;
7. Determinar tamanho funcional.

Para contagem das funções de transação podem ser utilizados: Diagrama de Caso de Uso, Diagramas de Sequência, Casos de Uso Expandidos, Lista de Requisitos e Telas do sistema. No caso do Sistema de Atendimento Psicológico Online, conta-se com: Lista de Requisitos, alguns Casos de Uso Expandido, alguns Diagramas de Sequencia, algumas Telas. A seguir está apresentada a Lista de Requisitos do sistema de Atendimento Psicológico Online:

- R1: Gerenciar Cliente
 - C1.1: Adicionar Cliente
 - C1.2: Pesquisar Cliente
 - C1.3: Alterar Cliente
- R2: Gerenciar Agenda
 - C2.1: Adicionar Agenda
 - C2.2: Remover Agenda
 - C2.3: Alterar Agenda
 - C2.4: Pesquisar Agenda
- R3: Agendamento de Consulta
 - C3.1: Adicionar Agendamento
 - C3.2: Remover Agendamento
 - C3.3: Alterar Agendamento
 - C3.4: Pesquisar Agendamento

- C3.5: Confirmar Agendamento
- C3.6: Cancelar Agendamento
- R4: Pagamento da Consulta
 - C4.1: Confirmar pagamento de consulta
 - C4.2: Pagar consulta
 - C4.3: Cancelar Pagamento

Todos os requisitos apresentados na lista de requisitos devem ser avaliados, identificados como processos elementares, ou não, e então, devem ser classificados como Entrada Externa (EE), Saída Externa (SE) ou Consulta Externa (CE). Após isso, as funções EE, SE, ou CE devem ser classificadas quanto a sua complexidade. O Quadro 15 apresenta a identificação de o requisito Adicionar Cliente, como um processo elementar.

Processo elementar Adicionar Cliente

O requisito Adicionar Cliente tem como objetivo principal inserir dados de um cliente no sistema, através da transação denominada Cadastro.

Quadro 15 - Identificar processo elementar – Requisito Adicionar Cliente

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim
2. Constitui uma transação completa?	Sim
3. É autocontido.	Sim
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Adicionar Cliente é um processo elementar.

O requisito Adicionar Cliente foi considerado um processo elementar, pois ele mantém o ALI Cliente, através de lógica de processamento. Agora é necessário verificar se o processo elementar Adicionar Cliente é único dentro do projeto sendo contado.

Quadro 16 – Determinar se o processo Adicionar Cliente é único.

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de TDs e	Não é possível identificar tal informação.
2. Requerem o mesmo conjunto de TRs e	Não é possível identificar tal informação.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não. Adicionar um Cliente tem um conjunto diferente de lógica de processamento em relação ao processo Editar um cadastro de Cliente.
4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	A função de Adicionar um Cliente consiste em um conjunto único de lógica de processamento que deixa o sistema em estado consistente.

Após análise pode-se perceber que o processo elementar Adicionar Cliente na aplicação sendo contada. O próximo passo é classificar o processo elementar como EE, SE, ou CE. O Quadro 17 apresenta as regras de classificação do processo elementar, para EE.

Quadro 17 – Classificar processo elementar.

Regras de contagem par a EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mais ALIs ou • Alterar o comportamento da aplicação 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como dados de controle. Não.
2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Sim. Informações de Controle de Cliente entram na fronteira da aplicação.

O processo elementar Adicionar Cliente envia dados para dentro da fronteira da aplicação para manter o ALI Cliente, logo, essa função transacional é considerada como uma Entrada Externa. Se o processo é uma EE, logo não pode ser um SE, ou CE. O próximo passo para a contagem é identificar os TDs e ALRs relacionados ao processo elementar. O quadro 18 apresenta regra para contagem de ALR.

Quadro 18 – Contagem de ALR

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Cliente é mantido pelo processo elementar.

Foi identificado um ALR para a transação Adicionar Cliente. O processo elementar envia dados para dentro da fronteira da aplicação para manter o ALI Cliente. Para identificação dos Tipos de Dados da função Adicionar Cliente, será considerada a Figura 18, apresentada anteriormente. O Quadro 19 apresenta as regras de contagem do TDs de um processo elementar.

Quadro 19 – Identificação de Tipos de Dados

Regras de contagem	A regra se aplica
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.	<ul style="list-style-type: none"> • Nome Cliente • Data de Nascimento • E-mail • Complemento
2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.	Mensagem de Sucesso para o usuário.
3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-	Botão de comando Cadastrar.

la(s).	
<p>4. Não contar os seguintes itens como TDs:</p> <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais como: atributos de data e hora. • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	<p>Nenhum dos literais existentes na tela de entrada das Informações de Controle é contado.</p> <p>Não existe nenhum item deste tipo.</p>

Após a análise foram identificados 4TDs para a transação Adicionar Cliente. Após a identificação dos tipos de Dados e Arquivos Referenciados é possível identificar a complexidade da Entrada Externa. O quadro 12 apresenta o resultado da contagem de TDs a ARLs.

Quadro 22 – Identificar processo elementar – Pesquisar Cliente

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim.
2. Constitui uma transação completa?	Sim. A Consulta de um Cliente com a exibição dos seus dados constitui uma transação completa no sistema.
3. É autocontido.	Sim. Esta atividade é significativa e autocontida.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado consistente.
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Pesquisar Cliente satisfaz todos os critérios acima.

Pesquisar Cliente foi considerado um processo elementar, agora é necessário determinar se esse processo elementar é único dentro da aplicação sendo contada. O Quadro 23 apresenta as regras para determinação de processos elementares únicos aplicadas ao processo elementar Pesquisar Cliente.

Quadro 23 – Determinar processo elementar único

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de TDs e	Não, pesquisar um cliente requer que apenas o nome do mesmo seja informado.
2. Requerem o mesmo conjunto de TRs e	Não, pesquisar um cliente requer que apenas o nome seja informado, as informações retornadas são apenas para visualização.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não.

4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	Os requisitos funcionais do usuário consideram que há uma única função. A função Consultar um Cliente não foi dividida em dois processos elementares.
--	---

Pesquisar Cliente é classificado como um processo elementar único. O próximo passo é verificar se o processo elementar é um EE, SE ou CE. As regras para verificação são apresentadas no Quadro 24.

Quadro 24 – Classificar como EE, SE ou CE.

Regras de contagem para EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mas ALIs ou • Alterar o comportamento da aplicação 	Não. Não.
2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Não.
3. O processo elementar realiza cálculos?	Não.
4. O processo elementar deriva dados?	Não.
5. O processo elementar mantém ou atualiza um ou mais ALI ou AIE?	Não.
6. O processo elementar modifica o comportamento do sistema?	Não.

O processo elementar não é considerado uma EE, pois não atualiza ou mantém nenhum ALI ou AIE no sistema, não é considerado um SE, pois não deriva dados ou realiza cálculos, logo é considerado uma Consulta Externa (CE). A seguir é necessário contar os TDs e ALRs, o Quadro 25 apresenta a contagem de ALR para o requisito Pesquisar Cliente.

Quadro 25 – Contagem de ALR.

Regras de Contagem de ALR	A regra se aplica?
---------------------------	--------------------

Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Cliente é acessado para apresentar as informações do cliente.
--	---

Foi identificado, um ALR relacionado ao processo elementar Pesquisar Cliente. O Quadro 26 apresenta a contagem dos TDs desse processo elementar.

Quadro 26 – Contagem de TDs.

Regras de contagem	A regra se aplica
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.	<ul style="list-style-type: none"> • Nome • Data de Nascimento • E-mail • Complemento
2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.	Mensagem de alerta.
3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).	Não existem entidades desse tipo.
4. Não contar os seguintes itens como TDs: <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais como: atributos de data e hora. 	<p>Nenhum dos literais existentes na tela de entrada das Informações de Controle é contado.</p> <p>Não existe nenhum item deste tipo.</p>

<ul style="list-style-type: none"> • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. 	Não existe nenhum item deste tipo.
<ul style="list-style-type: none"> • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. 	Não existe nenhum item deste tipo.
<ul style="list-style-type: none"> • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. 	Não existe nenhum item deste tipo.
<ul style="list-style-type: none"> • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	Não existe nenhum item deste tipo.

Foram identificados 5 tipos de Dados para a função Pesquisar Cliente. O resultado do número de TDs e ALRs é apresentado no Quadro 27.

Quadro 27 – Resultado da Contagem de ALRs e TDs.

ALR	Dado Elementar Referenciado
<ul style="list-style-type: none"> • ALI Cliente. 	<ul style="list-style-type: none"> • Nome • Data de Nascimento • E-mail • Complemento • Mensagem de alerta.
Total: 1 ALR	Total: 5 TDs

Depois de encontrado o número de ALR e TDs a complexidade (baixa, média ou alta) é determinada de acordo com a Tabela 6 apresentada na seção 2.2.6.3. Funções do Tipo Transação. De acordo com a Tabela 6 a complexidade dessa CE é baixa. O tamanho funcional

é definido de acordo com a Tabela 7 apresentada na seção 2.2.6.3 Funções do Tipo Transação. De acordo com Tabela 7, a Consulta Externa Pesquisar Cliente contribui com 3 pontos de função para o tamanho funcional da aplicação.

O Caso de Uso a ser avaliado é o Editar Cliente, o primeiro passo é verificar se ele constitui um processo elementar. O Quadro 28 apresenta as regras de identificação aplicadas a esse caso de uso.

Quadro 28 – Identificar processo elementar.

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim
2. Constitui uma transação completa?	Sim
3. É autocontido.	Sim
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Editar Cliente é um processo elementar.

A análise mostra que o caso de uso Editar Cliente é considerado um processo elementar. O Quadro 29 apresenta as regras para determinar se o processo elementar é único comparado com outros processos da mesma aplicação.

Quadro 29 – Determinar processo elementar único.

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de TDs e	Não é possível identificar tal informação.
2. Requerem o mesmo conjunto de TRs e	Não é possível identificar tal informação.

3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não. Editar um Cliente tem um conjunto diferente de lógica de processamento em relação ao processo Adicionar um Cliente.
4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	A função de Editar um Cliente consiste em um conjunto único de lógica de processamento que deixa o sistema em estado consistente.

O processo elementar Editar Cliente foi considerado único dentro da aplicação sendo contada. O próximo passo na contagem é determinar se ele é um EE, SE ou CE na aplicação. O Quadro 30 apresenta as regras para identificação de EE, SE ou CE.

Quadro 30 – Classificar como CE, SE ou EE.

Regras de contagem par a EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mais ALIs ou • Alterar o comportamento da aplicação 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como dados de controle para manter o ALI Cliente. Não.
2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Sim. Informações de Controle de Cliente entram na fronteira da aplicação.

O processo elementar Editar Cliente envia dados para dentro da fronteira da aplicação para manter/atualizar o ALI Cliente, logo, essa função transacional é considerada como uma Entrada Externa. O próximo passo é contar os ALRs e os TDs, o Quadro 31 apresenta a contagem de ALRs.

Quadro 31 – Contagem de ALRs.

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Cliente é atualizado/mantido pelo processo elementar.

Foi identificado um ALR para a transação Editar Cliente, o Quadro 32 apresenta contagem dos TDs para o processo elementar Editar Cliente.

Quadro 32 – Contagem de TDs para Editar Cliente.

Regras de contagem	A regra se aplica
<p>1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.</p>	<ul style="list-style-type: none"> • Nome Cliente • Data de Nascimento • E-mail • Complemento
<p>2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.</p>	<p>Mensagem de Sucesso para o usuário.</p>
<p>3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).</p>	<p>Botão de comando Editar.</p>
<p>4. Não contar os seguintes itens como TDs:</p> <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais como: atributos de data e hora. • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. 	<p>Nenhum dos literais existentes na tela de entrada das Informações de Controle é contado.</p> <p>Não existe nenhum item deste tipo.</p> <p>Não existe nenhum item deste tipo.</p> <p>Não existe nenhum item deste tipo.</p>

<ul style="list-style-type: none"> • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	<p>Não existe nenhum item deste tipo.</p> <p>Não existe nenhum item deste tipo.</p>
--	---

Foram identificados 4 TDs para a transação Adicionar Cliente. O Quadro 33 apresenta o resultado da contagem de ALRs e TDs.

Quadro 33 – Resultado da Contagem de TDs e ALRs.

ALR	Dado Elementar Referenciado
<ul style="list-style-type: none"> • ALI Cliente. 	<ul style="list-style-type: none"> • Nome • Data de Nascimento • E-mail • Complemento • Botão Editar • Mensagem de sucesso.
Total:1ALR	Total:6TDs

Após a identificação dos tipos de Dados e Arquivos Referenciados é possível identificar a complexidade da Entrada Externa. Depois de encontrado o número de ALR e TDs a complexidade (baixa, média ou alta) é determinada de acordo com a Tabela 4 apresentada na seção 2.2.6.3. Funções do Tipo Transação. De acordo com a Tabela 4 a complexidade dessa EE é baixa. O tamanho funcional é definido de acordo com a Tabela 7 apresentada na seção 2.2.6.3 Funções do Tipo Transação. De acordo com Tabela 7, a EE Editar Cliente contribui com 3 pontos de função para o tamanho funcional da aplicação.

A contagem de o requisito Gerenciar Cliente foi concluída, a seguir será feita a contagem de o requisito Gerenciar Agenda. Para contagem do Caso de Uso Adicionar Agenda, será utilizado o Diagrama de Sequencia de Projeto, apresentado a seguir, na Figura 19.

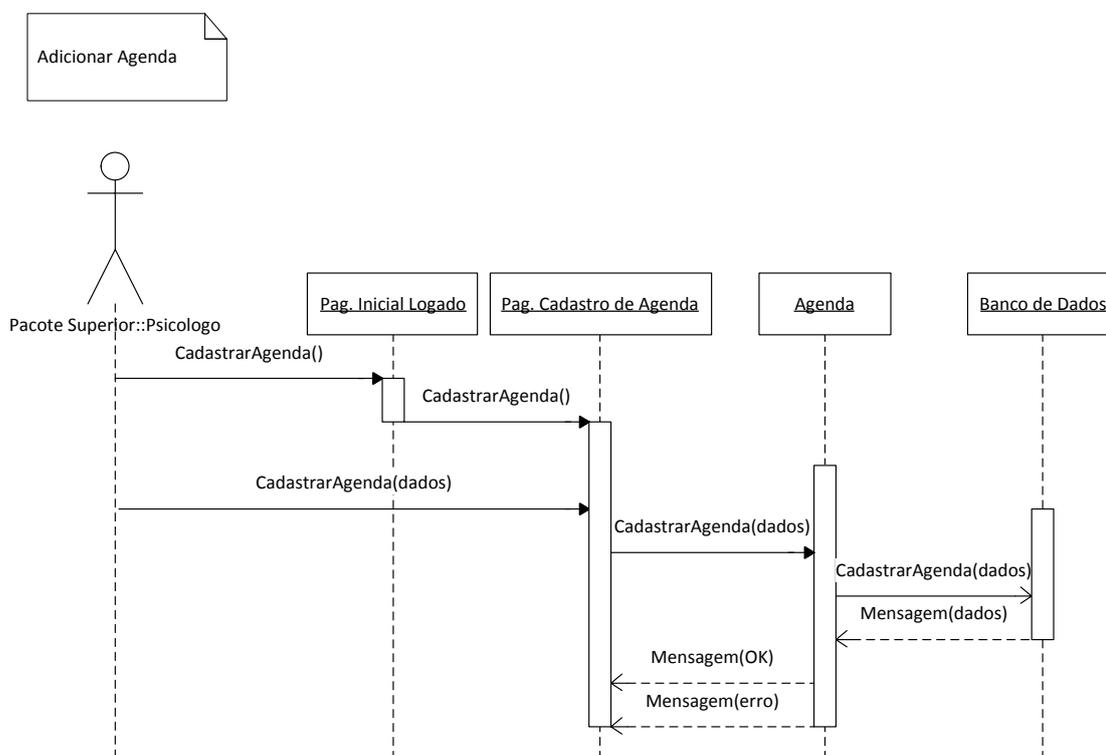


Figura 19 – Diagrama de Sequência de projeto – Adicionar Agenda

O Quadro 34 apresenta as regras de identificação de processo elementar.

Quadro 34 – Identificação do processo elementar.

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaz todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim, a adição de uma agenda é significativa para o usuário.
2. Constitui uma transação completa?	Sim. A adição de uma agenda constitui uma transação completa.
3. É autocontido.	Sim.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado consistente quando uma agenda é inserida.
5. Identificar um processo elementar para cada unidade de atividade identificada	Adicionar uma agenda constitui um processo elementar de acordo com os critérios acima.

que satisfaça todos os critérios acima.

Após a identificação do processo elementar é necessário determinar se o processo elementar é único de acordo com o Quadro 35.

Quadro 35 – Identificação de processo elementar único.

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de TDs e	Sim. Incluir uma Agenda requer o mesmo conjunto de TDs que Atualizar uma Agenda.
2. Requerem o mesmo conjunto de TRs e	Sim. Adicionar uma Agenda requer o mesmo conjunto de TRs que Atualizar uma Agenda.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não. Adicionar Agenda tem um conjunto diferente de lógica de processamento em relação ao processo Atualizar uma Agenda.
4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	Os requisitos funcionais do usuário consideram que há uma única função. A função Adicionar uma Agenda não foi dividida em dois processos elementares.

Depois de identificar o processo elementar e determina-lo como único para a aplicação sendo contada é necessário classifica-lo como umas funções transacionais: EE, SE ou CE. A classificação é feita de acordo com o Quadro 36.

Quadro 36 – Determinar tipo de transação.

Regras de contagem par a EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mas ALIs ou • Alterar o comportamento da aplicação 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como dados de controle. Não.

2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Sim. Informações de Controle de Agenda entram na fronteira da aplicação.
---	--

O processo elementar Adicionar Agenda envia dados para dentro da fronteira da aplicação através de lógica de processamento e é considerada uma Entrada Externa (EE). Os próximos passos são a identificação dos ALRs e a identificação dos TDs. Como apresentado no Quadro 37 foi identificado um ALR relacionado ao processo elementar.

Quadro 37 – Contagem de ALR

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Agenda é mantido pelo processo elementar.

Quadro 38 – Contagem de TDs

Regras de contagem	A regra se aplica
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.	Não é possível identificar tal informação.
2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.	Mensagem de retorno.
3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).	Não é possível identificar tal informação com base na documentação disponível.

<p>4. Não contar os seguintes itens como TDs:</p> <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais como: atributos de data e hora. • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	<p>Nenhum dos literais existentes na tela de entrada das Informações de Controle é contado.</p> <p>Não existe nenhum item deste tipo.</p>
--	---

Não foi possível identificar todos os tipos de dados relacionados à função transacional Adicionar Agenda. Nesse caso, como se trata de uma aplicação pequena, é muito provável que o número de TDs da função em questão não passe de 15, que é o limite para determinar a complexidade.

Quadro 39 - Resultado contagem TDs e ALRs

ALR	Tipo de Dados
<ul style="list-style-type: none"> • ALI Agenda. 	<ul style="list-style-type: none"> • Mensagem de retorno.
Total: 1 ALR	Total: 1 TDs

Depois de encontrado o número de ALR e TDs a complexidade (baixa, média ou alta) é determinada de acordo com a Tabela 4 apresentada na seção 2.2.6.3. Funções do Tipo Transação. De acordo com a Tabela 4 a complexidade dessa EE é baixa. O tamanho funcional é definido de acordo com a Tabela 5 apresentada na seção 2.2.6.3 Funções do Tipo Transação. De acordo com Tabela 5 a Entrada Externa Adicionar Agenda contribui com 3 pontos de função para o tamanho funcional da aplicação.

O próximo caso de uso a ser avaliado é o Alterar Agenda. Esse requisito tem como ação principal alterar dados de uma Agenda no sistema. O Quadro 40 apresenta a

Quadro 40 – Identificar processo elementar

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim, a Alteração de uma agenda é significativa para o usuário.
2. Constitui uma transação completa?	Sim. A Alteração de uma agenda constitui uma transação completa.
3. É autocontido.	Sim.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado consistente quando uma agenda é alterada.
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Alterar uma agenda constitui um processo elementar de acordo com os critérios acima.

Após a análise foi concluído que o Alterar Agenda é considerado como um processo elementar, as regras do Quadro 41 tem o intuito de verificar se o processo elementar Alterar Agenda é único dentro da aplicação.

Quadro 41 – Identificar unicidade do processo elementar.

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de TDs e	Sim. Alterar uma Agenda requer o mesmo conjunto de TDs que Adicionar uma Agenda.
2. Requerem o mesmo conjunto de TRs e	Sim. Alterar uma Agenda requer o mesmo conjunto de TRs que Adicionar uma Agenda.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não. Alterar Agenda tem um conjunto diferente de lógica de processamento em relação ao processo Adicionar uma Agenda.
4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	Os requisitos funcionais do usuário consideram que há uma única função. A função Alterar uma Agenda não foi dividida em dois processos elementares.

O processo elementar foi considerado como único para a aplicação sendo contato, o Quadro 42 apresenta as regras para classificar o processo como EE, SE ou CE.

Quadro 42 -

Regras de contagem par a EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mas ALIs ou • Alterar o comportamento da aplicação 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como dados de controle. Não.
2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Sim. Informações de Controle de Agenda entram na fronteira da aplicação.

O processo elementar Alterar Agenda envia dados para dentro da fronteira da aplicação através de lógica de processamento e é considerada uma Entrada Externa (EE). O Quadro 43 mostra a contagem dos ALRs.

Quadro 43 – Contagem ALRs.

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Agenda é mantido pelo processo elementar.

Foi identificado um ALR relacionado ao processo elementar Alterar Agenda. O Quadro 44 apresenta a contagem dos TDs.

Quadro 44 – Contagens de TDs.

Regras de contagem	A regra se aplica
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.	Não é possível identificar tal informação.
2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.	Não é possível identificar tal informação.
3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).	Não é possível identificar tal informação com base na documentação disponível.
4. Não contar os seguintes itens como TDs: <ul style="list-style-type: none"> • Literais, tais como: títulos de 	

<p>relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos.</p> <ul style="list-style-type: none"> • <i>Application generated stamps</i>, tais como: atributos de data e hora. • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	<p>Nenhum dos literais existentes na tela de entrada das Informações de Controle é contado.</p> <p>Não existe nenhum item deste tipo.</p>
--	---

Não foi possível identificar os tipos de dados relacionados à função transacional Adicionar Agenda. Nesse caso, como se trata de uma aplicação pequena, é muito provável que o número de TDs da função em questão não passe de 15, então, será atribuído valor 0 aos TDs.

Quadro 45 – Resultado contagem ALR e TDs

ALR	Tipo de Dados
-----	---------------

• ALI Agenda.	• Não foi possível identificar os TDs.
Total 1 ALR	Total 0TDs

Depois de encontrado o número de ALR e TDs a complexidade (baixa, média ou alta) é determinada de acordo com a Tabela 4 apresentada na seção 2.2.6.3. Funções do Tipo Transação. De acordo com a Tabela 4 a complexidade dessa EE é baixa. O tamanho funcional é definido de acordo com a Tabela 5 apresentada na seção 2.2.6.3 Funções do Tipo Transação. De acordo com Tabela 5 a Entrada Externa Adicionar Agenda contribui com 3 pontos de função para o tamanho funcional da aplicação.

O próximo Caso de Uso a ser avaliado é o Remover Agenda, que tem como intenção principal excluir uma agenda cadastrada no sistema, o documento utilizado para contagem é a Lista de requisitos. O primeiro passo é verificar se Remover Agenda é classificado como processo elementar, o que é demonstrado no Quadro 46.

Quadro 46 – Identificar Processo Elementar.

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim, a Remoção de uma agenda é significativa para o usuário.
2. Constitui uma transação completa?	Sim. A Remoção de uma agenda constitui uma transação completa.
3. É autocontido.	Sim.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado consistente quando uma agenda é removida.
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Remover uma agenda constitui um processo elementar de acordo com os critérios acima.

Segundo a análise apresentada no Quadro 46, a ato de remover uma agenda no sistema é considerada uma transação completa com significado para o usuário, ou seja, um processo elementar. O Quadro 47 apresenta as regras para determinar se esse processo elementar é único dentro da aplicação sendo contada.

Quadro 47 – Determinar processo elementar único.

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de TDs e	Não.
2. Requerem o mesmo conjunto de TRs e	Não.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não.
4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	Os requisitos funcionais do usuário consideram que há uma única função. A função de Remover agenda não está subdividida em mais funções.

De acordo com a análise o processo elementar Remover Agenda é considerado único dentro da aplicação sendo contada. O Quadro 48 apresenta as regras pra classificação do processo elementar em EE, SE ou CE.

Quadro 48 – Classificação da função.

Regras de contagem par a EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mas ALIs ou • Alterar o comportamento da aplicação 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como dados de controle. Não.
2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Não.

O processo elementar Remover Agenda é considerado uma EE, que Atualiza um ALI. Os Quadros 49 e 50 apresentam respectivamente a contagem dos ALRs e dos TDs relacionados ao processo elementar Remover Agenda.

Quadro 49 – Contar os ALRs

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Agenda é mantido/Atualizado pelo processo elementar.

Quadro 50 – Identificação dos TDs.

Regras de contagem	A regra se aplica
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.	Não é possível identificar tal informação.
2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.	Não é possível identificar tal informação.
3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).	Não é possível identificar tal informação com base na documentação disponível.
4. Não contar os seguintes itens como TDs: <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de 	Nenhum dos literais existentes na tela de entrada das Informações de

<p>atributos.</p> <ul style="list-style-type: none"> • <i>Application generated stamps</i>, tais como: atributos de data e hora. • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	<p>Controle é contado.</p> <p>Não existe nenhum item deste tipo.</p>
---	--

Não foi possível identificar os tipos de dados relacionados à função transacional Adicionar Agenda. Nesse caso, como se trata de uma aplicação pequena, é muito provável que o número de TDs da função em questão não passe de 15, então, será atribuído valor 0 aos TDs.

Quadro 51 – Resultado da Contagem de ALRs e TDs.

ALR	Tipos de Dados
<ul style="list-style-type: none"> • ALI Agenda. 	<ul style="list-style-type: none"> • Não foi possível identificar os TDs.
Total 1 ALR	Total:0TDs

Depois de encontrado o número de ALR e TDs a complexidade (baixa, média ou alta) é determinada de acordo com a Tabela 4 apresentada na seção 2.2.6.3. Funções do Tipo Transação. De acordo com a Tabela 4 a complexidade dessa EE é baixa.

O tamanho funcional é definido de acordo com a Tabela 5 apresentada na seção 2.2.6.3 Funções do Tipo Transação. De acordo com Tabela 5 a Entrada Externa Remover Agenda contribui com 3 pontos de função para o tamanho funcional da aplicação.

O próximo caso de uso a ser avaliado é o Pesquisar Agenda, a documentação utilizada na avaliação é apenas a Lista de Requisitos. O Quadro 52 apresenta a identificação do processo elementar.

Quadro 52 – Identificar processo elementar.

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim, a Consulta de uma agenda é significativa para o usuário.
2. Constitui uma transação completa?	Sim. A Consulta de uma agenda constitui uma transação completa, na qual dados são apresentados ao usuário.
3. É autocontido.	Sim.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado consistente.
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Consultar uma agenda constitui um processo elementar de acordo com os critérios acima.

Pesquisar agenda foi considerado como um processo elementar, agora é necessário classifica-lo como único, segundo as regras apresentadas no Quadro 53.

Quadro 53 – Determinar processo elementar único.

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de	Não.

TDs e	
2. Requerem o mesmo conjunto de TRs e	Não.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não.
4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	Os requisitos funcionais do usuário consideram que há uma única função. A função de Remover agenda não está subdividida em mais funções.

O processo elementar Pesquisar Agenda foi considerado único para a aplicação sendo contada, agora é necessário classifica-lo como umas das funções transacionais: EE, SE ou CE. O Quadro 54 mostra as regras para classificação.

Quadro 54 – Classificar como SE, CE ou EE.

Regras de contagem par a EE	A regra de aplica?
1. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Não.
2. O processo elementar realiza cálculos?	Não.
3. O processo elementar deriva dados?	Não.
4. O processo elementar mantém ou atualiza um ou mais ALI ou AIE?	Não.
5. O processo elementar modifica o comportamento do sistema?	Não.

O processo elementar Consultar Agenda é considerado uma CE, pois, não atualiza ou mantém nenhum ALI ou AIE e também não realiza cálculos para apresentar dados ao usuário. Os Quadros 55, 56 e 57 mostram respectivamente a contagem dos ALRs, TDs e o resultado das duas contagens.

Quadro 55 – Contagem de ALRS.

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Agenda é consultado pelo processo elementar.

Quadro 56 – Contar TDs.

Regras de contagem	A regra se aplica
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.	Não é possível identificar tal informação.
2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.	Não é possível identificar tal informação.
3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).	Não é possível identificar tal informação com base na documentação disponível.
4. Não contar os seguintes itens como TDs: <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais como: atributos de data e hora. 	Não é possível identificar tal informação.

<ul style="list-style-type: none"> • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	<p>Não é possível identificar tal informação..</p> <p>Não é possível identificar tal informação.</p> <p>Não é possível identificar tal informação.</p>
--	--

Não foi possível identificar os tipos de dados relacionados à função transacional Consultar Agenda. Como se trata de uma aplicação pequena, é provável que o número de TDs da função em questão não passe de 15, então, será atribuído valor 0 aos TDs.

Quadro 57 – Resultado Contagem TDs e ALRs.

ALR	Tipos de Dados.
<ul style="list-style-type: none"> • ALI Agenda. 	<ul style="list-style-type: none"> • Não foi possível identificar os TDs.
Total:1 ALR	Total:0TDs

Depois de encontrado o número de ALR e TDs a complexidade (baixa, média ou alta) é determinada de acordo com a Tabela 4 apresentada na seção 2.2.6.3. Funções do Tipo Transação. De acordo com a Tabela 4 a complexidade dessa CE é baixa. O tamanho funcional é definido de acordo com a Tabela 5 apresentada na seção 2.2.6.3 Funções do Tipo Transação.

De acordo com Tabela 5 a Consulta Externa Consultar Agenda contribui com 3 pontos de função para o tamanho funcional da aplicação.

O próximo conjunto de Casos de Uso que serão analisados faz parte dos requisitos: Gerenciar Agendamento e Gerenciar Pagamento. A única documentação disponível para realizar a contagem desses requisitos é a Lista de Requisitos. Possivelmente os casos de uso desses requisitos são mais complexos e exigem uma análise mais detalhada.

O caso de uso Adicionar Agendamento, tem como intuito principal adicionar um agendamento no sistema, provavelmente para realizar um agendamento são utilizados dados dos ALIs Agenda e Cliente, no entanto, pela documentação disponível não é possível perceber isso. O Quadro 58 apresenta as regras para identificação do processo elementar aplicada e esse caso de uso.

Quadro 58 – Identificar processo elementar.

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim
2. Constitui uma transação completa?	Sim.
3. É autocontido.	Sim.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado consistente.
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Adicionar um Agendamento constitui um processo elementar de acordo com os critérios acima.

A função de Adicionar um Agendamento é considerada um processo elementar, segundo as regras de contagem. O Quadro 59 apresenta as regras para definição de processos elementares únicos.

Quadro 59 - Determinar processo elementar único

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	

1. Requerem o mesmo conjunto de TDs e	Não é possível identificar tal informação.
2. Requerem o mesmo conjunto de TRs e	Não é possível identificar tal informação.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não é possível identificar tal informação, mas não outra funcionalidade na lista de requisitos que parece ser semelhante a essa funcionalidade.
4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	A função de Adicionar Agendamento consiste em um conjunto único de lógica de processamento que deixa o sistema em estado consistente.

O processo elementar é considerado único dentro de aplicação. O Quadro 60 apresenta a classificação do processo elementar como EE, SE ou CE.

Quadro 60 – Regras para classificação da função.

Regras de contagem para EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mais ALIs ou • Alterar o comportamento da aplicação 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como dados de controle. Não.
2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Sim. Informações de Controle de Cliente entram na fronteira da aplicação.

O processo elementar é classificado como uma Entrada Externa, já que mantém o ALI agendamento. O processo elementar Adicionar Agendamento Consulta dados do ALI Agenda e insere dados no ALI Agendamento, ou seja está relacionado com ambos. O Quadro 61 apresenta a contagem do ALRs e TDs relacionados ao processo elementar.

Quadro 61 – Contagem de ALR e TDs.

Regras de Contagem de ALR	A regra se aplica?
---------------------------	--------------------

Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Agendamento é mantido pelo processo elementar. Não há informação suficiente para determinar se o processo consulta dados de outro ALI como: Cliente ou Agenda.
Regras de Contagem para TDs	A regra se aplica?
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
<p>4. Não contar os seguintes itens como TDs:</p> <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais como: atributos de data e hora. • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. 	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.

<ul style="list-style-type: none"> • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	
--	--

Não é possível identificar os tipos de dados da função transacional Adicionar Agendamento. Sem a identificação dos tipos de dados da função Adicionar Agendamento, também não é possível determinar sua complexidade. Inferir que a quantidade de TDs é zero nesse caso, não é viável, pois provavelmente Adicionar um Agendamento envolve dados dos ALIs Agenda e Cliente, mas não é possível saber quais são os dados utilizados. Com base na contagem do IFPUG não há como determinar o tamanho funcional se não se sabe a complexidade da função. O Quadro 62 apresenta a avaliação da função Alterar Agendamento, para determinar se ela constitui um processo elementar.

Quadro 62 – Identificar processo elementar

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim
2. Constitui uma transação completa?	Sim.
3. É autocontido.	Sim.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado

	consistente.
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Alterar um Agendamento constitui um processo elementar de acordo com os critérios acima.

Alterar um agendamento é considerado um processo elementar, segundo as regras de contagem, agora é necessário determinar se ele é único para a aplicação sendo medida.

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de TDs e	Não é possível identificar tal informação.
2. Requerem o mesmo conjunto de TRs e	Não é possível identificar tal informação.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não é possível identificar tal informação, mas não outra funcionalidade na lista de requisitos que parece ser semelhante a essa funcionalidade.
4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	A função de Alterar Agendamento consiste em um conjunto único de lógica de processamento que deixa o sistema em estado consistente.

Alterar um agendamento é considerado um processo elementar único. O Quadro 63 apresenta a classificação do processo com EE, SE ou CE.

Quadro 63 – Classificar função.

Regras de contagem para EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mais ALIs ou 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como dados de controle.

<ul style="list-style-type: none"> • Alterar o comportamento da aplicação 	Não.
2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Sim. Informações de Controle entram na fronteira da aplicação.

Alterar um agendamento envia dados para dentro de fronteira da aplicação, com o intuito de manter/atualizar um ou mais ALIs, logo é uma EE. O Quadro 64 apresenta a contagem dos ALRs e TDs relacionados ao processo elementar.

Quadro 64 – Contagem de TDs e ARLs.

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Agendamento é mantido pelo processo elementar. Não há informação suficiente para determinar se o processo consulta dados de outro ALI como: Cliente ou Agenda.
Regras de Contagem para TDs	A regra se aplica?
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
4. Não contar os seguintes itens como	<ul style="list-style-type: none"> • Não foi possível identificar tal

<p>TDs:</p> <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais como: atributos de data e hora. • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	<p>informação.</p>
--	--------------------

O ALI Agendamento é mantido pela aplicação, logo é um ALR, mas não foi possível identificar os TDs de acordo com as regras de contagem. De acordo com o manual de práticas de contagem, se não foi possível identificar os TD e ALRs também não é possível determinar a complexidade da função. O tamanho funcional é determinado de acordo com a complexidade da função, se a função não foi classificada com uma determinada complexidade, logo não é possível atribuir um peso em PF para a mesma.

O próximo caso de uso avaliado é Confirmar Agendamento. O Quadro 65 apresenta a identificação do mesmo.

Quadro 65 – Identificar processo elementar.

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim
2. Constitui uma transação completa?	Sim.
3. É autocontido.	Sim.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado consistente.
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Alterar um Agendamento constitui um processo elementar de acordo com os critérios acima.

Confirmar um agendamento é considerado um processo elementar, segundo as regras de identificação, o Quadro 66 apresenta as regras que determinam se um processo elementar é único.

Quadro 66 - Determinar processo elementar único

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de TDs e	Não é possível identificar tal informação.
2. Requerem o mesmo conjunto de TRs e	Não é possível identificar tal informação.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não é possível identificar tal informação, mas não outra funcionalidade na lista de requisitos que parece ser semelhante a essa funcionalidade.

4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	A função de Confirmar Agendamento consiste em um conjunto único de lógica de processamento que deixa o sistema em estado consistente.
--	---

Após a análise pode-se considerar que Confirmar um Agendamento é um processo elementar único dentro de aplicação sendo contada. O Quadro 67 apresenta a análise para determinar se um processo elementar é uma função do tipo SE, CE ou EE.

Quadro 67 - Determinar tipo da função

Regras de contagem para EE	A regra de aplica?
3. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mais ALIs ou • Alterar o comportamento da aplicação 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como dados de controle. Não.
4. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Sim. Informações de Controle de Cliente entram na fronteira da aplicação.

É uma Entrada Externa, pois mantém ou atualiza um ou mais ALI. O Quadro 68 mostra contagem dos ALR e dos TDs relacionados ao processo elementar Confirmar Agendamento.

Quadro 68 - Contagem de TDs e ALRs

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Agendamento é mantido pelo processo elementar. Não há informação suficiente para determinar se o processo consulta dados de outro ALI como: Cliente ou Agenda.
Regras de Contagem para TDs	A regra se aplica?
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.

<p>processamento de uma função de transação.</p>	
<p>2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.</p>	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
<p>3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).</p>	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
<p>4. Não contar os seguintes itens como TDs:</p> <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais como: atributos de data e hora. • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de 	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.

<p>transação e armazenados em um ALI sem sair da fronteira.</p> <ul style="list-style-type: none"> • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	
---	--

Após a análise pode-se perceber que não foi possível identificar e contar os TDs e ALRs com base na documentação existente. De acordo com o Manual de Práticas de Contagem, se não foi possível identificar os TDs e ALRs também não é possível determinar a complexidade da função. O tamanho funcional é determinado de acordo com a complexidade da função, se a função não foi classificada com uma determinada complexidade, logo não é possível atribuir um peso em PF para a mesma.

O próximo requisito a ser analisado é o Cancelar Agendamento, o principal objetivo desse requisito é Cancelar um agendamento realizado no sistema. O Quadro 69 apresenta as regras para identificação de processo elementar aplicadas a esse requisito.

Quadro 69 - Identificar processo elementar

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim
2. Constitui uma transação completa?	Sim.
3. É autocontido.	Sim.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado consistente.
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Cancelar um Agendamento constitui um processo elementar de acordo com os critérios acima.

De acordo com a análise Cancelar um agendamento constitui um processo elementar. O Quadro 70 apresenta as regras que determinam se o processo elementar é único para aplicação sendo contada.

Quadro 70 – Determinar processo elementar único.

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de TDs e	Não é possível identificar tal informação.
2. Requerem o mesmo conjunto de TRs e	Não é possível identificar tal informação.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não é possível identificar tal informação, mas não outra funcionalidade na lista de requisitos que parece ser semelhante a essa funcionalidade.
4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	A função de Cancelar Agendamento consiste em um conjunto único de lógica de processamento que deixa o sistema em estado consistente.

De acordo com análise e com as regras Cancelar Agendamento constitui um processo elementar único para a aplicação sendo contada. O Quadro 71 determina se o processo elementar é classificado como SE, EE, ou CE.

Quadro 71 - Classificar como EE, SE ou CE

Regras de contagem para EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mais ALIs ou • Alterar o comportamento da aplicação 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como dados de controle. Não.
2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Sim. Informações de Controle de Cliente entram na fronteira da aplicação.

De acordo com a análise o processo elementar é considerado uma EE (Entrada Externa) parasistema sendo contado. No Quadro 72 é apresentada a contagem dos TDs e ALRs relacionados ao processo elementar.

Quadro 72 - Contagem de TDs e ALRs

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Agendamento é mantido pelo processo elementar.
Regras de Contagem para TDs	A regra se aplica?
5. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
6. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
7. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
8. Não contar os seguintes itens como TDs: <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais 	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.

<p>como: atributos de data e hora.</p> <ul style="list-style-type: none"> • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	
---	--

Com base na análise não é possível identificar tal informação com base na documentação existente. De acordo com o manual de práticas de contagem, se não foi possível identificar os TD e ALRs também não é possível determinar a complexidade da função. O tamanho funcional é determinado de acordo com a complexidade da função, se a função não foi classificada com uma determinada complexidade, logo não é possível atribuir um peso em PF para a mesma.

O próximo requisito a ser avaliado é o Pesquisar Agendamento, que tem como função primária, buscar um Agendamento Cadastrado no sistema e apresentar ao usuário. O Quadro 73 mostra identificação do processo elementar.

Quadro 73 - Identificar processo elementar

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfazer todos os critérios abaixo relacionados.	

1. É significativo para o usuário?	Sim
2. Constitui uma transação completa?	Sim.
3. É autocontido.	Sim.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado consistente.
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Pesquisar um Agendamento constitui um processo elementar de acordo com os critérios acima.

Após ter considerado a pesquisa de agendamento um processo elementar no sistema sendo contado, é necessário determinar se ele é único. O Quadro 74 mostra as regras para determinar se o processo elementar é único.

Quadro 74 - Determinar processo elementar único

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de TDs e	Não é possível identificar tal informação.
2. Requerem o mesmo conjunto de TRs e	Não é possível identificar tal informação.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não é possível identificar tal informação, mas não outra funcionalidade na lista de requisitos que parece ser semelhante a essa funcionalidade.
4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	A função de Pesquisar Agendamento consiste em um conjunto único de lógica de processamento que deixa o sistema em estado consistente.

Após a análise pode-se perceber que Pesquisar Agendamento é um processo elementar único no sistema sendo contado, agora é necessário classificar o processo como EE, SE ou CE de acordo com o Quadro 75.

Quadro 75 - Classificar função como SE, EE ou CE

Regras de contagem para EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mais ALIs ou • Alterar o comportamento da aplicação 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como dados de controle. Não.
2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Sim. Informações de Controle de Cliente entram na fronteira da aplicação.

O processo elementar é considerado uma Consulta Externa, onde dados do Agendamento são enviados para fora da fronteira da aplicação, através de uma solicitação do usuário. O Quadro 76 apresenta a contagem de ALRs e TDs relacionados a esse processo elementar.

Quadro 76 - Contagem de ALRs e TDs

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Agendamento é mantido pelo processo elementar.
Regras de Contagem para TDs	A regra se aplica?
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.

de resposta, mesmo que existam múltiplas mensagens.	
3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
<p>4. Não contar os seguintes itens como TDs:</p> <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais como: atributos de data e hora. • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem 	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.

sair da fronteira.	
--------------------	--

Com base na documentação disponível não é possível identificar todos os TDs e ALRs relacionados ao processo elementar. De acordo com o manual de práticas de contagem, se não foi possível identificar os TD e ALRs também não é possível determinar a complexidade da função. O tamanho funcional é determinado de acordo com a complexidade da função, se a função não foi classificada com uma determinada complexidade, logo não é possível atribuir um peso em PF para a mesma.

O próximo requisito a ser avaliado é o Pagar Consulta que tem como função primária realizar o pagamento de uma consulta no sistema. O Quadro 77 apresenta a identificação do processo elementar.

Quadro 77 - Identificar processo elementar

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim
2. Constitui uma transação completa?	Sim.
3. É autocontido.	Sim.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado consistente.
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Pagar uma consulta constitui um processo elementar de acordo com os critérios acima.

A função de Pagar uma Consulta é considerada um processo elementar, segundo as regras de contagem. O Quadro 78 apresenta as regras que determinam se um processo elementar é único.

Quadro 78 - Determinar processo elementar único

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	

1. Requerem o mesmo conjunto de TDs e	Não é possível identificar tal informação.
2. Requerem o mesmo conjunto de TRs e	Não é possível identificar tal informação.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não é possível identificar tal informação, mas não outra funcionalidade na lista de requisitos que parece ser semelhante a essa funcionalidade.
4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	A função de Pagaruma consulta consiste em um conjunto único de lógica de processamento que deixa o sistema em estado consistente.

O processo elementar é considerado único dentro de aplicação. O Quadro 79 apresenta as regras para determinar o tipo do processo elementar: EE, SE ou CE.

Quadro 79 - Classificar processo elementar como EE, SE ou CE

Regras de contagem para EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mais ALIs ou • Alterar o comportamento da aplicação 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como dados de controle. Não.
2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Sim. Informações de Controle de Cliente entram na fronteira da aplicação.

O processo elementar é classificado como uma Entrada Externa, já que mantem o ALI pagamento. O Quadro 80 apresenta a contagem do TDs e ALRs relacionados ao processo elementar.

Quadro 80 – Contagem de TDs e ALRs.

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função	O ALI Pagamento é mantido pelo processo

de dados única que é acessada (lida e/ou mantida) pela função de transação.	elementar.
Regras de Contagem para TDs	A regra se aplica?
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
<p>4. Não contar os seguintes itens como TDs:</p> <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais como: atributos de data e hora. • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. • Ajudas de navegação, tais como: a 	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.

<p>habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes.</p> <ul style="list-style-type: none"> • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	
--	--

Não foi possível identificar todos os TDs e ALRs relacionados ao processo elementar Pagar Consulta, com base na documentação existente. Sem a identificação dos tipos de dados da função, também não é possível determinar sua complexidade. Com base na contagem do IFPUG não há como determinar o tamanho funcional se não se sabe a complexidade da função.

O requisito que será avaliado agora é o Confirmar Pagamento, que tem como ação primária confirmar o pagamento de um consulta realizada no sistema. O Quadro 81 apresenta a identificação do processo elementar.

Quadro 81 – Identificar processo elementar.

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim
2. Constitui uma transação completa?	Sim.
3. É autocontido.	Sim.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado consistente.

5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Confirmar Pagamento constitui um processo elementar de acordo com os critérios acima.
--	---

Confirmar o pagamento de uma consulta o sistema é considerado um processo elementar, o Quadro 82 apresenta as regras para determinar se esse processo elementar é único.

Quadro 82 – Determinar processo elementar único.

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de TDs e	Não é possível identificar tal informação.
2. Requerem o mesmo conjunto de TRs e	Não é possível identificar tal informação.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não é possível identificar tal informação, mas não outra funcionalidade na lista de requisitos que parece ser semelhante a essa funcionalidade.
4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	O processo elementar não foi dividido.

A função de Confirmar Pagamento consiste em um conjunto único de lógica de processamento que deixa o sistema em estado consistente. O Quadro 83 apresenta a contagem de ALRs e TDs relacionados ao processo elementar Confirmar Pagamento.

Quadro 83 – Classificar Função

Regras de contagem para EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mais ALIs ou 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como

<ul style="list-style-type: none"> Alterar o comportamento da aplicação 	<p>dados de controle.</p> <p>Não.</p>
<p>2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.</p>	<p>Sim. Informações de Controle de Cliente entram na fronteira da aplicação.</p>

O processo envia dados para dentro de fronteira da aplicação, com o intuito de manter/atualizar um ou mais ALIs, logo é uma EE. O Quadro 84 apresenta a contagem de ALRs e TDs.

Quadro 84 – Contagem de ALRs e TDs

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Agendamento é mantido pelo processo elementar.
Regras de Contagem para TDs	A regra se aplica?
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o processamento de uma função de transação.	<ul style="list-style-type: none"> Não foi possível identificar tal informação.
2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.	<ul style="list-style-type: none"> Não foi possível identificar tal informação.
3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).	<ul style="list-style-type: none"> Não foi possível identificar tal informação.

<p>4. Não contar os seguintes itens como TDs:</p> <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais como: atributos de data e hora. • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de transação e armazenados em um ALI sem sair da fronteira. • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
--	--

A partir da documentação disponível e das regras de contagem não foi possível identificar os TDs e ALRs relacionados ao processo elementar Confirmar Pagamento. De acordo com o manual de práticas de contagem, se não foi possível identificar os TD e ALRs também não é possível determinar a complexidade da função. O tamanho funcional é determinado de acordo com a complexidade da função, se a função não foi classificada com uma determinada complexidade, logo não é possível atribuir um peso em PF para a mesma.

O próximo requisito a ser avaliado é Cancelar Pagamento, esse requisito tem como ação principal Cancelar um pagamento realizado no sistema. O Quadro 85 apresenta a identificação do processo elementar.

Quadro 85 – Identificar processo elementar

Identificar processo elementar	A Regra de aplica?
Compor e/ou decompor os Requisitos Funcionais do usuário na menor unidade de uma atividade, se satisfaça todos os critérios abaixo relacionados.	
1. É significativo para o usuário?	Sim
2. Constitui uma transação completa?	Sim.
3. É autocontido.	Sim.
4. Deixa o negócio da aplicação que está sendo medida em um estado consistente.	Sim. O negócio é deixado em um estado consistente.
5. Identificar um processo elementar para cada unidade de atividade identificada que satisfaça todos os critérios acima.	Cancelar Pagamento constitui um processo elementar de acordo com os critérios acima.

Cancelar pagamento é considerado um processo elementar de acordo com as regras de contagem e de identificação. O Quadro 86 determina se o processo elementar é único.

Quadro 86- Determinar processo elementar único.

Determinar Processos Elementares Únicos	A Regra se Aplica?
Quando comparado com um Processo Elementar já identificado, contar dois Processos Elementares similares como o mesmo Processo Elementar, se ambos:	
1. Requerem o mesmo conjunto de TDs e	Não é possível identificar tal informação.
2. Requerem o mesmo conjunto de TRs e	Não é possível identificar tal informação.
3. Requerem o mesmo conjunto de lógica de processamento para completar o processo elementar	Não é possível identificar tal informação, mas não outra funcionalidade na lista de requisitos que parece ser semelhante a essa funcionalidade.

4. Não dividir um processo elementar com múltiplas formas de lógica de processamento em múltiplos processos elementares.	Nenhuma função foi dividida em pedaços menores.
--	---

A função de Cancelar pagamento consiste em um conjunto único de lógica de processamento que deixa o sistema em estado consistente. O Quadro 87 determina o tipo do processo elementar EE, SE ou CE.

Quadro 87 - Classificar processo elementar como: SE, CE ou EE.

Regras de contagem para EE	A regra de aplica?
1. Possui como ação primária: <ul style="list-style-type: none"> • Manter um ou mais ALIs ou • Alterar o comportamento da aplicação 	Sim. Os dados que entram na fronteira da aplicação são utilizados eventualmente como dados de controle. Não.
2. Inclui a lógica de processamento para aceitar dados ou informações de controle que entram na fronteira da aplicação.	Sim. Informações de Controle de Cliente entram na fronteira da aplicação.

Segundo as regras de classificação o processo elementar é considerado uma EE. O Quadro 88 apresenta a contagem de ALRs e TDs relacionados ao processo elementar.

Quadro 88 – Contagem de ALRs e TDs.

Regras de Contagem de ALR	A regra se aplica?
Um ALR deve ser contado para cada função de dados única que é acessada (lida e/ou mantida) pela função de transação.	O ALI Agendamento é mantido pelo processo elementar. Não há informação suficiente para determinar se o processo consulta dados de outro ALI como: Cliente ou Agenda.
Regras de Contagem para TDs	A regra se aplica?
1. Contar um TD para cada atributo não repetido, reconhecido como único pelo usuário, que atravessa (entra e/ou sai) a fronteira durante o	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.

processamento de uma função de transação.	
2. Contar apenas um TD por função de transação para a habilidade da aplicação de enviar uma mensagem de resposta, mesmo que existam múltiplas mensagens.	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
3. Contar apenas um TD por função de transação para a habilidade de iniciar uma ou mais ações, mesmo que existam múltiplas formas para iniciá-la(s).	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.
<p>4. Não contar os seguintes itens como TDs:</p> <ul style="list-style-type: none"> • Literais, tais como: títulos de relatórios, identificadores de telas, cabeçalhos de colunas e títulos de atributos. • <i>Application generated stamps</i>, tais como: atributos de data e hora. • Variáveis de paginação, tais como: número de páginas e informações de posicionamento, i.e., ‘Linhas 37 a 54 de 211’. • Ajudas de navegação, tais como: a habilidade para navegar dentro de uma lista utilizando atalhos como “anterior”, “próximo”, “primeiro”, “último” e suas representações gráficas equivalentes. • Atributos gerados dentro da fronteira da aplicação por uma função de 	<ul style="list-style-type: none"> • Não foi possível identificar tal informação.

<p>transação e armazenados em um ALI sem sair da fronteira.</p> <ul style="list-style-type: none"> • Atributos recuperados ou referenciados de um ALI ou AIE para serem usados no processamento sem sair da fronteira. 	
---	--

Não é possível identificar os TDs e ALRs com base na documentação existente. De acordo com o manual de práticas de contagem, se não foi possível identificar os TD e ALRs também não é possível determinar a complexidade da função. O tamanho funcional é determinado de acordo com a complexidade da função, se a função não foi classificada com uma determinada complexidade, logo não é possível atribuir um peso em PF para a mesma.

Após identificar e classificar todas as funções de dados e transacionais é possível determinar o tamanho funcional da aplicação. O resultado obtido está apresentado no Quadro 89.

Quadro 89 – Resultado da Contagem.

Função	Tipo	Complexidade	Contribuição
Cliente	ALI	Baixa	7
Agenda	ALI	Baixa	7
Agendamento	ALI	Baixa	7
Pagamento	ALI	Baixa	7
Adicionar Cliente	EE	Baixa	3
Pesquisar Cliente	CE	Baixa	4
Alterar Cliente	EE	Baixa	3
Adicionar Agenda	EE	Baixa	3
Alterar Agenda	EE	Baixa	3

Remover Agenda	EE	Baixa	3
Adicionar Agendamento	EE	Baixa	Não identificado
Remover Agendamento	EE	Baixa	Não identificado
Alterar Agendamento	EE	Baixa	Não identificado
Confirmar Agendamento	EE	Baixa	Não identificado
Cancelar Agendamento	EE	Baixa	Não identificado
Pesquisar Agendamento	CE	Baixa	Não identificado
Pagar Consulta	EE	Baixa	Não identificado
Cancelar Pagamento	EE	Baixa	Não identificado
Confirmar Pagamento	EE	Baixa	Não identificado
Total			51

Logo o tamanho funcional da aplicação em PF não ajustados seria igual a 51 pontos de função. As funções de transação relacionadas aos ALIs Agendamento e Pagamento não foram contabilizadas devido a falta de documentação disponível. Isso introduziria uma margem de erro na contabilização dos pontos de função da aplicação. No entanto, o IFPUG indica, que, caso a documentação disponível seja insuficiente, devem ser feitas entrevistas com o cliente, ou com o gerente de projeto, para se obterem mais detalhes sobre as funcionalidades do sistema. Logo, a documentação reduzida não impossibilita, de fato, a utilização da abordagem de IFPUG, mas dificulta a mesma.

Como foi demonstrado na contagem dos requisitos Gerenciar Cliente e Gerenciar Agenda é possível realizar a contagem de Pontos de Função em um projeto ou aplicação que possuem uma documentação mais abrangente. A partir dessa documentação é possível identificar os tipos de dados, e tipos de registros utilizados para manter uma função de dados ou utilizado como informações de controle em uma função do tipo transação, e assim determinar a complexidade das funções e atribuir um peso em PF às mesmas.

No entanto, como apresentado na contagem dos requisitos Gerenciar Pagamento e Agendamento, com pouca documentação disponível não é possível concluir, com êxito, a contagem de PF de acordo com a abordagem do IFPUG. Ainda assim, é possível identificar as funções e classifica-las como ALIs, ou AIEs e CE, SE, ou EE. No entanto, não é possível identificar a quantidade de TDs, TRs ou ALR, e dessa forma também não é possível classificar em relação complexidade baixa, média ou alta e, sendo assim, não se pode atribuir um peso em PF para as funções.

Quando se diz que não é possível, refere-se ao fato de não ser possível contar a partir da documentação disponível, no entanto, o IFPUG indica que nesses casos devem ser feitas entrevistas com gerentes de desenvolvimento ou clientes para obtenção de maiores detalhes sobre as funcionalidades do *software*. O que acaba gerando mais tempo e custo.

Nas seções seguintes será apresentada uma análise das abordagens para contagem de PF da Associação de Métricas de *software* da Holanda (NESMA). As abordagens analisadas serão Contagem Indicativa e Contagem Estimativa. A Contagem de pontos de função Detalhada da NESMA não será apresentada nessa análise, pois o processo de contagem da mesma é equivalente à contagem de pontos de função do Manual de Práticas de Contagem.

4.2. Contagem Indicativa

Como foi apresentado da seção 2.3.1. Contagem Indicativa, para esse tipo de contagem é necessário identificar apenas as funções do tipo dado e determinar o tipo de cada função (Arquivo Lógico Interno ou Arquivo de Interface Externa). Sendo assim, os passos para contagem indicada do tamanho funcional de um projeto de *software* são: Identificar as funções do tipo dado; Classificar as funções do tipo dados em ALI ou AIE. Como foi mencionado anteriormente o sistema a ser medido com essa abordagem é o mesmo sistema que foi medido com a abordagem do IFPUG, o Sistema de Atendimento Psicológico Online. O Diagrama de componentes analisado é o apresentado na [Figura 17](#).

Analisando a Figura 17 é necessário a partir dela identificar as funções de dados, ou seja, aquelas que atendem os requisitos de armazenamento de dados do sistema. O Quadro 87 apresenta as regras para identificação das funções de dados.

Quadro 90 – Identificação das funções de dados.

Regras de Identificação	A regra se aplica?
1. Identifique todos os dados ou informações de controle logicamente	<ul style="list-style-type: none"> • Cliente • Agenda • Agendamento

relacionados e reconhecidos pelo usuário dentro do escopo da contagem.	<ul style="list-style-type: none"> • Pagamento • Chat • Videoconferência
2. Exclua entidades que não são mantidas por qualquer aplicação.	<ul style="list-style-type: none"> • Chat • Videoconferência
3. Agrupe entidades relacionadas que são entidades dependentes.	Não existem entidades desse tipo.
4. Exclua as entidades referidas como Dados de Código.	Não existem entidades desse tipo.
5. Exclua entidades que não possuem atributos reconhecidos pelo usuário	Não existem entidades desse tipo.
6. Remova as entidades associativas que contém atributos adicionais não requeridos pelo usuário e entidades associativas que contém somente chaves estrangeiras; agrupe os atributos chave estrangeira com as entidades primárias.	Não existem entidades desse tipo.

Aplicando as regras é possível identificar 4 funções de dados. Já que as funções de Chat e Videoconferência não são mantidas pela aplicação, através de processos elementares. Após encontradas as funções de dados é necessário classifica-las com ALIs ou AIEs. O Quadro 91 apresenta as regras de classificação.

Quadro 91 – Classificação da função de dados em ALI ou AIE.

Regra de classificação	A regra se aplica?
1. Classificar como ALI se os dados são mantidos pela aplicação sendo contada.	Cliente, Agenda, Agendamento, Pagamento, são mantidas através de processos elementares da aplicação.
2. Classificar como AIE se: É Referenciado, mas não é mantido pela aplicação sendo contada. É identificado como ALI em uma ou mais outras aplicações.	Todas são mantidas dentro da aplicação, logo não existem AIEs.

As funções de dados: Cliente, Agenda, Agendamento e Pagamento são consideradas ALIs. Após identificar as funções de dados e classifica-las como ALI ou AIE o tamanho funcional é obtido pelo Quadro 92:

Quadro 92 – Tamanho funcional da aplicação

Fórmula	PF = N° de ALIs * 35 + N° de AIEs *15
Número de ALIs	4
Número de AIEs	0
Total em PF	140

Utilizando essa fórmula o tamanho funcional da Aplicação de Atendimento Psicológico é igual a 140 PF não ajustados. É possível notar uma diferença no número de pontos obtido por essa abordagem e pela abordagem do UFPUG, mas é necessário ressaltar que essa diferença se dá pelo fato de não ter sido possível identificar as funções transacionais relacionadas aos ALIs Agendamento e Pagamento, devido a falta de documentação.

4.3. Contagem Estimativa

Na Contagem Estimativa da NESMA é preciso identificar todas as funções de dados e todas as funções transacionais e depois é necessário classificadas como ALI ou AIE no caso das funções de dados e EE, SE ou CE no caso das funções de transação. Logo, os passos a serem realizados nesse tipo de contagem são: Identificar as funções de dados; Classificar as funções de dados; Identificar as funções de transação; Classificar as funções de transação; Atribuir complexidade média para todas as funções de transação; Atribuir complexidade baixa para todas as funções de dados; Calcular tamanho funcional. Para representar a contagem de acordo com a abordagem estimativa da NESMA será apresentada a contagem do mesmo sistema contado nas abordagens anteriores, o Sistema de Atendimento Psicológico.

Os documentos utilizados para medição são os mesmo já utilizados nas abordagens anteriores: Diagrama de Componentes, Lista de Requisitos, Diagramas de Sequencia e Caso de Uso Expandido.

A partir do Diagrama de componentes, [Figura 17](#), é possível identificar as funções de dados, como apresentado no Quadro 93.

Quadro 93 – Identificar funções de dados.

Regras de Identificação	A regra se aplica?
1. Identifique todos os dados ou informações de controle logicamente relacionados e reconhecidos pelo usuário dentro do escopo da contagem.	<ul style="list-style-type: none"> • Cliente • Agenda • Agendamento • Pagamento • Chat • Videoconferência
2. Exclua entidades que não são mantidas por qualquer aplicação.	<ul style="list-style-type: none"> • Chat

	<ul style="list-style-type: none"> • Videoconferência
3. Agrupe entidades relacionadas que são entidades dependentes.	Não existem entidades desse tipo.
4. Exclua as entidades referidas como Dados de Código.	Não existem entidades desse tipo.
5. Exclua entidades que não possuem atributos reconhecidos pelo usuário	Não existem entidades desse tipo.
6. Remova as entidades associativas que contém atributos adicionais não requeridos pelo usuário e entidades associativas que contém somente chaves estrangeiras; agrupe os atributos chave estrangeira com as entidades primárias.	Não existem entidades desse tipo.

Aplicando as regras é possível identificar 4 funções de dados. Já que as funções de Chat e Videoconferência não são mantidas pela aplicação. As funções de dados: Cliente, Agenda, Agendamento e Pagamento são consideradas ALIs.

Para identificação das funções de transação será utilizada a lista de requisitos, apresentada a seguir:

- R1: Gerenciar Cliente
 - C1.1: Adicionar Cliente
 - C1.2: Pesquisar Cliente
 - C1.3: Alterar Cliente
- R2: Gerenciar Agenda
 - C2.1: Adicionar Agenda
 - C2.2: Remover Agenda
 - C2.3: Alterar Agenda
 - C2.4: Pesquisar Agenda
- R3: Agendamento de Consulta
 - C3.1: Adicionar Agendamento
 - C3.2: Remover Agendamento
 - C3.3: Alterar Agendamento
 - C3.4: Pesquisar Agendamento
 - C3.5: Confirmar Agendamento
 - C3.6: Cancelar Agendamento
- R4: Pagamento da Consulta

- C4.1: Confirmar pagamento de consulta
- C4.2: Pagar consulta
- C4.3: Cancelar Pagamento

Como já foi demonstrado nas abordagens anteriores, cada Caso de Uso da Lista de Requisitos foi considerado uma função do tipo transação, logo, as funções transacionais para a aplicação são: Adicionar Cliente, Pesquisar Cliente, Alterar Cliente, Adicionar Agenda, Remover Agenda, Alterar Agenda, Pesquisar Agenda, Adicionar Agendamento, Remover Agendamento, Alterar Agendamento, Pesquisar Agendamento, Cancelar Agendamento, Confirmar Agendamento, Confirmar Pagamento, Cancelar Pagamento e Pagar Consulta. O Quadro 94 apresenta o resultado da análise.

Quadro 94 – Resultado

Função	Tipo	Complexidade	Contribuição
Cliente	ALI	Baixa	7
Agenda	ALI	Baixa	7
Agendamento	ALI	Baixa	7
Pagamento	ALI	Baixa	7
Adicionar cliente	EE	Média	4
Alterar cliente	EE	Média	4
Adicionar Agenda	EE	Média	4
Alterar Agenda	EE	Média	4
Remover Agenda	EE	Média	4
Adicionar Agendamento	EE	Média	4
Alterar Agendamento	EE	Média	4
Remover Agendamento	EE	Média	4
Pagar Consulta	EE	Média	4
Remover Pagamento	EE	Média	4
Cancelar Pagamento	EE	Média	4
Confirmar Pagamento	EE	Média	4
Pesquisar Cliente	CE	Média	4
Pesquisar Agenda	CE	Média	4
Pesquisar Agendamento	CE	Média	4
Cancelar Agendamento	EE	Média	4
Total:			92 PF

Como pode ser observado existem diferenças nas contagens de PF de diferentes abordagens, o que não significa que as contagens estejam incorretas. Essa diferença tende a ser mais visível em *softwares* com poucas funcionalidades, como é o caso do Sistema de Atendimento Psicológico *online*. Nesse caso, essa diferença também é justificada pelo fato dos processos elementares, ou seja, as funções de transação, dos ALIs Agendamento e Pagamento não terem sido identificadas. Os resultados obtidos através das 3 contagens está apresentado na Tabela

Tabela 13 – Resultados das Contagens

Abordagem	Nº de funções avaliadas e contadas	Total em PF
IFPUG	9	51
Indicativa NESMA	4 – apenas funções de dados	140
Estimativa NESMA	20 – dados e transação	92

A contagem que apresentou o maior número de pontos de função foi a contagem Indicativa da NESMA, como foi apresentado na seção 2.3.1 – Contagem Indicativa, a fórmula para contagem de PFs utilizada na Contagem Indicativa é baseada na premissa de que cada ALI tem aproximadamente 3 EE, duas SE e uma CE, nesse caso o ALI Agenda possui apenas duas EE e uma CE e não possui nenhuma SE. O ALI Agendamento possui três EE, uma CE e nenhuma SE. Também não foi possível identificar as funções transacionais dos ALIs Agendamento e Pagamento, o que diminuiu consideravelmente o número de PFs obtidos na contagem do IFPUG.

4.3.1. Considerações finais das análises do IFPUG e NESMA.

Como foi demonstrado através de exemplos, as contagens indicativa e estimativa da NESMA podem ser utilizadas para meios de estimativas de projetos de *softwares*, mesmo quando não se possui um elevado nível de documentação ou de detalhamento do projeto a ser dimensionado.

Sendo assim, essas contagens seriam mais indicadas para realizar estimativas de tamanho em projetos de *software* desenvolvidos com metodologias ágeis, como o FDD, que possui foco na entrega frequente de valor ao cliente é guiado por funcionalidades, assim como a técnica de APF.

Já na abordagem do IFPUG, de acordo com as regras e o processo de contagem de PF do CPM, é necessário que haja alguns artefatos de *software* já no início do projeto. Os tipos de artefatos necessários variam de acordo com o tipo de contagem utilizada. No caso de projetos de desenvolvimento, onde se obtém uma estimativa do tamanho de um projeto a ser

desenvolvido, como foi visto na seção 2.2.6 – Processos de Contagem, é necessário identificar as Funções de Dado e Funções de Transação, e depois determinar a complexidade de cada uma delas, o que somente é possível através da identificação da quantidade de TDs (Tipo de Dado) e TRs (Tipos de Registro).

Para a determinação de um TD ou TR é necessário entender o modelo de dados de forma mais detalhada, e/ou obter telas do sistema. Em projetos desenvolvidos com metodologias ágeis, como o FDD, isso pode se tornar complicado, pois geralmente a quantidade de artefatos produzidos é pouca e o detalhamento desses artefatos não costuma ser muito grande. Por exemplo, na primeira fase do FDD os documentos produzidos são apenas uma lista de funcionalidades e um modelo de dados abrangente, além de definição de prioridades e equipe. Logo, não seria possível utilizar a abordagem do IFPUG em *softwares* desenvolvidos com FDD, sem a necessidade de ajuste em uma das técnicas.

Outro ponto que pode inviabilizar a utilização da técnica do IFPUG é que a contagem de Projetos de Melhoria pode, em projetos pequenos, dar margem a erros nas estimativas de tamanho do *software*, como foi abordado da seção 2.2.8.2 – Projeto de Melhoria IFPUG. Nesse ponto, a abordagem da NESMA para projetos de melhoria se mostra mais viável e mais concisa, como foi abordado na seção 2.2.8.3 – Projeto de Melhoria NESMA. Outro ponto que contribui para seleção da abordagem da NESMA foi o fato do grupo holandês contar com três tipos diferentes de contagem, que podem ser utilizados em momentos diferentes no ciclo de vida de um projeto, desde a fase inicial (Contagem indicativa) até a fase final (Contagem detalhada).

A seção seguinte apresenta uma proposta de utilização das Contagens Indicativa e Estimativa da NESMA em projetos desenvolvidos com a metodologia de desenvolvimento FDD. A proposta foi elaborada com base nos estudos realizados e nas análises apresentadas anteriormente.

4.4. Proposta de Adaptação do FDD

A proposta se baseia na metodologia FDD acrescida de dois processos que dizem respeito à realização de estimativas do tamanho de um *software*. Essas estimativas podem servir de base para cálculo de outros fatores, como custo e tempo. A Figura 20 apresenta graficamente essa proposta.

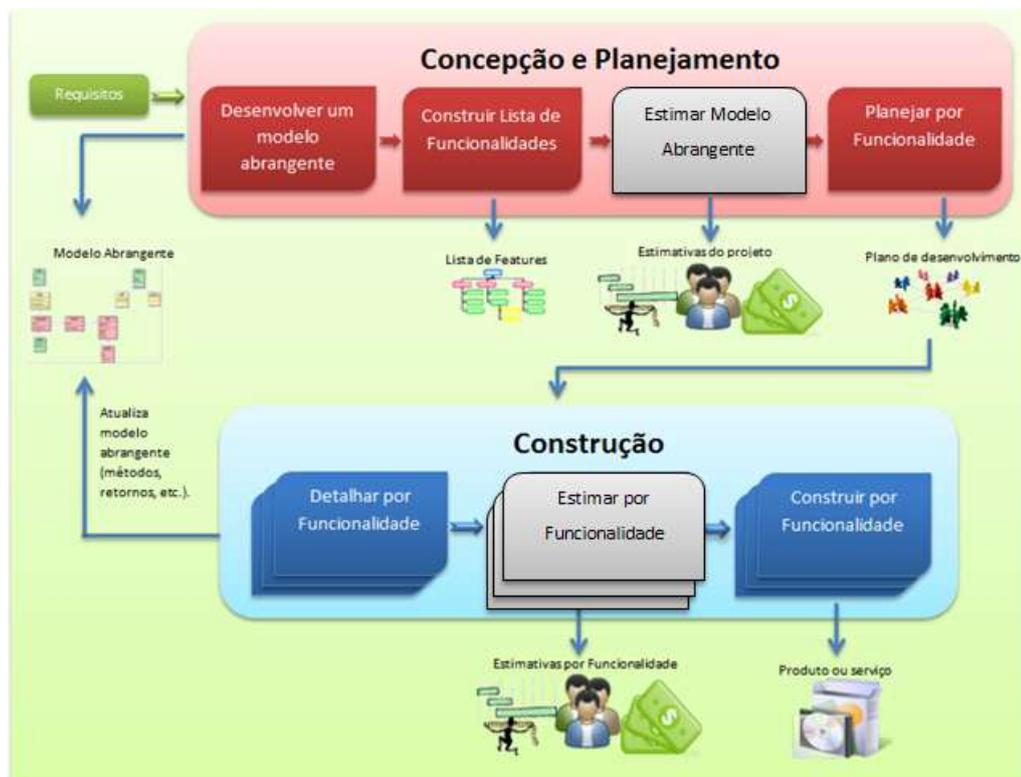


Figura 20 – Proposta de adaptação do FDD.

Como apresentado na Figura 20, foram inseridos dois novos processos na metodologia FDD, um na fase de Concepção e Planejamento e o outro na fase de Construção. O processo inserido na primeira fase é o Estimar Modelo Abrangente-EMA, que tem como objetivo obter uma estimativa geral do tamanho do projeto baseado no Modelo de Contagem Indicativa da NESMA. As seções seguintes apresentam de forma mais detalhada os processos inseridos.

4.4.1. Processo Estimar Modelo Abrangente

Essa seção apresentada o detalhamento do processo Estimar Modelo Abrangente, foi o processo inserido na primeira fase do FDD. A Figura 21 mostra o detalhamento do processo EMA.

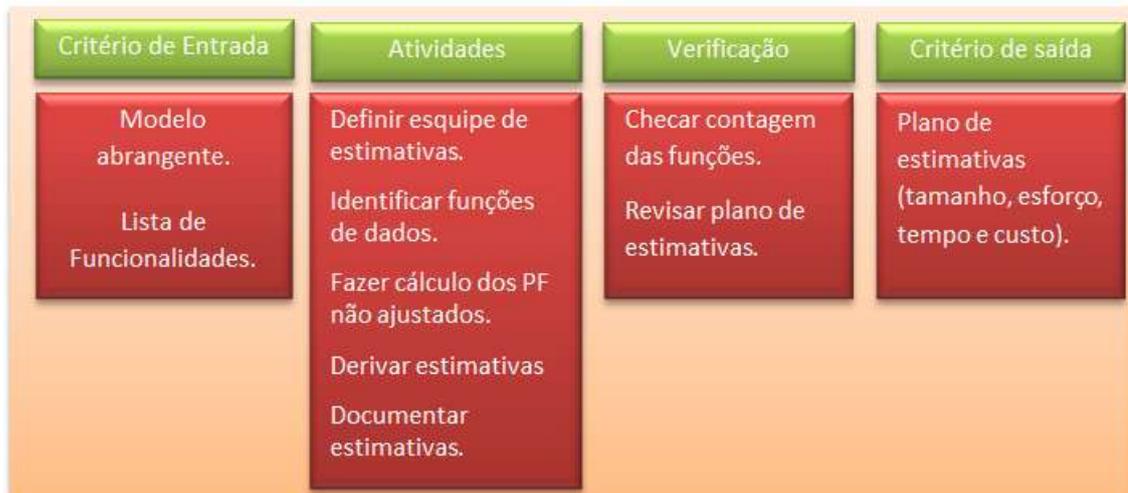


Figura 21 – Detalhamento do processo Estimar Modelo Abrangente.

Como pode ser visto na Figura 21 o critério de entrada para o processo EMA é o Modelo Abrangente do projeto e a Lista de Funcionalidades. As atividades são: formar a equipe de estimativas, identificar as funções de dados, obter tamanho indicado, derivar estimativas de esforço, prazo e custo e documentar as estimativas na base da organização. A verificação consiste em um *Check List* do modelo abrangente e das estimativas para garantir que todas as funções de dados foram somadas e que nenhuma delas foi somada de forma errônea ou contada duas vezes. O critério de saída é um plano de estimativas do projeto. Após a verificação, caso erros sejam encontrados, eles devem ser corrigidos e documentados na base histórica e as estimativas devem ser atualizadas na base.

A seguir serão apresentados detalhadamente os componentes do processo: Critérios de Entrada, Atividades, Verificação e Critérios de Saída.

a) Critérios de Entrada

Os Critérios de Entrada são o Modelo Abrangente, construído no processo DMA e a Lista de Funcionalidades, produzida no processo CLF. Esses documentos devem servir de base para a identificação e classificação das funções de dados do projeto de sistema.

b) Atividades

Definir equipe de estimativas: definir a equipe responsável pela realização das estimativas de tamanho, esforço, tempo e custo. Essa equipe pode ser composta por membros

externos à equipe de desenvolvimento ou por membros da própria equipe de desenvolvimento.

Identificar funções de dados: a equipe de estimativas tem a função de analisar a documentação existente do projeto e a partir dessa documentação e da aplicação das regras de contagem identificar as funções de dados do sistema. Além de identificar é necessário classificar as funções de dados como ALIs ou AIEs.

Fazer cálculo dos PFs não ajustados: essa atividade tange à aplicação da fórmula para contagem do tamanho funcional da abordagem Indicativa da NESMA, onde o número de ALI é multiplicado por 35, o número de AIS é multiplicado por 15 e os valores resultantes são somados, o resultado da soma representa o tamanho funcional estimado do projeto de *software*.

Derivar estimativas: atividade de responsabilidade da equipe de estimativas e seu objetivo é obter: a estimativa do esforço e do tempo necessário para desenvolvimento do projeto, estimativa de custo do projeto. Essas estimativas são derivadas com base em algum método de estimativas de *software* a ser definido pela equipe de desenvolvimento e que deve usar o tamanho em PF como critério de entrada.

Documentar Estimativas: as estimativas obtidas devem ser documentadas na base histórica da organização, juntamente com dados do projeto. A documentação pode ser feita pelo membro da equipe que assume o papel de documentador ou pela própria equipe de estimativas.

c) Verificação

Checar contagens das funções: é feita uma checagem geral da documentação e da contagem realizada, com um intuito de identificar possíveis irregularidades, como funções contadas duas vezes ou funções que deixaram de ser contadas. A checagem deve ser feita com toda documentação disponível.

Revisão do plano de estimativas: as estimativas obtidas devem ser revisadas. Essa revisão pode ser feita pela própria equipe de estimativas ou por uma equipe externa.

d) Critérios de Saída

Como critério de saída é gerado um plano de estimativas composto por estimativas de tamanho, esforço, tempo e custo. Esse documento será utilizado para ajudar no planejamento do projeto e na elaboração de cronogramas e orçamentos.

4.4.2. EPF - Processo Estimar por Funcionalidade

O processo Estimar por Funcionalidade – EPF foi inserido na fase de Construção e seu objetivo é obter o Tamanho Estimado de uma funcionalidade baseado no modelo de Contagem Estimativa da NESMA. A Figura 22 apresenta o detalhamento do processo Estimar por Funcionalidade.

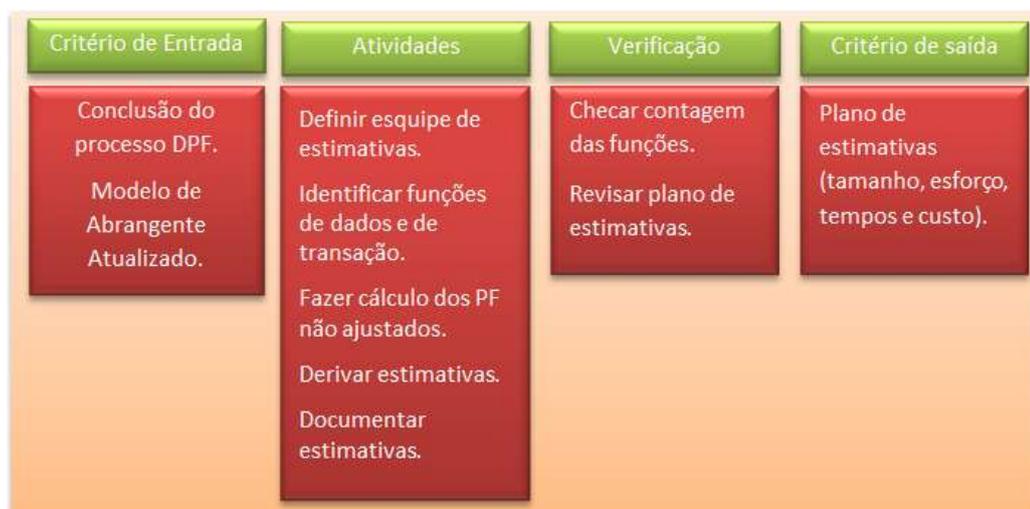


Figura 22 – Detalhamento do processo Estimar Por Funcionalidade

Como critério de entrada o processo recebe os documentos produzidos no processo DPF. As atividades do projeto consistem em medição do tamanho estimado da funcionalidade e realização de estimativas de esforço, prazo e custo da funcionalidade. Ao final do processo obtêm-se um plano de estimativa para a funcionalidade em questão.

a) Critérios de Entrada

O critério de entrada para esse processo é a conclusão do processo DPF (Detalhar por Funcionalidade). A conclusão do DPF deve fornecer subsídios suficientes para a identificação e classificação das funções de dados e das funções transacionais de cada funcionalidade. Os documentos que podem ser utilizados para esse tipo de medição são: Caso de Uso Expandido, Modelo Entidade Relacionamento, Diagramas de Sequencia, Lista de Funcionalidades, Diagrama de Classes.

b) Atividades

Definir equipe de estimativas: definir a equipe responsável pela realização das estimativas de tamanho, esforço, tempo e custo. Essa equipe pode ser composta por membros externos à equipe de desenvolvimento ou por membros da própria equipe de desenvolvimento.

Identificar e classificar funções de dados: a equipe de estimativas tem a função de analisar a documentação existente do projeto e a partir dessa documentação e da aplicação das

regras de contagem identificar as funções de dados do sistema. Além de identificar é necessário classificar as funções de dados como ALIs ou AIEs.

Identificar e classificar as funções de transação: é necessário nessa atividade identificar todas as funções de transação com base na documentação disponível. Depois de identificar cada função de transação é necessário classificá-la como EE, SE ou CE. Documentos como: Caso de uso Expandido ou Diagramas de Sequencia podem ser utilizados para classificação das funções de transação.

Fazer cálculo dos PFs não ajustados: nessa atividade, depois de todas as funções identificadas e classificadas com um tipo, é atribuída complexidade baixa a todas as funções de dados e complexidade média a todas as funções de transação, como definido pela contagem Estimativa da NESMA. A partir disso, os pontos são atribuídos de acordo com as tabelas de atribuição de PF apresentados na seção 2.2.6 - Processos de Contagem, e o tamanho funcional é obtido.

Derivar estimativas: atividade de responsabilidade da equipe de estimativas e seu objetivo é obter: a estimativa do esforço e do tempo necessário para desenvolvimento da funcionalidade ou do grupo de funcionalidade, estimativa de custo da funcionalidade. Essas estimativas são derivadas com base em algum método de estimativas de *software* a ser definido pela equipe de desenvolvimento e que deve usar o tamanho em PF como critério de entrada.

Documentar Estimativas: as estimativas obtidas devem ser documentadas na base histórica da organização, juntamente com dados do projeto e da funcionalidade. A documentação pode ser feita pelo membro da equipe que assume o papel de documentador ou pela própria equipe de estimativas.

c) Verificação

Checar contagens das funções: é feita uma checagem geral da documentação e da contagem realizada, com um intuito de identificar possíveis irregularidades, como funções contadas duas vezes ou funções que deixaram de ser contadas. A checagem deve ser feita com toda documentação disponível.

Revisão do plano de estimativas: as estimativas obtidas devem ser revisadas. Essa revisão pode ser feita pela própria equipe de estimativas ou por uma equipe externa.

d) Critérios de Saída

Como critério de saída é gerado um plano de estimativas composto por estimativas de tamanho, esforço, tempo e custo. Esse documento ser utilizado para ajudar no planejamento

da funcionalidade, elaboração de cronogramas e orçamentos, ou mesmo para determinar se a mesma deve ser dividida em funcionalidades menores.

Com a inserção desse processo é possível evitar que os prazos para construção de funcionalidades sejam definidos de forma errônea, ainda que seja necessário extrapolar o prazo máximo de duas semanas indicado pela FDD.

4.4.3. Artefatos obrigatórios do FDD

O FDD já define alguns artefatos obrigatórios como: lista de requisitos e modelo de objetos. Além disso, outros artefatos são sugeridos como: caso uso expandido, diagramas de sequencia, protótipos etc. No entanto, esses artefatos obrigatórios podem não ser suficientes para a realização da contagem, por isso, para que não haja problemas referentes a documentação disponível para contagem dos pontos de função, alguns artefatos foram definidos como obrigatórios nos processos do FDD.

No primeiro processo - DMA, além do modelo abrangente do projeto, que representa as necessidades de negócio do sistema, deve ser confeccionado, para fins de medição, um diagrama de base de dados, por exemplo, um Modelo Entidade Relacionamento. A partir desse modelo deve ser possível identificar as funções de dados, mas não necessariamente esse modelo deve ser detalhista, mostrando, por exemplo, a quantidade de atributos de uma determinada tabela. Sugere-se um documento contendo a descrição das funcionalidades do sistema do ponto de vista do usuário, por exemplo: requisitos de usuário ou documento de visão.

Já na segunda fase, o processo DPF, deve-se gerar documentos que permitam a identificação e classificação das funções de dados e transacionais. Esses documentos podem ser: protótipos de telas do sistema, caso de uso expandido ou diagrama de sequência. É importante ressaltar que para identificação e classificação das funções de transação, os documentos citados são equivalentes, ou seja, a equipe pode optar por um, ou mais, ou todos os documentos, mas é necessário que pelo menos um deles seja confeccionado.

4.5. Riscos de aplicação da proposta

Por se tratar de uma metodologia ágil o FDD tem foco na entrega frequente de valor ao cliente e a desburocratização dos processos, o que, ainda assim, não o impede de possuir processos definidos em seu ciclo de vida. No decorrer do trabalho foram levantadas questões pertinentes a possível perda de agilidade com a inserção dos novos processos que foram propostos. Questões essas que só podem ser devidamente respondidas com a aplicação da proposta, ou em um possível estudo de caso.

Os processos inseridos têm como objetivo contribuir com o planejamento e melhoria do processo de desenvolvimento de *software*, de forma que a equipe de planejamento possa contar com uma métrica funcional nas fases iniciais do projeto que permita a estimativa de esforço, tempo e custo do desenvolvimento do *software*.

Com a estimativa de tamanho, ainda na fase de Concepção e Planejamento, seria possível obter outras estimativas e, a partir dessas, ter uma projeção geral de quanto tempo o projeto irá durar e o quanto esforço será necessário para construí-lo, o que evitaria projetos com iterações infinitas provindas da subdivisão interminável das funcionalidades em funcionalidades menores. Além disso, as estimativas obtidas em cada projeto seriam armazenadas na base histórica da organização, servindo assim, de base para projetos futuros. Com o passar do tempo, as estimativas seriam feitas com base no tamanho estimado em PF calibradas com base nos dados históricos e do projeto atual, como foi proposto por Aguiar (2002) e por Hazan (2009).

5. CONSIDERAÇÕES FINAIS

Esse trabalho apresentou conceitos de técnicas de medição de *software*, bem como suas principais características, seus objetivos e benefícios. Dentre as técnicas de medição apresentadas, destaca-se a Análise de Pontos de Função, por ser uma técnica de medição funcional que pode ser utilizada nos momentos iniciais de um projeto de *software*.

Dentro dos conceitos da técnica de APF foram apresentadas 4 abordagens para contagem de PF, sendo que essas abordagens foram: abordagem do IFPUG, segundo o manual de práticas de contagem (CPM), Contagem Indicativa, Contagem Estimativa e Contagem Detalhada da NESMA.

Também foram apresentados conceitos de metodologias de *software* ágeis, destacando-se o FDD. Sobre o FDD foi apresentado seu ciclo de vida no desenvolvimento e gerenciamento de *software*, seus papéis, fases e processos, bem como o detalhamento dos processos, apresentando os critérios de entrada, as atividades, os critérios de verificação e critérios de saída.

A partir dos estudos realizados e conceitos apresentados foi realizada uma análise nas abordagens de contagem de pontos de função e no ciclo de vida do FDD. Essa análise foi feita com o intuito de identificar a abordagem de contagem que melhor se adaptaria ao desenvolvimento ágil de *software* com a metodologia FDD.

Após a realização da análise foi elaborada uma proposta de utilização conjunta de APF e FDD. A proposta elaborada tem o objetivo de permitir que *softwares* desenvolvidos com a metodologia ágil FDD possam ser dimensionados com a técnica de medição funcional APF. A utilização conjunta dos dois conceitos visa oferecer subsídios para realização de estimativas de esforço e tempo no desenvolvimento de *software*, o que pode ajudar no gerenciamento de projetos e cumprimento de orçamento e cronograma.

O trabalho pode ser utilizando tanto para uso profissional quanto acadêmico, sendo que a proposta foi elaborada pensando, principalmente, na utilização de APF para realizar estimativas no gerenciamento de projetos desenvolvidos com metodologia ágil FDD, pois ambos são orientados a funcionalidade e o FDD não possui em suas definições um passo, fase ou processo que envolva a estimação de tamanho e do tempo necessário para o desenvolvimento do projeto como um todo.

Para o uso comercial, em fundamentação de contratos, alguns cuidados precisam ser tomados para que, por exemplo, um ALI utilizado em duas iterações não seja contabilizado

duas vezes no tamanho final do projeto, e conseqüentemente cobradas do cliente. Para fundamentação de contrato, e preciso identificar bem a inserção de uma nova funcionalidade ou a alteração de uma já existente, em cada iteração.

O objetivo principal do trabalho foi a realização de uma proposta que permita a utilização de FDD e APF no processo de desenvolvimento de software. Acredita-se que o objetivo foi atingido, uma vez que, uma proposta de utilização de APF e FDD foi apresentada, como resultado do trabalho realizado.

Uma aplicação prática, através de um estudo de caso da proposta, é indicada como possibilidade de trabalhos futuros. A proposta pode ser aplicada, por exemplo, em disciplinas do curso de Sistemas de Informação, como Modelagem e Desenvolvimento de Sistemas de Informação, nas quais uma equipe trabalha para a modelagem e desenvolvimento de um sistema de informação, respectivamente.

Em caso de aplicação da proposta, os resultados obtidos na primeira e na segunda fase deviam ser confrontados com o intuito validar os resultados obtidos com as medições nas duas fases do desenvolvimento. Também se recomenda a utilização da Contagem Detalhada da NESMA para validação das estimativas obtidas pela contagem Indicativa e pela Contagem Estimativa, nesse caso, a intenção seria medir o software depois de sua conclusão, e o resultado obtido deveria ser armazenado na base histórica e utilizado para calibrar estimativas futuras.

6. REFERÊNCIAS

ABES - Associação Brasileira de Empresas de *software*. **Mercado Brasileiro de *software*: panorama e tendências**, São Paulo, ed. 1, p. 4 jun. 2011. Disponível em: <http://www.abes.org.br/UserFiles/Image/PDFs/Mercado_BR2011.pdf>. Acesso em: 04 abr. 2012.

AGUIAR, Mauricio. Pontos de Função ou Pontos por Caso de Uso? Como Estimar Projetos Orientados a Objeto. TI Métricas. Janeiro de 2003. Disponível em: <http://www.bfpug.com.br/Artigos/UCP/Aguiar-Pontos_de_Funcao_ou_Pontos_por_Caso_de_Uso.pdf> Acesso em: 13 de abr. 2012.

ANDRADE, Edméia Leonor Pereira de. **Pontos de Casos de Uso e pontos de Função na Gestão de Estimativa de *software* Orientado a Objetos**. Dissertação(Mestrado em Gestão do Conhecimento e Tecnologia da Informação). 2004.Universidade Católica de Brasília, Brasília, 132 p. Disponível em: <<http://www.bfpug.com.br/Artigos/UCP/Tese%20Edmeia.zip>>. Acesso em 15 abr. de 2012.

BARBOSA, António, AZEVEDO, Bruno, PEREIRA, Bruno, CAMPOS, Pedro,SANTOS, Pedro. **Metodologia ágil: *Feature Driven Development***. Disponível em: <http://paginas.fe.up.pt/~aaguiar/es/artigos%20finais/es_final_22.pdf>. Acessado em: 04 jun. de 2012.

BARTIÉ, Alexandre. **Garantia da Qualidade de *software***. Rio de Janeiro: Campos, 2002. 291 p.

BRAGA, Antônio. **Análise de pontos de função**. Rio de Janeiro: Infobook, 1996. 188 p.

COAD. Peter, LEFEBVRE. Eric, DE LUCA. Jeff. *Feature Driven Development*. **Java Modeling in Color with UML**: introduces both modelling in colour and feature-driven development. 1 ed. Prentice Hall PTR, 1999.183-203 p.

DE LUCA. Jeff. The original processes. Disponível em: <<http://www.nebulon.com/articles/fdd/originalprocesses.html>>. Acessado em: 07 jun. de 2012.

_____. The Latest FDD Processes. Nebulon. Disponível em: <<http://www.nebulon.com/articles/fdd/latestfdd.html>> Acessado em: 07 jun. de 2012

DICIONÁRIO da língua portuguesa. Lisboa: Priberam Informática, 1998. Disponível em: <<http://www.priberam.pt/dIDLPO>>. Acesso em: 28 jun. 1999.

HAZAN, Claudia. Análise de Pontos de Função - Uma aplicação nas estimativas de tamanho de Projetos de *software*. **Engenharia de *software* Magazine**. Rio de

Janeiro, n. 02, p. 25-31, 2009. Disponível em
<http://kameha.devmedia.com.br/site/engsoft/02/ESM02_baixa.zip> Acesso em: 16 abr. de 2012.

_____. Implantação de Um Processo de Medições de *software*. Disponível em:
<http://www.bfpug.com.br/Artigos/Palestra%20_Medicoes%20Claudia%20Hazan.pdf>
Acessado em: 10 de jun. de 2012.

HEPTAGON. Tecnologia da Informação. *Feature Driven Development*. Disponível em:<<http://www.heptagon.com.br/fdd>>. Acessado em: 04 jun. de 2012.

KOTLINSKY, Alessandro, CECHELERO, Deise, SELZER, Jean Carlos. **Técnicas de Estimativa de *software***: Análise de Pontos de Função. Disponível em:
<<http://www.inf.furb.br/~egrahl/disciplinas/engeharia/material/FPA.pdf>>. Acessado em: 08 mai. 2012.

LONGSTREET Consulting Inc. Análise de Pontos de Função Aplicada a Novas Tecnologias. Tradução: AGUIAR. Mauricio. Disponível em:
<<http://www.softwaremetrics.com/files/New%20Technologies%20-%20Portuguese.pdf>>
Acesso em: 26 de mai. 2012

_____. **Fundamentals of Function Point Analysis**. *Software Metrics - Measurement that's on Target*. Disponível em
<<http://www.softwaremetrics.com/files/Fundamentals%20of%20Function%20Point%20Analysis.pdf>>. Acesso em 26 de maio de 2012.

Nebulon. Feature Driven Development Processes Disponível em:
<<http://www.nebulon.com/articles/fdd/download/fddprocessesUSLetter.pdf>> Acessado em: 07 jun. de 2012.

NESMA. **Análise de Pontos de Função para melhoria de *software***: diretrizes. Holanda. 2009. Disponível em:
<http://www.portaisgoverno.pe.gov.br/c/document_library/get_file?uuid=066903b6-39e9-44c4-833f-e7155a1c68c9&groupId=335215>Acessado em: 03 mai. de 2012.

_____. **Contagem antecipada de pontos de função** Disponível em:
<<http://www.nesma.nl/download/artikelen/CONTAGEM%20ANTECIPADA%20DE%20PONTOS%20DE%20FUN%C3%87%C3%83O.pdf>> Acessado em: 11 de jun. de 2012.

OSIEK, Bruno Adam. CHAOS Report: Métodos Ágeis Aumentam Taxa de Sucesso em Projetos. Disponível em: <<http://blog.scrumhalf.com.br/2011/02/metodos-ageis-impactam-positivamente-no-sucesso-de-projetos/>> Acesso em: 19 mai. de 2012

PRESSMAN, Rober S. **Engenharia de *software***. São Paulo: Makron Books, 1995. 1088 p.

PURIFICAÇÃO. Mauricio Cesar Santos da. **FDWS: Uma Metodologia para Gerência e Desenvolvimento de Projetos Ágeis de Business Intelligence**. 2010. 155 f. Universidade

Federal da Bahia. Instituto de Matemática. Departamento de Ciência da Computação. Bahia. 2010.

RETAMAL. Adail Muniz. *Feature Driven Development*. Disponível em: <<http://www.heptagon.com.br/files/FDD-Processos.pdf>>. Acesso em: 25 de mai. de 2012.

Santos. Gustavo. **Guia de medição**. Disponível em: <http://www.portaisgoverno.pe.gov.br/c/document_library/get_file?p_1_id=567493&folderId=404286&name=DLFE-25582.pdf> Acessado em: 26 de mai. de 2012.

The Standish Group. **Chaos Report**. 1995. Disponível em: <<http://www.projectsart.co.uk/docs/chaos-report.pdf>>. Acesso em: 04 abr. 2012.

_____. **Chaos Report**. Boston – Massachusetts Abr. 2009. Disponível em: <http://www1.standishgroup.com/newsroom/chaos_2009.php>. Acessado em: 04 abr. 2012

VAZQUEZ, Carlos Eduardo, SIMÕES, Guilherme Siqueira e ALBERT, Renato Machado. **Análise de Pontos de Função: Medição, estimativas e gerenciamento de projetos de software** 10. ed. São Paulo: Érica, 2010. 232 p.

VIEIRA, Everton Luiz. **Uso do Conceito de Passos Obrigatórios para Aprimorar o Processo de Contagem do Método “Pontos De Caso De Uso”**. Dissertação (Mestrado em Ciência da Computação). 2007. Universidade Federal de Santa Catarina – Programa de Pós-Graduação em Ciência de Computação, Florianópolis, 88 p. <Disponível em: <http://www.bfpug.com.br/Artigos/UCP/Everton%20Luiz%20Vieira.pdf>>

Visual Studio News. **Chaos Report**. Disponível em: <<http://blogs.msdn.com/b/dannawi/archive/2009/05/15/2009-standish-chaos-report-we-aresuccessful-in-the-failure.aspx?PageIndex=12&PostSortBy=MostViewed>>. Acessado em: 04 abr. 2012