

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CÂMPUS CORNÉLIO PROCÓPIO  
DIRETORIA DE PESQUISA E PÓS - GRADUAÇÃO  
PROGRAMA DE PÓS - GRADUAÇÃO EM INFORMÁTICA

ALESSANDRO SILVEIRA DUARTE

**PROCESSO DE SISTEMATIZAÇÃO DE AMBIENTES DE  
RESIDÊNCIA EM SOFTWARE BRASILEIROS**

DISSERTAÇÃO – MESTRADO

CORNÉLIO PROCÓPIO

2015

ALESSANDRO SILVEIRA DUARTE

**PROCESSO DE SISTEMATIZAÇÃO DE AMBIENTES DE  
RESIDÊNCIA EM SOFTWARE BRASILEIROS**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná – UTFPR como requisito parcial para a obtenção do título de “Mestre Profissional em Informática”.

Orientador: Prof. Dr. José Augusto Fabri

CORNÉLIO PROCÓPIO

2015

---

Dados Internacionais de Catalogação na Publicação

---

- D812 Duarte, Alessandro Silveira  
Processo de sistematização de ambientes de residência em software brasileiros / Alessandro Silveira Duarte. – 2015.  
134 f. : il. ; 30 cm
- Orientador : José Augusto Fabri .  
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Informática. Cornélio Procópio, 2015.  
Referências: p. 119 – 121.
1. Software - Desenvolvimento. 2. Gestão do conhecimento. 3. Programa Brasileiro da Qualidade e Produtividade em Software. 4. Informática – Dissertações. I. Fabri, José Augusto, orient. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Informática. III. Título.

CDD (22. ed.) 004



**Título da Dissertação Nº 04:**

**“Processo de sistematização de ambientes de residência em software brasileiros”.**

por

**Alessandro Silveira Duarte**

Orientador: **Prof. Dr. José Augusto Fabri**

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM INFORMÁTICA – Área de Concentração: Computação Aplicada, pelo Programa de Pós-Graduação em Informática – PPGI – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 10h do dia 08 de maio de 2015. O trabalho foi APROVADO pela Banca Examinadora, composta pelos professores:

---

Prof. Dr. José Augusto Fabri  
(Presidente)

---

Prof. Dr. Marcelo Schneck de Paula Pessôa  
(USP-SP)

---

Prof. Dr. Luiz Ricardo Begosso  
(FEMA-SP)

Visto da coordenação:

---

**Carlos Nascimento Silla Junior**  
Coordenador do Programa de Pós-Graduação em Informática  
UTFPR Câmpus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Programa.

Dedico este trabalho a minha esposa Viviane e aos meus filhos  
Isabella e Matheus.

## **AGRADECIMENTOS**

Ao Professor Dr. José Augusto Fabri, pela dedicação e orientação deste trabalho.

Ao Professor Dr. Alexandre L'Erario pelas valiosas contribuições.

Aos amigos Fernando Cesar de Lima e José Antonio Gonçalves pelas várias discussões e incentivos, durante todas as fases que compõem este desafio chamado Mestrado Profissional em Informática.

Agradecendo a estes amigos, estendo os meus agradecimentos a todos os outros que diretamente ou indiretamente também fizeram parte deste processo.

I call my field “knowledge management” but you can’t manage knowledge. Nobody can. What you do - what a company does - is manage the environment that optimizes knowledge.

(PRUSAK, Laurence, 1997).

“Eu denomino meu campo de Gestão do Conhecimento, mas você não pode gerenciar conhecimento. Ninguém pode. O que você pode fazer - o que a empresa pode fazer - é gerenciar o ambiente que otimize o conhecimento”.

(PRUSAK, Laurence, 1997)

## RESUMO

DUARTE, Alessandro Silveira. **Processo de sistematização de ambientes de residência em software brasileiros**. 2015. 134 f. Dissertação de Mestrado (Mestrado Profissional em Informática), Programa de Pós - Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

No Brasil existe um número limitado de empresas com certificações de qualidade, a baixa qualificação na mão de obra pode confirmar este cenário. Uma equipe altamente qualificada pode ser uma das formas de se melhorar a qualidade das empresas e conseqüentemente a sua competitividade. Este trabalho apresenta um delineamento de um processo de sistematização de ambientes de residência em software, cujo objetivo é propiciar qualificação para os profissionais de empresas de software. As teorias de gestão de conhecimento e residência em software foram utilizadas em um método experimental para validar a sistematização apresentada neste trabalho. Foram executados quatro experimentos: três executados com alunos de programa de mestrado e um em uma empresa real. A execução dos experimentos evidência que é possível sistematizar um ambiente de residência em software. A contribuição deste trabalho é a validação do processo e conseqüentemente o ganho e a transferência de conhecimentos entre os funcionários de uma empresa de software.

**Palavras-chave:** Residência em Software. Processo de Desenvolvimento de Software. Treinamento em Transferência de Conhecimento.



## ABSTRACT

DUARTE, Alessandro Silveira. **Processo de sistematização de ambientes de residência em software brasileiros**. 2015. 134 f. Dissertação de Mestrado (Mestrado Profissional em Informática), Programa de Pós - Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

Brazil has a limited number of organizations with quality certification, and low unskilled workers can confirm this scenario. A well qualified team can be one of the ways to improve the quality of organizations and consequently increase their competitiveness. This work presents the process of systematizing a software residency environment, whose aim is to provide qualification for employees in software organizations. The knowledge management and software residency theories were adapted in an experimental method to validate the systematization proposed. Four experiments were conducted: three were executed with computer science master students and one within a real company. The performed experiments show evidences that it is possible to systematize a residency software environment. The contribution of this work is the validation of the process and, as a consequence, the knowledge gain and transfer among employees of a software organization.

**Keywords:** Software Residency. Software Development Process. Training Knowledge Transfer

## LISTA DE ILUSTRAÇÕES

Gráfico 1 - Resultado da avaliação pré-residência do segundo experimento.....	80
Gráfico 2 - Resultado da avaliação pós-residência do segundo experimento.....	81
Gráfico 3 - Resultado das avaliações pré e pós-residência do segundo experimento.....	81
Gráfico 4 - Resultado da avaliação pré-residência do terceiro experimento.....	92
Gráfico 5 - Resultado da avaliação pós-residência do terceiro experimento.....	93
Gráfico 6 - Resultado das avaliações pré e pós-residência do terceiro experimento.....	93
Gráfico 7 - Tempo investido por atividade do processo, quatro semanas.....	108

## LISTA DE FIGURAS

Figura 1 - O processo de gestão dos conhecimentos e suas capacitações.....	21
Figura 2 - Modelo de gestão de conhecimento gerado para a empresa.....	22
Figura 3 - Mapa conceitual apresentado como parte do conhecimento necessário para a execução da tarefa estabelecer primeiro contato com o cliente.....	23
Figura 4 - Modelo de gestão de conhecimento para a administração pública.....	25
Figura 5 - Ciclo de gestão de conhecimento – O Ciclo do KDCA.....	26
Figura 6 - Fases da residência médica do hospital Albert Einstein.....	30
Figura 7 - Modelo de Avaliação gerado a partir dos pressupostos estabelecidos pela norma 15504 (2004), SCAMPI (2006) e MA-MPS.BR (2006).....	43
Figura 8 - Modelo para classificação de ambientes para residência em software....	44
Figura 9 - Indicadores utilizados na Classificação de Ambientes para Residência em Software - PCP: Planejamento e Controle de Processo/Produção.....	46
Figura 10 - Operacionalização de variáveis.....	52
Figura 11 - Processo de Sistematização de Ambientes de Residência em Software Brasileiro.....	59
Figura 12 - Nova versão do Processo de Sistematização de Ambientes de Residência em Software Brasileiros.....	72
Figura 13 - Sugestão de alteração na ordem de execução dos estados 1, 2 e 3.....	82
Figura 14 - Estrutura de distribuições abordada pela empresa.....	85
Figura 15 - Processo de Sistematização com alterações nos estados 1, 2 e 3.....	87
Figura 16 - Diagrama de classes - parte de avaliação.....	89
Figura 17 - Divisão do estado 3 em cinco partes.....	99
Figura 18 - Especificações do boneco.....	101
Figura 19 - Processo (reestruturado) de produção de software da empresa.....	103
Figura 20 - Estrutura de dados utilizada para organizar informações do processo de construção de bonecos.....	104
Figura 21 - Estrutura de dados utilizada para organizar informações do processo de construção de software.....	105
Figura 22 - Nova composição do Processo de Sistematização.....	109
Figura 23 - Última versão do Processo de Sistematização.....	116

## LISTA DE QUADROS

Quadro 1 - Residência médica e residência em software.....	32
Quadro 2 - Estrutura do estudo de caso.....	35
Quadro 3 - Caracterização do processo, primeira parte.....	39
Quadro 4 - Demais caracterizações do processo de software, segunda parte.....	41
Quadro 5 - Classificação dos ambientes de residência.....	48
Quadro 6 - Apresentação da residência em software.....	63
Quadro 7 - Descrição do projeto genoma.....	65
Quadro 8 - Missão dos residentes.....	65
Quadro 9 - Entradas de dados.....	65
Quadro 10 - Saída de dados.....	66
Quadro 11 - Restrições.....	66
Quadro 12 - Andamento do desenvolvimento do projeto.....	66
Quadro 13 - Atividades que os residentes desenvolveram.....	67
Quadro 14 - Um exemplo de resultado da avaliação do tutor.....	67
Quadro 15 - Questionário para avaliação do primeiro experimento.....	69
Quadro 16 - Tabulação das respostas dos residentes.....	70
Quadro 17 - Observação direta do orquestrador e do tutor.....	70
Quadro 18 - Estados da máquina de estados.....	72
Quadro 19 - Transições da máquina de estados.....	73
Quadro 20 - Apresentação da definição da residência em software.....	77
Quadro 21 - Definição do ambiente de execução da residência em software.....	78
Quadro 22 - Atividade de execução do projeto.....	79
Quadro 23 - Atividade de gerenciamento e controle do software desenvolvido na residência.....	79
Quadro 24 - Estados da máquina de estados do terceiro experimento.....	87
Quadro 25 - Transições da máquina de estados do terceiro experimento.....	88
Quadro 26 - Questões da avaliação pré-residência aplicada do terceiro experimento.....	89
Quadro 27 - Roteiro utilizado nos levantamentos de informações da empresa.....	97
Quadro 28 - Cronograma resumido para execução da análise.....	97
Quadro 29 - Análise de SWOT da empresa.....	98
Quadro 30 - Atividades, tarefas, artefatos (entrada e saída), ferramentas e habilidades do processo da empresa.....	103
Quadro 31 - Modelo do LOG utilizado na empresa.....	106
Quadro 32 - Exemplo de um LOG preenchido.....	107
Quadro 33 - Súmula dos experimentos.....	111
Quadro 34 - Contribuições para a consolidação do processo de residência.....	113
Quadro 35 - Áreas de residência em software.....	117

## LISTA DE SIGLAS

BPMN	<i>Business Process Model and Notation</i>
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
DER	Diagrama Entidade-Relacionamento
EAP	Estrutura Analítica do Projeto
GTI	Grupo de Tecnologia em Informação
IHC	Interface Homem-Computador
JPA	<i>Java Persistence API</i>
PML	<i>Process Modeling Language</i>
PPGI	Programa de Pós-Graduação em Informática
RAD	<i>Rapid Application Development</i>
SWOT	<i>Strengths, Weakness, Opportunities, Threats</i>
TI	Tecnologia da Informação
UML	<i>Unified Modeling Language</i>
URA	Unidade de Resposta Audível
UTFPR	Universidade Tecnológica Federal do Paraná
XHTML	<i>eXtensible Hypertext Markup Language</i>

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>14</b>
<b>2 GESTÃO DE CONHECIMENTO.....</b>	<b>17</b>
2.1 REFERENCIAL TEÓRICO SOBRE GESTÃO DE CONHECIMENTO.....	17
2.2 CONHECIMENTO TÁCITO.....	19
2.3 CONHECIMENTO EXPLÍCITO.....	19
2.4 MODELOS DE GESTÃO DE CONHECIMENTO.....	20
2.4.1 Modelo de Gestão de Conhecimento de TRINDADE.....	22
2.4.2 Modelo de Gestão de Conhecimento de FABRI et al. ....	22
2.4.3 Modelo de Gestão de Conhecimento de BATISTA.....	24
<b>3 RESIDÊNCIA EM SOFTWARE .....</b>	<b>28</b>
3.1 REFERENCIAL TEÓRICO SOBRE RESIDÊNCIA EM SOFTWARE.....	28
3.2 AMBIENTES DE RESIDÊNCIA EM SOFTWARE NO BRASIL.....	33
3.2.1 Caracterização dos Ambientes.....	35
3.2.2 Caracterização do Processo de Software.....	38
3.2.3 Demais Caracterizações do Processo de Software.....	41
3.3 MODELO PARA CLASSIFICAÇÃO DE AMBIENTES.....	42
<b>4 MÉTODOS E PROCEDIMENTOS.....</b>	<b>49</b>
4.1 FORMULAÇÃO DO PROBLEMA.....	50
4.2 CONSTRUÇÃO DA HIPÓTESE.....	51
4.3 OPERACIONALIZAÇÃO DAS VARIÁVEIS.....	51
4.4 DEFINIÇÃO DO PLANO EXPERIMENTAL.....	53
4.4.1 Determinação dos Sujeitos.....	53
4.4.1.1 Sujeitos do primeiro experimento.....	53
4.4.1.2 Sujeitos do segundo experimento.....	54
4.4.1.3 Sujeitos do terceiro experimento.....	54
4.4.1.4 Sujeitos do quarto experimento.....	54
4.4.2 Determinação do Ambiente.....	55
4.4.2.1 Ambiente do primeiro experimento.....	55
4.4.2.2 Ambiente do segundo experimento.....	55
4.4.2.3 Ambiente do terceiro experimento.....	55
4.4.2.4 Ambiente do quarto experimento.....	56
4.5 COLETA DE DADOS.....	56
4.5.1 Coleta de Dados do Primeiro Experimento.....	56
4.5.2 Coleta de Dados do Segundo Experimento.....	56
4.5.3 Coleta de Dados do Terceiro Experimento.....	57
4.5.4 Coleta de Dados do Quarto Experimento.....	57

<b>5 PROPOSTA DO PROCESSO DESISTEMATIZAÇÃO DE AMBIENTES DE RESIDÊNCIA EM SOFTWARE BRASILEIROS.....</b>	<b>58</b>
5.1 DESCRIÇÃO DO PRIMEIRO EXPERIMENTO.....	59
5.1.1 Aplicação do Primeiro Experimento.....	62
5.1.2 Avaliação do Primeiro Experimento.....	68
5.2 DESCRIÇÃO DO SEGUNDO EXPERIMENTO.....	71
5.2.1 Aplicação do Segundo Experimento.....	75
5.2.2 Avaliação do Segundo Experimento.....	80
5.3 DESCRIÇÃO DO TERCEIRO EXPERIMENTO.....	83
5.3.1 Aplicação do Terceiro Experimento.....	86
5.3.2 Avaliação do Terceiro Experimento.....	92
5.4 QUARTO EXPERIMENTO.....	94
<b>6 ANÁLISE E INTERPRETAÇÃO DOS DADOS.....</b>	<b>94</b>
<b>7 APRESENTAÇÃO DAS CONCLUSÕES.....</b>	<b>114</b>
<b>REFERÊNCIAS.....</b>	<b>119</b>
<b>ANEXO A - MAPAS MENTAIS.....</b>	<b>122</b>
<b>ANEXO B - ENDEREÇOS DOS CONTEÚDOS.....</b>	<b>125</b>
<b>ANEXO C - AVALIAÇÃO PRÉ E PÓS-RESIDÊNCIA APLICADA.....</b>	<b>128</b>
<b>ANEXO D - DOCUMENTO DO PROCESSO DE SOFTWARE.....</b>	<b>133</b>

## 1 INTRODUÇÃO

Atualmente poucas empresas brasileiras produzem software, ou prestam serviços, com padrões de qualidade reconhecidos por algum modelo. Esta afirmação pode ser constatada ao ser analisada a quantidade de empresas certificadas no Modelo Referencial baseado em Maturidade, Capacidade e Integração (CMMI), CMMI-DEV e CMMI-SVC, o Brasil possui 221 empresas certificadas. Para efeitos comparativos, a China possui 3.316, os Estados Unidos da América 2.186 e a Índia 959<sup>1</sup>.

Este fato, aliado à alta carga tributária e a deficiência na formação da mão de obra na área de tecnologia da informação, mais precisamente na área de engenharia de software, caracteriza-se como fonte inibidora no processo de expansão externa do Brasil neste setor. Universidades, empresas e governo devem desenvolver mecanismos que alterem este cenário.

O *Standish Group International* (2013), mostra em seus estudos<sup>2</sup> que, a atividade de gestão de projetos é uma importante ferramenta para as empresas produtoras de software. Ao analisar os números, vide Tabela 1, apresentados pelo *Standish Group International*, é possível constatar que aproximadamente dois terços dos projetos de software não obtiveram o sucesso esperado.

**Tabela 1 – Situação de Desenvolvimento de Projeto de Software de 2004 a 2012**

	2004	2006	2008	2010	2012
Sucesso	29%	35%	32%	37%	39%
Falharam	18%	19%	24%	21%	18%
Mudaram	53%	46%	44%	42%	43%

**Fonte: CHAOS MANIFESTO 2013, Think Big, Act Small, p.1**

O *Standish Group International* classifica os projetos em três categorias:

a) sucesso: projetos que terminaram no prazo estipulado, dentro do orçamento e com escopo completo;

<sup>1</sup> Acessível em <http://cmminstitute.com/wp-content/uploads/2014/05/Maturity-Profile-Ending-March-2014.pdf>

<sup>2</sup> Acessível em <http://versionone.com/assets/img/files/ChaosManifesto2013.pdf>



b) falharam: projetos que foram cancelados; e

c) mudaram: projeto que atrasaram, estouraram ou tiveram o escopo reduzido.

Os dados mostrados refletem um cenário preocupante, pois, o número de projetos que obtiveram sucesso é menor que a soma dos que falharam e mudaram. Os dados de 2.012 não diferem significativamente dos de 2.004, o que leva a pensar que futuramente a situação não será diferente, se comparada a atual.

Com o objetivo de minimizar o cenário apresentado, algumas universidades e empresas criaram o conceito de Residência em Software. A residência em software pode ser comparada à residência médica. Neste tipo de atividade, ambientes são criados e/ou preparados para a participação dos alunos/residentes no intuito de promover especializações na área de engenharia de software.

Considerando o baixo número de empresas que desenvolvem software ou prestam serviços reconhecidos por algum modelo e o baixo número de projetos caracterizados pelo CHAOS como projetos que alcançaram sucesso, e considerando também a deficiência na formação da mão de obra na área de tecnologia da informação, mais precisamente na área de engenharia de software, **este trabalho tem como objetivo propor um Processo de Sistematização de Ambientes de Residência em Software Brasileiros.**

Além dos motivos já citados, a proposta do Processo de Sistematização levou em consideração os resultados positivos alcançados com a realização de quatro experimentos. Os profissionais treinados na residência em software, criada a partir dos pressupostos citados do processo de sistematização, alcançaram resultados significativos na questão de ganho de conhecimento. Os resultados sinalizam que, o Processo de Sistematização pode ter relevância para o setor produtivo de software, uma vez que contribui para a qualificação de mão de obra. Estes indícios reforçam a relevância desta contribuição e constituem a principal motivação para a realização deste trabalho.

O embasamento teórico deste trabalho (capítulos 2 e 3) ampara a construção de um processo de sistematização de residência em software.

Espera-se que a aplicação do Processo de Sistematização de Ambientes de Residência em Software Brasileiros possa contribuir minimamente para diminuir os números inerentes à qualidade do processo (apresentado no

relatório do CMMI, *Maturity Profile Reports*) e do projeto (apresentado no relatório do *Chaos Manifesto 2013, Think Big, Act Small*).

Dentro do contexto apresentado, este trabalho foi desdobrado em sete capítulos. Os capítulos 2 e 3 apresentam os referenciais bibliográficos sobre gestão de conhecimento e residência em software respectivamente. Estes apresentam os principais constructos teóricos que embasam a contribuição relatada no capítulo 7.

O método empregado para a validação do processo é apresentado no capítulo 4. Nele é delineada a pesquisa experimental como instrumento de pesquisa principal deste trabalho. A aplicação do protocolo de pesquisa apresentado neste é refletida no capítulo 5.

O capítulo 5, Proposta do Processo de Sistematização de Ambientes de Residência em Software Brasileiros apresenta três experimentos desenvolvidos em ambientes acadêmicos e um desenvolvido em ambiente empresarial, estes experimentos, juntamente com o embasamento teórico dos capítulos 2 e 3 alicerçaram a consolidação do Processo de Sistematização de Ambientes de Residência em Software Brasileiros.

A análise e interpretação dos dados são apresentadas no capítulo 6. Neste uma compilação dos dados levantados nos capítulos 5 são confrontados e examinados.

O capítulo 7 apresenta as análises finais, conclusões e contribuições deste trabalho, seguido pelas referências bibliográficas. Os resultados aqui apresentados, juntamente com os capítulos 2 e 3 compõem o conjunto de conhecimentos necessários para a consolidação do processo de sistematização.

## 2 GESTÃO DE CONHECIMENTO

Este capítulo é segmentado em quatro partes. A primeira tem como objetivo apresentar o referencial teórico sobre a gestão de conhecimento. A segunda foca na apresentação do conhecimento tácito. Já a terceira foca na apresentação do conhecimento explícito. E por fim, o capítulo apresenta três modelos de gestão de conhecimento.

Um dos problemas identificados durante a concepção deste trabalho foi a questão que tange a capacidade de mensurar o quanto evolui o nível de conhecimento que os residentes atingem quando treinados. A eficácia do processo de residência é determinada pela evolução dos residentes na capacidade de solucionar problemas semelhantes quando submetidos em um cenário real. Neste sentido, a gestão do conhecimento fundamenta a geração, a codificação, a disseminação e a apropriação de conhecimento. Estes elementos são fundamentais para mensurar as relações causais entre os envolvidos em uma residência em software.

### 2.1 REFERENCIAL TEÓRICO SOBRE GESTÃO DE CONHECIMENTO

A gestão de conhecimento caracteriza-se como a somatória das experiências e aprendizagens alcançadas pelo homem no decorrer de sua vida (TAVARES, 2004).

TERRA (1999) mostra que conhecimento é a habilidade do ser humano de escolher e compreender informações, obtidas por meio de aprendizagem proveniente de conceitos cognitivos, comportamentais e experientialistas.

RAMASUBRAMANIAN; JAGADEESAN (2002) e TRINDADE (2006) buscam organizar a definição de conhecimento como a união de especialidades, informações e experiências acumuladas ao longo do tempo. É importante observar a necessidade de se divulgar, manter e organizar essas informações.

Segundo RUS; LINDVALL (2002), a ideia de conhecimento está relacionada a dado e informação, cada um deles com seu grau de detalhamento. Dados podem ser considerados como algo sem grande significado e a ser trabalhado, sua relevância ocorrerá quando uma informação é gerada a partir deles. A informação torna-se importante quando as pessoas precisam tomar decisões ou

realizar determinadas tarefas, uma vez que, o conhecimento é oriundo de um conjunto de partes que interagem entre si, reflexão, experiência, percepção e ambiente.

DAVENPORT et al. (1998) tratam a gestão de conhecimento como instrumento de criação, distribuição, compartilhamento, aquisição e compreensão do conhecimento de uma empresa. Os autores salientam que:

- a) a criação está relacionada a capacidade de inovar e resolver problemas de cada pessoa;
- b) a distribuição se dá quando o conhecimento está preservado e escrito;
- c) o compartilhamento, atividade fundamental, quando o acesso é fácil para quem precisa;
- d) a aquisição por meio de leitura de manuais, de procedimentos, de documentos, de registro de boas práticas e participações em reuniões; e
- e) a compreensão alcançada com a participação em capacitações e em redes de profissionais especializados.

As atividades citadas transferem o conhecimento que, antes poderia ser considerado no nível individual, para o coletivo. A gestão de conhecimento aplicada na empresa traz benefícios imediatos, tais como: redução de custos; otimização no cumprimento de cronogramas; a não ocorrência do retrabalho; e aumento na qualidade e confiabilidade de seus produtos, pois, os erros tendem a diminuir à medida que os processos anteriores que obtiveram sucesso são conhecidos e praticados.

As tomadas de decisões que antes eram baseadas em conhecimentos individuais passam a ser caracterizadas em um nível colaborativo. Tem-se aqui, a gestão de conhecimento como estratégia competitiva, pois a ocorrência de tomadas de decisões corretas tende a aumentar expressivamente.

Concluída a contextualização da gestão de conhecimento, o autor deste trabalho considera importante uma abordagem sobre conhecimento tácito, conhecimento explícito e modelos de gestão de conhecimento, pois os mesmos são geradores de novos meios de preservação e transferência de conhecimentos, fato este relevante dentro de um ambiente de residência em software.

## 2.2 CONHECIMENTO TÁCITO

Para NONAKA; TAKEUCHI (1995) conhecimento tácito é aquele que o homem adquire ao longo de sua existência, pautado em suas experiências. Fatores tais como: religiosos, sociais, valores individuais, intuitivos, experimentais, perspectivas e motivações tendem a colaborar para o desenvolvimento do mesmo.

Considerando estes fatores, o conhecimento referido apresenta uma característica importante, ele é individual, o que o torna difícil de compreender, transferir e compartilhar.

POLANYI (1966) diz que o conhecimento tácito se caracteriza como um conhecimento organizacional e que, a sua construção ocorre de uma forma oculta. O autor salienta que é possível observá-lo por meio de suas manifestações e que, a necessidade de compreensão das mesmas se faz necessária em todo o seu processo de exploração.

COHEN et al. (1996) e KOGUT (1991) afirmam em seus pressupostos que existe uma grande dificuldade em se copiar os conhecimentos classificados como tácitos. A dificuldade está na complexidade da gestão deste tipo de conhecimento. Os autores destacam ainda que, por razão da complexidade da gestão, os conhecimentos tácitos possuem um lado positivo, as chances de ocorrências de vazamentos para os concorrentes são menores.

Para TSOUKAS (2003) a concepção que influencia o conhecimento tácito é objeto de estudo de vários autores, embora as definições sejam de fontes semelhantes, as interpretações variam. HALDIN-HERRARD (2000) diz que as ambiguidades do conhecimento tácito são devidas as diferenças na interpretação dos elementos ligados a articulação, a apropriação e a externalização. Já para GRANT (2002) e STEWART (2001) mesmo com as ambiguidades, o conhecimento tácito tem seus pontos fortes, elevado grau de valor e importância para pessoas e para empresas.

## 2.3 CONHECIMENTO EXPLÍCITO

O conhecimento explícito se constitui como o oposto do tácito, NONAKA; TAKEUCHI (1995) apresentam as características diferenciadas relacionadas às competências, às habilidades e aos conhecimentos de uma pessoa

ou empresa que podem ser transferidos, segundo o autor, facilmente sem a necessidade de grandes observações, uma vez que o mesmo pode ser documentado em manuais, procedimentos, guias de consulta rápida, revistas e livros. Este conhecimento também é classificado pelos autores como conhecimento codificado.

Para LYNN; AKGÜK (2000) o conhecimento explícito se caracteriza como declarativo, é o “saber sobre” o conhecimento que pode ser materializado. Os autores consideram também que, o conhecimento explícito é representado por objetivos, sentimentos e regras.

## 2.4 MODELOS DE GESTÃO DE CONHECIMENTO

Nesta seção serão apresentados modelos de gestão de conhecimento criados por TRINDADE (2006), FABRI et al. (2011) e BATISTA (2012).

TRINDADE (2006) mostra o modelo gestão de conhecimento organizado em 4 atividades:

- a) geração;
- b) codificação;
- c) disseminação; e
- d) apropriação.

FABRI et al. (2011) descrevem o modelo de gestão de conhecimento apresentando as atividades:

- a) capturação;
- b) confecção de mapa conceitual preliminar;
- c) estruturação;
- d) confecção de mapa conceitual armazenável;
- e) armazenamento;
- f) preservação em base de conhecimento;
- g) transferência; e
- h) verificação da compreensão dos conhecimentos.

BATISTA (2012) propõe em seu trabalho um modelo de gestão de conhecimento para a administração pública. Participam ativamente no que o autor considera seu processo de gestão, as atividades:

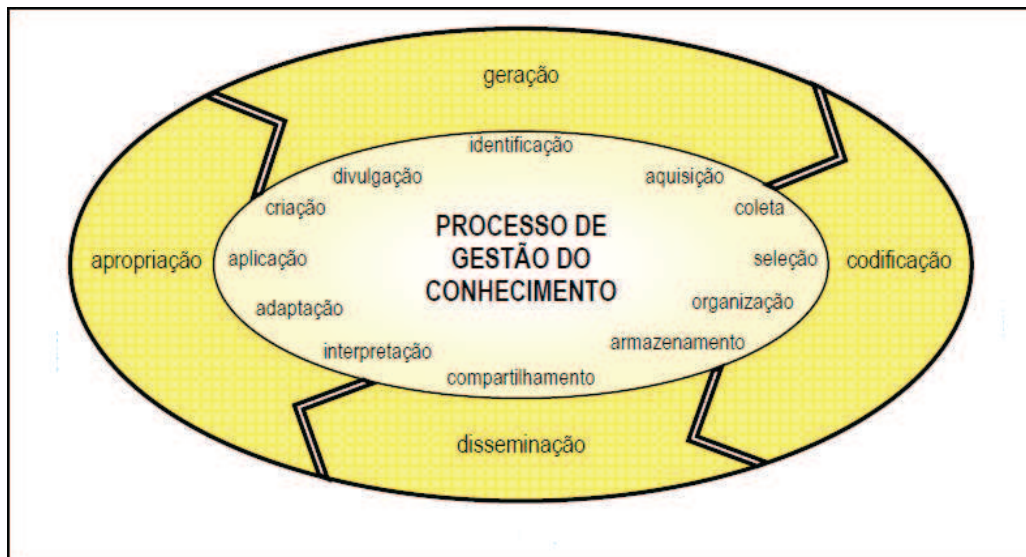
- a) identificar;

- b) criar;
- c) armazenar;
- d) compartilhar; e
- e) aplicar conhecimentos;

Nas próximas seções serão apresentados os modelos.

#### 2.4.1 Modelo de Gestão de Conhecimento de TRINDADE

O modelo apresentado por TRINDADE (2006), que pode ser observado na Figura 1, apresenta a gestão de conhecimento organizada em quatro atividades: geração, codificação, disseminação e apropriação.



**Figura 1 - O processo de gestão dos conhecimentos e suas capacitações.**  
**Fonte: Adaptado de TRINDADE (2006)**

- a) a geração se constitui na capacidade individual ou corporativa para criação, divulgação, identificação e até mesmo aquisição de conhecimento;
- b) a codificação inicia-se com a coleta, seleção, organização e armazenamento de conhecimento;
- c) a disseminação ocorre quando o compartilhamento e a interpretação do conhecimento geram valores à gestão e às circunstâncias na qual é aplicado; e
- d) a apropriação de experiências passadas são ações que colaboram com as futuras decisões, a aplicação dessas experiências representa uma chance de nova aprendizagem, por consequência cria-se um novo conhecimento.

Tal modelo explicita de forma clara e objetiva os passos para gestão de conhecimento. É importante salientar que os referidos passos são caracterizados de forma assíncrona. Este modelo fornece às empresas, um mecanismo de gestão no qual os colaboradores são os principais interlocutores capazes de desenvolver, aprimorar e aplicar práticas e técnicas já conhecidas ou recém-criadas.

#### 2.4.2 Modelo de Gestão de Conhecimento de FABRI et al.

FABRI et al. (2011) descrevem o modelo de gestão de conhecimento adotado por uma empresa de produção de software e como ocorre a transferência de conhecimento entre a matriz e a filial, a Figura 2 ilustra o referido modelo.

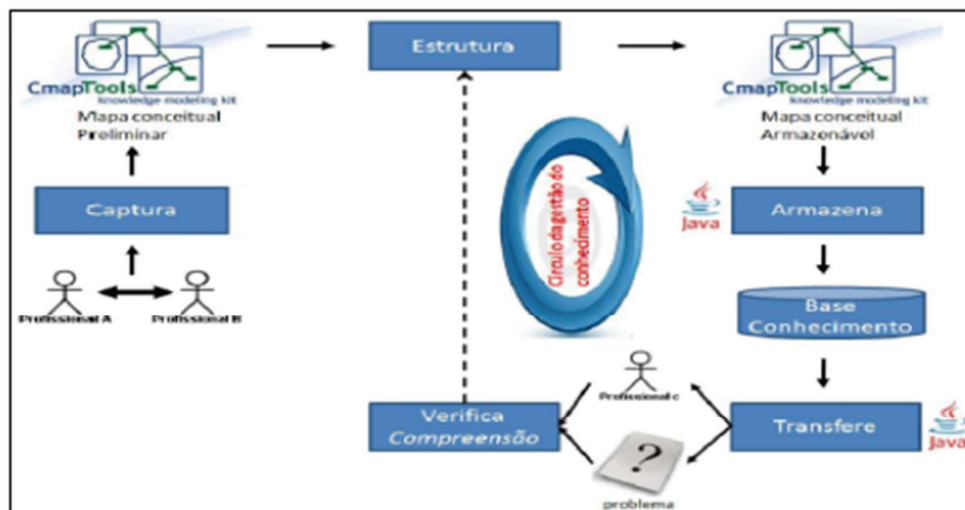


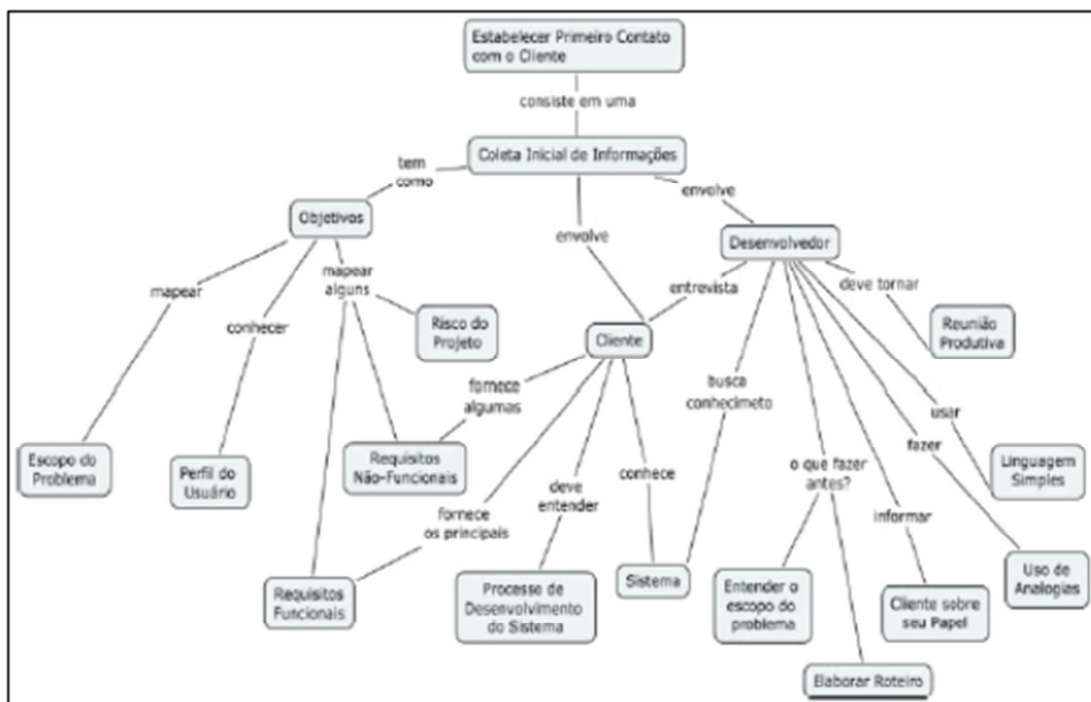
Figura 2 - Modelo de gestão de conhecimento gerado para a empresa  
Fonte: FABRI et al. (2011)

O referido modelo (vide Figura 2) destaca a forma que, a empresa utilizou para obter sucesso na transferência de conhecimento, trata as atividades de captura, confecção de mapa conceitual preliminar, estruturação, confecção de mapa conceitual armazenável, armazenamento, preservação em base de conhecimento, transferência e verificação da compreensão dos conhecimentos. Os autores salientam que tais atividades formam o círculo da gestão dos conhecimentos.

A execução do modelo tem como ponto de partida a ideia de que dois profissionais, A e B, estejam trabalhando em um projeto de software, a interpretação do contexto do projeto tem como uma das formas a técnica de *brainstorm*. Após a aplicação da técnica, os profissionais modelam um mapa conceitual, representando assim o conhecimento dentro de um determinado domínio.



Na atividade de estruturação, o engenheiro de conhecimento verifica a semântica, a sintaxe e a formatação no mapa (vide Figura 3). O passo seguinte caracteriza-se na utilização de uma ferramenta para o armazenamento do mapa na base de conhecimento. O profissional C, que receberá a transferência de conhecimentos gerados pelos profissionais A e B, após consulta a base de conhecimento, tem que estruturar um novo *brainstorm* para capturar informações de um novo projeto a ser desenvolvido pela filial.



**Figura 3 - Mapa conceitual apresentado como parte do conhecimento necessário para a execução da tarefa estabelecer primeiro contato com o cliente**  
 Fonte: FABRI et al. (2011)

Os resultados gerados pelo profissional C serão analisados pelo engenheiro de conhecimento, caso não atenda os critérios de qualidade, o mapa conceitual terá que ser refeito, dado a este fato, o círculo de gestão do conhecimento e a base de conhecimento serão atualizados.

Para os autores, o modelo de gestão de conhecimento proporcionou resultados qualitativos e quantitativos que podem ser observados:

a) aumento da qualidade do produto e do processo: com o desenvolvimento dos mapas conceituais, os profissionais que atuam na matriz realizaram algumas alterações no processo de produção de software, várias delas promoveram um ganho de qualidade e produtividade, o tempo para o desenvolvimento dos artefatos

(produtos) foi reduzido e a qualidade do software e a satisfação do cliente foi maximizada;

b) aumento da qualidade dos artefatos (produtos) gerados nas filiais também é um ponto a ser destacado com a aplicação do modelo. A cada círculo da gestão do conhecimento (vide Figura 2) realizado, os profissionais recém-contratados internalizavam o processo de software e os níveis produtividade e qualidade aumentavam;

c) as orientações sintáticas e semânticas aplicadas aos mapas foram de fundamental importância para o desenvolvimento das filiais, nesta etapa o papel do engenheiro do conhecimento foi de extrema importância, suas orientações foram eficientes e eficazes para o sucesso pretendido pela empresa;

d) a estratégia em utilizar os mapas conceituais está embasada em uma consulta realizada pelos consultores que participaram ativamente no desenvolvimento do processo. Nesta pesquisa, um mapa conceitual foi apresentado a oitenta e nove profissionais da área produtiva de software (engenheiros, programadores, analista, testadores, gerentes de projetos), 83% dos profissionais fizeram a leitura do mapa rapidamente e não tiveram problemas sobre a sua compreensão; e

e) por fim, o número de artefatos rejeitados, pelos clientes das filiais, diminuíram de acordo com a evolução qualitativa do conhecimento internalizado pelos profissionais e materializado nos mapas. Já no quarto mês de consultoria a quantidade de artefatos rejeitados foi 80% menor quando comparado ao primeiro mês de operação.

#### 2.4.3 Modelo de Gestão de Conhecimento de BATISTA

O modelo proposto por BATISTA (2012) pode ser observado na Figura 4, apresenta o modelo de gestão de conhecimento para a administração pública e como se dá a sua composição:

a) direcionadores estratégicos: visão, missão, objetivos estratégicos, estratégias e metas;

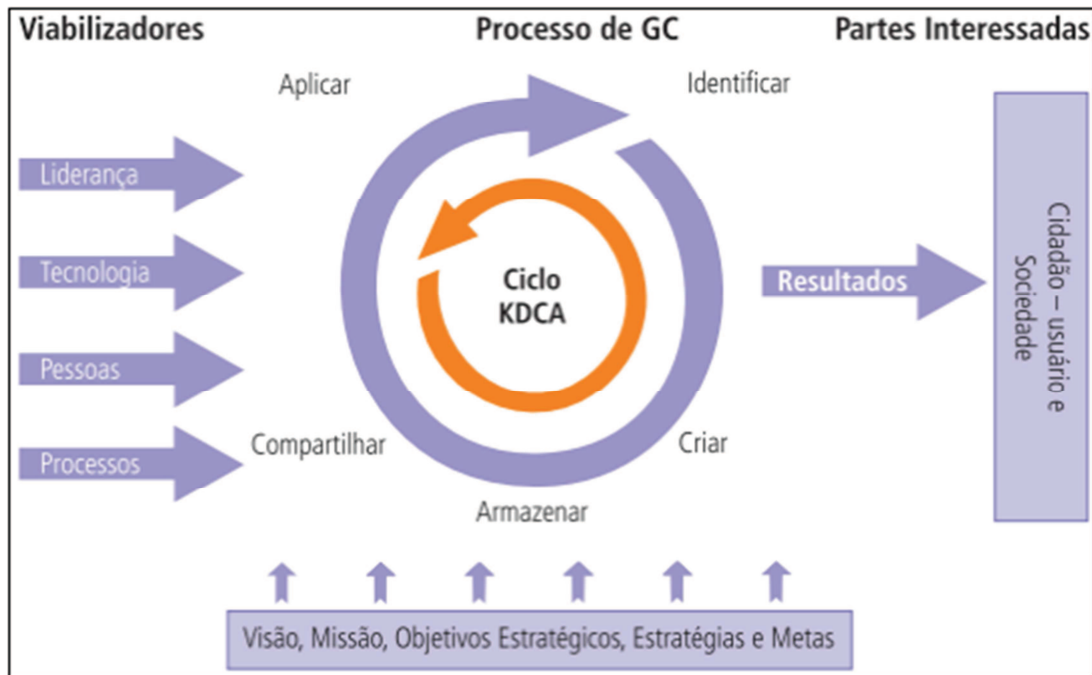
b) viabilizadores: liderança, tecnologia, pessoas e processos;

c) processo de gestão de conhecimento: identificar, criar, armazenar, compartilhar e aplicar;

d) ciclo do KDCA: *knowledge, do, check e act*;

e) resultados da gestão de conhecimento; e

f) partes interessadas: cidadão – usuário e sociedade.



**Figura 4 - Modelo de gestão de conhecimento para a administração pública**  
 Fonte: BATISTA (2012)

Para o autor, os direcionadores estratégicos devem estar bem alinhados com a gestão de conhecimento, para que os resultados da aplicação do modelo possam atender aos objetivos de uma determinada organização pública.

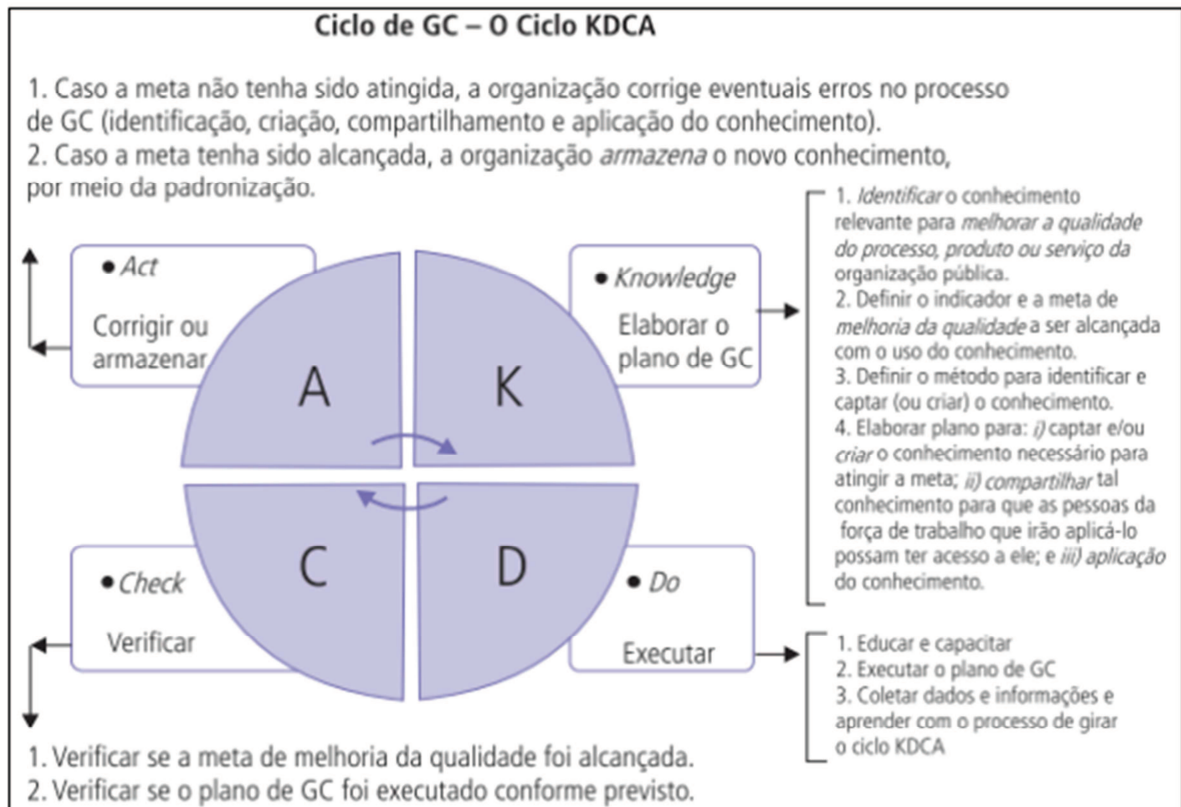
Os viabilizadores são considerados fatores críticos de sucesso, o que requer muita atenção. Segundo HEISIG (2009) eles são divididos em quatro grupos:

- a) fatores humanos: cultura, pessoas e liderança;
- b) organização: processo e estrutura;
- c) tecnologia: infraestrutura e aplicações; e
- d) processo de gestão: estratégia, objetivos e mensuração.

Para a construção do modelo de gestão de conhecimento para administração pública BATISTA (2012) observou a importância das atividades de identificação, criação, armazenamento, compartilhamento e aplicação de conhecimento no tocante ao processo de gestão de conhecimento.

O autor mostra que, as atividades do processo de gestão de conhecimento são executadas no ciclo do KDCA (vide Figura 5), na qual K (knowledge = conhecimento) substitui o P (*Plan* = planejar) do ciclo do PDCA, os

demais recebem as mesmas denominações, o D (*Do* = executar), o C (*Check* = verificar) e o A (*Act* = corrigir ou armazenar). Segundo o autor, o planejamento não está descartado, mas o foco do processo está no conhecimento.



**Figura 5 - Ciclo de gestão de conhecimento - O Ciclo do KDCA**  
Fonte: BATISTA (2012)

Os resultados da gestão de conhecimento segundo BATISTA (2012) podem ser classificados em imediatos ou finais.

Os resultados imediatos são aprendizagem e inovação e, como consequência, o incremento da capacidade de realização individual, da equipe, da organização e da sociedade de identificar, criar, armazenar, compartilhar e aplicar conhecimento. (BATISTA 2012 p.68).

Os resultados finais da gestão de conhecimento decorrem dos imediatos (aprendizagem e inovação; e aumento da capacidade de realização individual, de equipes, da organização e da sociedade) e aparecem na definição de gestão de conhecimento na administração pública... (BATISTA 2012, p. 69)

Ainda BATISTA (2012) afirmar que:

... a saber: aumentar a eficiência; melhorar a qualidade e a efetividade social; e contribuir para a legalidade, impessoalidade, moralidade e publicidade na administração pública é para o desenvolvimento brasileiro. (BATISTA 2012, p. 68).

O último componente do modelo é chamado por BATISTA (2012) de partes interessadas, composta pelo cidadão-usuário e sociedade. Os mesmos são os envolvidos diretos das ações e serviços prestados pela administração pública.

Para BATISTA (2012) a gestão do conhecimento exerce um papel importante na vida do cidadão-usuário ou da sociedade, pois, possibilita cumprir as indicações dos direcionadores de estratégias, exemplos:

- a) visão;
- b) missão;
- c) objetivos estratégicos; e
- d) estratégias e metas.

Em suma, este capítulo apresentou os referenciais teóricos sobre gestão de conhecimento, conceitos de conhecimentos tácitos, conceitos de conhecimentos explícitos e modelos de gestão de conhecimento. Tais conceitos e modelos alicerçaram a construção do processo de sistematização no que tange a definição da residência estrutural e aplicada (vide capítulo 3).

### 3 RESIDÊNCIA EM SOFTWARE

Este capítulo é segmentado em três partes. A primeira tem como objetivo apresentar o referencial teórico sobre a residência em software. Já a segunda foca a caracterização de oito ambientes de residência em software, que estão posicionados no território nacional. Para caracterizar os ambientes, o autor deste trabalho desenvolveu um Estudo de Caso. O referido estudo foi executado nos anos de 2.010, 2.011 e 2.012, e mapeou o processo de produção de software, o paradigma ágil ou tradicional, a gestão de projeto, a gestão da qualidade e o produto gerado.

Por fim, o capítulo apresenta uma proposta de um modelo de classificação dos referidos ambientes em quatro níveis:

- a) nível 1, neste o ambiente de residência encontra-se em um laboratório e os projetos de software são simulados;
- b) nível 2, neste o ambiente de residência encontra-se em um laboratório e os projetos são oriundos da indústria de software;
- c) nível 3, neste o ambiente de residência encontra-se em uma empresa desenvolvedora de software e os projetos de software são desenvolvidos para atender clientes reais; e
- d) nível 4, apresenta as mesmas características do nível 3, somando o propósito de exportar conhecimentos sobre processo de software, gestão de projetos e de qualidade e ferramenta.

#### 3.1 REFERENCIAL TEÓRICO SOBRE RESIDÊNCIA EM SOFTWARE

Com o intuito de minimizar a deficiência na formação da mão de obra para a área de engenharia de software, algumas empresas e universidades criaram a ideia da residência em software, algo semelhante à residência médica (FABRI et al. 2008).

A ideia de residência médica foi criada no Brasil pelo decreto número 80.281 de 05 de setembro de 1.977, e se caracteriza como uma modalidade de ensino de pós-graduação. Esta pós visa o aperfeiçoamento dos médicos sob a forma de um curso de especialização, na qual este está inserido em uma instituição ligada a área de saúde. Ao concluir o curso, o médico adquire o título de especialista em

alguma área da medicina. O tempo de residência médica varia de acordo com a especialização. Especialistas em cirurgias ficam quatro anos imersos no ambiente de residência. Já para obter as especialidades de ginecologista, o tempo de residência é de três anos. Outras especialidades possuem residência de dois anos. Na medicina o residente vive, grande parte do tempo, em um ambiente hospitalar, situações reais dentro de seu escopo de conhecimentos. Tal experiência é monitorada por um corpo de especialistas.

A residência em software tende a tomar mesma linha de raciocínio: proporcionar uma vivência ao aluno de pós-graduação ou ao profissional inserido na indústria, dentro de um ambiente real, que possua políticas de qualidade bem definidas, com o objetivo de promover a disseminação dos conceitos de qualidade, processo de produção e gestão em projetos na área de engenharia de software.

Para Fabri et al. (2008) a residência em software proporciona dentro de um ambiente real de desenvolvimento de software, uma experiência prática aos alunos de graduação, pós-graduação e/ou profissionais inseridos na indústria. Estes ambientes reais são preparados para que os residentes possam se especializar em uma determinada área.

Para Sampaio et al. (2005) a residência em software, como no contexto de residência médica, deve ser desenvolvida em centros de ensino, abordar conhecimentos específicos, conceitos relevantes e apresentar características de ensino formal. O desenvolvimento das práticas previstas nas indústrias de software deve ser o foco da atenção dos tutores que acompanham os residentes.

Considerando Fabri et al. (2008) e Sampaio et al. (2005) pode-se dizer que, a residência em software tem um paralelismo estreito com a residência médica, pois, ambas têm a mesma finalidade: especializar os alunos/profissionais interessados em uma determinada área. As referidas residências apresentam semelhanças na forma de execução. Por exemplo, as indústrias de software oferecem as chamadas de vagas para colaboradores *trainee*, para um período de experiência, estes passam por avaliações antes do ingresso, caso aprovados são acompanhados durante a fase de treinamento pelo(s) tutor(es) até a possível contratação.

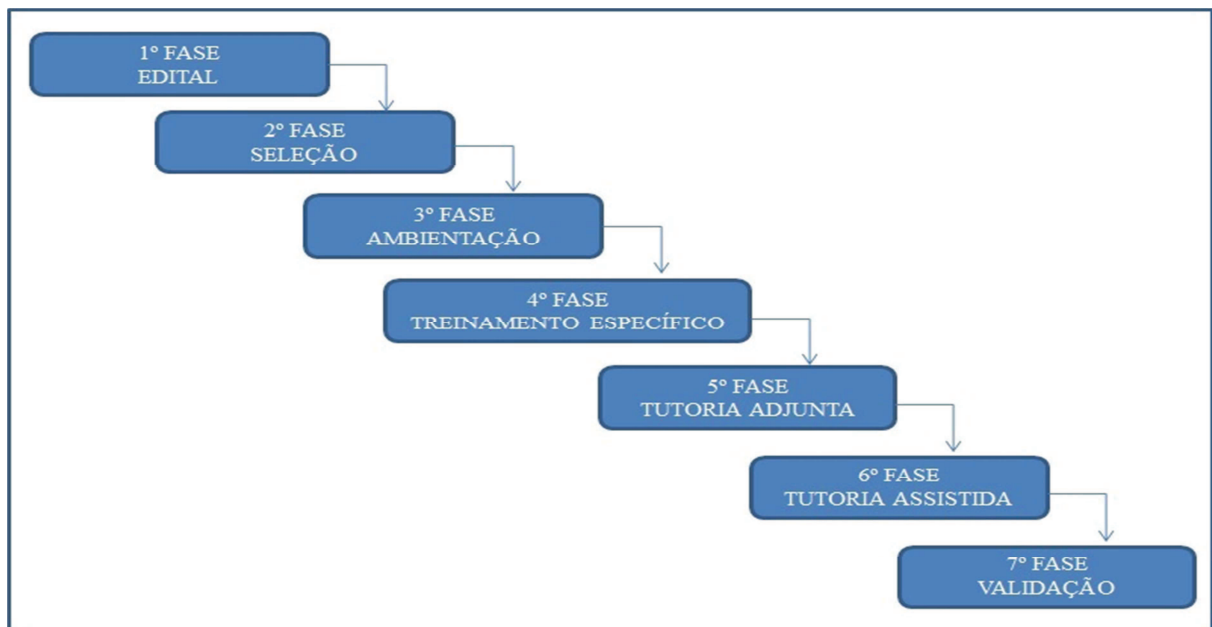
A residência médica inicia-se quando um edital é lançado, os médicos interessados, que já concluíram a graduação, se inscrevem na especialidade

desejada. Um exemplo de uma chamada para residência médica, pode ser a do Hospital Israelita Albert Einstein.

No Hospital Israelita Albert Einstein, um edital de abertura de inscrições para o programa de residência médica<sup>3</sup> tem por objetivo divulgar os métodos e os procedimentos inerentes ao processo de seleção e execução da referida residência. As inscrições dos interessados passam por um processo de validação, os que tiveram a inscrição validada são submetidos a uma avaliação eliminatória, esta composta por duas fases:

- a) a primeira por meio de uma prova escrita objetiva e;
- b) a segunda por meio de uma prova prática e análise de *curriculum vitae*.

Quando aprovados, os médicos residentes recebem acompanhamento de médicos supervisor(es) ou tutor(es) durante todo o tempo da residência. E ao término de dois, três ou quatro anos, dependendo do tempo da residência escolhida, são novamente avaliados. Por fim, os residentes receberem o título de especialista na área. A Figura 6 ilustra as fases da residência médica propostas no Hospital Israelita Albert Einstein.



**Figura 6 - Fases da residência médica do hospital Albert Einstein**  
**Fonte: Autoria própria**

<sup>3</sup> Acessível em <http://www.einstein.br/Ensino/residencia-medica/Documents/edital-residencia-medica-2014.pdf>



Ao analisar a Figura 6 é possível perceber as sete fases da residência médica oferecida pelo Hospital Albert Einstein. Considerando as fases citadas, este trabalho traça um paralelo entre a residência médica e a residência em software, apresentado a seguir:

a) chamada pública ou lançamento de edital: nesta fase, as indústrias de software ou centros de ensino que buscam promover o programa de residência em software, externam como e quando os interessados devem se proceder para participarem. Esta fase detalharia as condições de oferta:

a.1) a forma de inscrição;

a.2) os pré-requisitos necessários;

a.3) a remuneração para os residentes ou o investimento que os mesmos tem que fazer, dependendo do propósito da residência;

a.4) o programa da residência;

a.5) a forma de homologação das inscrições;

b) seleção: nesta fase, a seleção dos inscritos ocorre por meio de duas etapas eliminatórias. Na primeira, os interessados passariam por avaliações teóricas e práticas, caso fossem aprovados seguiriam para a segunda, a entrevista. Com os resultados das avaliações teóricas e práticas seria possível diagnosticar o grau de preparação e/ou conhecimentos gerais na área de engenharia de software que, o interessados apresentariam antes de participarem da residência em software. Já com entrevista seria possível observar a proatividade, a clareza de raciocínio, e capacidade de argumentação e comunicação do candidato;

c) ambientação ou inserção Inicial: nesta fase ocorre a apresentação das áreas e/ou departamentos da indústria de software, o processo de software e a política de gestão da qualidade utilizados nos ambientes de residência em software para os residentes;

d) treinamento específico: nesta fase busca-se treinar os residentes a fim de equalizar os seus conhecimentos para com os demais membros da equipe de produção;

e) tutorial adjunta, nesta fase forma-se equipes de residentes para resolverem uma situação problema proposto, o tutor trabalha ativamente juntos com às equipes na solução do problema;

f) tutorial assistida: nesta fase, o tutor acompanha e/ou tutoria as equipes na resolução do problema; e

g) validação: nesta fase, o tutor avalia e valida o produto software que os residentes desenvolveriam, levando-se sempre em consideração os aspectos relevantes, tais como:

- g.1) prazo;
- g.2) custo;
- g.3) organização;
- g.4) qualidade;
- g.5) eficiência; e
- g.6) eficácia.

Um comparativo entre a residência médica e a residência em software pode ser verificada no Quadro 1.

MEDICINA	SOFTWARE
Hospital	Empresas produtoras de software atuando no mercado (o hospital da residência em software)
Residentes (médicos)	Residentes (alunos do último ano de graduação ou profissionais da área de engenharia de software (aluno de pós-graduação))
Pacientes	Clientes/empresas que necessitam de software (o paciente da residência em software)
Médicos/Professores (tutores no ambiente de residência médica)	Engenheiros de Software/Professores (tutores no ambiente de residência em software)
Laboratórios (utilizado para realização de exames e diagnóstico)	Laboratório, Ambiente de produção de software (utilizado para construção do software e gestão do projeto)

**Quadro 1 - Residência médica e residência em software**  
**Fonte: FABRI et al. (2008)**

Os ambientes pesquisados por SILVEIRA DUARTE et al. (2013), vide seção 3.2, que desenvolveram residência em software apresentam na sua execução uma correlação com três fases da residência médica do Hospital Albert Einstein, esta correlação está ligada às fases: Treinamento Específico; Tutoria Adjunta; e Tutoria Assistida. Todos os ambientes pesquisados apresentaram foco em treinamentos e apenas um não apresentou foco em “módulo de vivência prática”<sup>4</sup>. O autor deste trabalho optou por chamar os treinamentos de Residência Estrutural e o “módulo de vivência prática” de Residência Aplicada.

<sup>4</sup> No módulo de vivência prática, os residentes são inseridos diretamente nas atividades simulada de desenvolvimento de software.

A Residência Estrutural pode ser comparada à quarta fase da residência médica do Hospital Israelita Albert Einstein, fase Treinamento Específico.

A Residência Estrutural ou treinamento é o início das atividades da residência em software, nesta os residentes são treinados por um ou mais especialistas da área de desenvolvimento de software. O tempo do treinamento depende do que foi estabelecido como conteúdos mínimos essenciais, para a área da residência escolhida. É possível existir residência estrutural com mais trezentas horas, porém, é importante destacar que a quantidade de horas não é o fator mais relevante desta, e sim o grau de instrução obtida pelos residentes, para que os mesmos se encontrem aptos a participarem da segunda atividade de uma residência em software, também denominada como residência aplicada. A definição apresentada neste parágrafo é compartilhada por FABRI (2010), SAMPAIO (2006) e BEGOSSO (2011).

A Residência Aplicada pode ser comparada à quinta e sexta fases da residência médica do referido hospital, fases Tutorial Adjunta e Tutoria Assistida.

A Residência Aplicada ou módulo de vivência prática inicia-se após o término da residência estrutural, nesta os residentes são inseridos diretamente nas atividades de desenvolvimento de software. A origem do software, a ser desenvolvido pelos residentes, depende da classificação do ambiente de residência. Nesta fase os instrutores acompanham e avaliam de perto o desenvolvimento do referido produto software.

Apresentados os referenciais teóricos sobre residência em software, a próxima seção tem por objetivo apresentar os ambientes de residência em software no Brasil.

### 3.2 AMBIENTES DE RESIDÊNCIA EM SOFTWARE NO BRASIL

Um ambiente de residência em software é criado e/ou preparado para que os residentes participem de um processo de especialização na área de engenharia de software. Esta criação envolve uma série de elementos, tais como:

- a) pessoas;
- b) laboratórios de informática;
- c) ferramentas;

- d) processo estruturado para execução da residência em software; e
- e) qualidade no processo.

SILVEIRA DUARTE et al. (2013), via estudo de caso, analisaram oito ambientes de residência em software implementados no Brasil. Para a concretização deste estudo, os autores executaram um estudo de múltiplos casos, alicerçados pelas teorias apresentadas por Yin (2005).

Segundo Yin (2005), o método estudo de caso é aderente às pesquisas cujo objetivo é relatar por meio da observação do ambiente, análise de documentos, de artefatos e fatos históricos como estão estruturados os ambientes econômicos, sociais, educacionais e produtivos. Tendo em vista que, um dos objetivos do trabalho é verificar como estão estruturados os ambientes de residência de software implementados no Brasil, a escolha do referido método é justificada.

Para executar um estudo de caso consistente é necessário percorrer as seguintes etapas:

- 1) Definir tipo de estudo de caso (caso único ou casos múltiplos?);
- 2) Definir as unidades de análise (quais casos serão analisados?);
- 3) Estruturar o protocolo (conjunto de regras) para realização do estudo de caso (o protocolo tem como objetivo auxiliar o pesquisador na coleta dos dados); e
- 4) Executar o estudo respeitando as diretrizes mapeadas no protocolo. A execução do estudo irá mostrar um relato dos ambientes de residência em software.

O Quadro 2 apresenta as informações que caracterizam as etapas 1, 2 e 3. Já a etapa 4 é apresentada nas seções 3.2.1, 3.2.2 e 3.2.3

<p><b>Tipo:</b> Casos Múltiplos.  <b>Justificativa:</b> O trabalho proposto irá realizar estudo de casos múltiplos, pois o mesmo não está sendo desenvolvido sob uma circunstância exclusiva ou o caso não representa um teste crucial à teoria existente.</p>				Etapa 1
<p><b>Unidades de Análise:</b> Foram analisados ambientes de residência em software espalhados de norte a sul do país.  <b>Justificativa:</b> A escolha destes ambientes se baseou nos seguintes critérios: a) Formalização do ambiente de residência e disponibilidade de informações sobre o mesmo. b) Possibilidade de acesso aos residentes. Os autores deste trabalho não possuem autorização para relacionar as informações mapeadas no estudo com os ambientes de residência em software, o trabalho cita (aleatoriamente) apenas os nomes dos ambientes, ou seja, a ordem de mapeamento das informações difere da ordem citação dos ambientes.  <b>Ambientes:</b> 1) Programa de Residência da Universidade Federal da Bahia. 2) Centro de Residência do Departamento de Engenharia de Produção da Escola Politécnica da Universidade de São Paulo. 3) Residência em Teste de Software do Centro de Informática da Universidade Federal de Pernambuco. 4) Proposta de Residência Software em um Curso de Pós-Graduação – Faculdade de Tecnologia de Ourinhos. 5) Programa de Residência em Software da Fundação Educacional do Município de Assis. 6) Programa de Residência em Software do Centro de Análise, Pesquisa e Inovação Tecnológica - Manaus. 7) Programa de Residência Alcatel-Lucent e Centro de Estudos e Sistemas Avançados de Recife. 8) Programa de Residência em Software da B2ML e da Universidade Federal de Itajubá.  É importante salientar que os ambientes de 1 a 8, acima, não tem ordem de relação com os mesmos de A à H.</p>				Etapa 2
<p><b>Protocolo:</b>  <b>Início</b> (da análise das unidades de análise): julho de 2010  <b>Término</b> (da análise das unidades de análise): agosto de 2012 (26 meses)</p>				
<b>Objetos observados:</b>				
Ambiente	Objetos Pesquisados (Residente ou Documentos)	Forma – Presencial; – Distância (via <i>call conference</i> )	Tempo de observação	
A	Residente.	Distância (via <i>call conference</i> )	1h	
B	Residente.	Distância (via <i>call conference</i> )	2h30m	
C	Publicações que relatam a estrutura do ambiente. Residente. Ambiente.	Presencial	8h	
D	Publicações que relatam a estrutura do ambiente. Residente. Ambiente.	Presencial	7h30m	
E	Publicações que relatam a estrutura do ambiente. Residente. Ambiente.	Presencial	9h	
F	Residente	Distância (via <i>call conference</i> )	1h30m	
G	Residente	Distância (via <i>call conference</i> )	2h	
H	Residente	Distância (via <i>call conference</i> )	2h	
<p><b>Roteiro utilizado na observação dos objetos:</b>  1 – Caracterização do ambiente (foco, objetivo, implementação da residência).  2 – Processo:  2.1. estrutura - a estrutura do processo pode ser caracterizada em três níveis: 1) processos curtos: produção e teste de código; 2) processos médios: produção de artefatos do projeto, código fonte e teste e; 3) processos longos: produção de toda modelagem de negócio, artefatos do projeto código fonte, teste e implantação;  2.2. aplicação da PML (<i>Process Modeling Language</i>);  2.3. documento de modelagem do processo (processo institucionalizado);  2.4. seleção de ferramentas;  2.5. treinamento nas ferramentas e no processo;  3 – Paradigma utilizado (ágil, tradicional, híbrido).  4 – Atividades de gestão do projeto.  5 – Atividades ligadas a gestão da qualidade.  6 – Produto gerado (orientação geral – desenvolver produtos para vários segmentos de mercado, orientação a domínio – desenvolver produtos para um único segmento de mercado, orientação a produto – desenvolver um único produto).  <b>Validação.</b>  A validação das informações foi efetuada somente para os ambientes C, D e E. Ambientes que os autores deste trabalho tiveram acesso direto. A validação foi consolidada pela verificação de evidências entre a opinião do residente e a forma de organização produtiva do software implementada no próprio ambiente.</p>				Etapa 3

#### Quadro 2 - Estrutura do estudo de caso

Fonte: Adaptado de SILVEIRA DUARTE (2013)

### 3.2.1 Caracterização dos Ambientes

Nesta seção será apresentada as características dos ambientes nos quais as residências em software foram realizadas. Os referidos ambientes foram citados no Quadro 2.

O ambiente A tem como foco o treinamento em diversas tecnologias para alunos do último ano e/ou recém-formados, promovendo formação teórica e orientação para desenvolvimento de atividades de pesquisa em tópicos avançados

relacionados ao tema de Governo Eletrônico. Neste ambiente participaram quinze estudantes, destes, oito se caracterizaram como residentes. Durante o programa os residentes desenvolveram uma monografia ligada a algum tópico de pesquisa visto durante o programa. As disciplinas: a) governo eletrônico; b) engenharia de software para aplicações web; c) arquiteturas para o desenvolvimento de aplicações distribuídas; d) integração de aplicações distribuídas; e) computação móvel; f) gestão do conhecimento; g) ontologias e web semântica; h) análise e modelagem de processos I; e i) análise e modelagem de processos II, totalizando duzentas horas, estas foram ministradas para especialização dos residentes.

O ambiente B tem como foco o treinamento em diversas tecnologias, busca formar especialista em arquitetura, gerência de configuração, análise de negócios e testes. Tem por objetivo a melhoria e aumento na formação de mão de obra na região norte do país. O projeto foi financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), desenvolvido dentro de uma fábrica de software internalizada pelo proponente. Os residentes recebiam bolsa de aproximadamente R\$ 1.000,00 (um mil reais). O programa se divide em duas fases: primeira – Formação Teórica; segunda – Formação Prática. Na primeira fase, os residentes participam de cursos teóricos modulares, totalizando uma carga horária de quinhentas e setenta e oito horas, com disciplinas relacionadas a: a) tecnologias Java; b) banco de dados; c) análise; d) arquitetura; e) teste e qualidade de software; e f) gerência de projetos de software. Na segunda é iniciado o módulo de vivência prática na qual os residentes serão inseridos diretamente nas atividades de desenvolvimento de software. A vivência prática se dá em três etapas: 1ª Identificação do perfil dos residentes e distribuição destes em grupos de acordo com a linha de formação específica dos orientadores; 2ª compreende um período de dois meses de simples observação e acompanhamento das atividades executadas pelos analistas, arquitetos e desenvolvedores seniores participantes desta proposta; e 3ª execução das atividades práticas de desenvolvimento acompanhadas pelos orientadores com duração de seis meses.

O ambiente C tem como foco o treinamento em diversas tecnologias, busca formar especialistas em teste de software. O projeto foi financiado por alguns órgãos de fomentos, nas esferas públicas e privadas, desenvolvido dentro de uma organização focada em teste de software, este projeto tem por objetivo a melhoria contínua da qualidade do produto. Neste ambiente o residente é inserido em um

ambiente real de teste, projetos advindos da indústria de software são testados e certificados quanto à qualidade do produto. As políticas de qualidade de projeto e processo são bem estabelecidas.

O ambiente D tem como foco o treinamento em diversas tecnologias, busca melhorar a formação na área de engenharia de software. Tem por objetivo implementar um ambiente simulado e uma formação diferenciada aos alunos que possuem curso superior na área de produção de software. O ambiente simulado é implementado em uma fábrica de software, com políticas de qualidade de processo bem definidas. Os residentes são alunos de um curso de pós-graduação e projetos advindos da indústria de software são desenvolvidos neste ambiente.

O ambiente E tem como foco o treinamento em diversas tecnologias, busca inserir alunos da graduação na indústria de software. Tem por objetivo implementar um ambiente simulado e uma formação diferenciada aos alunos da graduação. O ambiente é implementado com políticas de qualidade de processo previamente estabelecidas. Os residentes são alunos de um curso de graduação e os tutores são professores que promovem a construção de projetos simulados.

O ambiente F tem como foco o treinamento na solução dos problemas relacionados a processo de software. O ambiente atua em áreas como análise de dados e desenvolvimento para produção de relatórios, programação de Unidade de Resposta Audível (URA), desenvolvimento de estratégias e roteamento para encaminhamento de chamadas e integrações com as plataformas de *contact center* do cliente, além de soluções para centrais de atendimento, o ambiente tem como objetivo captar mão de obra qualificada. Este programa admite somente profissionais formados com mais três anos de experiência ou mestres, os residentes geralmente são bolsistas. Durante a execução do projeto, residentes serão acompanhados por tutores e inseridos no próprio ambiente. A plataforma de trabalho habilita o profissional a prestar serviços ou potencialmente ser contratado por clientes reais e seus parceiros, tanto no Brasil quanto no exterior.

O ambiente G tem como foco o treinamento em desenvolvimento de software. Tem por objetivo capacitar mão de obra especializada em desenvolvimento de software *web-based*. Os residentes possuem bolsa de R\$ 1.045,89 (um mil, quarenta e cinco reais, oitenta e nove centavos) participam de um curso de Especialização em Engenharia WEB e trabalham junto em um projeto dentro do próprio ambiente.

O ambiente H tem como foco o treinamento em diversas tecnologias, busca especializar profissionais da área de software. Tem por objetivo prover uma formação sólida para os profissionais de Tecnologia da Informação (TI), principalmente sob a ótica da qualidade de software. O programa admite somente profissionais formados. Durante a execução do projeto, residentes serão acompanhados por tutores e inseridos no próprio ambiente. O ambiente habilita o profissional a prestar serviços ou potencialmente ser contratado por clientes reais e seus parceiros.

### 3.2.2 Caracterização do Processo de Software

A caracterização do processo de software seguiu o roteiro utilizado na observação dos objetos (vide Quadro 2, p. 35). A caracterização do processo foi dividida em:

- a) estrutura;
- b) aplicação da *Process Modeling Language* (PML);
- c) documento de modelagem do processo (institucionalização do processo);
- d) seleção de ferramentas; e
- e) treinamento nas ferramentas e no processo. O Quadro 3 apresenta a mostra a divisão.



<b>Ambiente</b>	<b>2.1 Estrutura</b>	<b>2.2 PML</b>	<b>2.3 Modelagem Processo</b>	<b>2.4 Ferramentas</b>	<b>2.5 Treinamento</b>
<b>B</b>	Proposta de um processo longo.	Estrutura informal.	Modelado não-institucionalizado.	Propõe a utilização de ferramentas p/ gestão de projetos e gestão da qualidade.	Java. Banco de dados. Análise. Arquitetura. Teste e qualidade de software. Gerência de projetos.
<b>C</b>	Processo curto.	Estrutura informal.	Modelado atualizado e institucionalizado.	Gestão de Projetos. Ferramenta de teste de software.	Teste de software. Gestão de projetos e qualidade.
<b>D</b>	Processo médio.	Estrutura informal.	Modelado atualizado e institucionalizado.	Gestão de projetos. Métricas de software. Gestão da qualidade.	Processo de software. Gestão de projetos. Métricas de software. Qualidade de software.
<b>E</b>	Proposta de um processo longo.	Estrutura informal.	Modelado não-institucionalizado.	Propõe a utilização de ferramentas p/ gestão de projetos e gestão da qualidade.	UML. Garantia da qualidade de Software. Banco de dados. Ferramentas de produção de software.
<b>F</b>	Processo médio.	Estrutura informal.	Modelado atualizado e institucionalizado.	Gestão de projetos. Métricas de software. Gestão da qualidade.	Processo de software. Gestão de projetos. Métricas de software. Qualidade de software.
<b>G</b>	Proposta de um processo longo.	Estrutura informal.	Modelado não-institucionalizado.	Propõe a utilização de ferramentas p/ gestão de projetos.	Treinamento desenvolvido diretamente no ambiente de produção
<b>H</b>	Proposta de um processo longo.	Estrutura informal.	Modelado não-institucionalizado.	Propõe a utilização de ferramentas p/ gestão de projetos e de gestão da qualidade	Processo de software. Banco de dados. Java. Tecnologia .NET. Teste de software. Análise e projeto de sistemas. Qualidade de Software
<b>A</b>	Apesar de se intitular como ambiente de residência o referido ambiente não possui, ao menos, um ambiente simulado. Perceba, como descrito na seção 3.1.1, que somente treinamentos são desenvolvidos pelos supostos residentes. Este fato levou os autor deste trabalho a não classificar o Ambiente A como residência em software.				

**Quadro 3 - Caracterização do processo, primeira parte**

Fonte: SILVEIRA DUARTE et al. (2013)

O ambiente A como pode ser observado na última posição do Quadro 3, não foi classificado como um ambiente dentro da ótica da residência em software.

O ambiente B caracteriza seu processo como longo, nele ocorre a produção de toda modelagem de negócio, artefatos do projeto, código fonte, teste e implantação. A *Process Modeling Language* (PML) utilizada para modelar o processo possui uma estrutura informal e o documento do processo é modelado, mas não institucionalizado. Quanto às ferramentas o ambiente propõe a utilização para gestão de projetos e da qualidade. Quanto ao treinamento, os residentes foram instruídos em Java, banco de dados, análise, teste, qualidade e gerência de projeto.

O ambiente C possui um processo curto. O processo possui produção e teste de código. O referido ambiente aplica a PML de maneira informal, possui processo modelado, atualizado e institucionalizado. Quanto às ferramentas o ambiente propõe a utilização para gestão de projetos e de teste. Quanto ao treinamento, os residentes foram instruídos em teste, gestão de projetos e da qualidade.

O ambiente D possui um processo médio. O processo possui produção de artefatos do projeto, código fonte e teste. O referido ambiente aplica a PML de maneira informal, possui processo modelado, atualizado e institucionalizado. Quanto às ferramentas o ambiente propõe a utilização para gestão de projetos, de métricas e da qualidade. Quanto ao treinamento, os residentes foram instruídos em processo de software, gestão de projetos, de métricas e da qualidade.

O ambiente E possui um processo longo. O processo possui produção de toda modelagem de negócio, artefatos do projeto, código fonte, teste e implantação. O referido ambiente aplica a PML de maneira informal, possui processo modelado, mas não institucionalizado. Quanto às ferramentas o ambiente propõe a utilização para gestão de projetos e da qualidade. Quanto ao treinamento, os residentes foram instruídos em *Unified Modeling Language* (UML), garantia de qualidade, banco de dados e produção de software.

O ambiente F possui um processo médio. O processo possui produção de artefatos do projeto, código fonte e teste. O referido ambiente aplica a PML de maneira informal, possui processo modelado, mas não institucionalizado. Quanto às ferramentas o ambiente propõe a utilização para gestão de projeto, de métricas e da qualidade. Quanto ao treinamento, os residentes foram instruídos em processo de software, gestão de projetos, de métricas e da qualidade.

O ambiente G possui um processo longo. O processo possui produção de toda modelagem de negócio, artefatos do projeto código fonte, teste e implantação. O referido ambiente aplica a PML de maneira informal, possui processo modelado, mas não institucionalizado. Quanto às ferramentas o ambiente propõe a utilização para de gestão de projeto, de métricas e da qualidade. Quanto ao treinamento os residentes, foram instruídos diretamente no ambiente de produção.

O ambiente H possui um processo longo. O processo possui produção de toda modelagem de negócio, artefatos do projeto código fonte, teste e implantação. O referido ambiente aplica a PML de maneira informal, possui processo

modelado, mas não institucionalizado. Quanto às ferramentas o ambiente propõe a utilização para gestão de projeto e da qualidade. Quanto ao treinamento, os residentes foram instruídos em processo de software, banco de dados, tecnologia Java, tecnologia .NET, teste, análise e projeto de sistemas e qualidade.

É importante salientar, novamente, que todas as informações geradas, nos estudos, levaram em consideração as observações efetuadas pelos residentes. Somente para os ambientes C, D e E, os autores tiveram acesso a publicações que relatam a estrutura do ambiente e ao próprio ambiente.

### 3.2.3 Demais Caracterizações do Processo de Software

Nesta seção será apresentada a caracterização do paradigma ágil ou tradicional, da gestão de projeto, gestão da qualidade e do produto gerado dos ambientes B, C, D, E, F, G e H. O ambiente A não será caracterizado, pois, como já citado no Quadro 3, o mesmo não foi classificado como um ambiente de residência em software. O Quadro 4 apresenta as características.

	<b>3 Paradigma</b>	<b>4 Gestão de projetos</b>	<b>5 Gestão da Qualidade</b>	<b>6 Produto Gerado</b>
<b>B</b>	Tende a ser tradicional	Prevista	Prevista	Orientação geral
<b>C</b>	Ágil	Sim	Sim	Orientação geral
<b>D</b>	Tradicional	Sim	Prevista	Orientação geral
<b>E</b>	Tende a ser tradicional	Prevista	Prevista	Orientação geral
<b>F</b>	Ágil	Sim	Sim	Orientada a domínio
<b>G</b>	Tende a ser tradicional	Prevista	Prevista	Orientação Geral
<b>H</b>	Tende a ser tradicional	Prevista	Prevista	Orientação Geral

**Quadro 4 - Demais caracterizações do processo de software, segunda parte**

Fonte: SILVEIRA DUARTE et al. (2013)

O ambiente B utiliza o paradigma tradicional em seu processo de software, gestões de projeto e de qualidade são previstas e o produto gerado tem a característica de orientação geral - desenvolver produtos para vários segmentos de mercado.

O ambiente C adota o paradigma ágil, as gestões de projeto e qualidade são aplicadas e o produto gerado tem a característica de orientação geral - desenvolver produtos para vários segmentos de mercado.

O ambiente D adota o paradigma tradicional, a gestão de projeto é aplicada, já a de qualidade é prevista e o produto gerado tem a característica de orientação geral - desenvolver produtos para vários segmentos de mercado.

O ambiente E adota o paradigma tradicional em seu processo de software, às gestões de projeto e qualidade são aplicadas e o produto gerado tem a característica de orientação geral – desenvolver produtos para vários segmentos de mercado.

O ambiente F adota o paradigma ágil, às gestões de projeto e qualidade são aplicadas e o produto gerado tem a característica de orientada a domínio - desenvolver produtos para um único segmento de mercado.

O ambiente G adota o paradigma tradicional, às gestões de projeto e qualidade são previstas e o produto gerado tem a característica de orientação geral - desenvolver produtos para vários segmentos de mercado.

O ambiente H apresenta a mesma caracterização do ambiente G.

Nesta seção foram apresentadas as características de alguns ambientes de residência em software implementados no Brasil. Além do citado, o autor propôs em seu trabalho um Modelo para Classificação de Ambientes de Residência em Software.

Na seção a seguir o autor deste trabalho apresentada o referido modelo.

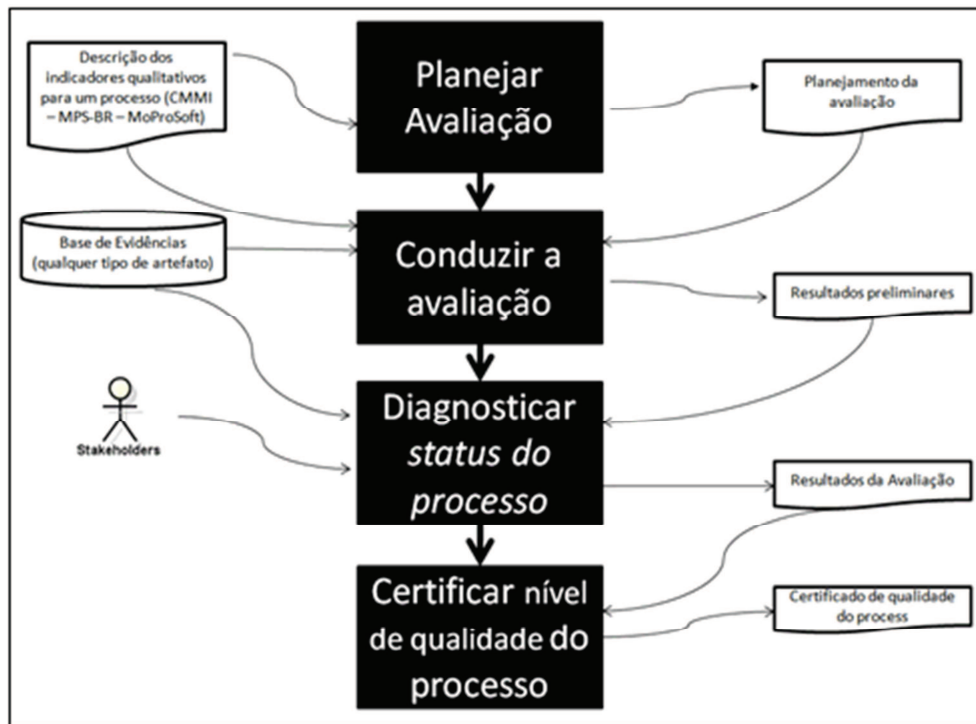
### 3.3 MODELO PARA CLASSIFICAÇÃO DE AMBIENTE

O Modelo para Classificação de Ambientes para Residência em Software tem como base o Modelo de Avaliação de Processo das normas ISO/IEC 15504 (2004), SCAMPI (2006) e MA-MPS.BR (2006).

Um modelo de avaliação tem como objetivo apresentar requisitos para que um processo de software atenda a certo nível de qualidade. O modelo deve prover uma estrutura de medição quantitativa e qualitativa, independente e aceita pela indústria de software.

Geralmente um modelo de avaliação é composto basicamente por requisitos, atividades e indicadores (quantitativos e qualitativos) necessários para obtenção de certo grau de qualidade em um processo de software (ISO/IEC 15504 (2004), SCAMPI (2006) e MA-MPS.BR (2006)).

Analisando os modelos propostos pela norma ISO/IEC 15504 (2004), pelo SCAMPI (2006) e pelo Modelo de Avaliação do MA-MPS.BR (2006) é possível verificar que a avaliação de um processo de software é composta, normalmente, por quatro atividades: 1 – Planejamento; 2 – Condução; 3 – Diagnóstico; 4 – Certificação. A Figura 6 apresenta as quatro atividades, seus artefatos de entrada (por exemplo: base de evidências) e saída (por exemplo: resultados preliminares).



**Figura 7 – Modelo de Avaliação gerado a partir dos pressupostos estabelecidos pela norma 15504 (2004), SCAMPI (2006) e MA-MPS.BR (2006)**  
**Fonte: SILVEIRA DUARTE et al. (2013)**

É importante salientar que a Figura 7 caracteriza-se como um modelo de avaliação genérico de processo, gerado a partir dos pressupostos estabelecidos pela norma 15504 (2004), pelo SCAMPI (2006) e pelo MA-MPS.BR (2006).

Ao analisar o modelo de avaliação proposto a partir da norma 15504 (2004), do SCAMPI (2006) e do MA-MPS.BR (2006), é possível verificar a presença de quatro atividades: 1 - Planejar; 2 – Conduzir; 3 – Diagnosticar; 4 – Certificar. Já o modelo classificatório, vide Figura 8, para residência em software instancia 3 das 4 atividades delineadas: 1 – Planejar, 2 – Conduzir, 3 – Classificar. Não é o objetivo do modelo certificar qualquer tipo de ambiente, este fato exclui a inserção da atividade 4, delineada na Figura 7.

No modelo de classificação dos ambientes de residência, a atividade de planejamento possui como artefato de entrada os indicadores qualitativos para classificação do ambiente (vide Figura 9, p. 46). A partir destes indicadores é gerado um plano de trabalho. Este plano irá conduzir a classificação do ambiente, gerando resultados preliminares. Após a coleção destes resultados o ambiente é classificado pela sua caracterização em um dos quatro níveis delineados pelo modelo:

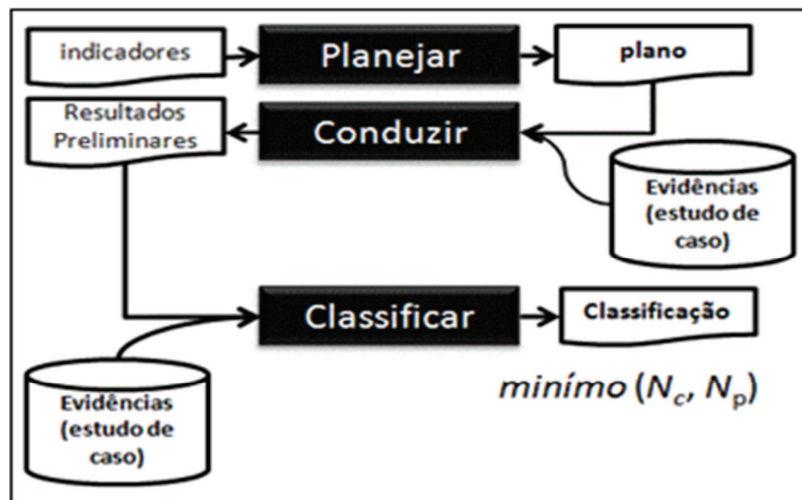


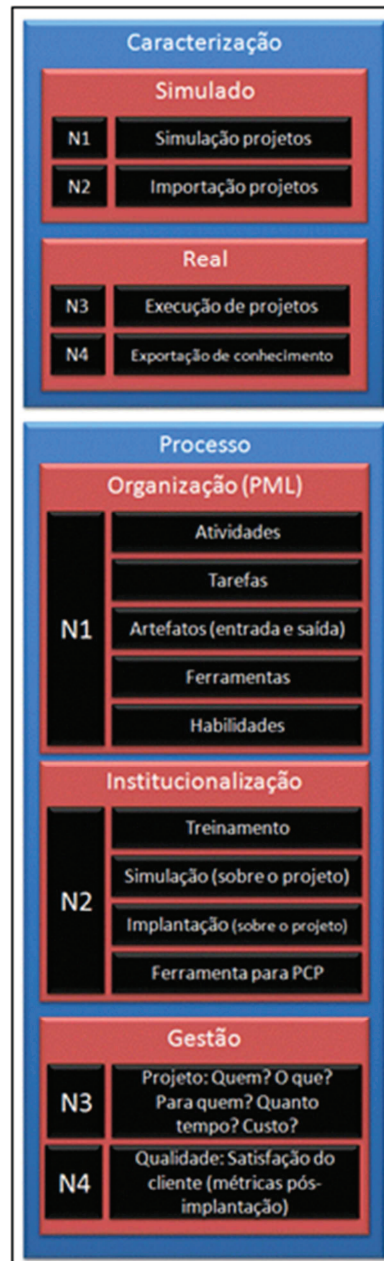
Figura 8 - Modelo para classificação de ambientes para residência em software  
Fonte: SILVEIRA DUARTE et al. (2013)

- Nível 1 - Totalmente simulado: O ambiente de residência está incubado em laboratório e os projetos de software são simulados.
- Nível 2 - Parcialmente simulado: O ambiente de residência está incubado em laboratório e os projetos de software são importados da indústria. Neste caso o ambiente também é responsável pela entrega do software.
- Nível 3 - Ambiente real de execução de projeto de software: O ambiente de residência em software é caracterizado na indústria (empresa do setor produtivo de software) e os projetos de software são desenvolvidos para atender clientes reais.
- Nível 4 - Ambiente real de execução de projetos de software e exportação de conhecimento: O ambiente de residência em software é caracterizado na indústria (empresa do setor produtivo de software) e os projetos de software são desenvolvidos para atender clientes reais. Além de atender estes clientes, o ambiente exporta conhecimento sobre processo de software, gestão de projetos e de qualidade e ferramentas.

Além da classificação dos ambientes pela sua caracterização, o modelo de classificação de ambientes de residência também prevê a classificação pela organização do processo de software, adotado pelo ambiente. Esta classificação pela organização do processo é apresentada em quatro níveis:

- Nível 1 - Nos ambientes deste nível existe a organização da *Process Modeling Language*: atividades; tarefas; artefatos de entrada e saída; ferramentas e habilidades;
- Nível 2 - Nos ambientes deste nível existe a institucionalização do treinamento, simulação (sobre o projeto), implantação (sobre o projeto) e ferramentas de planejamento e controle de processo/produção;
- Nível 3 - Nos ambientes deste nível encontra-se a gestão do projeto.
- Nível 4 - Nos ambientes deste nível encontra-se a gestão da qualidade: Satisfação do cliente (métricas pós-implantação).

É importante salientar que a classificação dos ambientes em um nível específico depende fortemente da organização do processo. Não basta o ambiente, segundo a sua caracterização (N1 – Simulação de Projeto de Software, N2 – Importação de Projeto de Software, N3 Execução de Projetos de Software, N4 – Exportação de Conhecimento), ser nível 3 por exemplo, e não possuir um processo institucionalizado (segundo a ótica do processo), neste caso o ambiente será classificado no nível 1, pois o processo está apenas organizado e modelado (via *Process Modeling Language* - PML). Este fato é explicitado por meio da Figura 8, nela a classificação se dá sobre duas óticas, a de caracterização do ambiente e de caracterização do processo.



**Figura 9 - Indicadores utilizados na Classificação de Ambientes para Residência em Software – PCP: Planejamento e Controle de Processo/Produção**  
 Fonte: SILVEIRA DUARTE et al. (2013)

Por fim, o autor salienta que a classificação é obtida por meio da relação mínima entre os níveis de caracterização de ambiente e da caracterização do processo. Um ambiente pode ser classificado no nível 2 quanto a caracterização, e no nível 1 quanto ao processo, levando-se em consideração a proposta do modelo de classificação de ambiente, o ambiente em questão será classificado no nível 1, pois o nível do processo é menor que o da caracterização.



A aplicação do modelo de classificação dos ambientes de residência em software foi caracterizado a partir das três atividades delineadas na Figura 8:

a) **Planejamento:** Conforme apresentado na Figura 8, o artefato de entrada para execução do planejamento é caracterizado pelos indicadores classificatórios de ambientes para residência de software (vide Figura 9). Estes indicadores caracterizaram a elaboração do protocolo que norteia a execução dos estudos de caso. Perceba que a organização do processo nos indicadores classificatório possui uma relação direta com os itens estrutura do processo (2.1), aplicação da PML (2.2), documento de modelagem do processo (processo institucionalizado) (2.3) e ferramentas (2.4) estabelecidas no protocolo (Quadro 2, p. 37). A institucionalização do processo apresentada nos indicadores possui relação os itens treinamento nas ferramentas e no processo (2.5) estabelecida no protocolo. A gestão de projetos e de qualidade caracterizadas nos indicadores se relacionam diretamente com as seções 4 e 5 delineadas no protocolo. Por fim os indicadores relacionados à caracterização do processo possuem relação direta com o item caracterização do ambiente (1). Por meio destas relações é possível concluir que o protocolo gerado pelos autores do trabalho se caracteriza como o artefato de saída (plano de trabalho) da atividade planejamento.

b) **Condução:** A condução da avaliação ocorreu de duas formas:

b.1) presencialmente: Os autores deste trabalho tiveram acesso aos ambientes C, D e E e neles puderam entrevistar os residentes e analisar as publicações que descrevem o processo.

b.2) a distância: Os autores localizaram os envolvidos (via contato telefônico com as instituições promotoras dos ambientes) que participaram da residência – caracterizados neste artigo como residentes e aplicaram o plano de trabalho via *call conference*.

É importante salientar que em ambas as formas de condução as informações geradas foram gravadas na íntegra, analisadas e posteriormente formatadas.

c) **Classificação:** A classificação dos ambientes é configurada com a aplicação do modelo proposto na seção 3 e nas informações estruturadas nos Quadros 3, 4 e 5. Ao analisar o modelo de classificação é possível perceber a presença de dois eixos classificatórios, o primeiro a caracterização do ambiente e o segundo o processo do processo. Ao analisar o ambiente B é possível verificar que parte da residência é feita

na indústria e os projetos são desenvolvidos para atender clientes reais (N3), porém este ambiente possui o processo modelado e não institucionalizado (N1) (vide Quadro 4), neste caso este ambiente é caracterizado no nível 1. A classificação dos demais ambientes segue a mesma sistemática e pode ser verificada por meio do Quadro 5.

NÍVEL DE CARACTERIZAÇÃO				NÍVEL DO PROCESSO			
Nível 1	Nível 2	Nível 3	Nível 4	Nível 1	Nível 2	Nível 3	Nível 4
	B			B			
		C					C
	D					D	
E				E			
		F					F
		G		G			
		H		H			
MÍNIMO (N <sub>C</sub> <sup>5</sup> , N <sub>P</sub> <sup>6</sup> )							
A	B	C	D	E	F	G	H
	Nível 1	Nível 3	Nível 2	Nível 1	Nível 3	Nível 1	Nível 1

**Quadro 5 - Classificação dos ambientes de residência**

Fonte: Adaptado de SILVEIRA DUARTE et al. (2013)

Em suma, este capítulo apresentou os referenciais teóricos sobre residência em software, as características de ambientes que desenvolveram residência em software no Brasil e um modelo para classificação dos mesmos. O autor deste trabalho no próximo capítulo apresentará uma proposta de sistematização de ambientes de residência em software brasileiros.

<sup>5</sup> Nível de caracterização: ambiente real ou ambiente simulação

<sup>6</sup> Nível do processo: organização da PML, institucionalização e gestão, vide figura (indicadores)

## 4 MÉTODOS E PROCEDIMENTOS

Para GIL (2002) e WOHLIN (2012) a classificação de uma pesquisa científica está relacionada aos seus objetivos gerais ou aos procedimentos técnicos adotados na sua elaboração.

Segundo GIL (2002) as pesquisas podem ser classificadas quanto aos objetivos em três grupos:

- a) exploratória;
- b) descritiva; e
- c) explicativa,

GIL (2002) afirma ainda que, esta classificação é útil para determinar o marco teórico ou uma aproximação conceitual.

Quanto à classificação pelos procedimentos adotados, as pesquisas estão ligadas ao delineamento<sup>7</sup>, e segundo o autor, a coleta de dados é o elemento mais importante para identificação do mesmo. Os delineamentos de uma pesquisa podem ser classificados em dois grandes grupos:

- a) o primeiro grupo, na qual a coleta de dados se dá pelas fontes chamadas pelo autor, de fontes de “papel”, neste estão a pesquisa bibliográfica e pesquisa documental; e
- b) o segundo grupo, na qual os dados são fornecidos por pessoas, neste estão à pesquisa experimental, a pesquisa *ex-post facto*, o levantamento e o estudo de caso; o autor afirma, ainda que, a pesquisa ação e pesquisa participante com uma certa controvérsia pertencem a este grupo.

Para GIL (2002, p. 47) a pesquisa experimental mostra-se como a melhor forma de se realizar uma pesquisa científica, fato adotado neste trabalho,

De modo geral, o experimento representa o melhor exemplo de pesquisa científica. Essencialmente, a pesquisa experimental consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto.

Para GIL (2002) no planejamento de uma pesquisa experimental, uma série de passos devem ser desenvolvidos para atender aos seus propósitos, a seguir, tais passos são relacionadas:

- a) formulação do problema (vide seção 4.1);

<sup>7</sup> Planejamento da pesquisa

- b) construção das hipóteses (vide seção 4.2);
- c) operacionalização das variáveis (vide seção 4.3);
- d) definição do plano experimental (vide seção 5.4);
  - d.1) determinação do sujeitos (vide seção 4.4.1);
  - d.2) determinação do ambiente (vide seção 4.4.2);
- e) coleta de dados (vide seção 4.5);
- f) análise e interpretação dos dados; e
- g) apresentação das conclusões.

Observadas as classificações das pesquisas, delineadas por GIL (2002) e tendo em vista a definição dos passos para a realização de uma pesquisa experimental, este capítulo tem como objetivo apresentar os passos de (a) à (e). Já os passos (f) e (g), análise e interpretação dos dados e conclusões serão delineadas nos capítulos 6 e 7 respectivamente.

#### 4.1 FORMULAÇÃO DO PROBLEMA

Para GIL (2002) o ponto inicial para realizar qualquer tipo de pesquisa científica é a identificação de um problema. O pesquisador ao escolher uma pesquisa experimental deve expressar de forma objetiva, precisa e clara o problema.

GIL (2002, p. 23) afirma que a definição de problema está relacionada a uma “questão não solvida e que é objeto de discussão, em qualquer domínio do conhecimento”. O autor afirma ainda que, existem problemas que não são passíveis de tratamento científicos.

Formular ou elaborar um problema de pesquisa científica é uma tarefa árdua (GIL, 2002). Segundo autor, existem regras práticas oriundas das experiências do pesquisador, que podem contribuir na difícil tarefa, o autor cita dentre as várias práticas que “o problema deve ser formulado como uma pergunta”.

Considerando a regra prática citada no parágrafo anterior, as colocações de GIL (2002), o autor deste trabalho apresenta a formulação do problema de pesquisa com uma pergunta: É possível propor um processo que possibilite sistematizar ambientes de residência na indústria de software?

A formulação do problema, como já citado, é o primeiro passo da estruturação da pesquisa experimental, dado ao término deste, nas seções seguintes serão apresentados os próximos passos.

## 4.2 CONSTRUÇÃO DA HIPÓTESE

A pesquisa experimental caracteriza-se pela clareza, precisão e parcimônia, este tipo de pesquisa por muitas vezes envolve uma única hipótese e que, essa por consequência confunde-se com o próprio problema (Gil, 2002). Segundo o autor, o que diferencia é a forma interrogativa para o problema e afirmativa para a hipótese.

Para GIL (2002, p. 94) nas pesquisas experimentais “as hipóteses referem-se, geralmente, ao estabelecimento de relações causais entre variáveis. Sugere-se que essas relações sejam definidas pela fórmula se... então”.

WAZLAWICK (2009, p. 65) esclarece que:

Um aspecto que diferencia o trabalho científico do trabalho técnico é a existência de uma hipótese de pesquisa. A hipótese é uma afirmação da qual não se sabe a princípio se é verdade ou falsa. O trabalho de pesquisa consiste justamente em tentar provar a veracidade ou falsidade da hipótese.

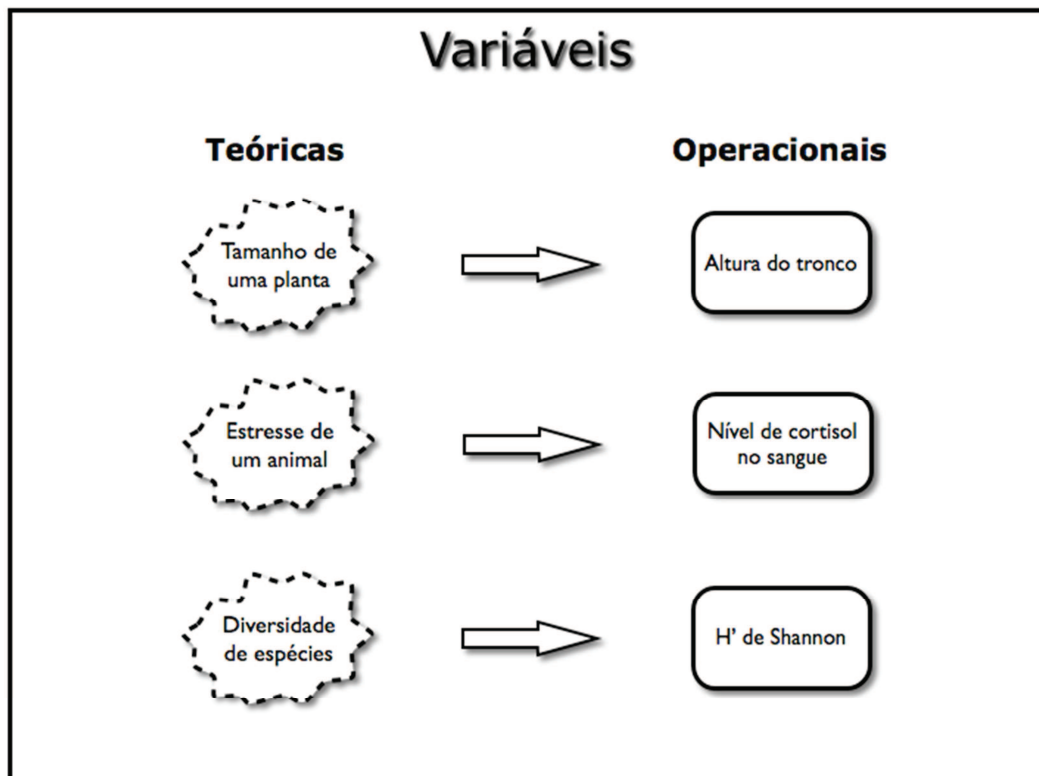
De posse dos pressupostos estabelecidos por GIL (2002) e WAZLAWICK (2009), este trabalho apresenta a seguinte hipótese, **se existem ambientes que implementam o conceito de residência em software então é possível estabelecer um processo para sistematização destes ambientes.**

## 4.3 OPERACIONALIZAÇÃO DAS VARIÁVEIS

Após a definição da hipótese é necessário caracterizar as variáveis que compõem o experimento.

Segundo GIL (2002), variável é um dos termos mais usado na linguagem dos pesquisadores, tem por objetivo proporcionar maior precisão para as hipóteses e podem assumir diferentes valores dependendo dos casos particulares ou circunstâncias experimentais.

Para MELLO (2012), um dos grandes desafios ao se conceber um projetos de pesquisa é a operacionalização da hipótese, para autor, isto é, tornar as ideias palpáveis. O autor coloca ainda que “a operacionalização é o processo de derivar previsões a partir de hipóteses, ou seja, representar uma variável teórica através de uma variável operacional”, vide Figura 10.



**Figura 10 - Operacionalização de variáveis**  
 Fonte: MELLO (2002)

Ao analisar a Figura 10 é possível encontrar a variável teórica Estresse de um animal, para mapear se o animal está estressado o pesquisador pode medir o nível de cortisol no sangue, variável esta que operacionaliza a variável teórica.

Gil (2002) afirma que as variáveis contidas na pesquisa experimental proveem o esclarecimento do que se pretende investigar, bem como sua comunicação de forma literal e não ambígua. Para o autor, isto pode ser alcançados mediante a operacionalização das variáveis.

Buscando o esclarecimento do que se pretende investigar, este trabalho tem o objetivo validar a hipótese de que, **se existem ambientes que implementam o conceito de residência em software então é possível estabelecer um processo para sistematização destes ambientes.** Para tal validação foram considerada duas variáveis operacionais:

- a) tendo em vista que um ambiente de residência em software busca a transferência de conhecimento, o ganho de conhecimento dos participantes da residência em software, se caracteriza como uma das variáveis.
- b) a aplicação do processo em ambientes de residência em software com diferentes níveis de classificação (vide seção 3.3).

#### 4.4 DEFINIÇÃO PLANO EXPERIMENTAL

Para SILVA (2005), o plano do experimento é “o conjunto completo das decisões e ações que devem ser tomadas e procedidas para a execução do experimento, estabelecido antecipadamente em forma escrita”.

GIL (2002, p. 94) expõe que “um experimento é uma pesquisa em que se manipulam uma ou mais variáveis independentes e os sujeitos são designados aleatoriamente a grupos experimentais”. Além de designar os sujeitos é necessário definir questões ambientais para a inserção dos sujeitos e a forma de coleta de dados, oriundas da execução do experimento.

##### 4.4.1 Determinação dos Sujeitos

Para GIL (2002, p. 98) “é necessário determinar com grande precisão a população a ser estudada, para isso devem ser consideradas as características que são relevantes para a clara e precisa definição da população”. Para o autor a determinação do sujeito é uma tarefa fundamental na efetivação do experimento.

GIL (2002, p. 98) afirma ainda que “a pesquisa tem por objetivo generalizar os resultados obtidos para população da qual os sujeitos pesquisados constituem uma amostra”. Para o autor, os pesquisadores encontram, com certa frequência, população tão ampla que se torna inviável considerá-la no todo. Neste caso o autor indica a técnica de randomização para os pesquisadores, cabe os mesmos escolher e estudar os sujeitos.

##### 4.4.1.1 Sujeitos do primeiro experimento

Os sujeitos do primeiro experimento foram nove alunos matriculados na disciplina Engenharia de Software, do primeiro semestre de 2013 do Programa de Pós-Graduação em Informática (PPGI) da Universidade Tecnológica Federal do Paraná, campus Cornélio Procópio.

Os sujeitos eram profissionais de ambos os sexos, com experiência em desenvolvimento de software, e como já citado alunos do PPGI, que participaram da aplicação do primeiro experimento realizado do Processo de Sistematização de Ambientes de Residência em Software Brasileiros.

#### 4.4.1.2 Sujeitos do segundo experimento

Os sujeitos do segundo experimento foram doze alunos matriculados na disciplina Engenharia de Software, do primeiro semestre de 2014 do PPGI da UTFPR, campus Cornélio Procópio.

Os sujeitos eram profissionais de ambos os sexos, com experiência em desenvolvimento de software, e como já citado alunos do PPGI, que participaram da aplicação do segundo experimento realizado do Processo de Sistematização de Ambientes de Residência em Software Brasileiros.

#### 4.4.1.3 Sujeitos do terceiro experimento

Os sujeitos do terceiro experimento foram treze alunos matriculados na disciplina Processo de Software, do segundo semestre de 2014 do PPGI da UTFPR, campus Cornélio Procópio.

Os sujeitos eram profissionais, de ambos os sexos, com experiência em desenvolvimento de software, e como já citado alunos do PPGI, que participaram da aplicação do terceiro experimento realizado do Processo de Sistematização de Ambientes de Residência em Software Brasileiros.

#### 4.4.1.4 Sujeitos do quarto experimento

Os sujeitos de quarto experimento foram os funcionários de uma empresa desenvolvedora de software. A referida empresa encontra-se sediada em uma cidade no norte do estado do Paraná.

Os sujeitos eram profissionais, de ambos os sexos, com experiência em desenvolvimento de software, analistas de sistemas e programadores, que participaram de um residência em software criada a partir dos pressupostos do Processo de Sistematização de Ambientes de Residência em Software Brasileiros.



#### 4.4.2 Determinação do Ambiente

Para GIL (2002) as pesquisas experimentais tem como ambiente o laboratório ou o campo. Independentemente do escolhido, para o autor o mesmo deve oferecer condições para que, a variável independente possa ser manipulada e os efeitos que ela causa nos sujeitos verificados.

Segundo GIL (2002, p. 99) a pesquisa experimental “quando é realizada em laboratório, a possibilidade de controle das variáveis é bem maior, já que o ambiente pode ser preparado de forma que permita a maximização dos efeitos das variáveis”.

“Nos experimentos de campo, o controle das variáveis é bastante reduzido, tanto por constituir empreendimento custoso quanto por poder artificializar situações que se desejam naturais” (GIL 2002, p. 100).

##### 4.4.2.1 Ambiente do primeiro experimento

O ambiente utilizado para a realização do primeiro experimento foi um laboratório de informática da UTFPR, campus Cornélio Procópio. O referido laboratório possuía vinte e quatro computadores, nos mesmos encontravam-se instalados os softwares necessários para a execução do primeiro experimento, entre eles é possível destacar: *Cmap*, *Freemind*, *Astah*, *Kanbanize* e *Grant Project*.

##### 4.4.2.2 Ambiente do segundo experimento

O ambiente utilizado para a realização do segundo experimento foi idêntico ao do primeiro.

##### 4.4.2.3 Ambiente do terceiro experimento

O ambiente utilizado para a realização do terceiro experimento foi um laboratório de informática da UTFPR, campus Cornélio Procópio. O referido laboratório possuía quarenta e cinco computadores, nos mesmos encontravam-se instalados os softwares necessários para a abordagem de tema relacionados a:

*JavaServer Faces, Java Persistence API (JPA), Subversion e Business Process Model and Notation (BPMN).*

#### 4.4.2.4 Ambiente do quarto experimento

O ambiente utilizado para a realização do quarto experimento foi um o que GIL (2002) considera experimento de campo. O referido ambiente foi as instalações de uma empresa desenvolvedora de software, já citada. O ambiente contava com uma sala de vinte metros, mesas, cadeiras e nove computadores.

### 4.5 COLETA DE DADOS

Para GIL (2002, p. 100) “a coleta de dados na pesquisa experimental é feita mediante a manipulação de certas condições e a observação dos efeitos produzidos”.

#### 4.5.1 Coleta de Dados do Primeiro Experimento

Para a coleta dos dados do primeiro experimento o autor deste trabalho optou pela aplicação de um questionário, o mesmo é composto por onze questões, estas por sua vez compostas por cinco alternativas das quais somente uma pode ser assinalada. Além da aplicação do questionário foram realizadas observações diretas juntos aos demais participantes. Estes no caso chamados de orquestrador e de tutor.

O questionário citado e a tabulação das respostas do mesmo podem ser observados nas páginas 69 e 70 respectivamente, já as observações diretas realizadas podem ser observadas na página 70.

#### 4.5.2 Coleta de Dados do Segundo Experimento

Para a coleta dos dados do segundo experimento o autor deste trabalho optou pela aplicação de um questionário, o mesmo tem por objetivo avaliar os conhecimentos dos sujeitos durante o processo de residência em software. O referido questionário é composto por quatro questões, estas exigindo por sua vez respostas dissertativas dos sujeitos. Além da aplicação do questionário foi realizada, como no

primeiro experimento, observações diretas juntos aos demais participantes, neste segundo experimento chamado somente de tutor.

O questionário citado e o gráfico que mostram os resultados das tabulações das respostas do mesmo podem ser observados no Anexo C e na página 81 respectivamente, já as observações diretas realizadas podem ser observadas na página 57.

#### 4.5.3 Coleta de Dados do Terceiro Experimento

Para a coleta dos dados do terceiro experimento o autor deste trabalho optou pela aplicação de um questionário, o mesmo tem por objetivo avaliar os conhecimentos dos sujeitos durante o processo de residência em software. O referido questionário é composto por vinte e seis questões, estas exigindo por sua vez respostas dissertativas dos sujeitos. Além da aplicação do questionário foi realizada, como no segundo experimento, observações diretas juntos aos demais participantes, neste terceiro experimento chamado somente de tutor.

O questionário citado e o gráfico que mostram os resultados das tabulações das respostas do mesmo podem ser observados nas páginas 89 e 93 respectivamente, já as observações diretas realizadas podem ser observadas na página 91.

#### 4.5.4 Coleta de Dados do Quarto Experimento

Para a coleta de dados do quarto experimento, este que diferentemente dos demais foi executado em uma empresa do setor produtivo de software, optou-se então pela observação dos relatórios de atividades desenvolvidas pelos analistas de sistemas e pelos programadores.

No próximo capítulo será apresentada a aplicação do Processo de Sistematização de Ambientes de Residência em Software Brasileiros seguindo os métodos e procedimentos aqui apresentados.

## **5 PROPOSTA DO PROCESSO DE SISTEMATIZAÇÃO DE AMBIENTES DE RESIDÊNCIA EM SOFTWARE BRASILEIROS**

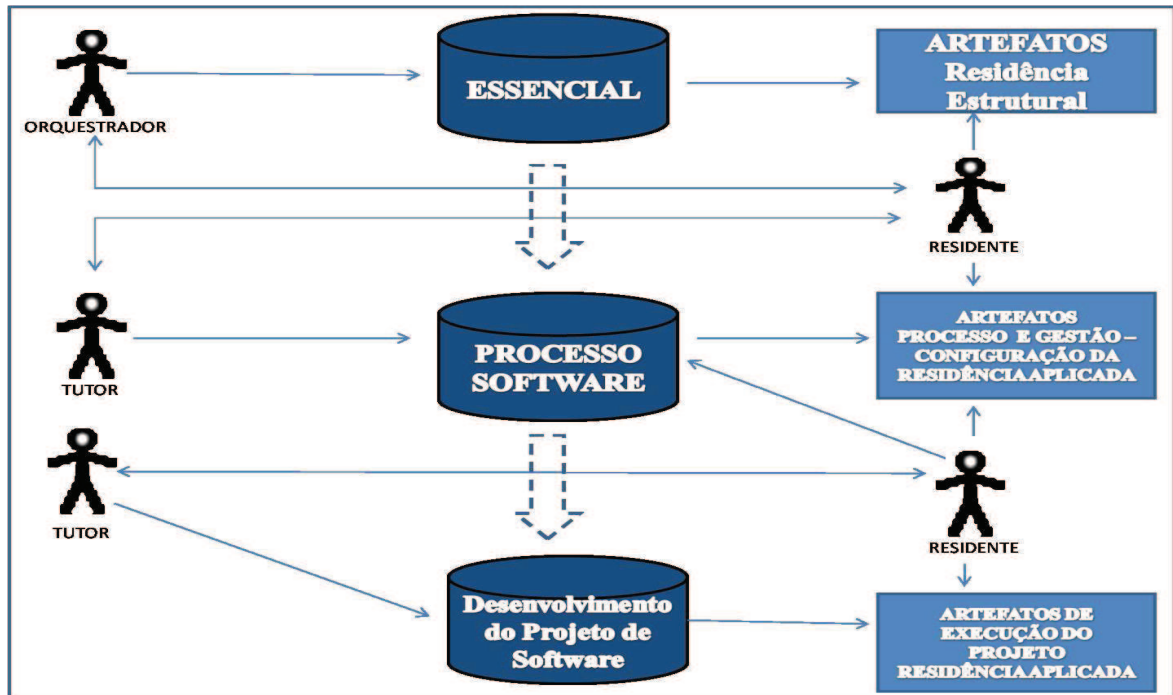
Como citado no primeiro capítulo, este trabalho tem por objetivo propor um Processo de Sistematização de Ambientes de Residência em Software Brasileiros. Para alcançar este objetivo foram realizados quatro experimentos. O primeiro e segundo foram realizados em ambientes de residência classificados no nível 1, o terceiro em um ambiente classificado no nível 2 e o quarto em um ambiente classificado no nível 3. É importante salientar que a classificação dos ambientes foi realizada de acordo com os pressupostos apresentados no capítulo 3.

Foi possível, por meio da aplicação do Processo de Sistematização, encontrar os pontos positivos e negativos e com eles reestruturar o mesmo visando sempre a otimização. Além dos pontos citados, esperava-se também, verificar se o referido Processo de Sistematização poderia atender minimamente aos seus propósitos.

Nas próximas seções serão apresentadas as descrições, as aplicações e as avaliações de cada experimento realizado.

### **5.1 DESCRIÇÃO DO PRIMEIRO EXPERIMENTO**

O primeiro experimento do processo, vide Figura 11, teve como objetivo inicial sistematizar um ambiente de residência em software, nível 1 (vide seção 3.3).



**Figura 11 - Processo de Sistematização de Ambientes de Residência em Software Brasileiros**  
 Fonte: Autoria Própria

O referido processo apresenta vários componentes, como pode ser observado. Alguns chamados, pelo autor deste trabalho, de componentes “humanos” e outros caracterizados como componentes de “armazenamento de conhecimento”. Os chamados “humanos” são: o Orquestrador, o Tutor e o Residente. Já os chamados de “armazenamento de conhecimento” são: a Base de Conhecimento Essencial, a Base de Conhecimento de Processo de Software e a Base de Conhecimento Desenvolvimento do Projeto de Software.

O Orquestrador é um professor ou especialista em desenvolvimento de software, com a finalidade de:

- a) definir e instanciar as bases de conhecimentos;
- b) aplicar a residência estrutural, os conceitos inerentes à residência podem ser verificados no capítulo 3; e
- c) treinar os residentes.

O Residente é um aluno de pós-graduação que busca se especializar em conceitos de qualidade, processo de produção e gestão de projetos na área de engenharia de software.

A Base de Conhecimento Essencial possui os conteúdos considerados como conhecimentos essenciais para os residentes. No modelo de gestão de conhecimento apresentado por TRINDADE (2006) a Base de Conhecimento

Essencial é gerada durante e execução da atividade de codificação. Para FABRI (2011) a referida base é gerada durante a atividade de estruturação. Já para BATISTA (2012) a base é gerada durante a execução da atividade de armazenamento.

A Base de Conhecimento Essencial é instanciada pelo orquestrador, esta instanciação proporciona a geração de artefatos úteis que os residentes podem recorrer ao participarem da residência aplicada. Os conhecimentos essenciais que compõem a referida base estão relacionados a:

- a) requisitos de software;
- b) formas de especificação de requisitos;
- c) especificação de projeto;
- d) ferramentas para especificação de projeto;
- e) processo de software;
- f) gestão de processo de software; e
- g) ferramentas de organização de processo;

O Tutor pode ser caracterizado como professor ou profissional da área, especialista em desenvolvimento de software. A função principal do tutor é acompanhar a aplicação da base de conhecimento de processo de software e da base de conhecimento do desenvolvimento do projeto de software, quando os residentes participavam da residência aplicada (vide seção 3). É importante salientar que, a diferença entre o Orquestrador e o Tutor é o papel que exercem. O primeiro ligado a residência estrutural e o segundo a residência aplicada.

A Base de Conhecimento do Processo de Software tem por objetivo proporcionar aos residentes a instanciação de artefatos, ferramentas, modelos úteis na prática da residência aplicada<sup>8</sup>. No modelo de gestão de conhecimento apresentado por TRINDADE (2006) a Base de Conhecimento do Processo de Software é gerada durante e execução da atividade de disseminação. Para FABRI (2011) a referida base é gerada durante a execução da atividade estruturação. Já para BATISTA (2012) a base é gerada durante a execução da atividade de compartilhamento.

A Base de Conhecimento do Processo de Software possui conhecimentos relacionados a:

---

<sup>8</sup> Os conceitos sobre residência aplicada podem ser verificados no capítulo 3.

- a) processo: na base de conhecimento é apontado o modelo de processo que podem ser utilizados pelos residentes. Exemplo: cascata, *Rapid Application Development* (RAD), incremental ou espiral (SOMMERVILLE, 2011). Os paradigmas tradicionais e ágeis de processos também são apresentados;
- b) atividade: todo processo é fracionado em subprocessos, neste trabalho um subprocesso é denominado como uma atividade. Exemplos: levantar requisitos, projetar software e implementar projetos;
- c) tarefa: toda atividade é fracionada em tarefas, na atividade de levantamento de requisitos é possível identificar duas tarefas: 1ª) identificar usuários chaves e 2ª) identificar requisitos funcionais. Estas tarefas pertencem à atividade de levantamento de requisitos.
- d) artefatos: toda tarefa possui artefatos de entrada ou saída. Exemplos: diagramas, documentos e código fonte;
- e) habilidade: toda tarefa a ser executada requer a identificação das habilidades dos residentes, para que os mesmos tenham condições de executá-la; e
- f) ferramentas: a resolução de uma tarefa requer uso de ferramentas. Exemplos: *Cmap*<sup>9</sup>, *Freemind*<sup>10</sup>, *Astah*<sup>11</sup>, *Kanbanize*<sup>12</sup>, *Grant Project*<sup>13</sup>.

O último componente proposto no processo é a Base de Conhecimento do Desenvolvimento do Projeto de Software. No modelo de gestão de conhecimento apresentado por TRINDADE (2006) a Base de Conhecimento do Desenvolvimento do Projeto de Software é gerada durante a execução da atividade de apropriação. Para FABRI (2011) a referida base é gerada durante a atividade de verificação de compreensão. Já para BATISTA (2012) a base é gerada durante a execução da atividade de aplicação.

Na Base de Conhecimento do Desenvolvimento do Projeto de Software é possível encontrar informações sobre as atividades executadas no projeto. A confecção das estimativas de custos, prazos, recursos, cronograma e a estrutura analítica do projeto (EAP) são os primeiros artefatos a serem instanciados, nesta base pelos residentes. Estes artefatos são utilizados ativamente na gestão de

<sup>9</sup> Disponível para *download* <http://cmapttools.softonic.com.br/download>

<sup>10</sup> Disponível para *download* <http://freemind.softonic.com.br/download>

<sup>11</sup> Disponível para *download* <http://astah.net/download>

<sup>12</sup> Acessível em <https://kanbanize.com>

<sup>13</sup> Disponível para *download* <http://www.ganttproject.biz/download.php>

projeto de software. Logo após a instanciação dos artefatos os residentes podem instanciar:

- a) descrição de caso de uso;
- b) diagrama entidade-relacionamento;
- c) diagrama de classe e sequência; e
- d) projeto de interface.

É importante observar que a instanciação dos artefatos acima citados são somente para projetos de software do mesmo domínio de conhecimento.

Nas próximas seções será apresentadas a aplicação e a avaliação do Processo de Sistematização de Ambientes de Residência em Software.

### 5.1.1 Aplicação do Primeiro Experimento

A aplicação do primeiro experimento do Processo de Sistematização de Ambientes de Residência em Software Brasileiros foi realizada no primeiro semestre de 2013 do Programa de Pós-Graduação em Informática (PPGI) da Universidade Tecnológica Federal do Paraná (UTFPR), campus Cornélio Procopio. Nove profissionais, de ambos os sexos, com experiência em desenvolvimento de software, neste caso alunos do programa, participaram da aplicação do referido processo. A aplicação foi realizada em ambiente totalmente simulado, nível 1 (vide seção 3.3)

A aplicação do processo foi dividida em duas fases:

- a) residência estrutural; e
- b) residência aplicada.

A residência estrutural consiste na aplicação de treinamentos aos residentes, na aplicação foram utilizadas trinta e duas horas. Este número de horas foi definido pelo orquestrador, o mesmo levou em consideração a área da residência em software escolhida. A referida quantidade de horas foi distribuída em blocos de quatro horas, com início às 08h e término às 12h, nos dias 1º, 8, 15 e 22 do mês de março e 05, 12 de mês de abril do ano de 2.013, e no dia 19 de abril o treinamento teve duração de oito horas, das 8h às 12h e das 13h às 17h.

O treinamento, de responsabilidade do orquestrador, como já citado, teve como foco o aprimoramento dos conhecimentos teóricos e práticos relacionados à:



- a) especificação de requisitos e uso ferramentas para tal;
- b) processo de software;
- c) gestão de projeto de software; e
- d) gestão da qualidade do produto desenvolvido.

Os conhecimentos citados são considerados essenciais, pelo orquestrador, para a questão de desenvolvimento de software e por este motivo, os mesmos são preservados em repositórios de dados.

A residência aplicada (vide seção 3) de responsabilidade do tutor foi realizada no dia 26 de abril de 2013 no período das 8h 20m às 12h 20m e das 13h 20m às 17h20m, exatamente uma semana após o término da residência estrutural. Os residentes formaram dois grupos de trabalho, um chamado de empresa Alfa e o outro de empresa Beta, e depois foram instruídos sobre os trabalhos que deveriam realizar. Estes ligados a gestão de projeto, projeto de software e implementação.

É importante salientar que os conceitos sobre residência em software apresentados no capítulo 3, não foram apresentados formalmente aos residentes na residência estrutural. Fato este que levou o tutor a apresentar o referido conceito por meio do Quadro 6.

#### **Residência em Software**

Pessoal, nesta atividade você irá participar de uma "micro" de residência em software. Para entender melhor este conceito é de fundamental importância que você leia o texto apresentado no link: <http://engenhariasoftware.wordpress.com/2008/05/13/um-paralelo-entre-residencia-medica-e-a-residencia-em-software/>

Após a leitura do texto você irá se deparar com o documento de requisitos de software e o processo de desenvolvimento. O processo conta com duas atividades, projeto de software e implementação, além claro, da gestão de projetos. No processo você irá verificar a presença das tarefas, artefatos de entrada e de saída, além, das ferramentas que você irá utilizar (estas podem ser baixadas diretamente pelo modle)

Neste ambiente - o moodle - é caracterizado com uma base essencial de conhecimento. Nesta base você encontra todos os subsídios para sanar as possíveis dúvidas que surgiram. Além desta base, alguns artefatos de saída, pertinentes ao processo, são disponibilizados nos links abaixo, ele, com certeza, o ajudarão, a confeccionar os referidos artefatos contemplados pelo processo.

Além de confeccionar o produto, você deve gerir as atividades pertinentes ao processo. Nesta situação você irá utilizar a técnica Kanban.

O produto deve ser construindo em JAVA, C, PHP ou PASCAL.

Um bom trabalho a todos.

#### **Quadro 6 - Apresentação da residência em software**

**Fonte: Autoria Própria**

Após a apresentação dos conceitos de residência em software foi apresentado aos residentes às atividades que os mesmos deveriam executar. Todas descritas e à disposição no ambiente virtual de aprendizagem. Além das atividades

havia documentos que exemplificavam como processo de desenvolvimento deveria ser executado.

Antes de executarem as atividades solicitadas, os residentes deveriam fazer a leitura de três mapas mentais que detalhavam o processo de software<sup>14</sup>. O primeiro mapa apresenta os dados da atividade de gestão do projeto, o segundo do projeto de software e o terceiro da implementação, os mapas podem ser observados no Anexo A. É importante salientar que a opção de descrever o processo de software por meio de um mapa mental foi do tutor. Depois de ler os três mapas, os residentes deveriam instalar todas as ferramentas delineadas pelo referido documento em duas máquinas, durante a instalação os residentes deveriam em um *post-it*<sup>15</sup>, anotar quem, está fazendo o que e em quanto tempo (em minutos).

Terminada as instalações e anotações, os alunos acessavam o documento de requisitos que, nesta residência caracterizou-se como a situação/problema. A referida situação/problema escolhida foi “Projeto Genoma<sup>16</sup>”, (vide Quadro 7). Além da descrição do Projeto Genoma, o documento de requisitos possuía outras informações, a seguir:

- a) Missão (vide Quadro 8);
- b) Entrada de Dados (vide Quadro 9);
- c) Saída de Dados (vide Quadro 10); e
- d) Restrições (vídeo Quadro 11).

Os residentes deveriam considerar todas as informações citadas nos Quadros 7, 8, 9, 10 e 11 para a confecção do produto software.

---

<sup>14</sup> Para HERMAN; BOVO (2005) “um mapa mental é essencialmente um diagrama hierarquizado de informações, na qual podemos facilmente identificar as relações e os vínculos entre as informações”.

<sup>15</sup> Pequeno pedaço de papel para deixar lembretes.

<sup>16</sup> Este foi a primeira tarefa do caderno de tarefa da Olimpíada Brasileira de Informática de 1999. Acessível em [www. http://olimpiada.ic.unicamp.br/passadas/pdf/provas/ProvaOBI1999.pdf](http://olimpiada.ic.unicamp.br/passadas/pdf/provas/ProvaOBI1999.pdf)

“Um grande projeto mundial está em curso para mapear todo o material genético do ser humano: o Projeto Genoma Humano. As moléculas de DNA (moléculas que contêm material genético) podem ser representadas por cadeias de caracteres que usam um alfabeto de apenas 4 letras: ‘A’, ‘C’, ‘T’ e ‘G’. Um exemplo de uma tal cadeia é:

TCATATGCAAATAGCTGCATACCGA

Você deverá produzir uma ferramenta muito utilizada projeto Genoma: um programa que procura ocorrências de uma pequena cadeia de DNA (que vamos chamar de  $p$ ) dentro de uma outra cadeia de DNA (que vamos chamar de  $t$ ). Você deverá procurar dois tipos de ocorrência: a "direta" e a "complementar invertida".

Uma ocorrência direta é quando a cadeia  $p$  aparece como subcadeia dentro de  $t$ .

Por exemplo, se  $p = \text{CATA}$

$t = \text{TCATATGCAAATAGCTGCATACCGA}$ ,

então  $p$  ocorre na forma direta na posição 2 e na posição 18 de  $t$ .

Uma ocorrência complementar invertida depende da seguinte correspondência entre as letras do DNA: ‘A’  $\leftrightarrow$  ‘T’ e ‘G’  $\leftrightarrow$  ‘C’. "Complementar o DNA" significa trocar as letras de uma cadeia de DNA seguindo essa correspondência. Se complementarmos a cadeia CATA, vamos obter GTAT. Mas além de complementar, é preciso também inverter, ou seja, de GTAT obter TATG. E é esta cadeia que deverá ser procurada, no caso da ocorrência complementar invertida. Assim, se  $p$  e  $t$  são as mesmas cadeias do exemplo anterior, então  $p$  ocorre na forma complementar invertida na posição 4 de  $t$ .

#### **Quadro 7 - Descrição do projeto genoma**

**Fonte: Olimpíadas Brasileira de Informática, Caderno de Tarefas (1999)**

#### **1. Missão**

Sua tarefa é escrever um programa que, dadas  $n$  cadeias  $p$  e  $t$ , onde o comprimento de  $p$  é menor ou igual ao comprimento de  $t$ , procura todas as ocorrências diretas e todas as ocorrências complementares invertidas de  $p$  em  $t$ .

#### **Quadro 8 - Missão dos residentes**

**Fonte: Olimpíadas Brasileira de Informática, Caderno de Tarefas (1999)**

#### **2. Entrada de Dados**

Você deve construir uma interface para gerar o arquivo GENOMA.IN - o referido arquivo contém vários testes. Cada conjunto de teste é composto por três linhas. A primeira linha contém dois inteiros positivos,  $M$  e  $N$ ,  $M \leq N$ , que indicam respectivamente o comprimento das cadeias de DNA  $p$  e  $t$ , conforme descrito acima. A segunda linha do conjunto de teste contém a cadeia  $p$ , e a terceira linha contém a cadeia  $t$ , onde  $p$  e  $t$  são compostas utilizando apenas os caracteres ‘A’, ‘C’, ‘G’ e ‘T’. O final do arquivo de testes é indicado quando  $M = N = 0$  (este último de testes não é válido e não deve ser processado). O arquivo GENOMA.IN contém ao menos um conjunto de teste que deve ser processado.

Exemplo de Entrada: GENOMA.IN

2 4

AC

TGGT

4 25

CATA

TCATATGCAAATAGCTGCATACCGA

0 0

#### **Quadro 9 - Entradas de dados**

**Fonte: Olimpíadas Brasileira de Informática, Caderno de Tarefas (1999)**

### 3. Saída de Dados

Seu programa deve produzir um arquivo de saída chamado GENOMA.OUT. Para cada conjunto de teste do arquivo de entrada seu programa deve produzir quatro linhas no arquivo de saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n", onde  $n$  é numerado a partir de 1. Na segunda linha deve aparecer a lista, em ordem crescente, com a posição inicial de cada ocorrência, na forma direta, do padrão  $p$  na sequência  $t$ . Na terceira linha deve aparecer a lista, em ordem crescente, com a posição inicial de cada ocorrência, na forma complementar invertida, do padrão  $p$  na sequência  $t$ . A quarta linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída – GENOMA.OUT

Teste 1

ocorrência direta: 0

ocorrência complementar invertida: 3

Teste 2

ocorrência direta: 2 18

ocorrência complementar invertida: 4

(esta saída corresponde ao exemplo de entrada acima)

IMPORTANTE: você deve gerar uma segunda interface que abra o arquivo .OUT e exiba as informações para o usuário.

#### Quadro 10 - Saída de dados

Fonte: Olimpíadas Brasileira de Informática, Caderno de Tarefas (1999)

### 4. Restrições

$1 \leq M \leq 15000$

$1 \leq N \leq 15000$

$M \leq N$

$M = 0$  e  $N = 0$  apenas para indicar o fim do arquivo de entrada"

#### Quadro 11 - Restrições

Fonte: Olimpíadas Brasileira de Informática, Caderno de Tarefas (1999)

Os residentes, após a leitura do documento de requisitos, acessaram um *template* confeccionado pelo tutor. O *template* (vide Quadro 12) solicita aos residentes o nome do projeto, a contagem dos pontos por função, o cronograma e o custo. É importante salientar que os conceitos citados foram estabelecidos na residência estrutural.

UTFPR – UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – Campus Cornélio Procópio  
PPGI – PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA  
Engenharia de Software

Nome do projeto: <<inserir o nome do projeto>>

Tamanho: <<em pontos por função>>

Cronograma:

<<insira o diagrama de gantt aqui – lembre-se que o diagrama deve ser visível>>

#### Quadro 12 - Andamento do desenvolvimento do projeto

Fonte: Autoria Própria

Os residentes a partir deste ponto começaram a desenvolver o software, como detalhado no documento de requisitos (vide nos Quadro 7, 8, 9,10 e

11), e também a postar as atividades solicitadas pelo tutor (vide Quadro 13) em horários previamente definidos.

	<b>Atividades</b>	<b>Horário Disponível para Postagem</b>	<b>Área do Processo de Software</b>
1	Mapear orçado	das 8h20m às 9h 40m	Gestão
2	Interface do projeto	das 8h20m às 12h15m	Projeto
3	Diagrama de Caso de Uso, Classes, Sequência e Entidade-Relacionamento	das 8h20m às 12h 15m	Projeto
4	Informar o estado do projeto	das 11h às 11h 30m	Gestão
5	Informar o estado do projeto	das 14h às 14h 30m	Gestão
6	Mapear o realizado	das 15h às 15h 45m	Gestão
7	Registrar as informações anotadas no <i>Post-it</i> um ferramenta automatizada	das 15h 30m às 16h	Gestão

**Quadro 13 - Atividades que os residentes desenvolveram**

Fonte: Autoria Própria

Além das atividades apresentadas no Quadro 13, os residentes apresentaram a implementação aos tutores às 16h para avaliação.

Os residentes foram acompanhados e avaliados pelo tutor, na avaliação foram observados fatores relacionados ao processo de software, ao tempo, a produção e qualidade do produto software desenvolvido.

Durante a execução da residência aplicada, esta ligada à engenharia de software, o especialista (tutor) apresentou os resultados positivos e negativos encontrados aos residentes. Um exemplo pode ser observado no Quadro 14.

**Atividade postada:** Diagrama de Caso de Uso, Classes, Sequência e Entidade-Relacionamento.

**Resultados:** No seu Diagrama de Sequência existem objetos sem classe. Vocês poderiam melhorar a organização da árvore de navegação. Vocês consultaram o exemplo. Vocês não preencheram a descrição do caso de uso.

**Quadro 14 - Um exemplo de resultado da avaliação do tutor**

Fonte: Autoria própria

Em suma, esta seção apresentou o primeiro experimento do Processo de Sistematização de Ambientes de Residência em Software Brasileiros, em um ambiente nível 1 (vide seção 3.3).

### 5.1.2 Avaliação do Primeiro Experimento

Na avaliação do primeiro experimento mapeou-se as impressões dos residentes, do orquestrador e do tutor sobre a execução da residência estrutural e da residência aplicada.

A coleta de dados sobre as impressões dos residentes treinados pelos pressupostos do referido processo, ocorreu por meio de um questionário.

As respostas do questionário baseava-se na escala proposta por Rensis Likert. AMARO (2005, p. 3) em sua obra salienta que “a escala de Likert apresenta uma série de cinco proposições, das quais o inquirido deve seleccionar uma, podendo estas ser: concorda totalmente, concorda, sem opinião, discorda, discorda totalmente”. A seguir a escala adota para respostas do questionário:

- a) 0 - Nunca;
- b) 1 - Quase Nunca;
- c) 2 - Às Vezes;
- d) 3 - Quase Sempre; e
- e) 4 - Sempre.

O questionário aplicado pode ser observado no Quadro 15.

Prezado (a).

Com o intuito de minimizar a deficiência na formação da mão de obra para a área de engenharia de software, algumas empresas e universidade criaram a ideia da residência em software (algo semelhante à residência médica). A residência em software busca proporcionar ao aluno de pós-graduação ou ao profissional inserido na indústria, uma vivência prática dentro de um ambiente real que possua políticas de qualidade bem definidas, com o objetivo de promover a disseminação dos conceitos de qualidade, processo de produção e gestão de projetos na área de engenharia de software.

Considerada a sua experiência como residente em software na residência aplicada na última aula da disciplina de Engenharia de Software (1º semestre de 2013) do Programa de Pós-Graduação em Informática da UTFPR campus Cornélio Procópio, onde foi utilizada a base de conhecimento essencial responda as questões abaixo:

a) Idade: \_\_\_\_\_

b) Qual a formação: \_\_\_\_\_

c) Quanto tempo de formado na graduação: \_\_\_\_\_ d) Possui especialização: \_\_\_\_\_

e) Quanto tempo está na área de computação: \_\_\_\_ anos

f) Quanto tempo de experiência em desenvolvimento de software: \_\_\_\_ anos

01) A base de conhecimento essencial forneceu instrumentação para realização de procedimento documentado para estimativa de tamanho de projeto?

( ) 0-Nunca ( ) 1-Quase nunca ( ) 2-Às vezes ( ) 3-Quase sempre ( ) 4-Sempre

02) A base de conhecimento essencial forneceu instrumentação para realização de procedimento de estimativa de custo, cronograma e esforços?

( ) 0-Nunca ( ) 1-Quase nunca ( ) 2-Às vezes ( ) 3-Quase sempre ( ) 4-Sempre

03) A base de conhecimento essencial forneceu instrumentação para rastrear requisitos?

( ) 0-Nunca ( ) 1-Quase nunca ( ) 2-Às vezes ( ) 3-Quase sempre ( ) 4-Sempre

04) A base de conhecimento essencial forneceu instrumentação para desenvolver projetos adotando conceito dos modelos de qualidade de software?

( ) 0-Nunca ( ) 1-Quase nunca ( ) 2-Às vezes ( ) 3-Quase sempre ( ) 4-Sempre

05) A base de conhecimento essencial forneceu instrumentação clara sobre que não se deve iniciar projeto sem realizar planejamento?

( ) 0-Nunca ( ) 1-Quase nunca ( ) 2-Às vezes ( ) 3-Quase sempre ( ) 4-Sempre

06) A base de conhecimento essencial forneceu instrumentação que proporciona integração do trabalho em equipe?

( ) 0-Nunca ( ) 1-Quase nunca ( ) 2-Às vezes ( ) 3-Quase sempre ( ) 4-Sempre

07) A base de conhecimento essencial forneceu instrumentação para melhor documentar os sistemas?

( ) 0-Nunca ( ) 1-Quase nunca ( ) 2-Às vezes ( ) 3-Quase sempre ( ) 4-Sempre

08) A base de conhecimento essencial forneceu instrumentação suficiente para que um novo membro/colaborador se integre facilmente ao grupo de desenvolvedores?

( ) 0-Nunca ( ) 1-Quase nunca ( ) 2-Às vezes ( ) 3-Quase sempre ( ) 4-Sempre

09) A base de conhecimento essencial forneceu ferramentas para especificação de projeto?

( ) 0-Nunca ( ) 1-Quase nunca ( ) 2-Às vezes ( ) 3-Quase sempre ( ) 4-Sempre

10) O quanto o processo de Residência em Software contribuiu para o seu conhecimento em desenvolvimento de software?

( ) 0-Nada ( ) 1-Pouca ( ) 2-Média ( ) 3-Alta ( ) 4-Muito Alta

11) Você indicaria a Residência em Software para outros profissionais ou para indústria de software?

( ) 0-Não ( ) 4-Sim

Obrigado pela sua participação.

**Quadro 15 - Questionário para avaliação do primeiro experimento**

**Fonte: Autoria Própria**

No Quadro 16 é possível observar a tabulação das respostas dos residentes.

Questões	Nunca	Quase Nunca	Às Vezes	Quase Sempre	Sempre
01	0	0	0	4	5
02	0	0	0	6	3
03	0	0	0	3	6
04	0	0	1	1	7
05	0	0	0	1	8
06	0	0	0	1	8
07	0	0	0	3	6
08	0	0	1	4	4
09	0	0	0	1	8
10	0	0	0	5	4
11	-	-	-	-	9

**Quadro 16 - Tabulação das respostas dos residentes**

**Fonte: Autoria Própria**

Além da aplicação do questionário, o orquestrador e o tutor também avaliaram o processo por meio da observação direta. No Quadro 17 são descritas as três observações que o orquestrador e o tutor fizeram.

1. O Orquestrador e o Tutor concordam que, o Processo de Sistematização de Ambientes de Residência em Software Brasileiros pode ser conduzido ou aplicado por um único especialista, uma vez que, ambos apresentam as mesmas especializações em desenvolvimento de projeto de software, e neste caso podem desempenhar os mesmos papéis.
2. O Orquestrador e o Tutor sugerem que, o Processo de Sistematização de Ambientes de Residência em Software Brasileiros tenha uma orientação por fluxos de ações na sua dinâmica de execução, pois, segundo os mesmos o referido processo com uma nova sequencialização orientada por fluxos de ações poderiam tornar a execução mais otimizada.
3. O Orquestrador e o Tutor observaram que a residência em software aplicada buscou permear todos os conceitos de Engenharia de Software. Para eles seriam interessante fatiar estes conceitos. Dentro desta premissa a residência seria efetuada dentro de um conceito específico, como por exemplo: Gestão de Projeto de Software.

**Quadro 17 - Observação direta do orquestrador e do tutor.**

**Fonte: Autoria Própria**

Considerando as respostas dos residentes e as observações do orquestrador e do tutor, o autor deste trabalho realizou alterações no processo.

Na próxima seção serão apresentadas as alterações e a nova forma do processo.



## 5.2 DESCRIÇÃO DO SEGUNDO EXPERIMENTO

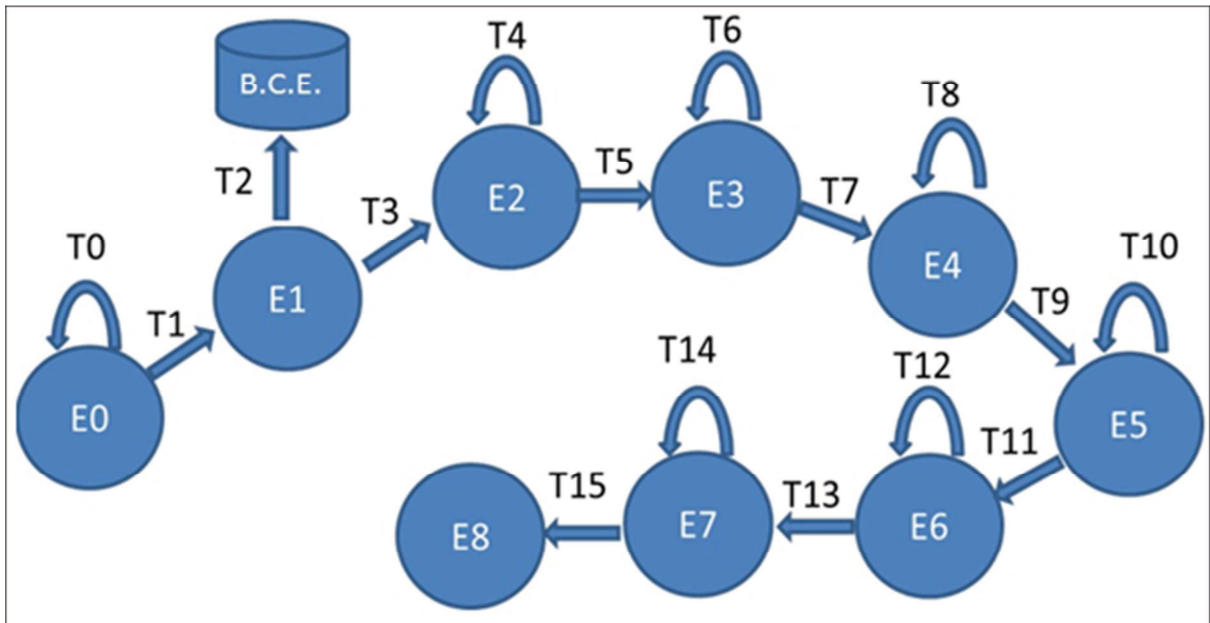
O segundo experimento do Processo de Sistematização de Ambientes de Residência em Software Brasileiros iniciou com a construção de uma nova versão do processo.

As sugestões que alicerçaram a confecção da nova versão, conforme já citadas, foram oriundas das avaliações e/ou observações dos alunos/residentes, do orquestrador e do tutor que participaram da primeira aplicação do processo, e por solicitação da organização do Processo em estados, isto, pelos membros da banca que avaliaram a proposta de dissertação do autor deste trabalho. A seguir o autor deste trabalho apresenta as duas alterações realizadas no Processo de Sistematização de Ambientes de Residência em Software Brasileiros.

A primeira alteração foi a eliminação da figura do orquestrador, isto ocorreu porque na aplicação do primeiro experimento percebeu-se que, tanto o tutor quanto o orquestrador podiam desempenhar o mesmo papel, por esta razão optou-se pela manutenção o tutor.

A segunda alteração foi a sequencialização das atividades do processo, na nova versão a aplicação passou a ser orientada por fluxo de ações, estas foram tratadas como estado, pelo autor deste trabalho. Esta alteração, segundo o tutor, tende a proporcionar um maior entendimento do Processo de Sistematização de Ambientes de Residência em Software Brasileiros e com as descrições de cada estado, a uma maior facilidade na operacionalização dos trabalhos. É importante destacar que o tutor da segunda aplicação foi o mesmo da primeira.

Com a adoção da sequencialização das atividades do Processo de Sistematização de Ambientes de Residência em Software Brasileiros, o autor deste trabalho utilizou uma Máquina de Estado para representá-lo. A escolha de uma Máquina de Estados ocorreu, pois, como acima citado, as atividades passaram a ser orientadas por fluxos de ações, e estas tratadas como estado. É importante destacar que, o objetivo deste trabalho é a aplicação do Processo de Sistematização de Ambientes de Residência em Software Brasileiros e não discutir se a uma Máquina de Estado é a melhor forma de representação do referido processo. A Figura 12 ilustra a nova versão do Processo de Sistematização de Ambientes de Residência em Software Brasileiros.



**Figura 12 - Nova versão do Processo de Sistematização de Ambientes de Residência em Software Brasileiros**

Fonte: Autoria Própria

A nova versão adotada para o Processo de Sistematização de Ambientes de Residência em Software Brasileiros não exhibe formalmente a imagem do tutor, mas, é importante salientar que o mesmo participa assiduamente de toda a execução. A seguir será apresentada a composição da máquina de estado e as execuções que cada estado desenvolve.

A máquina de estado é composta por nove estado, de E0 a E8. A seguir no Quadro 18, os estados são relacionados.

Abreviação	Descrição
E0	Estado 0, Definição da Área de Residência
E1	Estado 1, Construção da Base de Conhecimento Essencial
E2	Estado 2, Residência Estrutural
E3	Estado 3, Avaliação Pré-Residência Aplicada
E4	Estado 4, Definição da Residência Aplicada
E5	Estado 5, Execução da Residência Aplicada
E6	Estado 6, Avaliação Pós-Residência Aplicada
E7	Estado 7, Avaliação do Ganho de Conhecimento
E8	Estado 8, Análise de Resultados

**Quadro 18 - Estado da máquina de estados**

Fonte: Autoria Própria

Quinze transições de estados proporcionam a mudança dos estados que formam o Processo, no Quadro 19 estão relacionadas transições.

Abreviação	Descrição
T0	Transição 0, Área da Residência em Definição
T1	Transição 1, Área Definida
T2	Transição 2, Base Essencial em Instanciação
T3	Transição 3, Base Essencial Instanciada
T4	Transição 4, Aplicando a Residência Estrutural (Treinamento)
T5	Transição 5, Treinamento Aplicado
T6	Transição 6, Avaliação Pré-Residência em Execução
T7	Transição 7, Avaliação Pré-Residência Executada
T8	Transição 8, Residência Aplicada em Definição
T9	Transição 9, Definição da Residência Aplicada Concluída
T10	Transição 10, Residência Aplicada em Execução
T11	Transição 11, Residência Aplicada Concluída
T12	Transição 12, Avaliação da Pós-Residência em Execução
T13	Transição 13, Avaliação da Pós-Residência Executada
T14	Transição 14, Comparando os Resultados da Pré e Pós em Execução
T15	Comparação entre os Resultados da Pré e Pós Concluída

**Quadro 19 - Transições da máquina de estados**

**Fonte: Autoria Própria**

A seguir os estados e transições serão detalhados:

- a) E0, Definição da Área da Residência: a escolha da área da residência em software tem um paralelismo estreito com a escolha da área da residência em medicina, enquanto a de software poder ser, por exemplo, em: teste de software, processo de software, gestão de projeto de software entre outros; a da medicina pode ser, por exemplo em: ortopedia, cardiologia, ginecologia, pediatra entre outros. Neste estado duas transições ocorrem, a primeira, T0, área da residência em software em definição, e a segunda, T1, quando a área é definida;
- b) E1, Construção da Base de Conhecimento Essencial: o tutor, após a escolha da área da residência, instância a base de conhecimento essencial com conteúdos relacionados a área. Um ambiente virtual de aprendizagem é utilizado como meio de acesso as informações para os residentes. Neste estado duas transições ocorrem, a

primeira, T2, que instancia a base de conhecimento essencial, e a segunda, T3, quando a instanciação é finalizada.

c) E2, Residência Estrutural: o tutor, após a instanciação da base de conhecimento essencial, inicia os treinamentos utilizando os conteúdos da referida base, a partir deste momento o tutor e os residentes começam a se interagir. É importante salientar que, a quantidade de horas destinadas aos treinamentos está diretamente ligada a área de residência escolhida. Neste estado duas transições ocorrem, a primeira, T4, quando o treinamento em laboratório é aplicado, e a segunda, T5, quando o treinamento é finalizado. É importante destacar que o tempo de treinamento é definido durante a execução dos estados E1 e E2.

d) E3, Avaliação Pré-Residência Aplicada: o tutor elabora e aplica uma avaliação individual para mensurar os conhecimentos que os residentes apresentam antes da realização da residência aplicada. Esta avaliação pode ser em questionário de papel ou eletrônico. Neste estado duas transições ocorrem, a primeira, T6, com aplicação da avaliação pré-residência, e a segunda, T7, quando terminada a aplicação da avaliação.

e) E4, Definição da Residência Aplicada: o tutor apresenta aos residentes uma situação/problema, este, relacionado a área da engenharia de software delineada no E0, o processo que será utilizado para a solução do problema e o ambiente de execução. Em um paralelismo com a residência médica, o médico tutor apresenta aos residentes a possível patologia do paciente e os procedimentos prévios que devem ser realizados. Neste estado duas transições ocorrem, a primeira, T8 caracterizada como a residência aplicada em definição, e a segunda, T9, quando concluída.

f) E5, Execução da Residência Aplicada, de posse da definição da residência aplicada, inicia-se a execução da mesma, os residentes formam equipes de trabalho, para resolverem a situação/problema apresentada no Estado 4. Neste estado duas transições ocorrem, a primeira, T10, com a execução da residência aplicada, os participantes tem um número de horas determinada pelo tutor para concluírem os trabalhos. É importante salientar que número de horas depende, como nos treinamentos, da área de residência escolhida. A segunda transição, T11, ocorrem quando execução da residência aplicada é concluída. Durante a residência aplicada, assim como a residência médica, o tutor, ao perceber falhas, intervém no desenvolvimento da solução da situação/problema, proposta pelos residentes;

- g) E6, Avaliação Pós-Residência Aplicada, o tutor aplica a mesma avaliação individual, definida no Estado 3, para mensurar os conhecimentos que os residentes apresentam após a realização da residência aplicada, esta, como a avaliação pré-residência pode ser em questionário em papel ou eletrônicos. Na residência médica, a avaliação dos residentes é realizada ao final de cada ano de residência. Neste estado duas transições ocorrem, a primeira, T12, com a aplicação avaliação pós-residência aplicada, e a segunda, T13, quando terminada a aplicação da avaliação.
- h) E7, Avaliação do Ganho de Conhecimento, o tutor, de posse das avaliações pré e pós-residência, mensura o ganho de conhecimento que os residentes obtiveram. A avaliação ocorre por comparações da qualidade das respostas apresentadas pelos residentes, nos dois questionários aplicados, um no Estado 3 e outro no Estado 6. Neste estado duas transições ocorrem, a primeira, T14, com as comparações respostas a respostas, e a segunda, T15, quando terminada a comparações da avaliação do Ganho de Conhecimento.
- i) E8, Análise de Resultados, o tutor e demais envolvidos, empresa e universidade analisam a aplicação da residência como um todo, do primeiro ao penúltimo estado. Os dados coletados proporcionam o encontro dos pontos fortes e pontos de melhorias do Processo de Sistematização de Ambientes de Residência em Software Brasileiros. Este estado retroalimenta o processo.

Finaliza-se aqui, as apresentações das alterações na forma e da descrição do Processo de Sistematização de Ambientes de Residência em Software Brasileiros. Na próxima seção será apresentada a aplicação do mesmo.

### 5.2.1 Aplicação do Segundo Experimento

O segundo experimento do Processo de Sistematização de Ambientes de Residência em Software Brasileiros iniciou quando a última versão foi aplicada na disciplina de Engenharia do Software, no primeiro semestre de 2014, do Programa de Pós-Graduação em Informática da UTFPR, campus Cornélio Procópio. Treze profissionais, de ambos os sexos, com experiência em desenvolvimento de software, neste caso alunos do programa, participaram da referida aplicação. É importante observar que a segunda aplicação, bem como a primeira, aconteceu em ambiente totalmente simulado, nível 1 (vide seção 3.3).

A execução do Estado 0, Definição da Área de Residência, a área de

aplicação da residência foi Gestão de Projeto de Software. A definição da área de aplicação foi definida pelos profissionais imersos no ambiente, aqueles que participaram como residentes. Os residentes afirmaram que, a área definida poderia trazer maior valor agregado junto a mercado produtivo de software, na qual estes se encontram imersos.

A execução do Estado 1, Construção da Base de Conhecimento Essencial, neste estado o tutor instanciou a base de conhecimento para a concepção da residência de software. A base instanciada foi disponibilizada para os residentes por meio do software moodle, este se caracteriza como um ambiente virtual de aprendizagem. Na referida base foi disponibilizado: textos, filmes, exemplos e figuras sobre o arcabouço conceitual da gestão de projeto de software. É possível acessar de forma linear por meio dos endereços disponíveis Anexo B .

A execução do Estado 2, Residência Estrutural, neste estado o tutor treinou os residentes utilizando os objetos de aprendizagem disponibilizados na base de conhecimento essencial. Na residência em gestão de projetos de software o treinamento teve duração de trinta e duas horas, estas foram distribuídas em blocos de quatro horas, com início às 08h e término às 12h, nos dias 25 do mês de janeiro, 08, 22 do mês de fevereiro, 01º, 08, 15 e 22 do mês de março do ano de 2.014. É importante salientar que o número de horas para o treinamento foram definidas pelo tutor, o mesmo observou o exemplo da primeira aplicação e a área de residência escolhida.

A execução do Estado 3, Avaliação Pré-Residência Aplicada, após o término do treinamento os residentes participaram de uma avaliação formativa, esta realizada por meio de um questionário. As questões da avaliação pré-residência aplicada, do ambiente simulado da UTFPR podem ser observadas no Anexo C. O tutor para quantificar o desempenho dos residentes adaptou a escala proposta por Rensis Likert e adotou os seguintes critérios para a correção das questões:

- a) 0, quando não respondida a questão, respostas em branco;
- b) 1, quando se nota a tentativa de respostas, mesmo que errada;
- c) 2, quando a resposta está parcialmente correta; e
- d) 3, quando totalmente correta.

A execução do Estado 4, Definição da Residência Aplicada, neste estado o tutor definiu como a residência em gestão projeto de software seria realizada. É importante salientar que os conceitos sobre residência em software

apresentados no capítulo 3, não foram apresentados formalmente aos residentes na residência estrutural. Fato este que levou o tutor a apresentar o referido conceito por meio do Quadro 20.

#### **Residência em Gestão de Projetos de Software**

Nesta semana iremos trabalhar na prática os conceitos sobre planejamento, execução e controle de um projeto de software.

Antes de iniciar as atividades é de extrema importância que todos leiam o texto que caracteriza a residência em software.

<https://engenhariasoftware.wordpress.com/2008/05/13/um-paralelo-entre-residencia-medica-e-a-residencia-em-software/>

Para realização desta tarefa iremos instanciar o conceito de residência médica e aplicá-lo na engenharia de software.

O processo de residência em software iniciou-se em janeiro, quando trabalhamos ativamente de objetos de aprendizagem ligados a engenharia de software - enfatizando questões inerentes ao planejamento, execução (processo) e o controle.

Todas as informações postadas neste ambiente caracterizam a base de conhecimento essencial que será utilizada durante a execução da residência.

Importante: Todo processo de residência, possui em sua estrutura, a concepção de duas avaliações:

- a primeira tem como objetivo mapear o conhecimento do residente antes da aplicação da residência (60 minutos).

- a segunda tem como objetivo mapear o conhecimento do residente após a aplicação a residência (60 minutos). É importante salientar que estas avaliações são formativas.

#### **Quadro 20 - Apresentação da definição da residência em software**

**Fonte: Autoria Própria**

A execução do Estado 5, Execução da Residência Aplicada, neste estado os residentes após a leitura do apresentado no Quadro 19. Formaram três equipes de trabalho para desenvolverem e gerenciarem o software. Os residentes foram informados que, os detalhes do software a ser construído encontravam-se em um documento de requisitos postado no ambiente virtual de aprendizagem. Em uma analogia a residência médica, os residentes recebem os exames pré-cirúrgicos do paciente, analisam o histórico do mesmo e planejam como será realizada a cirurgia.

Após a formação dos grupos, os residentes tiveram acesso ao segundo conjunto de instrução (vide Quadro 21). Estas instruções apresentavam a definição do ambiente da residência.

**Definição do ambiente de execução da residência.**

Neste ponto iremos configurar o ambiente de execução da residência. Informações inerentes ao ambiente:

- configuração das equipes: 5 integrantes - defina o líder da equipe.
- todos integrantes das equipes devem possuir acesso a ferramenta kanbanize - é importante que todos os integrantes realizem o registro no site - <http://kanbanize.com/>.
- todos integrantes do grupo devem possuir acesso a ferramenta wbstools - para isto acesse o site: <http://www.wbstool.com/> - importante: Não é necessário o registro dos alunos para utilizar esta ferramenta.
- as equipes devem possuir, ao menos uma máquina, com a IDE ou ambiente de programação instalada.

Faça a gestão destas tarefas: mapeie em 1 post-it no kanbanize.  
modelo do post-it

```
+-----+
| configuração do ambiente |
| responsável             |
| início << em minutos>>  |
| término << em minutos>> |
+-----+
```

Tempo de configuração do ambiente: 20 minutos

**Quadro 21 - Definição do ambiente de execução da residência em software**

Fonte: Autoria própria

O tutor informou aos residentes que, após a configuração do ambiente era importante que todos acessem o documento de requisitos do produto software a ser construído, o referido documento foi o mesmo da primeira aplicação (vide Quadro 7, 8, 9, 10 e 11). O tutor informou ainda que, de posse dos requisitos é importante que todos acessem o documento de processo de software (vide Anexo D), pois este nortearia a construção do produto de software.

Os residentes a partir deste ponto, já sabendo como seria a residência em software, receberam as demais atividades e foram informados da necessidade de postá-las no ambiente virtual de aprendizagem. A seguir, as atividades solicitadas que caracterizam o planejamento do projeto serão apresentadas:

- a) mapear a capacidade produtiva do grupo, ou seja, construir a base histórica;
- b) definir a quantidade de pontos por caso de uso necessário para construir o produto; e
- c) construir a Estrutura Analítica Projeto (EAP), com a apresentação do tempo de cada pacote de trabalho.

Terminada a parte do planejamento do projeto, os residentes receberam as atividades de execução, no Quadro 22 a íntegra da solicitação do tutor.



**Atividade de execução do projeto.**

Sua equipe realizou o planejamento do projeto. A EAP apresenta o tempo para a construção de cada pacote de trabalho.

Nesta atividade iremos executar o projeto, ou seja, você implementarão o software.

Durante a atividade de implementação e teste o controle da produção deve ser feito utilizando o *kanbanize*<sup>17</sup>. Lembre-se que cada pacote de trabalho da EAP irá gerar um *post* no *kanban*.

Tempo para realização da atividade: 90 minutos

**Quadro 22 - Atividade de execução do projeto**

Fonte: Autorial Própria

Os residentes após o término da atividade de execução do projeto receberam as últimas instruções. O Quadro 23 apresenta as instruções solicitadas pelo tutor na íntegra.

Pessoal,

Neste momento você executou o processo, gerando o produto de software. Durante a execução você mapeou o tempo gasto para execução de cada pacote.

De posse destas informações gere uma nova EAP, nela deve estar contida o tempo efetivo do projeto.

Poste também uma cópia do quadro *Kanban*, representando a ideia de controle do projeto.

**Quadro 23 - Atividade de gerenciamento e controle do software desenvolvido na residência**

Fonte: Autorial Própria

As atividades desenvolvidas neste estado foram filmadas e estão disponibilizadas no endereço eletrônico<sup>18</sup>.

A execução do Estado 6, Avaliação Pós-Residência Aplicada, neste estado o tutor aplicou a mesma avaliação delineada no estado E3.

A execução do Estado 7, Avaliação do Ganho de Conhecimento. O tutor de posse das avaliações pré e pós-residência aplicada, mensurou o ganho de conhecimento que os residentes obtiveram. Ao analisar as repostas dos residentes, pode-se concluir que, o processo de residência em software trouxe ganhos consideráveis a gestão de projeto de software. Na próxima seção será apresentado um gráfico que mostra detalhes dos ganhos de conhecimento.

A execução do Estado 8, Análise dos Resultados, neste estado o tutor verificou se os resultados, oriundos da residência estrutural e aplicada, contribuíram no sucesso da realização da residência em software.

Na próxima será apresentada a avaliação do mesmo.

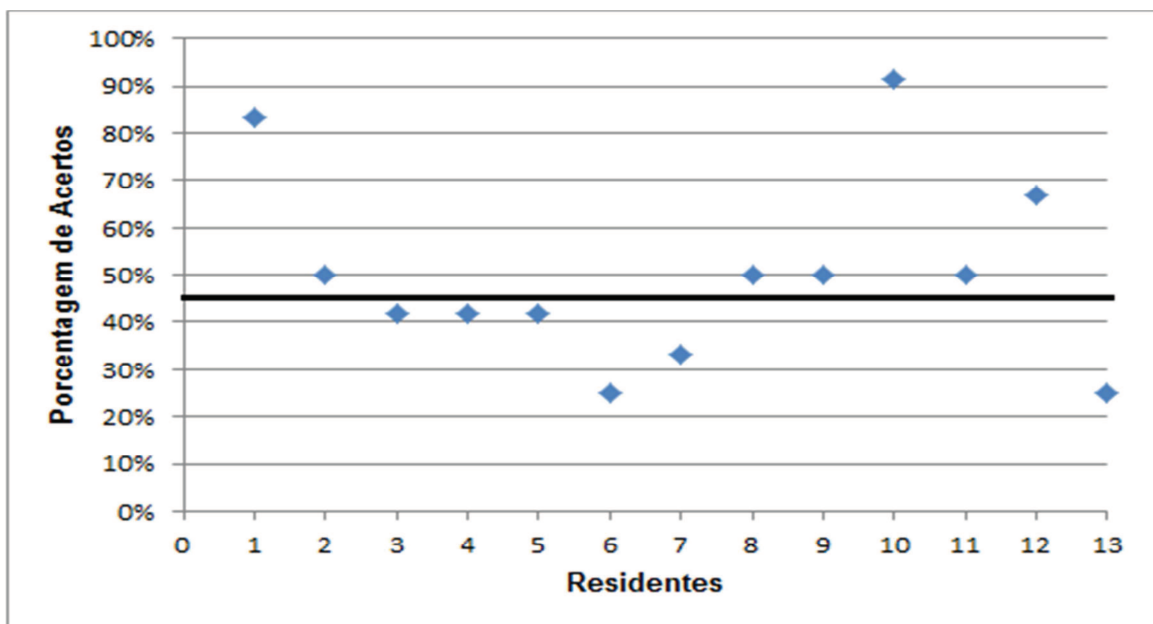
<sup>17</sup> Acessível em <https://kanbanize.com>

<sup>18</sup> Acessível em <http://residenciaemsoftware.wordpress.com/>

### 5.2.2 Avaliação do Segundo Experimento

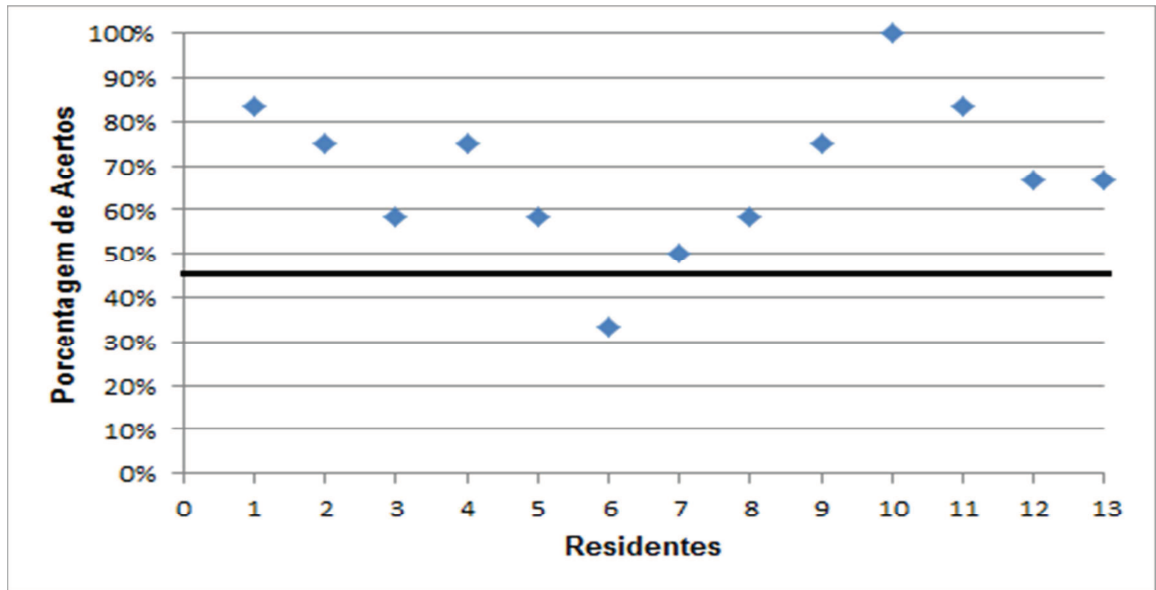
Para a avaliação do segundo experimento, além de mapear as impressões dos residentes e do tutor, buscou-se também, saber se o Processo de Sistematização de Ambientes de Residência em Software Brasileiros promove ganho de conhecimento para os residentes.

Participaram deste segundo experimento treze residentes, todos conseguiram resolver a situação/problema proposta na residência aplicada e, como já citado nos Estado 3 e 6, os mesmos foram avaliados pré e pós-residência. O Gráfico 1, caracterizado como dispersão, mostra os resultados que os residentes alcançaram na avaliação pré-residência, Estado 3. Neste é possível observar que, a média de acertos ficou na casa dos 50%.



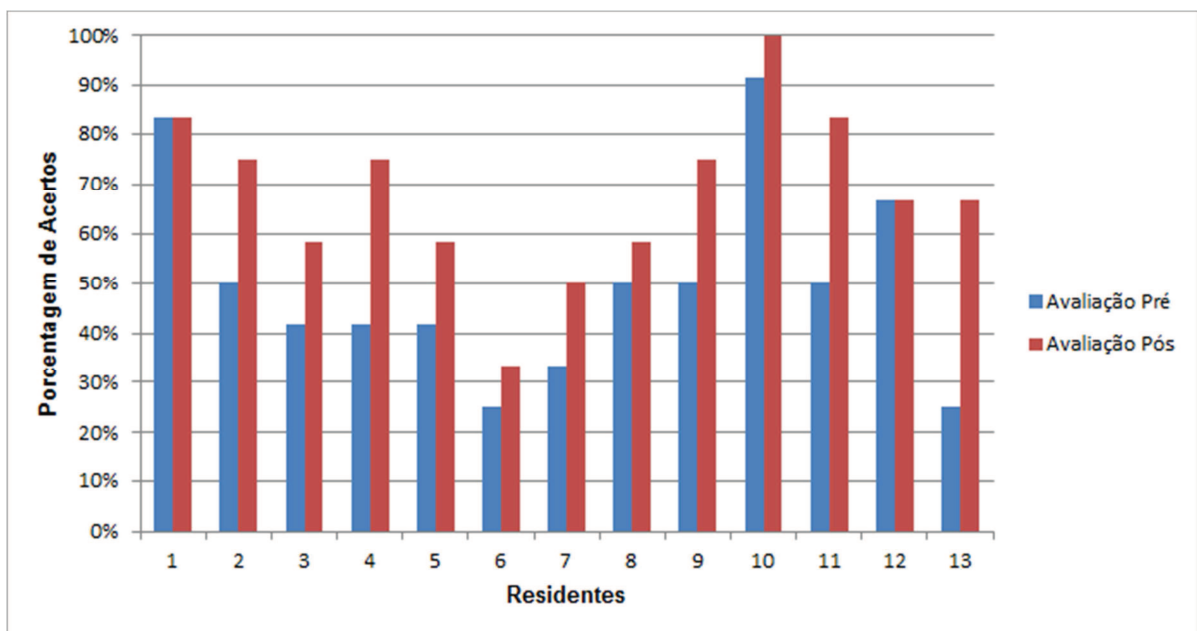
**Gráfico 1 - Resultado da avaliação pré-residência do segundo experimento**  
**Fonte: Autoria Própria**

O Gráfico 2, também caracterizado como dispersão, mostra os resultados que os residentes alcançaram na avaliação pós-residência, Estado 6. Neste é possível observar que, a média de acertos ficou na casa dos 68%.



**Gráfico 2 - Resultado da avaliação pós-residência do segundo experimento**  
 Fonte: Autoria Própria

Ao comparar os resultados apresentados nos Gráficos 1 e 2 nota-se que houve aumento de 36% no número de acertos. No Gráfico 3 é apresentado uma comparação entre a avaliação pré e pós-residência.



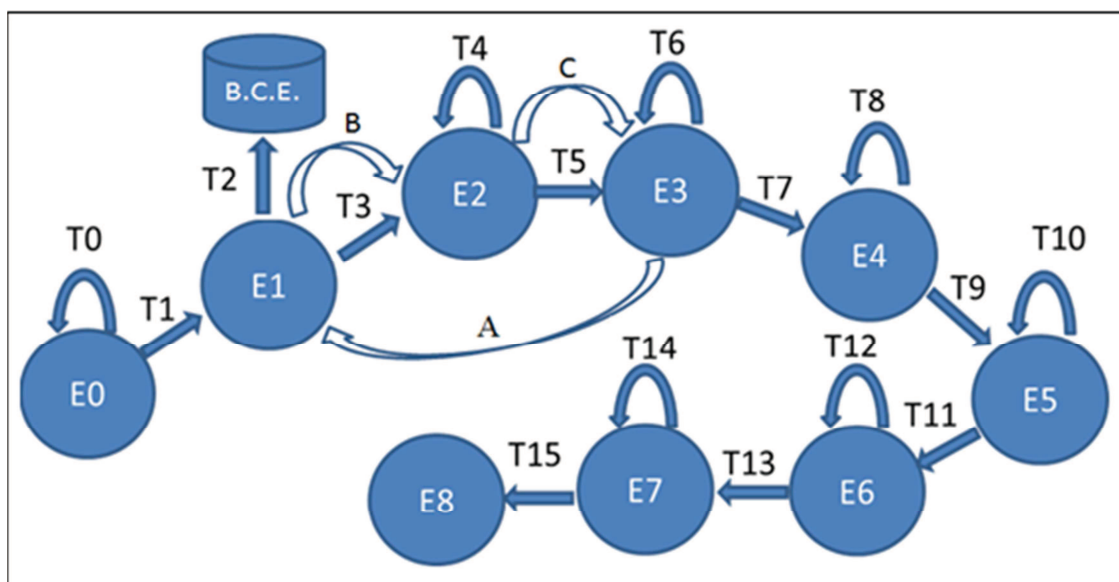
**Gráfico 3 - Resultado das avaliações pré e pós-residência do segundo experimento**  
 Fonte: Autoria Própria

As séries Avaliação Pré e Avaliação Pós indicadas na legenda do Gráfico 3 representam os acertos das avaliações pré e pós-residência respectivamente.

Em suma, a tabulação dos acertos das avaliações mostrou que, para onze residentes o Processo de Sistematização de Ambientes de Residência em Software Brasileiros proporcionou ganho de conhecimento e para dois foi indiferente.

Além das avaliações executadas nos Estados 3 e 6, o autor deste trabalho mapeou as impressões do tutor. Este salientou que, o processo ficou mais operacional e as ações ficaram melhor delineadas proporcionando, assim, otimização aos trabalhos.

Em outra observação, o tutor sugere que o Estado 3, Avaliação Pré-Residência Aplicada, seja executado antes do Estado 1, Construção da Base de Conhecimento. Pois, segundo ele, é importante mapear o grau de conhecimento dos residentes antes que os mesmos sejam treinados, isto, colabora na instanciação da base de conhecimento e na elaboração do plano de treinamento. A Figura 13 ilustra a sugestão, percebe-se que as setas A, B e C indicam a ordem das alterações sugeridas.



**Figura 13 - Sugestão de alteração na ordem de execução dos estados 1, 2 e 3.**  
**Fonte: Autoria Própria.**

### 5.3 DESCRIÇÃO DO TERCEIRO EXPERIMENTO

O terceiro experimento do Processo de Sistematização de Ambientes de Residência em Software Brasileiros foi motivado por dois fatores. O primeiro foi por em prática a sugestão do tutor, quanto a mudança na ordem de execução de dois estados da Máquina, esta sugestão pode ser observada na seção 4.2.2, Avaliação da Segunda Fase do Processo. O segundo foi aplicar o processo em um ambiente de residência de software classificado no nível 2 (vide seção 3.3), pois, as duas primeiras aplicações, como já citado, foram realizadas em ambientes de residência em software classificados no nível 1 (vide seção 3.3).

Para atender as duas motivações de uma só vez, o autor deste trabalho contactou o Grupo de Gestão de Tecnologia em Informática (GTI) da UTFPR, primeiro por saber que, o mesmo tem como objetivo prover direcionamentos na área de melhoria de processo de software e trabalhar ativamente, junto às empresas de produção de software, na implantação de processo de software, auxiliando-as nas avaliações dos modelos CMMI e MPS-BR; e segundo porque essa proximidade com as empresas poderia “fornecer” o ambiente classificado no nível 2 (vide seção 3.3).

O GTI é formado por seis professores da área de engenharia de software da UTFPR, campus Cornélio Procópio. Durante as consultorias e treinamentos em empresas desenvolvedoras de software, o GTI implementa a residência em software para institucionalizar ou melhorar o processo de produção de software de um determinado ambiente.

Os professores do GTI defendem a ideia que a residência em software pode, nos ambientes produtivos de software, promover resultados mais eficientes e eficazes durante a institucionalização e melhoria de um processo.

O grupo de professores do GTI identificou em uma consultoria, uma empresa com um problema crítico de gestão de configurações. Com a execução da residência, o problema foi resolvido e a solução foi implementada com sucesso. Embora tenha autorização para divulgar o problema, a empresa não autorizou a divulgação de seu nome e por esta razão não será identificada. O problema da gestão de configuração é típico e pode ser considerado como recorrente em várias empresas do setor.

A ideia era conduzir aos discentes dois aspectos do problema acima citado: a caracterização da empresa e um produto com as mesmas propriedades

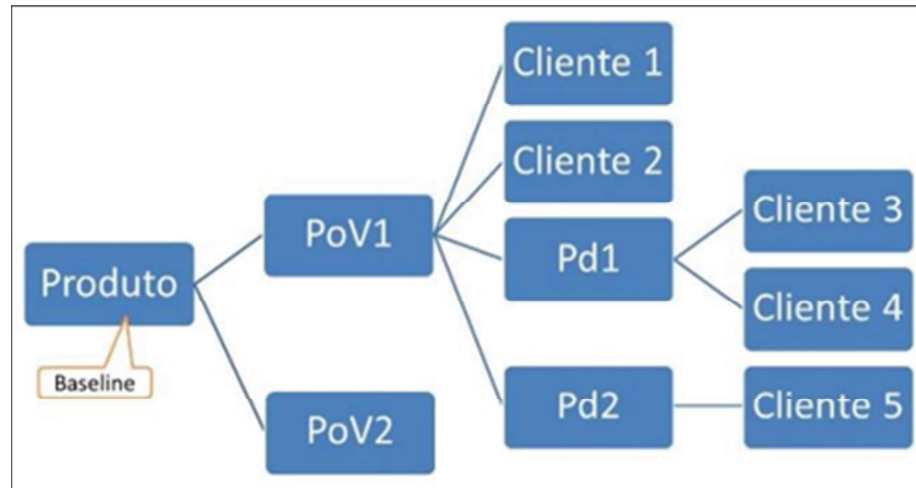
arquitetônicas. O produto original, tratado pelo GTI é de domínio da organização e por este motivo uma versão minimalista, mas com a significância arquitetônica idêntica poderia ser conduzida para a residência. A diferença entre o produto original e o produto conduzido para a residência era a implementação da quantidade de casos de uso. O produto tratado pelos alunos implementaria apenas um único caso de uso que representava um cadastro, enquanto que o produto original implementa cerca de setenta casos. A seguir será apresentada a caracterização da empresa e do produto.

A empresa em questão atua exclusivamente no ramo de desenvolvimento de software, conta com um time de nove desenvolvedores e possui um produto, aqui chamado de P. O produto P é o principal produto da empresa, está implantado em muitos clientes e representa uma parcela significativa do faturamento.

O produto em questão foi desenvolvido por um arquiteto de software que não está mais na organização. Neste caso o conhecimento referente à arquitetura, de domínio da organização, restringiu-se apenas a algumas documentações disponíveis.

Para a empresa, o produto P pode ser comercializado para um cliente em sua forma natural, entretanto há clientes que precisam de customizações. A customização de um cliente pode não atender outro, além disso, novas atualizações (melhoria ou correções de *bugs*) poderão ser disponibilizadas para todos os clientes e novas versões poderão ser lançadas, estas últimas, sobre forma de uma negociação contratual. Na empresa real, a customização para um cliente dura cerca de duas semanas.

A estrutura de distribuições do produto para seus clientes, de acordo com a empresa, está explícita na Figura 14. A partir de um *baseline* é criada uma primeira versão do produto, denominada de PoV1 (Produto Original Versão 1). Este representa o produto de software pronto para distribuição. A distribuição PoV1, no caso da Figura 14, foi entregue para os clientes 1 e 2. No caso dos clientes 3, 4 e 5, foi necessário efetuar uma customização no sistema, sendo que a primeira customização, representada pelo retângulo Pd1 (Produto Distribuição 1), atendeu aos clientes 3 e 4 enquanto que outra customização, Pd2 (Produto Distribuição 2), atendeu ao cliente 5.



**Figura 14 - Estrutura de distribuições abordada pela empresa**  
**Fonte: Autoria Própria**

A Figura 14 representa como a empresa distribui seu produto para seus clientes. Em um cenário real a quantidade de clientes é bem mais significativa e não apenas 5 como indicado na Figura 14. A empresa trata cada implantação em um cliente como uma nova distribuição do produto. Correções de *bugs*, ou melhorias ao produto podem ser implantadas. Neste caso quando existe uma modificação em sua versão original, PoV1 da Figura 14 e todos os nós associados a esta precisam receber tais atualizações. No exemplo da figura, uma modificação em PoV1, significa modificar o produto dos clientes de 1 a 5.

Novas versões do produto poderão ser lançadas. Na Figura 14, representada por PoV2, ou seja, produto original versão 2. Pode haver migração de um cliente que está na versão 1 para esta, mas depende das cláusulas contratuais entre a empresa e seu cliente.

Após apresentação deste cenário o seguinte conjunto de questões seria conduzido para os discentes:

- a) Como novos desenvolvedores iniciam seus trabalhos?
- b) Como novas funcionalidades poderão ser implementadas e implantadas?
- c) Como especificar os processos de gestão de configurações para suprir tal cenário?

Para o autor deste trabalho e para o GTI, esta situação “conjunto de questões que seria conduzida para os discentes” era ideal. Um problema real foi definido e poderia ser desenvolvido em laboratório. Estabeleceu-se então uma “parceira” formada entre o GTI e o autor deste trabalho. Tal “parceria” buscou

determinar se o conhecimento obtido a partir de um processo de residência em software era suficiente para que um grupo de discentes pudesse resolver um problema advindo do mundo real.

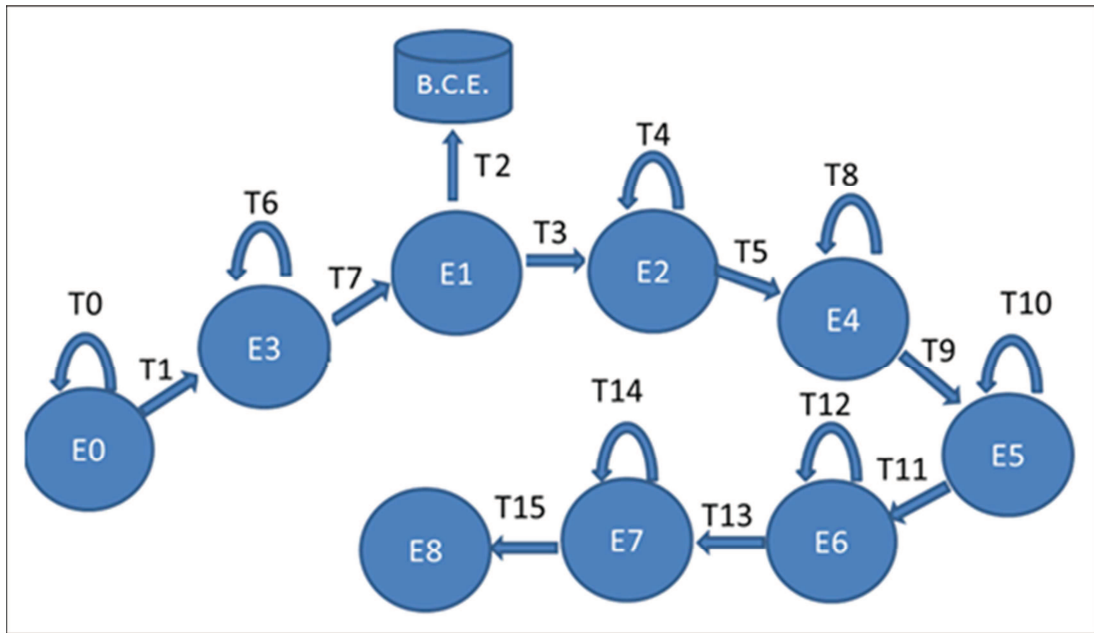
Considerando os fatores motivadores e a “parceria” que oportunizou o ambiente classificado no nível 2 (vide seção 3.2), como desejado, foi possível realizar a terceira aplicação do Processo de Sistematização de Ambientes de Residência em Software Brasileiros. Na próxima seção será apresentada a aplicação do processo.

### 5.3.1 Aplicação do Terceiro Experimento

A aplicação do terceiro experimento do Processo de Sistematização de Ambientes de Residência em Software Brasileiros foi realizada no segundo semestre de 2014, na disciplina de Processo de Software do PPGI da UTFPR. Doze profissionais, de ambos os sexos, com experiência em desenvolvimento de software, neste caso alunos do programa, participaram da referida aplicação. A residência em software criada a partir do Processo estava centrada na linha de produto e gerenciamento de controle de versões.

A aplicação, como esperada, aconteceu em ambiente parcialmente simulado, nível 2 (vide seção 3.3). O problema crítico, já citado na seção 4.3, foi conduzido aos alunos e a alteração na ordem de execução do estado da Máquina foi realizada. A figura 15 ilustra como ficou a ordem de execução dos estados da Máquina, após a alteração.





**Figura 15 - Processo de Sistematização com alterações nos estados 1, 2 e 3**  
**Fonte: Autoria Própria**

A máquina de estado da terceira aplicação do experimento é composta por nove estados, de E0 a E8. A seguir no Quadro 24 os estados são relacionados.

Abreviação	Descrição
E0	Estado 0, Definição da Área de Residência
E3	Estado 3, Avaliação Pré-Residência Aplicada
E1	Estado 1, Construção da Base de Conhecimento Essencial
E2	Estado 2, Residência Estrutural
E4	Estado 4, Definição da Residência Aplicada
E5	Estado 5, Execução da Residência Aplicada
E6	Estado 6, Avaliação Pós-Residência Aplicada
E7	Estado 7, Avaliação do Ganho de Conhecimento
E8	Estado 8, Análise de Resultados

**Quadro 24 - Estados da máquina de estados do terceiro experimento**  
**Fonte: Autoria Própria**

Quinze transições de estados proporcionam a mudança dos estados que formam o Processo, no Quadro 25 estão relacionadas transições.

Abreviação	Descrição
T0	Transição 0, Área da Residência em Definição
T1	Transição 1, Área Definida
T6	Transição 6, Avaliação Pré-Residência em Execução
T7	Transição 7, Avaliação Pré-Residência Executada
T2	Transição 2, Base Essencial em Instanciação
T3	Transição 3, Base Essencial Instanciada
T4	Transição 4, Aplicando a Residência Estrutural (Treinamento)
T5	Transição 5, Treinamento Aplicado
T8	Transição 8, Residência Aplicada em Definição
T9	Transição 9, Definição da Residência Aplicada Concluída
T10	Transição 10, Residência Aplicada em Execução
T11	Transição 11, Residência Aplicada Concluída
T12	Transição 12, Avaliação da Pós-Residência em Execução
T13	Transição 13, Avaliação da Pós-Residência Executada
T14	Transição 14, Comparando os Resultados da Pré e Pós em Execução
T15	Comparação entre os Resultados da Pré e Pós Concluída

**Quadro 25 - Transições da máquina de estados do terceiro experimento**

Fonte: Autoria Própria

A seguir será apresentada a execução dos estados do Processo de Sistematização de Ambientes de Residência em Software Brasileiros.

A execução do Estado 0, Definição da Área de Residência, área de aplicação da residência foi a Linha de Produto e Gerenciamento de Controle de Versões, tal definição foi justificativa uma vez que os membros do grupo de Pesquisa GTI, como já citado, indentificaram tais problemas na empresa.

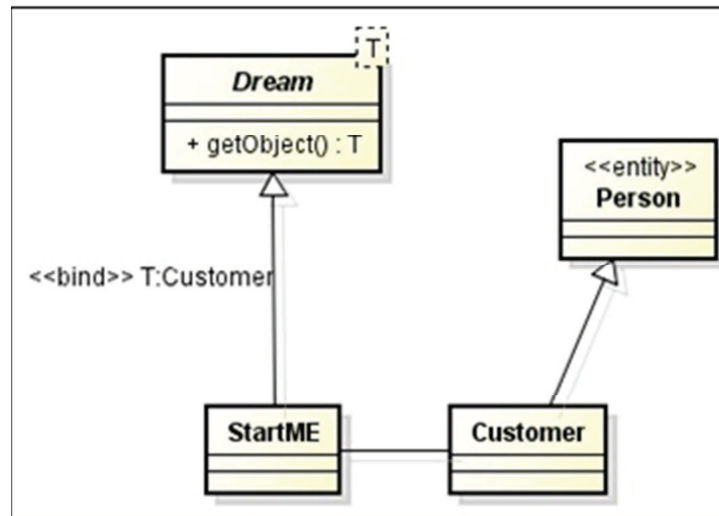
A execução do Estado 3, Avaliação Pré-Residência Aplicada, este como já citado, ocorreu antes da Residência Estrutural. Neste estado os residentes participaram de uma avaliação dissertativa. Para a realização da avaliação, o tutor elaborou vinte e seis questões (vide Quadro 26) que aborda os seguintes temas: *JavaServer Faces*<sup>19</sup>, *Java Persistence API (JPA)*<sup>20</sup>, *Subversion*<sup>21</sup> e *Business Process Model and Notation (BPMN)*<sup>22</sup>. A Figura 16 apresenta um diagrama de classe que os alunos observaram para responder as questões 21, 22 e 23 da avaliação.

<sup>19</sup> Acessível em <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>

<sup>20</sup> Acessível em <http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html>

<sup>21</sup> Acessível em <https://subversion.apache.org/>

<sup>22</sup> Acessível em <http://www.bpmnquickguide.com/viewit.html>



**Figura 16 - Diagrama de classes - parte de avaliação**  
**Fonte: Autoria Própria**

01. Defina controle de configurações.
02. Defina Controle de Versões.
03. Quais procedimentos deverão ser adotados para que o sistema de controle de versões funcione adequadamente em um processo de software?
04. Qual a importância do *Baseline* para o sistema de controle de versões?
05. Em um sistema de controle de versões, diferencie o conceito Build e Release.
06. Em um sistema de controle de versões, defina os conceitos de repositório e diretório de trabalho.
07. Em um sistema de controle de versões defina os conceitos de *check-out*, *add*, *commit*.
08. Em um sistema de controle de versões defina *branch*, *trunk* e *merge*.
09. Explique a arquitetura/funcionamento do JSF2.
10. O que é XHTML?
11. O que é *ManagedBean*? Explique.
12. O que são *UIComponents*? Explique seu funcionamento?
13. Explique todos os componentes desta instrução: `<h:inputText value="#{meuBean.login}" required="true" size="20"/>`.
14. Quais os escopos possíveis do *ManagedBean*? Explique.
15. Defina BPMN.
16. O que é um processo? Quais são seus componentes?
17. Defina e explique: tarefa, subprocesso e transação.
18. Diferencie: *user task* e *manual task*.
19. Explique os eventos de start: *none*, *message*, *timer*.
20. *Pool* e *Lane*, em BPMN estão associados a:
21. O que indica a letra T, sobre a classe *Dream*?
22. A classe *Dream* pode ser instanciada?
23. O que significa *entity* para o diagrama?
24. *StartME* pode ser instanciada dentro de *Customer*?
25. Em *StartME* *getObject* retorna:
26. Qual a importância da especificação de processos?

**Quadro 26 - Questões da avaliação pré-residência aplicada do terceiro experimento**

**Fonte: Autoria Própria**

Para quantificar o desempenho dos residentes, o tutor baseou-se em uma adaptação da escala proposta por Rensis Likert, como na segunda aplicação, e adotou os seguintes critérios para a correção das questões:

- a) 0, quando não respondida a questão, respostas em branco;
- b) 1, quando se nota a tentativa de respostas, mesmo que errada;
- c) 2, quando a resposta está parcialmente correta; e
- d) 3, quando totalmente correta.

A execução do Estado 1, Construção da Base de Conhecimento Essencial. Para a construção base de conhecimento essencial o tutor, neste caso um professor membro do GTI e professor da disciplina de Processo de Software do PPGI da UTFPR, especialista em linha de produto e gerenciamento de versões, realizou a instanciação da referida base para a concepção da residência de software. Na base foram disponibilizadas as video aulas e notações de aulas, “slides” referentes aos conteúdos objetos da residência.

A execução do Estado 2, Residência Estrutural. O tutor treinou os residentes utilizando os objetos de aprendizagem disponibilizados na base de conhecimento essencial. Na residência em questão, o treinamento teve duração de oito horas e por objetivo nivelar os conhecimentos dos alunos residentes, este treinamento foi realizado no dia 09/08/2.014 das 8h às 12h e das 13h as 17h. É importante ressaltar que os participantes, já eram atuantes no mercado e por este motivo o treinamento pode ser otimizado, haja visto que, todos já possuíam conhecimentos na linguagem Java e principalmente em orientação a objetos.

A execução do Estado 4, Definição da Residência Aplicada. O tutor apresentou o cenário, juntamente com o problema de gestão de controle de versão e linha de produto, citados no início deste capítulo. O objetivo foi conduzir os grupos a resolverem os problemas impostos.

A execução do Estado 5, Execução da Residência Aplicada, esta foi realizada no dia 23 de agosto de 2.014, com duração de oito horas das 8h às 12h e das 13h às 17h. É importante observar que a quantidade de horas foi definida pelo tutor. Para a execução da residência aplicada, os residentes formaram três grupos de trabalho para desenvolverem o problema citado no Estado 4. Os grupos deveriam apresentar duas versões do relatório que contém as especificações e explicações do processo que a empresa deve utilizar, isto, antes do término das oito horas. Ao final, entregar a especificação dos processos em *Business Process Model and Notation* (BPMN), juntamente com o diretório de versionamento do *Subversion*. Os grupos tiveram mais três dias para elaborar um relatório sobre tais processos, sem a modificação deste para compor toda a documentação.

A execução do Estado 6, Avaliação Pós-Residência Aplicada. O tutor aplicou a mesma avaliação delineada no estado E3.

A execução do Estado 7, Avaliação do Ganho de Conhecimento. O tutor de posse das avaliações pré e pós-residência aplicada, mensurou o ganho de conhecimento que os residentes obtiveram. Ao analisar as repostas dos residentes pode-se concluir que o processo de residência em software trouxe ganhos consideráveis a linha de produto e sistema de controle de versão.

A execução do Estado 8, Análise dos resultados. O tutor verificou que os resultados da residência estrutural e da residência aplicada são passíveis de serem replicados em outros ambientes, pois, todos os alunos que participaram do processo aprovaram a sistemática de trabalho utilizada durante a realização da residência em software. Estas aprovações foram levantadas por meio de uma entrevista com os residentes.

### 5.3.2 Avaliação do Terceiro Experimento

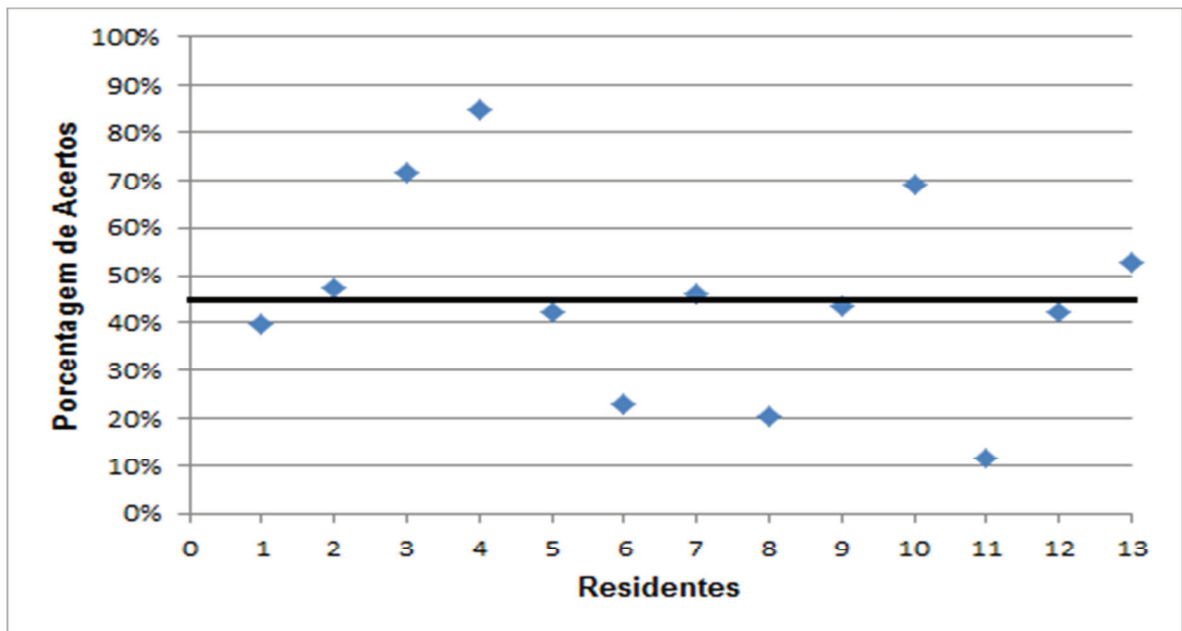
Para a avaliação do terceiro experimento, buscou-se mapear:

- a) as impressões dos residentes e do tutor;
- b) se o conhecimento obtido, a partir de um processo de residência em software, era suficiente para que um grupo de discentes pudesse resolver um problema advindo do mundo real; e
- c) se o Processo de Sistematização de Ambientes de Residência em Software Brasileiros promove ganho de conhecimento para os residentes.

Os três grupos de residentes conseguiram desenvolver com eficiência o problema proposto na residência aplicada, isto sinaliza que, o Processo de Sistematização de Ambientes de Residência de Software Brasileiros pode ser aplicado em ambiente nível 2 (vide seção 3.3).

O tutor, desta terceira aplicação, concorda com as sugestões que o tutor da segunda aplicação fez, quanto à alteração na ordem de execução do Estado 1, 2 e 3 da Máquina de Estado. Segundo tutor desta terceira aplicação mapear os conhecimentos dos residentes contribui na elaboração do plano de treinamento a ser realizado. E ainda expôs que, a orientação dos trabalhos por fluxos de ações traz agilidade na execução da residência.

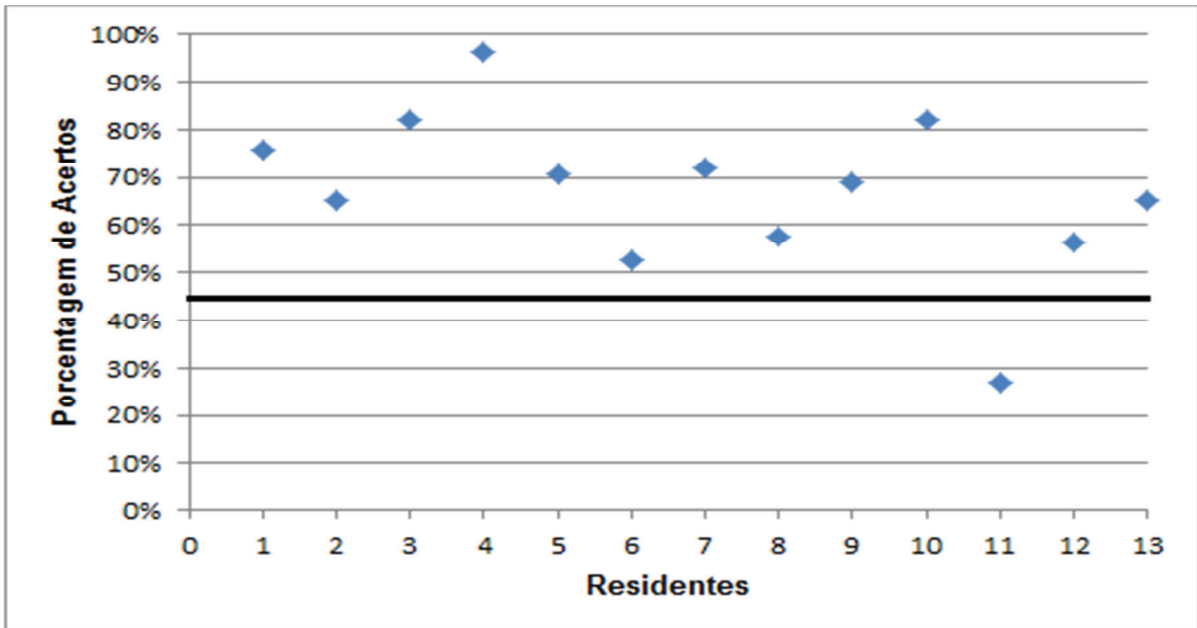
Os residentes, que neste terceiro experimento foram treze como já citado, foram submetidos a duas avaliações, Estado 3 e Estado 6, estas avaliações objetivaram mensurar os conhecimentos que os residentes possuíam pré e pós-residência em gestão de configuração e linha de produto. A diferença entre o número de acertos das avaliações pós e pré indica uma possível evolução no grau de conhecimento alcançado pelos residentes. O gráfico 4, caracterizado como dispersão, mostra os resultados que os residentes alcançaram na avaliação pré-residência. Neste é possível observar que, a média de acertos ficou na casa dos 46%.



**Gráfico 4 - Resultado da avaliação pré-residência do terceiro experimento**

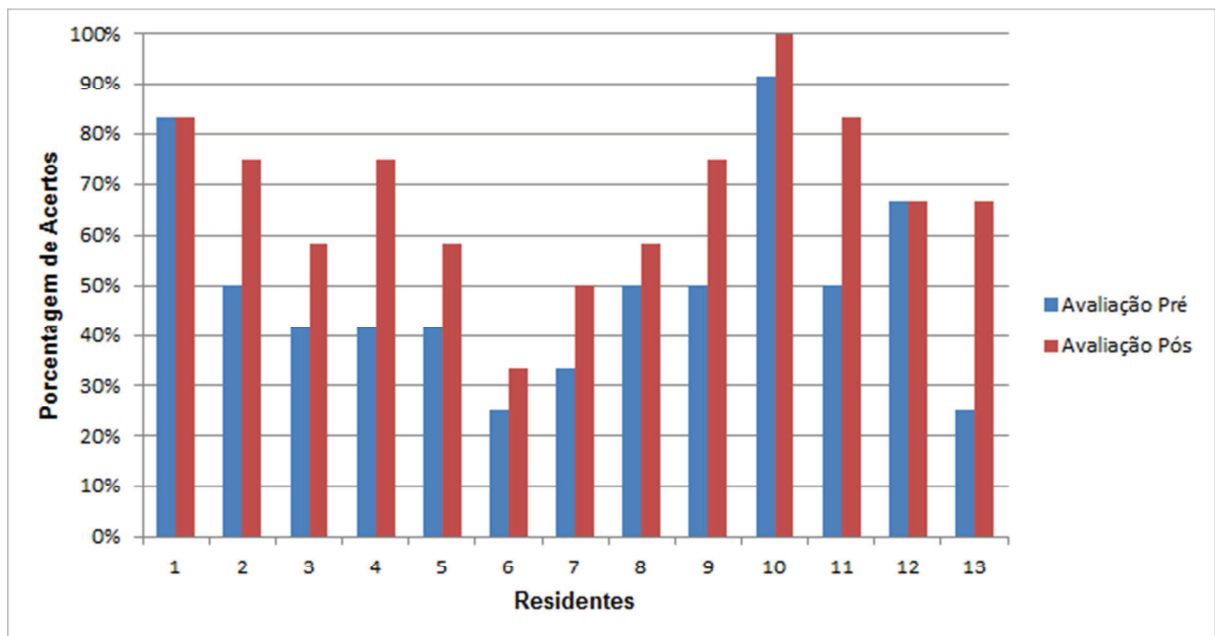
Fonte: Autoria Própria

O gráfico 5, caracterizado como, mostra os resultados que os residentes alcançaram na avaliação pós-residência. Neste é possível observar a média de acertos ficou na casa dos 67%. Ao comparar os resultados apresentados nos Gráficos 4 e 5 pode-se observar que ocorreu um aumento de 46% número de acertos.



**Gráfico 5 - Resultado da avaliação pós-residência do terceiro experimento**  
 Fonte: Aatoria Própria

No gráfico 6 é apresentado uma comparação entre a avaliação pré e pós-residência dos residentes.



**Gráfico 6 - Resultado das avaliações pré e pós-residência do terceiro experimento**  
 Fonte: Aatoria Própria

As séries Avaliação Pré e Avaliação Pós indicadas na legenda do gráfico representa os acertados da avaliação pré e pós-residência respectivamente.

#### 5.4 DESCRIÇÃO DO QUARTO EXPERIMENTO

O Processo de Sistematização de Ambientes de Residência em Software Brasileiros após duas aplicações em ambientes de residência classificados no nível 1 e uma em ambiente classificado no nível 2, foi conduzido para uma aplicação em um ambiente classificado no nível 3.

A condução desta aplicação foi para atender a uma necessidade do Grupo de Tecnologia da Informática (GTI) da UTFPR, campus Cornélio Procópio. O referido grupo foi contratado para realizar uma consultoria em uma empresa de desenvolvimento de software.

A referida empresa nasceu no ano de 2008, atualmente está sediada em uma cidade do Estado do Paraná e está instalada em um prédio de cerca de 500m<sup>2</sup>. O foco de suas atividades é o desenvolvimento de software para segmento de marketing de rede, para isso a empresa conta nove funcionários, além dos dois sócios proprietários.

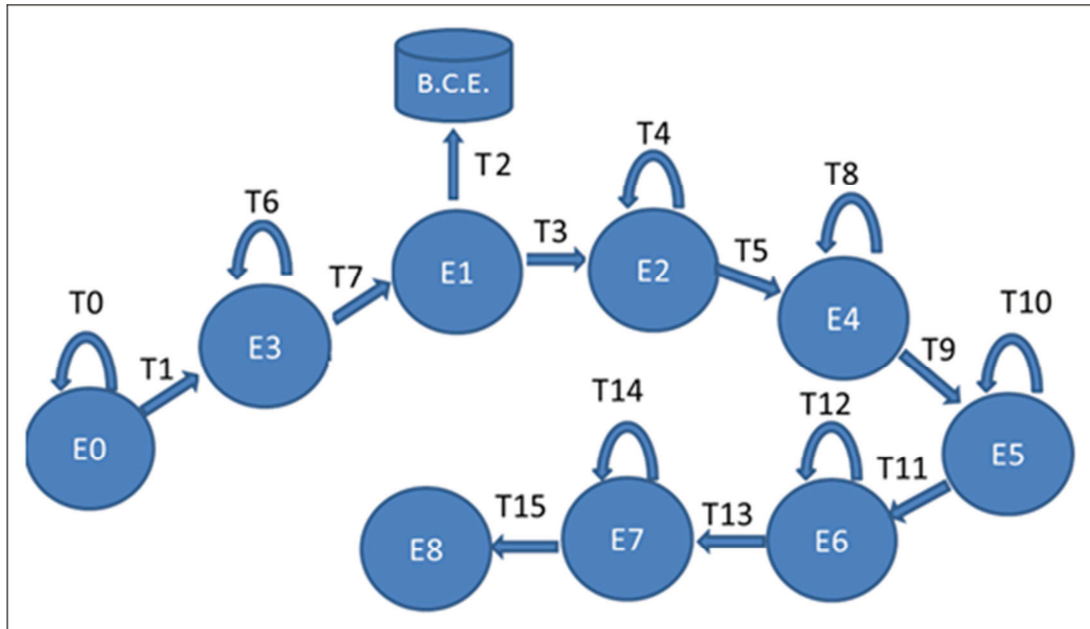
Os diretores da empresa com o propósito de otimizar o processo de software, já institucionalizado na empresa, contrataram os consultores do GTI. O objetivo dos trabalhos era alcançar patamares mais elevados na qualidade do processo e do produto software desenvolvido pela empresa.

O GTI, em sua primeira ação desenvolveu uma pré-consultoria, nela os consultores caracterizam a análise de SWOT (*Strenghts, Weakness, Opportunities, Threats*), (Forças, Fraquezas, Oportunidades e Ameaças). A análise SWOT é considerada, pelo GTI, como um dos melhores métodos de análise de um ambiente empresarial (DAYCHOUW, 2010). Os resultados obtidos pela análise SWOT foram apresentados aos diretores da empresa. As quatro fases da análise de SWOT será apresentada na execução do Estado 3, Avaliação Pré-Residência do Processo de Sistematização de Ambientes de Residência em Software Brasileiros.

De posse dos resultados obtidos com a análise SWOT, o GTI decidiu pela criação de um ambiente de residência em software na empresa, durante a execução da referida criação, o GTI adotou pela primeira vez, o Processo de Sistematização de Ambientes de Residência em Software Brasileiros proposto neste trabalho.



Para a aplicação do Processo de Sistematização de Ambientes de Residência em Software Brasileiros o GTI utilizou a máquina de estado (vide Figura 15) com as sugestões que os tutores delinearam na seção 5.2.2.



**Figura 15 - Processo de Sistematização com alterações nos estado 1, 2 e 3**  
 Fonte: Autoria Própria

Ressalta-se novamente que o Processo de Sistematização de Ambientes de Residência em Software Brasileiros é composto por nove estados, (vide Quadro 24).

Abreviação	Descrição
E0	Estado 0, Definição da Área de Residência
E3	Estado 3, Avaliação Pré-Residência Aplicada
E1	Estado 1, Construção da Base de Conhecimento Essencial
E2	Estado 2, Residência Estrutural
E4	Estado 4, Definição da Residência Aplicada
E5	Estado 5, Execução da Residência Aplicada
E6	Estado 6, Avaliação Pós-Residência Aplicada
E7	Estado 7, Avaliação do Ganho de Conhecimento
E8	Estado 8, Análise de Resultados

**Quadro 24 - Estados da máquina de estados do terceiro experimento**  
 Fonte: Autoria Própria

Além dos Estados, o Processo de Sistematização de Ambientes de Residência em Software Brasileiros é composto quinze transições que proporcionam a mudança dos estados (vide Quadro 25).

Abreviação	Descrição
T0	Transição 0, Área da Residência em Definição
T1	Transição 1, Área Definida
T6	Transição 6, Avaliação Pré-Residência em Execução
T7	Transição 7, Avaliação Pré-Residência Executada
T2	Transição 2, Base Essencial em Instanciação
T3	Transição 3, Base Essencial Instanciada
T4	Transição 4, Aplicando a Residência Estrutural (Treinamento)
T5	Transição 5, Treinamento Aplicado
T8	Transição 8, Residência Aplicada em Definição
T9	Transição 9, Definição da Residência Aplicada Concluída
T10	Transição 10, Residência Aplicada em Execução
T11	Transição 11, Residência Aplicada Concluída
T12	Transição 12, Avaliação da Pós-Residência em Execução
T13	Transição 13, Avaliação da Pós-Residência Executada
T14	Transição 14, Comparando os Resultados da Pré e Pós em Execução
T15	Comparação entre os Resultados da Pré e Pós Concluída

**Quadro 25 – Transições da máquina de estados do terceiro experimento**

Fonte: Autoria Própria

Execução do Estado 0, Definição da Área de Residência: o GTI foi contratado pela empresa para melhorar o processo de produção de software, fato este que delimitou a área da residência.

Execução do Estado 3, Avaliação Pré-Residência Aplicada: esta avaliação ocorreu por meio da análise SWOT, o GTI adotou quatro fases:

1) **Pré-análise**. Nesta fase o GTI solicitou aos diretores da empresa o organograma, o histórico da mesma, em qual segmento do mercado de software a empresa atuava, como era a carteira de clientes e o número de profissionais na área de desenvolvimento de software. De posse das informações, os professores do GTI desenvolveram um plano de trabalho para execução da análise de SWOT.

O plano de trabalho foi estruturado pelo GTI, da seguinte forma:

a) definição do roteiro das entrevistas. O GTI adotou os pressupostos do MPS.BR nível G para estruturar as entrevistas, para o GTI tais pressupostos permitem

conhecer como está a gestão de processos e gestão de requisitos praticadas na empresa, e por ser de interesse da empresa pleitear futuramente a certificação no referido nível. Uma cópia do roteiro desenvolvido pelos pesquisadores pode ser visualizado no Quadro 27.

<b>Levantamentos de informações da empresa</b>
<p>Na entrevista foram levantadas informações pertinentes a: 1) requisitos, 2) planejamento de projeto, 3) controle, 4) gestão de configuração; 5) e garantia da qualidade do processo e do produto.</p> <ol style="list-style-type: none"> <li>1. As Informações sobre <b>Requisitos</b> foram coletas em cinco partes: levantamento, forma de validação, identificação de inconsistência, rastreabilidade e gestão de requisitos;</li> <li>2. As Informações sobre <b>Planejamento de Projeto</b> foram coletas em quatro partes: estimativas, plano de projeto, aceitação dos stakeholders e ciclo de vida;</li> <li>3. As informações sobre <b>Controle</b> foram coletas em cinco partes: dos parâmetros, riscos, marcos e revisões, comprometimento e revisões de progresso;</li> <li>4. As informações sobre <b>Garantia da Qualidade do Processo e do Produto</b> foram coletas em duas partes: avaliação objetiva do processo e avaliação objetiva do produto.</li> </ol>

**Quadro 27 - Roteiro utilizado nos levantamentos de informações da empresa**

Fonte: Autoria própria

b) cronograma de trabalho para a execução da análise. Os pesquisadores do GTI enviaram aos diretores da empresa um cronograma para execução da análise. Uma cópia resumida do referido cronograma pode ser verificada no Quadro 28.

<p>Este artefato tem como objetivo apresentar o cronograma da execução da análise.</p>
<p>1º – Atividades:</p> <ol style="list-style-type: none"> <li>1º - Apresentar os membros do GTI e a configuração dos trabalhos da consultoria(1h)</li> <li>2ª - Entrevistar os diretores e o coordenador de projetos de software (3h)</li> <li>3ª - Entrevistar os analistas de sistemas e programadores (3h)</li> </ol> <p>2ª – Atividade:</p> <ol style="list-style-type: none"> <li>01ª - Conhecer os documentos que o coordenador de projetos de software utiliza para estabelecer relacionamento com os clientes. (4h)</li> </ol>

**Quadro 28 - Cronograma resumido para execução da análise.**

Fonte: Autoria própria

2) **Execução da análise.** Nesta fase, o GTI entrevistou os diretores, o coordenador de projetos de software, os analistas de sistemas e os programadores. O GTI também teve acesso aos documentos que a coordenação de projeto de software utiliza para estabelecer o relacionamento com os clientes, quanto as questões de elaboração de orçamentos, levantamento de requisitos de software, solicitações de customizações e relatos de falhas.

Durante a execução da análise, o GTI de forma empírica observou que: o coordenador de projetos de software, os analistas de sistemas e programadores

têm o conhecimento formal do processo de software desenvolvido na empresa, porém, este não aplicado de forma sistêmica. É importante salientar que as entrevistas foram gravadas com autorização dos entrevistados e que, os documentos acima citados foram armazenados em um repositório de dados

3) **Redação da análise.** Nesta fase, o GTI confeccionou um relatório descrevendo o processo de software atual da empresa e o conhecimento dos residentes sobre o mesmo. A notação utilizada para a descrição do processo foi a *Unified Modeling Language* (UML). É importante salientar que o processo de software era parcialmente documentado, o conhecimento do referido processo utilizado foi classificado como tácito, e os analistas de sistemas e programadores com mais tempo de trabalho na empresa eram os que mais tinham compreensão do processo de software. Além da descrição do processo de software e o conhecimento dos residentes sobre o mesmo, é possível encontrar no relatório: as fortalezas, as fraquezas, as oportunidades e as ameaças que o GTI encontrou.

4) **Apresentação dos resultados.** Nesta fase, o GTI em um reunião de trabalho, entregou e apresentou o relatório da análise de SWOT para os diretores. Uma visão do relatório e da apresentação pode ser observado no Quadro 29.

	Ajuda	Atrapalha
Ambiente Interno	<ul style="list-style-type: none"> <li>▪ Posicionamento no mercado</li> <li>▪ Crescimento.</li> <li>▪ Infraestrutura de produção.</li> <li>▪ Pessoal qualificado (contínua de qualificação).</li> <li>▪ Sinergia de grupo.</li> <li>▪ Gestão de pessoas.</li> <li>▪ Tecnologia de produção.</li> <li>▪ Metas qualitativas claras e definidas.</li> <li>▪ Foco e objetivo bem traçados. (Fortalezas)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Processo em estruturação.</li> <li>▪ Ausência de um processo de fidelização.</li> <li>▪ Ineficiência no processo de comunicação</li> <li>▪ Ineficiência na gestão da produção.</li> <li>▪ Ausência de métricas qualitativas e quantitativas.</li> <li>▪ Estimativas mal elaboradas.</li> <li>▪ Precificação inconsistente.</li> <li>▪ Gestão de conhecimento ineficiente.</li> <li>▪ Processo cascata. (Fraquezas)</li> </ul>
Ambiente Externo	<ul style="list-style-type: none"> <li>▪ Reestruturar o Produto.</li> <li>▪ Estruturar processo de produção.</li> <li>▪ Atingir níveis qualitativos do processo.</li> <li>▪ Melhorar o processo de gestão da produção.</li> <li>▪ Estabelecer uma base histórica de projetos.</li> <li>▪ Organizar de maneira efetiva o tempo de produção.</li> <li>▪ Possibilitar uma maior flexibilidade produtiva dos colaboradores.</li> <li>▪ Atingir mercados externos. (Oportunidades)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Único produto.</li> <li>▪ Concorrência.</li> <li>▪ Rotatividade de pessoal.</li> <li>▪ Legislação. (Ameaças)</li> </ul>

Quadro 29 - Análise de SWOT da empresa

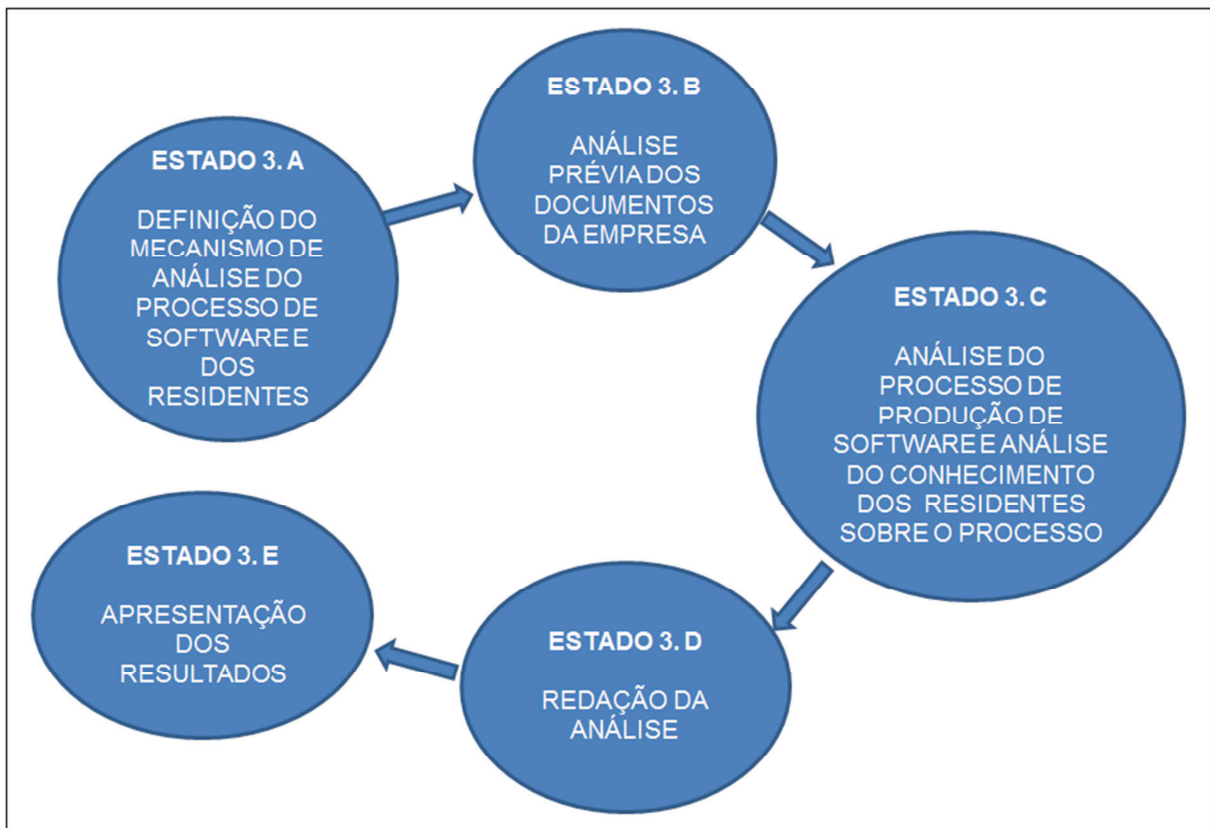
Fonte: Adaptado do GTI – UTFPR, campus Cornélio Procópio

O Quadro 29 é fracionado em quatro partes. As duas partes no nível superior foca o ambiente interno da empresa. As duas inferiores mostra o ambiente externo, as fortalezas e oportunidades estão posicionadas a esquerda enquanto que as fraquezas e ameaças estão posicionadas a direita.

É perceptível que o Estado 3 do Processo de Sistematização de Ambientes de Residência em Software Brasileiros foi fracionado em cinco partes:

- a) 3. A - definição do mecanismo de análise do processo de software e dos residentes;
- b) 3. B - análise prévia dos documentos da empresa;
- c) 3. C - análise do processo de produção de software e análise do conhecimento dos residentes sobre o processo;
- d) 3. D - redação da análise; e
- e) 3. E - apresentação dos resultados.

A Figura 17 seguir ilustra a divisão do Estado 3.



**Figura 17 - Divisão do estado 3 em cinco partes**  
Fonte: Autoria Própria

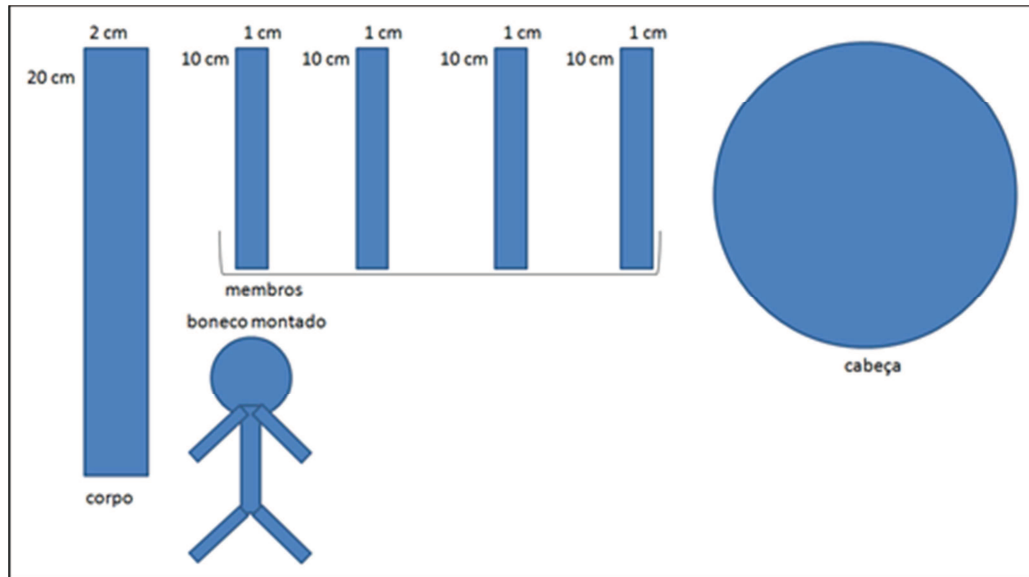
Por fim, o quadrante fraquezas forneceu subsídios para o GTI definir questões inerentes a construção da base de conhecimento essencial e delinear a residência estrutural.

Execução do Estado 1, Construção da Base de Conhecimento Essencial: O GTI definiu os conteúdos necessários para que o treinamento pudesse ser realizado e que um professor, membro do grupo, especialista em processo de software, instanciasse a referida base. Na referida base foram instanciados os conteúdos referentes a teoria de processo. O GTI teve o cuidado de gravar o treinamento e disponibilizar na base de conhecimento. Estas video aulas podem ser acessadas em <https://residenciaemsoftware.wordpress.com>.

Execução do Estado 2, Residência Estrutural, o professor do GTI, após a instanciação da base de conhecimentos, iniciou os treinamentos utilizando os conteúdos da referida base. O treinamento de quarenta horas foi desenvolvido com o objetivo de contextualizar o conceito de processo de software. Para a referida contextualização o GTI utilizou um processo de construção de boneco (FABRI et al. 2015).

A execução da residência estrutural iniciou-se com a caracterização do ambiente. Nesta caracterização os pesquisadores do GTI solicitaram aos diretores da empresa uma sala com cerca de vinte metros quadrados, três mesas (para trabalho em equipe) com quadro cadeiras cada e um quadro branco.

Todos os colaboradores, oito programadores e o coordenador de projetos de software, foram separados em três grupos, cada um com três integrantes. Cada grupo recebeu uma folha de papel sulfite. De posse da folha, o grupo tinha como tarefa construir um boneco (vide especificações do boneco na Figura 18). Perceba que os colaboradores saíram da esfera da construção de um produto caracterizado como software e passaram a construir um boneco.



**Figura 18 - Especificações do boneco**  
**Fonte: FABRI et al. (2015)**

Os consultores do GTI limitaram os recursos para construção dos bonecos, os colaboradores da empresa possuíam somente a folha e a cola – restrições quanto ao uso de réguas, lápis foram impostas pela equipe de consultoria.

O GTI dividiu a construção do boneco em duas atividades:

- a) Recortar: Cabeça, corpo e membros devem ser recortadas – utilizando técnicas de dobradura; e
- b) Montar: União das partes cortadas.

O produto final gerado por uma das equipes é apresentado na Fotografia 1. É importante salientar que durante a construção do boneco, os colaboradores da empresa cronometraram o tempo utilizado em cada atividade. Essa informação é de extrema importância na composição das métricas e da base histórica de projetos.

Após a construção, os colaboradores da empresa constataram os seguintes fatos:

- a) a especificação do produto é concisa e consistente;
- b) sem ferramentas é impossível construir o produto idêntico a especificação. Perceba que a cabeça ficou quadrada (vide Fotografia 1), nas especificações era redonda.
- c) a falta de um número maior de ferramenta prejudicou a construção do produto final.

De posse das constatações, os colaboradores foram questionados: Como melhorar a qualidade do produto final?

A resposta emitida por um dos colaboradores e compartilhada por todos: *“Basta termos alguns recursos adicionais para construir o boneco: caneta, régua, tesoura e compasso. Com esses recursos teremos a presença de mais uma atividade no processo – Desenhar”.*



**Fotografia 1 - Produto final gerado por um dos grupos**  
**Fonte: FABRI et al. (2015)**

De posse das respostas dos colaboradores, os consultores estabeleceram a seguinte analogia: A empresa possui um processo cascata com três atividades distintas – requisitos, produção e teste, percebeu que a empresa poderia aplicar uma gama maior de recursos e ferramentas para melhorar o processo. Dado as devidas proporções, atualmente a empresa constrói boneco somente com o papel e cola, por que não construir boneco com caneta, régua, tesoura e compasso.

Com a analogia, os colaboradores e consultores iniciaram a reestruturação do processo de software da empresa. Este fato culminou no processo espelhado pela Figura 19.

Na Figura 19 é possível perceber que as atividades do processo foram mantidas (requisitos, produção e teste). Essas atividades foram fracionadas em tarefas, por exemplo: a atividade de levantamento de requisitos foi fracionada em duas tarefas: especificação e validação. Para cada atividade foram definidos os artefatos de entrada, os artefatos de saída, as ferramentas e as habilidades necessárias (vide Quadro 30). O processo passou a contemplar também a atividade de gerenciamento de projeto – essa atividade é fracionada em planejamento e controle, a execução é caracterizada pelas atividades e tarefas.



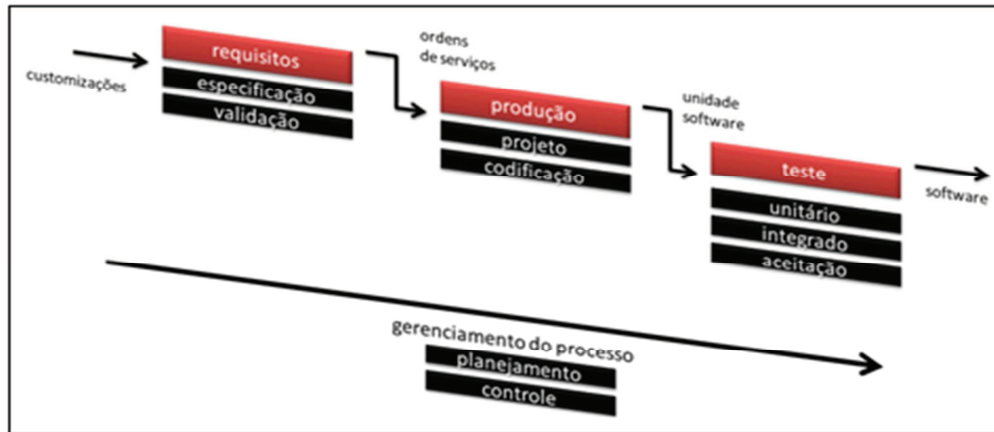


Figura 19 - Processo (reestruturado) de produção de software da empresa  
Fonte: FABRI et al. (2015)

Atividade	Tarefa	Artefato de Entrada	Artefato de Saída	Ferramenta	Habilidades
Levantamento de Requisitos	Especificação	Questionário.	Diretrizes sobre os requisitos.	Formulário WEB.	Prospecção de requisitos. Venda de software.
	Validação	Diretrizes sobre os requisitos.	Diretrizes validadas.	Software para call conference e de trocas de mensagem (síncrona e assíncrona).	Prospecção de requisitos. Venda de software. Analista de sistemas.
Produção	Projeto do software	Diretrizes Validadas	Projeto de software	astah (astah.net).	Analista de sistemas.
	Codificação	Projeto de software.	Código fonte.	Ruby.	Programador.
Teste	Teste unitário	Código fonte. Projeto de Software. Roteiro de teste.	Código fonte testado de forma unitária.	Astah Ruby Ferramenta para automatização do teste	Testador.
	Teste integrado	Código fonte. Projeto de software. Unidade executável integrada. Roteiro de teste.	Código fonte testado de forma integrada.	Astah. Ruby. Ferramenta para automatização do teste.	
	Teste de aceitação	Código fonte. Projeto de software. Unidade de software integrada. Roteiro de teste.	Código testado junto ao usuário	Astah. Ruby. Ferramenta para automatização do teste	Testador. Usuário.

Modelo de Processo: Incremental

Quadro 30 - Atividades, tarefas, artefatos (entrada e saída), ferramentas e habilidades do processo da empresa

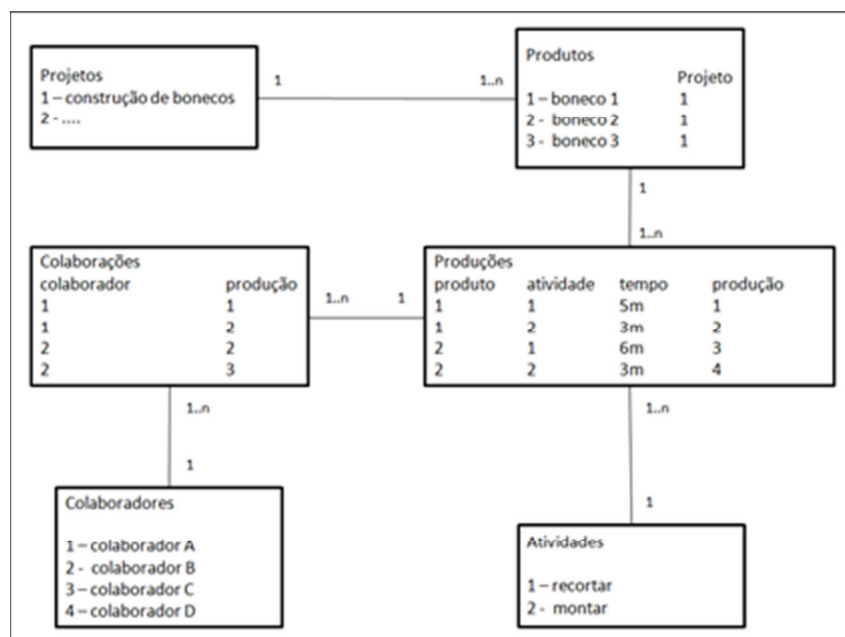
Fonte: FABRI et al. (2015)

Durante a construção do boneco os colaboradores executaram as atividades de recorte e montagem. Além de executar as referidas atividades, foi solicitado aos colaboradores que cronometrassem o tempo utilizado em cada atividade.

De posse destas informações os consultores do GTI mapearam no quadro branco as seguintes informações:

- foi solicitados a vocês um projeto para construção de três produtos (boneco 1, 2 e 3);
- para produzir os produtos vocês percorreram as atividades de recorte e montagem do boneco.
- no grupo 1 participaram da atividade de recorte, o colaborador A e da Montagem o Colaborador A e B.

O referido mapeamento das informações norteia a inferência da estrutura de dados que responde uma questão crucial em qualquer projeto: Quem está fazendo o que para quem em quanto tempo? (vide Figura 20).



**Figura 20 - Estrutura de dados utilizada para organizar informações do processo de construção de bonecos**

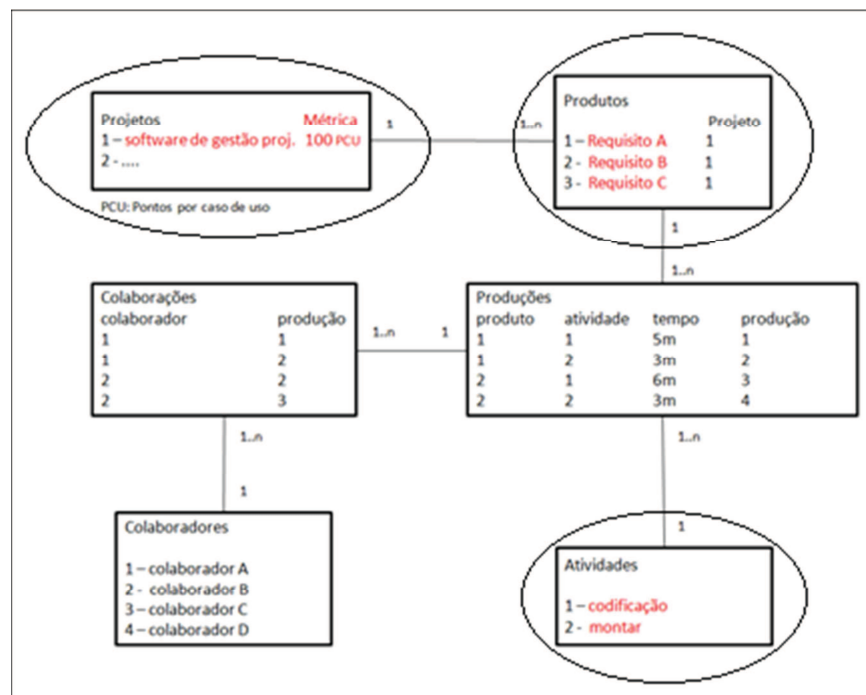
Fonte: FABRI et al. (2015)

Terminada a construção da Figura 20, os consultores do GTI/UTFPR questionaram toda equipe:

Ao analisar a Figura 20 é possível extrair:

- o tempo que a equipe levou para construir todos os bonecos?
- o tempo de execução da atividade de montagem do boneco?
- o tempo investido na atividade de recorte? e
- o colaborador que percorreu mais rapidamente a atividade de montagem?

Todas as questões foram respondidas de forma positiva. De posse das repostas os consultores propuseram pequenas alterações no preenchimento das informações mapeadas na Figura 20, vide Figura 21.



**Figura 21 - Estrutura de dados utilizada para organizar informações do processo de construção de software**

Fonte: FABRI et al. (2015)

Ao analisar a Figura 21 é possível perceber que ao invés de organizar as informações sobre a produção de bonecos, a estrutura de dados passou a organizar as informações sobre a produção de software.

Execução do Estado 4, Definição da Residência Aplicada. O GTI apresentou como seria a residência aplicada aos diretores, ao coordenador de projetos de software, aos analistas de sistemas e aos programadores. Os pesquisadores do GTI decidiram iniciar a execução da residência aplicada por meio do monitoramento do processo de software da empresa. Para estabelecer este monitoramento, o GTI desenvolveu um artefatos para capturar, quem, está fazendo o



Colaborador	Data	Projeto	Tarefa Realizada	Minutos
Fulano 1	27/10/14	Geral	Análise de execução de regras	7
Fulano 1	27/10/14	Geral	Estudo	16
Fulano 1	27/10/14	F. D.	Teste assunto do e-mail renovação de contrato	23
Fulano 1	27/10/14	Geral	Ajustes ao deletar qualificação	77
Fulano 1	27/10/14	Geral	Teste gerais	57
Fulano 1	27/10/14	Geral	Removido tela confirmação no editar dados	10
Fulano 1	27/10/14	Geral	Removido opção de editar id da rede	10
Fulano 1	27/10/14	Geral	Ajuste no retorno dos <i>titles</i> nas legendas das redes	5
Fulano 1	27/10/14	Geral	Ajuste no botão do relatório premiações	13
Fulano 1	27/10/14	Geral	Análise de qual ferramenta será liberada para o admin	28
Fulano 1	27/10/14	Geral	Removido limite de indicações por mês	18
Fulano 1	27/10/14	Geral	Reunião com o gerente financeiro	139

**Quadro 32 - Exemplo de um LOG preenchido**

Fonte: Autoria Própria

Os participantes da residência tinham o compromisso de preencher e postar o LOG semanalmente, durante todo o período da execução da residência. Os LOGs foram postados no ambiente moodle.

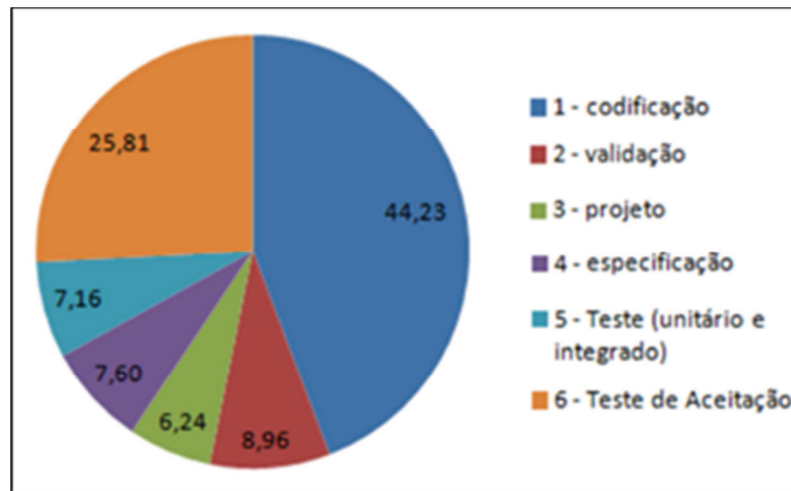
Execução do Estado 6, Avaliação Pós-Residência Aplicada, o GTI apresentou para os diretores da empresa, resultados aferidos a partir das análises dos LOGs preenchidos. Na análise foi mapeada as informações sobre o tempo investido para a execução de cada atividade do processo (vide Figura 19):

- a) especificação;
- b) validação;
- c) projeto;
- d) codificação;
- e) teste unitário ou integrado; e
- f) teste de aceitação

As informações apontadas nos LOGs, durante quatro semanas, pelos participantes da residência, mostram o seguinte cenário quanto a utilização do tempo de produção de software da empresa:

- a) 44,23% do tempo na atividade de codificação de projeto;
- b) 8,96% na atividade de validação;
- c) 6,24% na atividade de projeto;
- d) 7,60% na atividade de especificação;
- e) 7,16% na atividade de teste (unitário e integrado); e
- f) 25,81 do tempo na atividade de teste de aceitação.

O Gráfico 7 a seguir ilustra os dados acima descritos.



**Gráfico 7 – Tempo investido por atividade do processo, quatro semanas**  
**Fonte: Autoria própria**

Estado 7, Avaliação do Ganho de Conhecimento. Neste o GTI mensurou, após a execução da residência aplicada, o ganho de conhecimento que os residentes e a empresa alcançaram. Participaram desta avaliação o GTI, os diretores e coordenador de projeto de software.

Para os residentes, o fato de gerar o Log mostra que os memos obtiveram um conhecimento prático sobre o processo de produção de software e passaram a coloca-lo em prática.

Já para a empresa, o ganho de conhecimento ocorreu quando ela encontrou as suas fragilidades no processo de software utilizado. Em uma análise o Gráfico 7 é perceptível que a empresa passa a maior parte do tempo executando a atividade de codificação 44,23%. Questionado sobre este índice, o coordenador de projetos de software afirma que uma grande fatia deste tempo é devido às inconsistências nas atividades de especificações de projeto, fato este que gera o retrabalho.

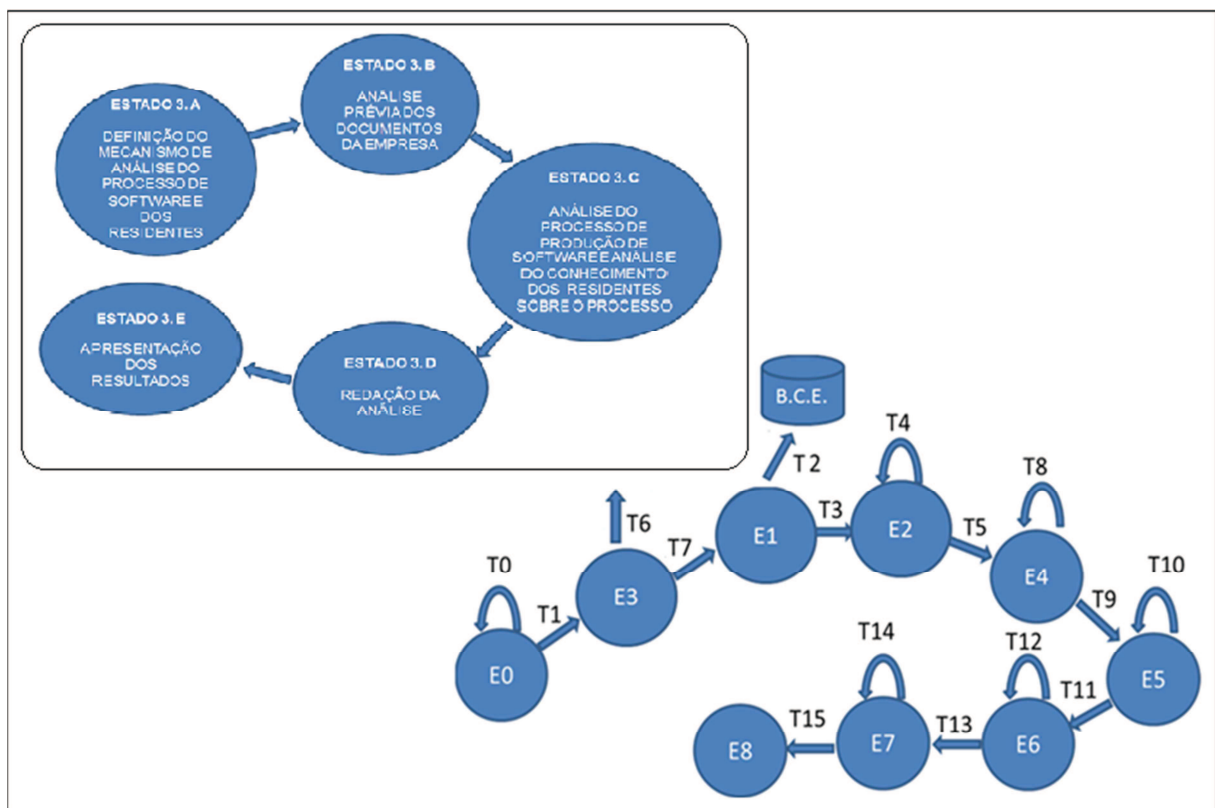
De posse desta informação o GTI junto com os diretores e o coordenador de projeto de software observaram que:

a) a empresa possui sérios problemas na definição de requisitos e especificação de projeto;

b) a atividade de teste de software, alicerçada com o levantamento de requisitos e projetos de software consistente também poderia minimizar o retrabalho apontado pelo coordenador de projeto.

As observações delineadas pelo GTI, diretores e coordenador de projetos de software mapeiam a execução do Estado 8, Análise dos Resultados.

Após a execução dos nove estados do Processo de Sistematização na empresa, leva o autor deste trabalho a inferir uma nova composição do processo (vide Figura 22).



**Figura 22 - Nova composição do Processo de Sistematização**  
 Fonte: Autoria própria

A nova composição do Processo de Sistematização de Ambientes de Residência em Software Brasileiros apresenta os mesmos estados e transições do Processo aplicado no terceiro experimento. Houve somente uma alteração no Estado 3, por se tratar de uma aplicação em um ambiente de residência real.

O Estado 3, Avaliação Pré-Residência foi desdobrado em cinco partes, a primeira é a definição do mecanismo de análise do processo de software, a segunda é a análise de documentos da empresa, a terceira é a análise do processo

de software, a quarta é a redação da análise e a quinta é a apresentação dos resultados.

Em suma, este capítulo apresentou os quatro experimentos que o Processo de Sistematização de Ambientes de Residência em Software Brasileiros percorreu, suas aplicações e as suas avaliações também foram citadas. Os experimentos foram desenvolvidos em ambientes acadêmicos e empresarial, visando sempre a otimização da execução do processo. Considerando os resultados que podem ser observados nas avaliações, é possível constatar que o Processo de Sistematização de Ambientes de Residência em Software Brasileiros atendeu aos seus propósitos. Com a aplicação do processo foi possível sistematizar os ambientes de residência em software de nível 1, de nível 2 e de nível 3 (vide seção 3.3) e promover ganho de conhecimento para a maioria dos residentes.



## 6 ANÁLISE E INTERPRETAÇÃO DOS DADOS

Neste trabalho foram executados quatro experimentos para a validação do Processo de Sistematização de Ambientes de Residência em Software Brasileiros. O Quadro 33 apresenta uma súmula dos mesmos. A primeira coluna do quadro identifica em qual parte deste documento estão às informações que detalham os experimentos. A segunda e terceira colunas apresentam o período e o nível de classificação em que os experimentos foram executados. A quarta coluna apresenta como as informações foram coletadas durante a execução dos experimentos.

Experimento	Período	Nível	Coleta de Dados
1 (seção 5.1)	2013/1	1	Avaliação dos itens da residência estrutural e a forma da aplicação da residência em software pelos residentes (seção 5.1.2) e observação direta.
2 (seção 5.2)	2014/1	1	Avaliações aplicadas aos residentes (Anexo C) e observação direta.
3 (seção 5.3)	2014/2	2	Avaliações aplicadas aos residentes (seção 5.1.3) e observação direta.
4 (seção 5.4)	2014/2	3	Análise SWOT da empresa e observação direta dos LOGs, os registros dos trabalhos dos colaboradores da empresa.

**Quadro 33 - Súmula dos experimentos**

Fonte: Autoria própria

Considerando o Quadro 33 é possível afirmar que o Processo de Sistematização proposto neste trabalho foi experimentado:

- a) Na sistematização de dois ambientes de residência em software nível 1, neste ambiente de residência encontra-se em um laboratório e os projetos de software são simulados;
- b) Na sistematização de um ambiente de residência em software nível 2, neste o ambiente de residência encontra-se em um laboratório e os projetos são oriundos da indústria de software; e
- c) Na sistematização de um ambiente de residência em software nível 3, neste o ambiente de residência encontra-se em uma empresa desenvolvedora de software e os projetos de software são desenvolvidos para atender clientes reais.

Após a realização dos experimentos, os dados foram tabulados. Considerando as observações diretas realizadas e as avaliações (vide Quadro 33) é possível observar que no primeiro experimento:

- a) a base de conhecimento essencial, Estado 1 do Processo de Sistematização forneceu instrumentação suficiente para 91% dos residentes, segundo os mesmos as instruções foram importantes para desenvolvimento dos trabalhos realizados na residência aplicada.
- b) o Orquestrador e o Tutor concordam e sugerem que, o Processo de Sistematização de Ambientes de Residência em Software Brasileiros pode ser conduzido ou aplicado por um único especialista, uma vez que ambos apresentam as mesmas especializações em desenvolvimento de projeto de software, e neste caso podem desempenhar os mesmos papéis.

Já com a tabulação dos dados, observações diretas e avaliações do segundo experimento pode-se observar:

- a) a qualidade das respostas dos residentes evoluiu em 36%, após a aplicação da residência em software criada a partir dos pressupostos do Processo de Sistematização. Os dados da referida evolução podem ser observadas no Gráfico 3.
- b) o tutor sugere que o Estado 3, Avaliação Pré-Residência Aplicada, seja executado antes do Estado 1, Construção da Base de Conhecimento. Pois, segundo ele é importante mapear o grau de conhecimento dos residentes antes que os mesmos sejam treinados, isto colabora na instanciação da base de conhecimento e na elaboração da residência estrutural.

Com a tabulação dos dados, observações diretas e avaliações do terceiro experimento pode-se observar:

- a) a qualidade das respostas dos residentes evoluiu em 46%, após a aplicação da residência em software criada a partir dos pressupostos do Processo de Sistematização. Os dados da referida evolução podem ser observadas no Gráfico 6.
- b) o tutor, desta terceira aplicação, concorda com as sugestões que o tutor da segunda aplicação fez quanto à alteração na ordem de execução do Estado 1, 2 e 3 da Máquina de Estado.
- c) o Processo de Sistematização de Ambientes de Residência em Software pode ser aplicado em ambientes de nível 2 (vídeo seção 3.3), pois todos os residentes

conseguiram desenvolver todas as tarefas proposta na residência em software.

Já com as observações diretas dos LOGs, no caso os registros dos trabalhos dos analistas de sistemas e programadores, foi possível avaliar a aplicabilidade do Processo de Sistematização de Ambientes de Residência em Software Brasileiros em um ambiente de residência real. A aplicação do quarto experimento mostrou que: o Processo de Sistematização contribuiu ativamente na melhoria no processo de produção de software da empresa; e promoveu para os residentes o ganho do conhecimento prático do processo de software, pois, os mesmos passaram a por em praticada os conhecimentos que tinham, mas, não os praticava.

Ao analisar os dados tabulados, a partir das aplicações do Processo de Sistematização de Ambientes de Residência em Software Brasileiros em ambientes parcialmente ou totalmente simulado e real, pode-se observar que:

- a) o Processo de Sistematização proporcionou ganho de conhecimento aos residentes. Os Gráficos 3 e 6 gerados, a partir da tabulação dos dados, mostram resultados significantes, quanto ao aumento dos índices de ganho de conhecimento.
- b) o Processo de Sistematização contribuiu na transferência de conhecimento para os residentes e colaboradores de uma empresa desenvolvedora de software.
- c) o Processo pode ser aplicado nos ambientes de nível 1, nível 2 e nível 3.

Desta maneira, os quatro experimentos contribuíram sobre diferentes aspectos quanto à consolidação do processo de residência. O Quadro 34 apresenta de maneira resumida as contribuições dos quatro experimentos.

Experimento	Contribuições
Experimento 1	Importância da base de conhecimento. Possível unificação do tutor com o orquestrador. Relevância do processo orientado a fluxo
Experimento 2	Avaliação pré e pós-residência como mecanismo de mensurar o ganho de conhecimento. A avaliação pré-residência é um mecanismo que viabiliza a construção da base de conhecimento.
Experimento 3	Composição do processo em uma máquina de estado, versão de teste do Processo de Residência.
Experimento 4	O processo de sistematização de residência é aderente com organizações de software. Os dados coletados constataram a eficácia do processo uma vez que após a residência a organização obteve ganho no que tange a elaboração de orçamentos, previsão de prazos e captação de novos clientes.

**Quadro 34 - Contribuições para a consolidação do processo de residência**

Fonte: Autoria Própria

## 7 APRESENTAÇÃO DAS CONCLUSÕES

Atualmente poucas empresas brasileiras produzem software, ou prestam serviços, com padrões de qualidade reconhecidos por algum modelo. Esta afirmação pode ser constatada ao ser analisada a quantidade de empresas certificadas no Modelo Referencial baseado em Maturidade, Capacidade e Integração (CMMI), CMMI-DEV e CMMI-SVC, o Brasil possui 221 empresas certificadas. Para efeitos comparativos, a China possui 3.316, os Estados Unidos da América 2.186 e a Índia 959<sup>23</sup>.

Este fato, aliado à alta carga tributária e a deficiência na formação da mão de obra na área de tecnologia da informação, mais precisamente na área de engenharia de software, caracteriza-se como fonte inibidora no processo de expansão externa do Brasil neste setor.

Com o objetivo de contribuir minimamente para que o número de empresas certificadas reconhecidas por algum modelo referencial baseado em Maturidade, Capacidade e Integração aumente, algumas universidades e empresas criaram o conceito de Residência em Software. Neste tipo de residência, ambientes são preparados para que os residentes participem no intuito de se especializarem na área de engenharia de software.

Considerando a necessidade de preparação de ambientes para a execução de residência em software, o objetivo deste trabalho foi à construção de processo para sistematização de ambientes de residência. Este, voltado para as organizações do setor de software no mercado nacional. O referido objetivo foi alcançado, alicerçado pelos embasamentos teóricos e pela validação da hipótese.

A hipótese apresentada no capítulo 4 afirma que: **se existem ambientes que implementam o conceito de residência em software então é possível estabelecer um processo para sistematização destes ambientes**

Para analisar a hipótese foram definidas duas variáveis dependentes:

- a) o ganho de conhecimento dos participantes da residência em software; e
- b) a aplicação do processo em ambientes de residência em software com diferentes níveis de classificação.

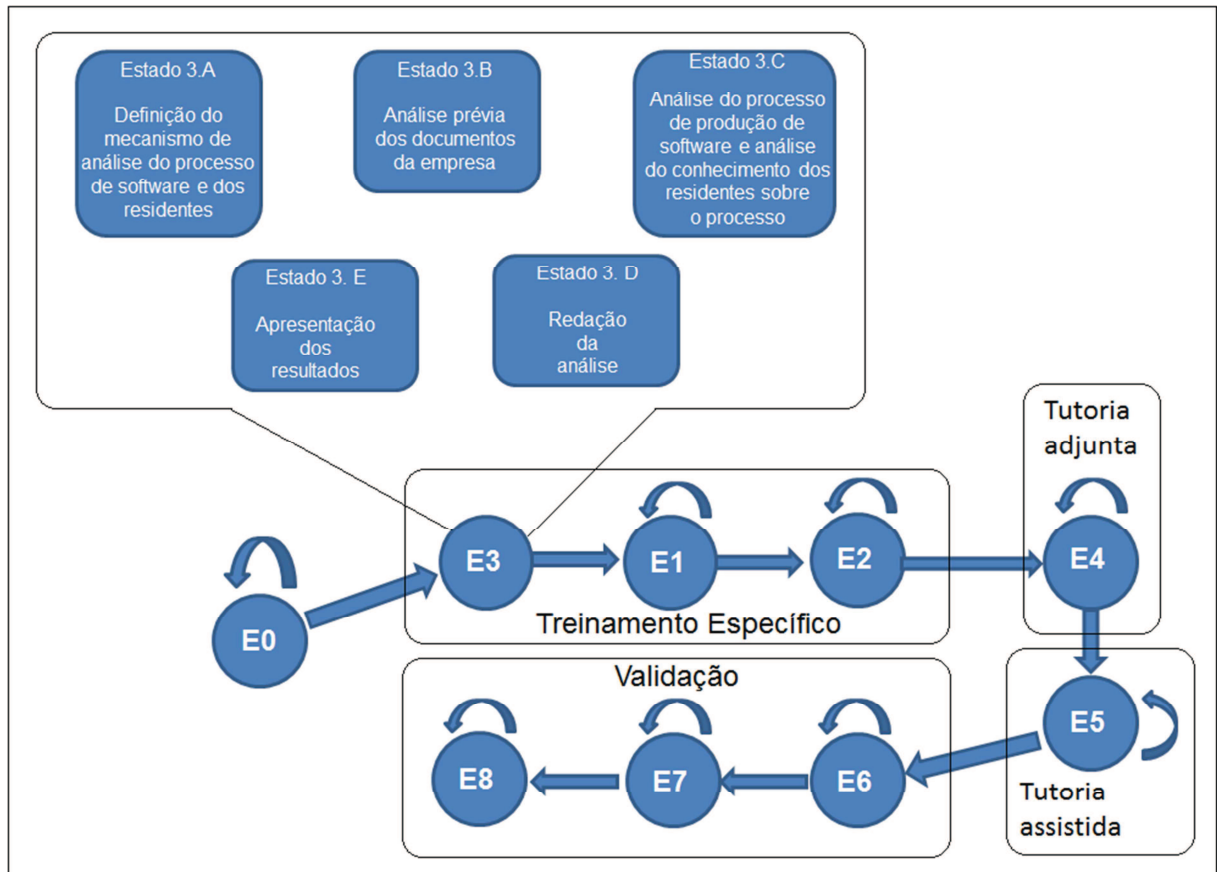
<sup>23</sup> Acessível em <http://cmminstitute.com/wp-content/uploads/2014/05/Maturity-Profile-Ending-March-2014.pdf>

A métrica utilizada para compor o conteúdo da primeira variável (a) foi determinada a partir dos resultados dos residentes em tratar problemas reais, após a residência em software. Nos três primeiros experimentos foi constatado um ganho de conhecimento com a aplicação de uma avaliação pré-residência e outra pós-residência. Esta escala pode ser mensurada em percentagem (vide Gráfico 3 e Gráfico 6). No quarto experimento, em um ambiente real, a avaliação pré-residência foi configurada como uma análise SWOT. Tal análise determinou os limites dos colaboradores relativos ao conhecimento sobre o ambiente em que atuam, identificando problemas relativos ao nicho de mercado em que a organização atua.

No quarto experimento foi possível identificar que, a organização conseguiu mapear questões inerentes a orçamento, a prazo e a captação de clientes, questões estas estruturadas dentro da residência aplicada. Tal observação pode constatar a eficácia do processo, uma vez que houve, nestas dificuldades pontuais, por parte dos colaboradores um ganho significativo em lidar com problemas advindos do cenário real de mercado em que a organização está inserida.

A variável (b) tem seu resultado definido como verdadeiro. Este resultado é alicerçado pelos três primeiros experimentos, dois deles realizados em ambientes classificados no nível 1 (vide seção 3.3) e um no nível 2. O experimento realizado no nível 3 também colabora com o referido resultado. Os experimentos mostram que o Processo de Sistematização evoluiu em relação a composição e organização dos seus estados (vide Figura 11, Figura 12, Figura 15, Figura 22 e Figura 23). A Figura 23 apresenta a última versão do Processo de Sistematização de Ambientes de Residência em Software mapeado neste trabalho.

Dados os valores das variáveis (a) e (b) é possível afirmar que a hipótese apresentada neste trabalho é verdadeira, portanto, **se existem ambientes que implementam o conceito de residência em software então é possível estabelecer um processo para sistematização destes ambientes.** A última versão deste Processo de Sistematização é contemplada pela Figura 23.



**Figura 23 - Última versão do Processo de Sistematização**  
**Fonte: Autoria Própria**

A Figura 23 apresenta a última versão do Processo de Sistematização de Ambientes de Residência em Software, após a realização do experimento em uma empresa do setor produtivo de software. Nesta, houve uma reordenação dos estados, indicando que a avaliação pré-residência é um elemento essencial. Além disso, tal elemento E3 tem uma especialização, representada por um ciclo de vida (Estado E3A, E3B, E3C, E3D e E3E). Este ciclo de vida é contemplado pela definição do mecanismo de análise do processo de software até a apresentação do resultado. Lembrando que estas etapas antecedem a residência estrutural. A residência estrutural é denominada na Figura 23 como treinamento específico. As tutorias são indicadas pelos estados E4 e E5. No estado E4, o tutor executa atividades juntamente com os residentes, caracterizando uma atividade colaborativa, enquanto que no estado E5, o tutor observa a conduta dos residentes, como uma forma de avaliação.

Os estados E6, E7 e E8 representam a validação da residência. Neste os residentes são avaliados em um cenário final que, no caso de acadêmico trata-se

de aplicação de questões práticas e teóricas, enquanto que em um ambiente real tal avaliação é aferida pelos resultados da organização.

A residência em software atende áreas específicas dentro do domínio da engenharia de software. Os experimentos realizados almejavam diferentes áreas de residência do processo como um todo, conforme Quadro 35. Embora os experimentos tenham constatado que o processo é válido, não há evidências que os resultados de uma residência sejam idênticos à de outra em uma mesma área e organizações diferentes. Tal fato ocorre, pois, cada organização tem seu processo definido norteado pelo nicho de mercado que atende e por esta razão, variáveis como quantidade de clientes, perfil de clientes, perfil dos colaboradores, podem intervir no conteúdo da base de conhecimento, tornando-a replicável entre residências, mas com certa restrição.

Experimento	Área de residência
Experimento 1	Residência em engenharia de software
Experimento 2	Residência em gestão de projetos de software
Experimento 3	Residência em controle de versão e linha de produto
Experimento 4	Residência em requisitos (escopo, custo e prazo)

**Quadro 35 - Áreas de residência em software**  
**Fonte: Autoria Própria**

O papel central, delineado pelo processo de residência, foi o tutor. O mecanismo de tutoria, orientado pela base de conhecimento, juntamente com o conhecimento tácito do tutor tornaram as experiências das residências viáveis, com uma fluidez natural. A viabilidade está no fato do tutor conduzir seus treinamentos utilizando um conjunto de ferramentas tecnológicas (o moodle, por exemplo), consideradas pelos residentes de fácil uso. A fluidez natural desta está no fato do tutor intervir junto aos residentes sem causar impacto no trabalho natural que estes precisam exercer na organização. Tal afirmativa pode ser constatada no experimento.

Não foram executados experimentos em que a tutoria tenha sido executada a distância. Este cenário pode ocorrer quando a organização apresenta quadros de *offshore* e/ou *homeoffice* (escritório em casa), por exemplo. Atividades relacionadas a distribuição da produção citados por CARMEL (2009), tornam comum o desenvolvimento de um mesmo produto de software por equipes dispersas

(equipes compostas por um ou mais colaboradores). Neste sentido, como trabalho futuro deste, pretende-se permear os conceitos de residência de software em organizações que desenvolvem seus produtos de maneira distribuída.

Os residentes podem tornar-se tutores em um mesmo ambiente organizacional, dado a absorção de conhecimento e a capacidade de resolver impasses oriundos do próprio nicho de atuação, fato este não observado neste trabalho.

Na residência médica, o intuito principal do residente é manter-se em contato com os pacientes. Desta mesma maneira é extremamente importante que o residente em software mantenha contato constante com seu cliente (seja interno ou externo a organização) não descartando a necessidade de uma boa relação com ele. Somente desta maneira, o residente em software pode, após o processo de residência conhecer melhor seu cenário e agir proativamente de maneira a maximizar a eficácia de seu trabalho.

Além do trabalho futuro já citado nos parágrafos acima, outros podem contribuir para a consolidação do processo de sistematização de ambientes de residência em software brasileiros: a) Investigar se a análise de SWOT é forma mais indicada para analisar uma empresa desenvolvedora de software; e b) qual o melhor método de avaliação do ganho de conhecimento dos residentes no processo de residência em software.



## REFERÊNCIAS

AMARO, A. et al. Relatório **A arte de fazer questionário** disponível em: [http://www.unisc.br/portal/upload/com\\_arquivo/a\\_arte\\_de\\_fazer\\_questionario.pdf](http://www.unisc.br/portal/upload/com_arquivo/a_arte_de_fazer_questionario.pdf). Acesso em: 18 de fevereiro de 2015.

BATISTA, Fabio Ferreira. **Modelo de gestão do conhecimento para a administração pública brasileira**: como implementar a gestão do conhecimento para produzir resultados em benefício do cidadão. Brasília: IPEA, 2012.

BEGOSSO, L. R. et al. Programa de residência em software. In: Workshop sobre Educação em Computação da Sociedade Brasileira de Computação, 19., 2011, Natal. **Anais...** Natal: CSBC, 2011 p. 1666-1669.

CARMEL, E. et al. Follow the sun software development: new perspectives, conceptual foundation, and exploratory field study. In International Conference on System Sciences 42., **Proceedings...**Hawaii, 2009.

COHEN, M. D. et al. **Routines and other recurring action patterns of organizations**: contemporary research issues. Disponível em <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.138.7555&rep=rep1&type=pdf>. Acesso em 08 jan. 2.014

DAVENPORT, T. et al. **Successful knowledge management projects**. Disponível em <http://cesumar.br/mestrado-em-gestao-do-conhecimento/arquivos/successful-knowledge-kanagement-projects.pdf>. Acesso em 20 out. 2013

DAYCHOUW, Merhi. **40 ferramentas e técnicas de gerenciamento**. 3ed. Brasport, 2010.

FABRI, J. A. et al. Implementation of software residency at a graduation course. In: 40th Frontiers in Education Conference, **Proceedings...** Washington, DC 2010. p. F1H1-F1H6.

\_\_\_\_\_. **Knowledge management and concept maps applied to software process improvement**. Disponível em <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5974286>  
Acesso em: 08 jan. 2014

\_\_\_\_\_. Implantando melhorias no processo de software de forma lúdica. **Revista Engenharia de Software**. No prelo, 2015

FABRI, J. A. et al.. Residência em software: um caso real e uma proposta de genérica para a normatização de novos programas. Disponível em <https://engenhariasoftware.files.wordpress.com/2008/05/fabriiisic.pdf>  
Acesso em: 15 out. 2013

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 4ed. São Paulo: Atlas, 2002.

GRANT R. M.. **The knowledge-based view of the firm**. In The Strategic Management of Intellectual Capital and Organizational Knowledge. C. W. Choo, and N. Bontis, Eds. New York: Oxford University Press, 2002. p. 133-148.

HALDIN-HERRGARD T.. Difficulties in diffusing of tacit knowledge in organization. **Journal of Intellectual Capital**. v. 1, n. 4, 2000, p. 357-363

HEISIG, P.. Harmonisation of knowledge management: comparing 160 KM frameworks around the globe. **Journal of knowledge Management**, v. 13, n. 4, 2009, p. 4-31

HERMANN, W., BOVO, V.. **Mapas mentais enriquecendo inteligências**: manual de aprendizagem e desenvolvimento de inteligências: captação, seleção, organização, síntese, criação e gerenciamento de conhecimento. 2ed., Campinas: Walther Herman, 2005

International Standard 15504, Information Technology - Process Assessment, Reference No. ISO/IEC 15504, 2004

KOGUT, B.. Country capabilities and the permeability of borders. **Strategic Management Journal**, v. 12, 1991. p. 33-47.

LYNN, G. S.; AKGÜK, A. E.. A new product development learning model: antecedents and consequences of declarative and procedural knowledge. In **Journal Technology Management**, v. 20, n. 5, 6, 7 e 8, 2000. p. 490-510.

MA-MPS.BR - Melhoria de Processo do Software Brasileiro. Guia Geral. Versão 1.1, 2006.

NONAKA, I.; TAKEUCHI, H., 1995, **The knowledge-creating company**: how japanese companies create the dynamics of innovation. NY: Oxford University Press, 1995

MELLO, Marco A. R.. **Operacionalizando uma hipótese**. 2012. Disponível em: <<http://marcoarmello.wordpress.com/2012/03/13/operacionalizando-uma-hipotese/>>. Acesso em: 11 de janeiro de 2015.

POLANYI. **Personal knowledge**: towards a post-critical philosophy. The University of Chicago Press: Chicago, IL, 1966.

RAMASUBRAMANIAN, S.; JAGADEESAN, G.. **Knowledge management at infosys**. In IEEE Software, USA, may/june 2002.

RUS, I.; LINDVALL, M.. **knowledge management in software engineering**. In IEEE Software, May-June 2002.

SAMPAIO, A. et al.. **Software test program a software residency experience**. Disponível em <http://imprensa.cesar.org.br/artigo.pdf>. Acesso em: 20 out. 2013

SAMPAIO, C. A. S.; LIMA, J. M.. Residência em software. **Revista PROQUALIT**: qualidade na produção de software. Editora UFLA. v. 2, n. 1. Maio de 2006.

SCAMPI - Standard CMMI Appraisal Method for Process Improvement. Software. 2006.

SILVEIRA DUARTE, A. et al. **Proposal of a model to classify software residency environments**. Disponível em <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6615725>. Acesso em: 22 out. 2013

SOMMERVILLE, Ian. **Engenharia de software**. 9ed. São Paulo: Pearson Education, 2011

STEWART T. A., **The wealth of knowledge**: intellectual capital and the twenty-first century organization. Random house Inc.: New York, 2001

TAVARES, Edval S.. **Uma contribuição para os processos de gerência de projetos através da gerência do conhecimento**. Tese de doutorado apresentada ao Departamento de Engenharia de Produção da Escola Politécnica da USP. São Paulo. 2004.

TERRA, José C. C.. **Gestão do conhecimento**: aspectos conceituais e estudo exploratório sobre as práticas de empresas brasileiras. Tese de doutorado apresentada ao Departamento de Engenharia de Produção da Escola Politécnica da USP. São Paulo. 1999.

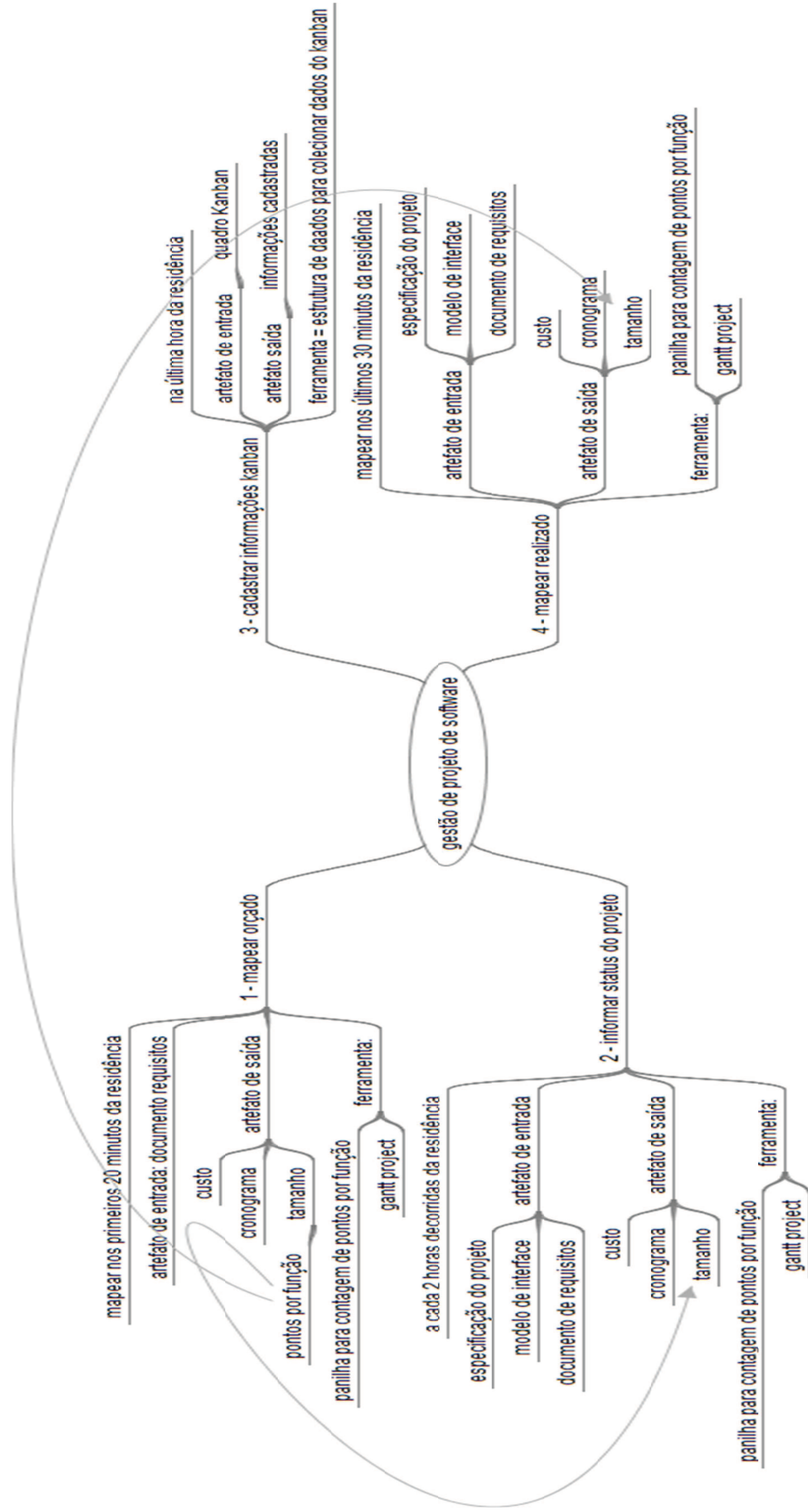
TRINDADE, A. L. P.. **Uma contribuição para o entendimento da ensinagem na preservação do conhecimento em ambientes de fábrica de software**. Tese de doutorado apresentada ao Departamento de Engenharia de Produção da Escola Politécnica da USP. São Paulo 2006.

TSOUKAS H.. **Do we really understand tacit knowledge?** In The Blackwell Handbook of Organizational Learning and Knowledge Management, M. Easterby-Smith and M. A. Lyles Eds.. Oxford: Blackwell Publishing Ltd., 2003. p. 410-427

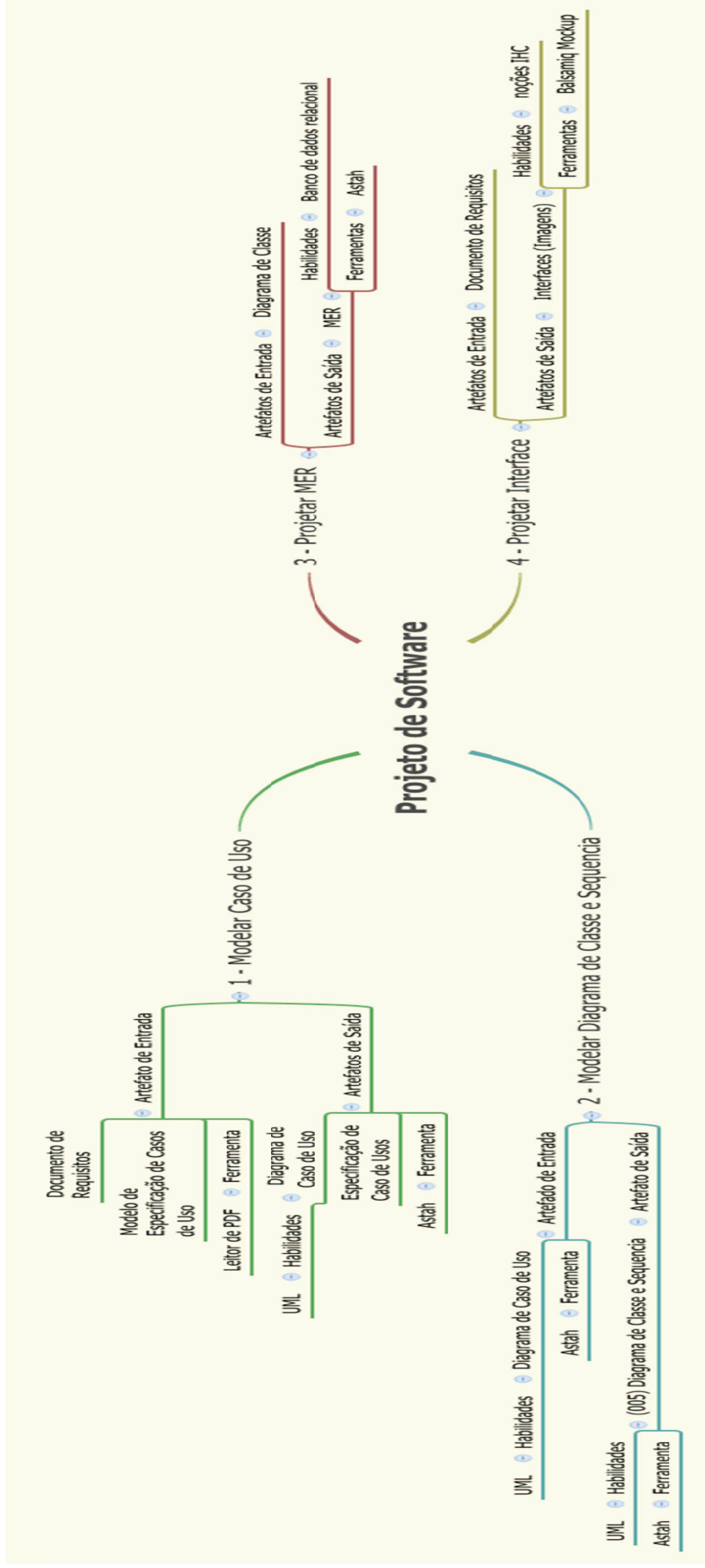
WOHLIN, Claes et al. **Experimentation in software engineering.** Springer Science & Business Media, 2012

YIN, Robert. **Estudo de caso.** 3ed. São Paulo: Bookman, 2005

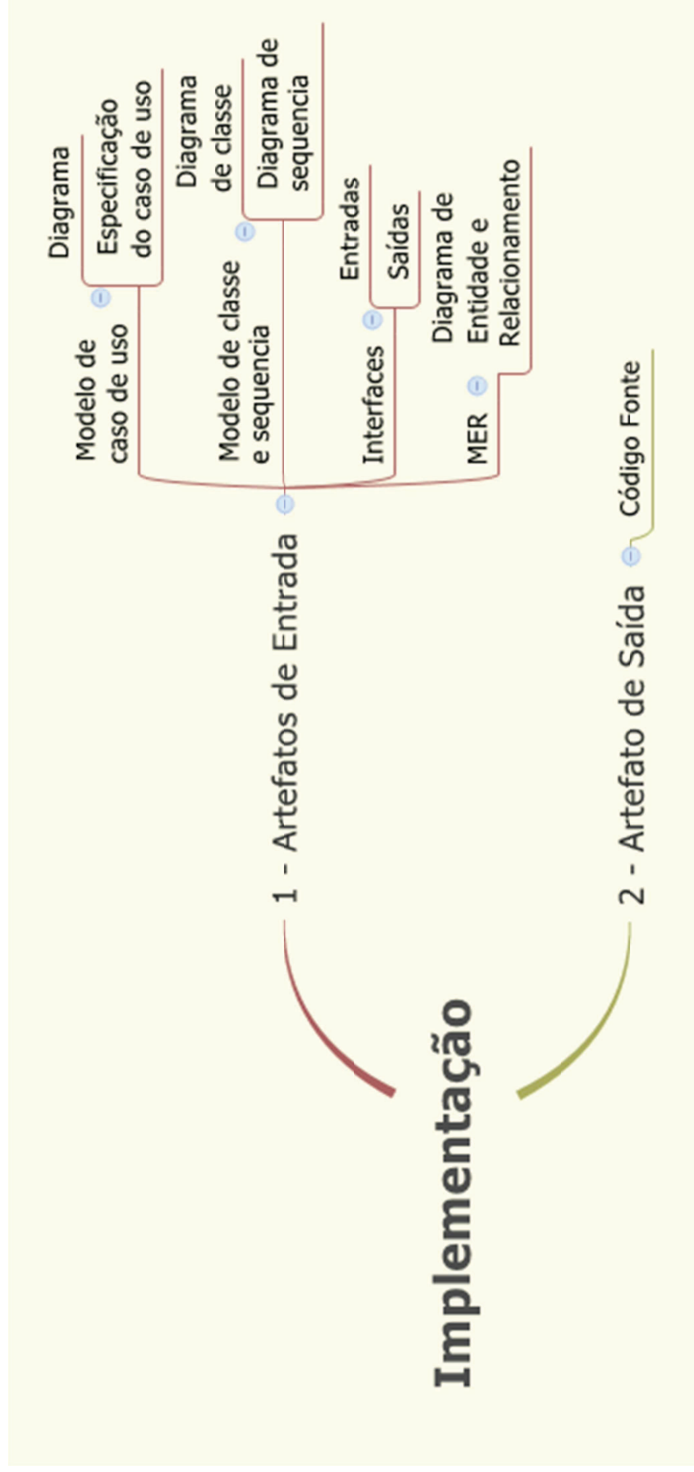
# ANEXO A – MAPAS MENTAIS ATIVIDADE GESTÃO DE PROJETO



## ANEXO A – MAPAS MENTAIS ATIVIDADE PROJETO DE SOFTWARE



## ANEXO A – MAPAS MENTAIS ATIVIDADE IMPLEMENTAÇÃO



## ANEXO B – ENDEREÇOS DOS CONTEÚDOS SOBRE O ARCABOUÇO CONCEITUAL DA GESTÃO DE PROJETO DE SOFTWARE

### REQUISITOS DE SOFTWARE

01. A relação entre sistema e software

<https://enghariasoftware.wordpress.com/2008/05/03/>

02. A incapacidade das empresas na gestão de projetos de software

<https://enghariasoftware.wordpress.com/2011/02/16/a-incapacidade-das-empresas-em-gerir-projetos-de-software/>

03. Figura clássica sobre requisitos de software

<https://enghariasoftware.wordpress.com/2010/04/06/engenharia-de-requisitos/>

04. Comunicação como fator determinante na gestão de projeto de software

<https://enghariasoftware.wordpress.com/2010/09/21/a-comunicacao-como-fator-determinante-na-gestao-de-projetos-de-software/>

05. Problemas mais frequentes em gestão de projetos

<https://enghariasoftware.wordpress.com/2014/01/07/problemas-mais-frequentes-na-gestao-de-projeto/>

06. O problema do mapeamento de requisitos no teatro do milênio

<https://enghariasoftware.wordpress.com/2012/11/13/comunicacao-e-gestao-de-projetos/>

07. O estoque de conhecimento

<https://enghariasoftware.wordpress.com/2010/03/30/a-melhor-saida-esta-no-estoque-do-conhecimento/>

08. Videos sobre mapas mentais

<https://mapamentalconceitual.wordpress.com/category/mapas-mentais/videos-mm/>

09. Materializando escopo de projetos com mapas mentais

<https://enghariasoftware.wordpress.com/2009/07/06/materializando-o-escopo-de-um-projeto-com-mapas-mentais/>

### FORMAS DE LEVANTAMENTO DE REQUISITOS

01. Formas de levantar requisitos - *Brainstorm*

<https://enghariasoftware.wordpress.com/2010/12/01/tecnicas-para-desenvolver-um-brainstorm/>

02. Formas de levantar requisitos - *Workshop*

<https://enghariasoftware.wordpress.com/2012/12/04/desenvolvendo-um-workshop-para-levantar-requisitos-de-software/>

03. Formas de levantar requisitos - Entrevista

<https://enghariasoftware.wordpress.com/2012/12/05/estruturando-uma-entrevista-para-o-levantamento-de-requisitos-de-software/>

04. Formas de levantar requisitos – Etnografia

<https://enghariasoftware.wordpress.com/2012/12/11/a-etnografia-como-processo-de-levantamento-de-requisitos/>

## ESPECIFICAÇÃO DE PROJETOS DE SOFTWARE E DE FERRAMENTAS

01. Formas de especificação de requisitos (projeto de software)

<https://engenhariasoftware.wordpress.com/2012/11/20/3cs-na-especificacao-de-requisitos-de-software-ers/>

02. Qual a melhor forma de especificação de requisitos

<https://engenhariasoftware.wordpress.com/2012/11/26/qual-a-melhor-forma-para-especificar-um-requisito-funcional-de-software/>

03. Especificação de requisitos – ferramenta Astah

<http://astah.net/download>

04. Tutorial da ferramenta Astah Professional

<https://engenhariasoftware.wordpress.com/2013/06/11/tutorial-sobre-aplicabilidade-do-astah-professional-no-projeto-de-software/>

05. Ferramenta Cmap

<http://cmap.ihmc.us/>

06. Ferramenta Xmin

<http://www.xmind.net/download/win/>

07 Ferramenta Freemind – download

<http://freemind.sourceforge.net/wiki/index.php/Download>

## PROCESSO DE SOFTWARE

01. Guia para instanciação de processo

<https://engenhariasoftware.wordpress.com/2011/08/27/um-guia-para-instanciacao-de-um-processo-de-software/>

02. Modelo Cascata

<https://engenhariasoftware.wordpress.com/2013/01/24/a-aplicabilidade-do-modelo-cascata-na-engenharia-de-software/>

03. Modelo Rapid Application Development

<https://engenhariasoftware.wordpress.com/2013/01/24/rapid-application-development-rad/>

04. Modelo Incremental

<https://engenhariasoftware.wordpress.com/2013/01/26/os-modelos-evolutivos-de-processo-de-software-a-aplicabilidade-do-modelo-incremental/>

05. Modelo Espiral

<https://engenhariasoftware.wordpress.com/2013/02/01/os-modelos-evolutivos-de-processo-de-software-espiral/>

04. Open UP

<https://engenhariasoftware.wordpress.com/2011/12/14/uma-alternativa-ao-rup-o-open-up/>

05. Scrum

<https://engenhariasoftware.wordpress.com/2013/03/19/scrum-na-integra/>

06. *Extreme Programming* XP

<https://engenhariasoftware.wordpress.com/2013/03/19/scrum-na-integra/>



### GESTÃO DE PROJETO DE SOFTWARE

01. Processo e complexidade de software - caso de uso.

<https://engenhariasoftware.wordpress.com/2013/08/28/aplicando-os-pontos-por-caso-de-uso-para-mapear-a-complexidade-de-um-software/>

02. Quadrinhos.

<http://stripgenerator.com/strip/813770/quadrinhos-x/#share>

03. Técnica do Pomodoro.

<https://engenhariasoftware.wordpress.com/2009/12/09/utilizando-tomates-no-planejamento-de-seu-tempo/>

04. Ferramenta de gestão de tempo focada na técnica do pomodoro.

<https://engenhariasoftware.wordpress.com/2013/03/21/ferramenta-de-gestao-de-tempo-focada-na-tecnica-pomodoro/>

### GESTÃO DE PROJETO DE SOFTWARE – VISÕES

01. Estrutura de dados para organização de processo.

<https://engenhariasoftware.wordpress.com/2011/08/29/estrutura-de-dados-para-a-gestao-de-projetos-de-software/>

02. A relação entre conhecimento e gerência: Uma visão peculiar.

<https://engenhariasoftware.wordpress.com/2009/06/22/a-relacao-entre-conhecimento-e-gerencia-uma-visao-peculiar/>

03. Construindo resultados com gerenciamento de projetos.

<https://engenhariasoftware.wordpress.com/2010/09/15/construindo-resultados-com-o-gerenciamento-de-projetos/>

04. Pontos por funções, é possível comprar a implementação de 1.000 pontos por função.

<https://engenhariasoftware.wordpress.com/2008/06/18/e-possivel-comprar-a-implementacao-de-1000-pontos-por-funcao-para-um-projeto-de-software/>

05. Interpretação dos pontos por função

<https://engenhariasoftware.files.wordpress.com/2012/05/relacao-contagem-pontos-por-funcao.png>

### GESTÃO DE PROJETO DE SOFTWARE – FERRAMENTAS

01. Como aplicar o Kanban na gestão de projeto de software.

<https://engenhariasoftware.wordpress.com/2013/05/15/como-aplicar-o-kanban-na-gestao-de-projetos-de-software>

02. Ferramenta *on line* para Kanban

<https://kanbanflow.com/>

03. Ferrament Kanbanize.

<https://kanbanize.com/>

04. Ferramenta Kanban – trello.

<https://trello.com/>

## ANEXO C – AVALIAÇÃO PRÉ-RESIDÊNCIA APLICADA

Universidade Tecnológica Federal do Paraná – Campus Cornélio Procopio  
Programa de Pós-Graduação em Informática PPGI  
Área: Engenharia de Software  
Residência em Planejamento, Execução e Controle de um Projeto de Software.  
Avaliação – Formativa

Nome: \_\_\_\_\_ Data: 29-03 - Hora: \_\_\_\_: \_\_\_\_

Formação: \_\_\_\_\_ Tempo de Formado: \_\_\_\_\_ anos

Quanto tempo de experiência em desenvolvimento de software? \_\_\_\_\_ anos Possui especialização: ( ) Sim ( ) Não

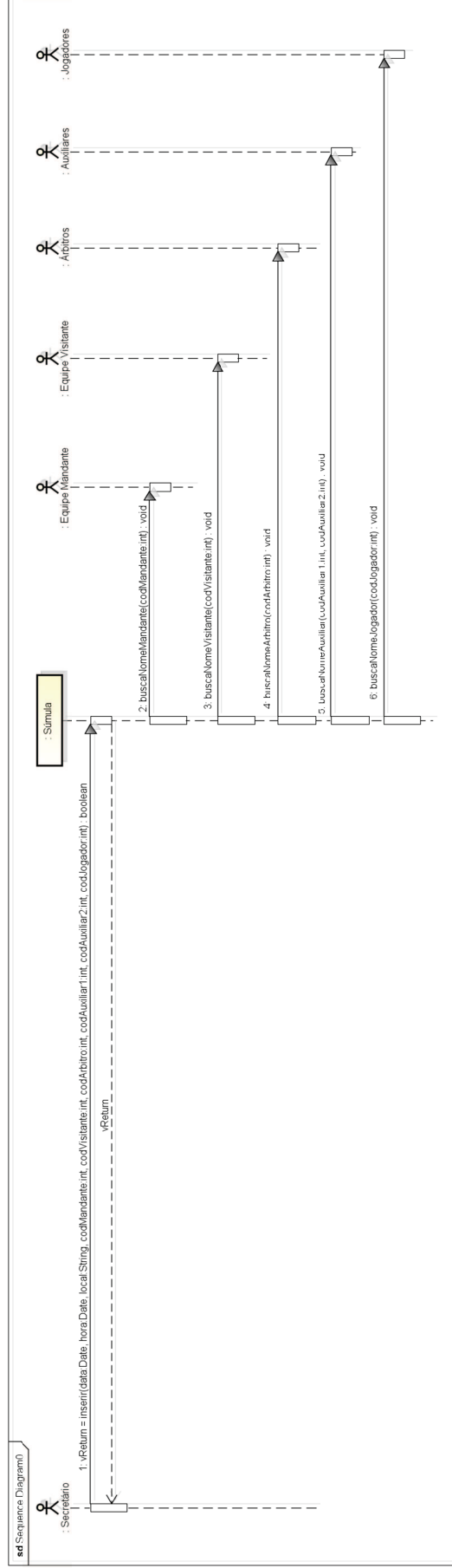
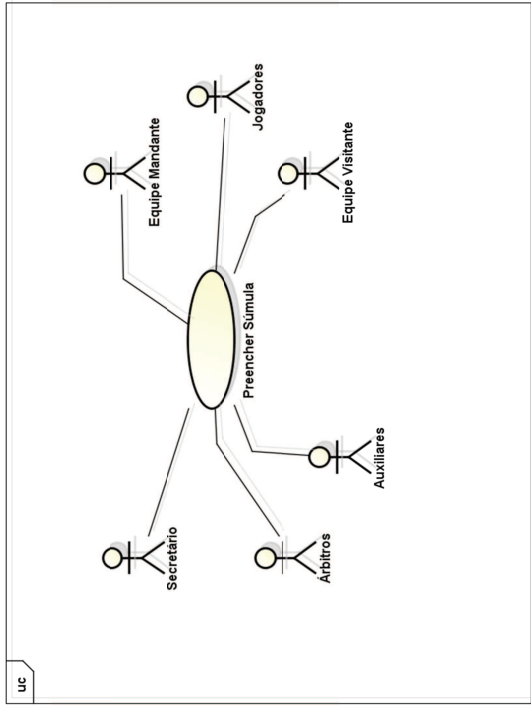
Profissão: \_\_\_\_\_ Tempo de atuação.: \_\_\_\_\_ anos

### Instruções para realização da avaliação

- 1 – Esta avaliação possui caráter formativo
- 2 – A avaliação é composta por 3 questões (1 teórica e 2 práticas)
- 3 – O tempo de realização da avaliação é de 60 minutos
- 4 – As respostas devem ser grafadas a tinta

### Questões

- 1 – Um projeto de software possui em seu escopo dois artefatos:
  - Diagrama de caso de uso;
  - Diagrama de sequencia.



De posse dos diagramas estabeleça o número de pontos por caso de uso para o projeto.

Utilize a tabela abaixo para efetuar o cálculo:

Ator	Interface	Peso
Simple	Interface de programa (API)	1
Médio	Protocolo (Ex.:TCP/IP) ou interface em modo texto	2
Complexo	Interface gráfica	3

Caso de Uso	Descrição	Peso
Simple	<= 3 transações ou < 5 classes de análise	1
Médio	4-7 transações ou 5 a 10 classes de análise	2
Complexo	> 7 transações ou > 10 classes de análise	3

Total de pontos por caso de uso não ajustados: \_\_\_\_\_

2 – Um elemento de extrema importância para a atividade de planejamento é a base histórica de projetos de software. Dentro deste contexto escreva um algoritmo que leia 2 número inteiros e apresente qual deles é maior. Durante o desenvolvimento do algoritmo registre as informações que você considera importante para a base histórica.

Algoritmo:

Registro das informações



## ANEXO D – PROCESSO DE SOFTWARE DA SEGUNDA APLICAÇÃO

### Universidade Tecnológica Federal do Paraná – Campus Cornélio Procopio

Programa de Pós-Graduação em Informática PPGI

Área: Engenharia de Software

Residência em Planejamento, Execução e Controle de um Projeto de Software.

Avaliação – Formativa

#### Processo

Durante a residência aplicada, os residentes deverão construir um produto caracterizado como software – a construção deste produto passa pelo processo. O processo é dividido em atividade e as atividades são fracionadas em tarefas – cada tarefa possui artefatos de entrada e de saída, as atividades também são automatizadas por ferramentas. É de extrema importância que você faça uma leitura sobre o conceito de processo antes de iniciar a residência aplicada.

Texto que define a ideia de processo: <http://wp.me/pcuYv-px>

#### Processo da residência aplicada

Atividades	Tarefas	Artefatos de Entrada	Artefatos de Saída	Habilidades	Ferramentas
Codificação	---	Documento de requisitos	Código Fonte	Programação	Linguagens ou IDES
	Definir casos de teste	Documento de requisitos	Casos de teste	Testador	Editor de texto
Definir dados de teste	Casos de teste	Dados de teste	Editor de Texto, IDE, Linguagem		
Testar	Dados de teste e programa executável	Programa executável testado	Editor de texto		
Teste	Relatar o teste	Programa executável testado	Relatório de teste		Editor de texto

