

**UNIVERSIDADE FEDERAL DE SERGIPE  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO**

**UEF-WEB: *FRAMEWORK* DE APOIO À ENGENHARIA DE  
USABILIDADE PARA APLICAÇÕES *WEB***

**DIEGO SANTANA SILVEIRA**

**SÃO CRISTÓVÃO/SE**

**2015**

**UNIVERSIDADE FEDERAL DE SERGIPE  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO**

**DIEGO SANTANA SILVEIRA**

**UEF-WEB: *FRAMEWORK* DE APOIO À ENGENHARIA DE  
USABILIDADE PARA APLICAÇÕES *WEB***

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal do Sergipe (UFS) como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

**Orientador:** Prof. Dr. Henrique Nou Schneider

**SÃO CRISTÓVÃO/SE**

**2015**

**FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL  
UNIVERSIDADE FEDERAL DE SERGIPE**

Silveira, Diego Santana  
S587u UEF-WEB : *framework* de apoio à engenharia de usabilidade  
para aplicações *WEB* / Diego Santana Silveira ; orientador  
Henrique Nou Schneider. – São Cristóvão, 2015.  
127 f. : il.

Dissertação (Mestrado em Ciência da Computação) –  
Universidade Federal de Sergipe, 2015.

1. Framework (Programa de computador). 2. World Wide Web  
(Sistema de recuperação da informação). 3. Interfaces  
(computadores). 4. Internet. I. Schneider, Henrique Nou, orient. II.  
Título.

CDU 004.738.5

**DIEGO SANTANA SILVEIRA**

**UEF-WEB: *FRAMEWORK* DE APOIO À ENGENHARIA DE  
USABILIDADE PARA APLICAÇÕES *WEB***

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal do Sergipe (UFS) como parte de requisito para obtenção do título de Mestre em Ciência da Computação.

**BANCA EXAMINADORA**

Prof. Dr. Henrique Nou Schneider, Orientador  
Universidade Federal de Sergipe (UFS)

Profa. Dra. Adicinéia Aparecida de Oliveira, Membro  
Universidade Federal de Sergipe (UFS)

Prof. Dr. Sérgio Crespo Coelho da Silva Pinto, Membro  
Universidade Federal Fluminense (UFF)

# UEF-WEB: *FRAMEWORK* DE APOIO À ENGENHARIA DE USABILIDADE PARA APLICAÇÕES *WEB*

Este exemplar corresponde à redação final da Dissertação de Mestrado, sendo o Exame de Defesa do mestrando **DIEGO SANTANA SILVEIRA** para ser aprovada pela Banca Examinadora.

São Cristóvão, 13 de Março de 2015

---

Prof. Dr. Henrique Nou Schneider  
Orientador

---

Profa. Dra. Adicinéia Aparecida de Oliveira  
Membro

---

Prof. Dr. Sérgio Crespo Coelho da Silva Pinto  
Membro

Dedico esta dissertação a minha família pelo contínuo apoio e incentivo recebidos durante o desenvolvimento desta pesquisa.

## AGRADECIMENTOS

---

A Deus, pela graça, oportunidade e força concedidas. Confesso que sem a Tua proteção e ajuda não teria conseguido concluir este projeto.

A minha família – meus pais Thomaz e Maria José e minhas irmãs Grazielle, Vanessa e Andreza – por não somente acreditar nas minhas realizações acadêmicas, mas também pelo apoio e incentivo recebidos neste período de qualificação profissional.

Ao meu orientador professor Henrique Nou Schneider, pela confiança depositada, pela orientação fornecida e pelas lições aprendidas no decorrer desta caminhada. Ao Grupo de Estudos e Pesquisa em Informática na Educação (GEPIED/UFS), pelo incentivo concedido.

Aos professores Adicinéia Aparecida de Oliveira e Sérgio Crespo Coelho da Silva Pinto, pelas contribuições apresentadas durante os exames de qualificação e de defesa desta dissertação de mestrado.

Aos docentes e funcionários do Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal de Sergipe (UFS), pelo apoio recebido de forma direta ou indireta ao longo desta jornada. Aos colegas do mestrado, pela convivência e amizade durante o curso.

Aos gestores do Departamento de Comunicação Social da UFS, pela compreensão e apoio aos meus estudos.

Obrigado!

Quem segue a justiça e a lealdade encontra vida,  
justiça e honra (Provérbios 21:21).

Nos últimos anos, a demanda por aplicações *Web* vem crescendo rapidamente para atender às necessidades das pessoas e empresas. Essas aplicações têm proporcionado benefícios que vão desde melhorias no processo de comunicação e maior velocidade ao realizar tarefas até a redução de custos. No entanto, inúmeros problemas relacionados à usabilidade de interfaces, a exemplo da alta complexidade e baixa eficácia de uso têm dificultado e, em muitos casos, inviabilizado a utilização de aplicações *Web*. Esse fato é preocupante, pois a usabilidade é apontada pelos usuários como o atributo de qualidade de *software* mais importante e por constatar que muitas organizações ainda não empregam os recursos de suporte à usabilidade sob vários motivos, tais como: desconhecimento de técnicas, métodos e ferramentas; restrição de orçamento; e dificuldade de seleção e uso sistemático desses recursos. Desenvolver interfaces com maior qualidade de uso é um desafio que tem se tornado crítico para o sucesso das aplicações *Web* junto às empresas e aos usuários. Pesquisas foram realizadas, mas há lacunas quanto ao fornecimento de soluções integradas de suporte à usabilidade nas diversas fases de um projeto de aplicação *Web*. Diante dessa necessidade, este trabalho propõe o *Framework* UEF-WEB de apoio à Engenharia de Usabilidade. Este *framework*, composto por fases, atividades, recursos e artefatos, tem por objetivo principal auxiliar as organizações a introduzir, de maneira sistematizada, recursos de usabilidade no processo de planejamento, desenvolvimento e avaliação de interfaces de aplicações *Web*, com vistas à ampliação da qualidade de uso das interfaces construídas.

**Palavras-chave:** *Framework* UEF-WEB, Engenharia de Usabilidade, Aplicações *Web*, Problemas de Usabilidade, Processo de Desenvolvimento de Interfaces.

## ABSTRACT

---

In recent years, the demand for Web applications is growing rapidly to meet the needs of individuals and businesses. These applications have provided benefits such as improvement in communication speed and greater process to perform tasks to cost reduction. However, numerous problems related to the usability of interfaces, such as the high complexity and low efficiency of use have made it difficult, and in many cases made impossible the use of Web applications. This fact is worrisome because the usability is pointed out by users as quality attribute of software and most important to note that many organizations do not employ the resources to support usability under various reasons, such as lack of techniques, methods and tools; budget constraint; and difficulty selection and systematic use of these resources. Developing interfaces with higher quality of use is a challenge that has become critical to the success of Web applications with companies and users. Researches have been conducted, but there are gaps in the provision of integrated support in usability in various stages of a Web application project. Given this need, this paper proposes the UEF-WEB Framework to support Usability Engineering. This framework, composed of stages, activities, resources and artifacts, has the main objective to help organizations to introduce a systematic way, usability features in the planning, development and evaluation of Web application interfaces process, in order to expand the quality using the built in interfaces.

**Keywords:** UEF-WEB Framework, Usability Engineering, Web Applications, Usability Issues, Development Process Interfaces.

---

## LISTA DE FIGURAS

---

<b>Figura 1.1</b>	Enquadramento Metodológico da Pesquisa .....	24
<b>Figura 2.1</b>	Modelo de Ciclo de Vida Estrela .....	32
<b>Figura 2.2</b>	Atividades da Fase de Análise de Requisitos .....	33
<b>Figura 2.3</b>	Atividades da Fase de Projeto, Testes e Implementação .....	34
<b>Figura 2.4</b>	Atividades da Fase de Instalação .....	34
<b>Figura 2.5</b>	Modelo de Projeto Centrado no Ser Humano para Sistemas Interativos .....	36
<b>Figura 3.1</b>	Arcabouço de Processo WebE .....	54
<b>Figura 3.2</b>	Diferentes Visões de Qualidade de <i>Software</i> .....	70
<b>Figura 3.3</b>	Modelo de Qualidade Externa e Interna .....	71
<b>Figura 3.4</b>	Modelo de Qualidade em Uso .....	72
<b>Figura 4.1</b>	<i>Framework</i> UEF-WEB .....	77
<b>Figura 4.2</b>	Atividades e Artefato da Fase de Planejamento do <i>Framework</i> UEF-WEB..	80
<b>Figura 4.3</b>	Atividades e Artefatos da Fase de Desenvolvimento do <i>Framework</i> UEF- WEB .....	87
<b>Figura 4.4</b>	Atividade e Artefato da Fase de Avaliação do <i>Framework</i> UEF-WEB .....	98
<b>Figura 4.5</b>	Escalas de Classificação e a Pontuação do SUS .....	100

## LISTA DE QUADROS

---

<b>Quadro 1.1</b>	Resumo Comparativo entre os Trabalhos Relacionados e o <i>Framework</i> UEF-WEB.....	19
<b>Quadro 3.1</b>	Categorias de Aplicações <i>Web</i> .....	51
<b>Quadro 3.2</b>	Arcabouço de Arquitetura de Aplicação <i>Web</i> (AAAW) .....	61
<b>Quadro 3.3</b>	Princípios de Projeto para Aplicações <i>Web</i> .....	65
<b>Quadro 4.1</b>	Fases, Atividades, Recursos e Artefatos do <i>Framework</i> UEF-WEB .....	78
<b>Quadro 4.2</b>	Roteiro Básico de Questões para Entrevista e Reunião .....	83
<b>Quadro 4.3</b>	Princípios, Diretrizes e Heurísticas de Usabilidade para Interfaces .....	88
<b>Quadro 4.4</b>	Artefato Formulário de <i>Feedback</i> do <i>Framework</i> UEF-WEB .....	89
<b>Quadro 4.5</b>	Perspectivas de Projeto <i>Web</i> e o Foco na Usabilidade .....	91
<b>Quadro 4.6</b>	Artefato Questionário de Satisfação do <i>Framework</i> UEF-WEB – Baseado no Questionário SUS ( <i>System Usability Scale</i> ) de Brooke (1986) .....	101

## LISTA DE TABELAS

---

<b>Tabela 2.1</b>	As Dez Heurísticas de Usabilidade de Nielsen .....	42
<b>Tabela 2.2</b>	Escala de Severidade para os Problemas de Usabilidade .....	43
<b>Tabela 4.1</b>	Tabela de Especificação de Requisitos de Usabilidade .....	84
<b>Tabela 4.2</b>	Relação entre Heurísticas e Perspectivas na Técnica WDP .....	92
<b>Tabela 4.3</b>	Artefato Formulário de Inspeção do <i>Framework</i> UEF-WEB .....	96

## LISTA DE SIGLAS

---

AAAW	Arcabouço de Arquitetura de Aplicação <i>Web</i>
ABNT	Associação Brasileira de Normas Técnicas
ACM	<i>Association for Computing Machinery</i>
DBC	Desenvolvimento Baseado em Componentes
HFI	Integração de Fatores Humanos
IEC	<i>International Electrotechnical Commission</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IHC	Interação Humano-Computador
ISO	<i>International Standards Organization</i>
IU	Interface de Usuário
MCP	Modelo Conceitual do Projeto
NBR	Norma Brasileira
PDIU	Projeto Detalhado da Interface de Usuário
PPT	Padrões de Projeto de Telas
QUIS	<i>Questionnaire for User Interaction Satisfaction</i>
SUMI	<i>Software Usability Measurement Inventory</i>
SUS	<i>System Usability Scale</i>
UEF-WEB	<i>Framework</i> de apoio à Engenharia de Usabilidade para Aplicações <i>Web</i>
USEMATE	Sistema de Gestão de Usabilidade
WAMMI	<i>Web Analysis and Measurement Inventory</i>
WDP	<i>Web Design Perspectives-based Usability Evaluation</i>
Webapps	Aplicações <i>Web</i>
WebE	Engenharia <i>Web</i>
WE-QT	<i>Web Evaluation – Question Technique</i>
W3C	<i>World Wide Web Consortium</i>

<b>Capítulo 1 – Introdução .....</b>	<b>15</b>
1.1 Contextualização .....	15
1.2 Problemática e Hipótese .....	17
1.3 Objetivos da Pesquisa .....	17
1.4 Trabalhos Relacionados .....	18
1.5 Justificativa .....	22
1.6 Metodologia .....	24
1.7 Organização da Pesquisa .....	25
<b>Capítulo 2 – Engenharia de Usabilidade .....</b>	<b>27</b>
2.1 Usabilidade na Interação Humano-Computador (IHC) .....	27
2.2 Atributos de Usabilidade .....	29
2.3 Engenharia de Usabilidade .....	30
2.3.1 Modelos de Ciclos de Vida .....	31
2.3.2 Métodos .....	39
2.3.2.1 Entrevista e Reunião .....	39
2.3.2.2 Prototipação .....	40
2.3.2.3 Avaliação Heurística .....	40
2.3.2.4 Inspeção de Usabilidade Baseada em Perspectivas .....	43
2.3.2.5 Teste de Usabilidade .....	44
2.3.2.6 Questionário de Satisfação .....	45
2.4 Considerações Finais do Capítulo .....	47
<b>Capítulo 3 – Usabilidade em Aplicações <i>Web</i> .....</b>	<b>49</b>
3.1 Aplicações <i>Web</i> : conceito, atributos e categorias .....	49
3.2 WebE: Um Arcabouço de Processo para a Engenharia <i>Web</i> .....	52

3.3 <i>Framework</i> como Instrumento de Apoio ao Desenvolvimento e Avaliação de Aplicações <i>Web</i> .....	56
3.3.1 O Conceito <i>Framework</i> .....	57
3.3.2 <i>Framework</i> no Processo de Engenharia de <i>Software Web</i> .....	61
3.4 Usabilidade em Aplicações <i>Web</i> .....	62
3.5 Princípios e Diretrizes de Projeto de Interfaces para Aplicações <i>Web</i> .....	64
3.6 Importância da Usabilidade para a Qualidade das Aplicações <i>Web</i> .....	70
3.7 Considerações Finais do Capítulo .....	73
<b>Capítulo 4 – <i>Framework</i> UEF-WEB .....</b>	<b>75</b>
4.1 Introdução .....	75
4.2 <i>Framework</i> UEF-WEB .....	76
4.2.1 Fase de Planejamento .....	80
4.2.2 Fase de Desenvolvimento .....	86
4.2.3 Fase de Avaliação .....	98
4.3 Considerações Finais do Capítulo .....	103
<b>Capítulo 5 – Considerações Finais .....</b>	<b>105</b>
<b>REFERÊNCIAS .....</b>	<b>108</b>
<b>ANEXO A – Questionário SUS (<i>System Usability Scale</i>) .....</b>	<b>116</b>
<b>APÊNDICE A – Contexto de Uso e Requisitos de Usabilidade .....</b>	<b>117</b>
<b>APÊNDICE B – Formulário de <i>Feedback</i> .....</b>	<b>125</b>
<b>APÊNDICE C – Formulário de Inspeção .....</b>	<b>126</b>
<b>APÊNDICE D – Questionário de Satisfação .....</b>	<b>127</b>

## INTRODUÇÃO

### 1.1 Contextualização

Nos últimos anos, a demanda por aplicações *Web* vem crescendo rapidamente para atender às variadas necessidades das pessoas e organizações como, por exemplo, apoiar a realização de atividades laborais, comerciais, educacionais e de comunicação, entre outras (PRESSMAN, 2011). Essa expansão pode ser percebida não somente em relação à quantidade e diversidade, como também no nível de complexidade das aplicações.

É importante acrescentar que a facilidade de acesso a informações e serviços proporcionada pela *Web* tem possibilitado o uso em larga escala dessas aplicações na sociedade, como suporte à realização de atividades cotidianas. Mais do que apoiar atividades, as aplicações *Web* estão contribuindo para influenciar a forma como as pessoas efetuam compras (comércio eletrônico), procuram informações (ferramentas de busca e portais de notícias), comunicam-se (*email* e *chats*), expressam opiniões (*blogs* e redes sociais), estudam (educação à distância) e realizam atividades de entretenimento (jogos eletrônicos *online*) (PRESSMAN; LOWE, 2009).

O termo aplicação *Web* é definido como um sistema de *software* baseado em tecnologias e padrões do *World Wide Web Consortium* (W3C)<sup>1</sup> que fornece recursos específicos da *Web* tais como conteúdo e serviços, por meio de uma interface de usuário denominada *browser Web* (KAPPEL *et al.*, 2006). Uma aplicação *Web*, no entanto, não está restrita a um navegador *Web* como elemento indispensável de interface, pois existem aplicações que se utilizam da infraestrutura da *Web*, mas não fazem uso de um *browser* para apresentar e explorar informações, a exemplo de aplicações envolvendo *workflow* científico e características de ubiquidade<sup>2</sup> computacional (SILVA, 2011).

Os benefícios obtidos através das aplicações *Web* são inúmeros e vão desde a automatização, maior velocidade na execução de tarefas, facilidades e melhorias no processo de comunicação até a redução de custos (PRESSMAN, 2011). Entretanto, diversos problemas

---

<sup>1</sup> *World Wide Web Consortium* (W3C). O W3C é um consórcio internacional no qual organizações filiadas trabalham juntas para desenvolver tecnologias e padrões para a *Web* (W3C, 2014).

<sup>2</sup> Ubiquidade: paradigma de computação que explora a integração crescente de dispositivos de computação com o mundo físico do dia-a-dia (COULOURIS; DOLLIMORE; KINDBERG, 2005).

relativos à ergonomia das interfaces de aplicações *Web* têm sido observados com frequência. Problemas como alta complexidade, ausência de padrões, baixa flexibilidade, facilidade e eficácia de uso têm dificultado e, em muitos casos, inviabilizado o uso de aplicações *Web* (NIELSEN; LORANGER, 2007; NASCIMENTO; AMARAL, 2010).

A ergonomia, em suma, é definida como a ciência que estuda a adequação do trabalho às características do ser humano para garantir que sistemas e dispositivos estejam adaptados ao modo como o usuário pensa, comporta-se e exerce atividades laborais (CYBIS; BETIOL; FAUST, 2010). No que diz respeito à Interação Humano-Computador (IHC), a ergonomia de interfaces pode ser mensurada através do nível de usabilidade fornecido.

Usabilidade, por seu turno, pode ser entendida como um atributo de qualidade associado à facilidade de se utilizar e ao nível de satisfação em realizar algo (CYBIS; BETIOL; FAUST, 2010). O usuário, ao fazer uso de um *software* ou aplicação *Web*, espera encontrar interfaces simples, intuitivas, sem erros, eficientes e agradáveis. O atendimento a esse objetivo é condição fundamental para o uso contínuo e para sucesso de uma aplicação *Web*, tendo em vista que qualquer barreira encontrada pelo usuário pode causar a rejeição a uma interface computacional (NIELSEN; LORANGER, 2007).

Uma aplicação *Web* para ser bem sucedida deve, além de ser eficiente e segura, dispor de um bom nível de usabilidade (NASCIMENTO; AMARAL, 2010). Quando os mecanismos empregados para exibir, coletar informações e realizar a interação com as interfaces são bem projetados, por exemplo, os usuários desenvolvem suas atividades de forma correta e sem maiores esforços. No momento em que isso não ocorre, as atividades deixam de ser realizadas adequadamente, o que contribui para reduzir a produtividade e elevar a frustração dos usuários (NIELSEN; LORANGER, 2007).

As dificuldades enfrentadas diariamente pelas pessoas ao interagir com interfaces computacionais de baixa qualidade ergonômica constituem um fato preocupante, pois a usabilidade é apontada como o atributo de qualidade de *software* mais relevante na percepção dos usuários (DE LA VARA *et al.*, 2011). Essa importância se revela maior no caso das aplicações *Web*, notadamente interativas e centradas no usuário.

Diante disso, a Engenharia de Usabilidade, que compreende o processo de desenvolvimento de interfaces com vistas à facilidade de uso e à satisfação dos usuários, tem fornecido técnicas e métodos para reduzir os problemas de usabilidade das interfaces (ROCHA; BARANAUSKAS, 2003). Todavia, ainda há organizações que não fazem uso dessas contribuições ao projetar e construir aplicações *Web* (INSFRAN; FERNANDEZ, 2008). O desconhecimento de técnicas e métodos, a limitação de orçamento, assim como

dificuldades no processo de seleção e implantação sistemática de recursos de usabilidade são algumas das causas apontadas para explicar essa situação (NIELSEN; LORANGER, 2007; CONTE, 2009; CYBIS; BETIOL; FAUST, 2010).

## 1.2 Problemática e Hipótese

O problema discutido nesta dissertação está relacionado à dificuldade de se introduzir e utilizar, de forma sistematizada, recursos da Engenharia de Usabilidade no processo de desenvolvimento de interfaces de aplicações *Web*. Como visto, além do fato de muitas organizações não fazerem uso de recursos de suporte à usabilidade por vários motivos, a qualidade das aplicações *Web* tem sido, com frequência, avaliada de forma *ad hoc*, tendo por base as percepções, o senso comum e os conhecimentos dos desenvolvedores (ABRAHÃO *et al.*, 2003; NASCIMENTO; AMARAL, 2010). Diante disso, observa-se a necessidade de se desenvolver, avaliar e implantar adequadamente abordagens de apoio à usabilidade nas organizações, com vistas a garantir qualidade no uso das interfaces.

Esta dissertação pretende responder a seguinte **questão de pesquisa**: como auxiliar as organizações a introduzir no processo de desenvolvimento de interfaces de aplicações *Web* recursos da Engenharia de Usabilidade de forma sistematizada, a fim de promover maior qualidade de uso para as interfaces construídas?

A **hipótese de trabalho** desta dissertação consiste no seguinte argumento: a utilização de um *framework* de apoio à Engenharia de Usabilidade pode auxiliar as organizações a introduzir, de forma sistematizada, recursos de usabilidade no processo de desenvolvimento de interfaces de aplicações *Web* e promover maior qualidade de uso para as interfaces construídas.

## 1.3 Objetivos da Pesquisa

O **objetivo geral** desta pesquisa é propor um *framework* de apoio à Engenharia de Usabilidade para as fases de planejamento, desenvolvimento e avaliação de interfaces de aplicações *Web*. Este *framework*, denominado de UEF-WEB (*Framework* de apoio à Engenharia de Usabilidade para aplicações *WEB*), busca promover a sistematização de princípios e recursos de usabilidade nas organizações, como forma de possibilitar a ampliação

da qualidade de uso das interfaces produzidas. Para alcançar o objetivo geral, os seguintes **objetivos específicos** foram estabelecidos:

- Conhecer os princípios, atividades e recursos norteadores da Engenharia de Usabilidade, assim como modelo de processo, *framework*, princípios e diretrizes de projeto de interfaces empregados no desenvolvimento de aplicações *Web*;
- Especificar o *Framework* UEF-WEB de apoio à Engenharia de Usabilidade para as fases de planejamento, desenvolvimento e avaliação de interfaces de aplicações *Web*.

As principais contribuições esperadas com o desenvolvimento desta pesquisa são apresentadas, a seguir: (i) a definição de um *framework* de suporte à Engenharia de Usabilidade para as fases de planejamento, desenvolvimento e avaliação de interfaces de aplicações *Web*; (ii) a disponibilização deste *framework* para a comunidade profissional e acadêmica como forma de auxiliar as organizações na adoção sistemática e facilitada de princípios e recursos de usabilidade, com vistas à ampliação da qualidade de uso das interfaces de aplicações *Web* construídas.

## 1.4 Trabalhos Relacionados

Os trabalhos relacionados a esta dissertação foram identificados através de uma pesquisa bibliográfica. Esta pesquisa foi realizada utilizando *strings* de busca<sup>3</sup> nos idiomas português e inglês e nas seguintes bases de dados científicas disponíveis na Internet:

- IEEE *Xplore* (<http://ieeexplore.ieee.org>);
- ACM *Digital Library* (<http://dl.acm.org>);
- Springer (<http://link.springer.com>);
- Biblioteca Digital de Teses e Dissertações (<http://bdtd.ibict.br>).

---

<sup>3</sup> *Strings* de busca no idioma português: (*framework* ou processo) e (suporte ou avaliação ou teste) e (usabilidade ou “Engenharia de Usabilidade”) e (interfaces ou aplicações ou *Web*); e (método ou ferramenta ou *software*) e (suporte ou avaliação ou teste) e (usabilidade ou “Engenharia de Usabilidade”) e (interfaces ou aplicações ou *Web*). *Strings* no idioma inglês: (*framework or process*) and (*support or evaluation or test*) and (*usability or “Usability Engineering”*) and (*interface or applications or Web*); e (*method or tool or software*) and (*support or evaluation or test*) and (*usability or “Usability Engineering”*) and (*interface or applications or Web*).

Através desta pesquisa bibliográfica, foram identificados nove trabalhos relacionados. Um resumo comparativo desses trabalhos com esta dissertação é apresentado no Quadro 1.1. Para realizar essa comparação, foi levado em consideração o critério de enfoque adotado em cada um dos trabalhos. Em outras palavras, procurou-se observar se a contribuição apresentava um arcabouço de atividades e em quais fases do processo de desenvolvimento de interfaces as propostas estavam inseridas, a saber: fase de planejamento, fase de desenvolvimento e fase de avaliação.

**Quadro 1.1** – Resumo Comparativo entre os Trabalhos Relacionados e o *Framework* UEF-WEB.

Trabalhos	Enfoque			
	Arcabouço de atividades	Planejamento	Desenvolvimento	Avaliação
Zahler e Bardach (2004)	X			
Gena e Weibelzahl (2007)		X	X	X
Al-Wabil e Al-Khalifa (2009)				X
Conte (2009)				X
Hart e Portwood (2009)				X
Ahmad, Sulaiman e Johari (2010)				X
Ferre e Bevan (2011)		X	X	X
Fernandes, Conte e Bonifácio (2012)				X
Fischer (2012)	X			
<i>Framework</i> UEF-WEB (2015)	X	X	X	X

A pesquisa de Zahler e Bardach (2004) apresenta um arcabouço de atividades de Engenharia de Usabilidade para as seguintes fases do processo de desenvolvimento: preparação, análise, projeto, implementação e teste. A atividade de análise de tarefas deve ser realizada durante as fases de preparação e de análise. Esta atividade envolve a participação de três a cinco representantes de usuário e tem por objetivo levantar informações sobre como os usuários trabalham, seu ambiente de uso e expectativas quanto ao futuro sistema. A atividade de sessão de revisão de protótipo, com início na fase de análise e continuação na fase de projeto, tem por finalidade refinar as especificações de interface de usuário. A atividade de

sessão de revisão de interface de usuário, realizada na fase de implementação, tem o intuito de garantir a consistência e a coerência das interfaces em desenvolvimento. Por fim, na fase de teste, deve ser conduzida a atividade de execução de caso de teste. Esta atividade tem por objetivo verificar a conformidade das interfaces construídas em relação aos requisitos e expectativas dos usuários. Para isso, devem ser criados e utilizados casos de teste de interface de usuário. Esta pesquisa, no entanto, não apresenta maiores detalhes acerca de cada uma dessas atividades.

Gena e Weibelzahl (2007) discutem uma abordagem de Engenharia de Usabilidade para o *design* e avaliação de sistemas adaptativos baseados na *Web*. Esta abordagem consiste em apresentar uma série de métodos para três fases de avaliação, a saber: fase de requisitos, fase de avaliação preliminar e fase de avaliação final. Na fase de requisitos, métodos são utilizados para reunir informações sobre os usuários típicos da aplicação (como, por exemplo, características, comportamentos, ações, necessidades, etc.), o domínio da aplicação, os objetivos e recursos do sistema. Alguns métodos desta fase incluem Análise de Tarefas, Grupo de Discussão e Observação Sistemática. Os métodos da fase de avaliação preliminar, a exemplo da Avaliação Heurística, Prototipação e Avaliação Participativa, devem ser aplicados durante o processo de desenvolvimento do sistema. Este tipo de avaliação visa reduzir os problemas de usabilidade presentes nas interfaces. Na fase de avaliação final, métodos devem ser empregados para avaliar a qualidade geral do sistema. Esta avaliação deve ser feita com a participação de usuários executando tarefas no sistema. Os métodos Teste de Usabilidade e Experimento Controlado fazem parte desta fase.

Al-Wabil e Al-Khalifa (2009) propõem um *framework* para a integração de métodos de avaliação de usabilidade. Este *framework* faz uso integrado dos métodos Teste de Usabilidade, Grupo de Discussão, Arranjo de Cartas e Rastreamento Ocular com o propósito de aproveitar os pontos fortes de cada um desses métodos e, com isso, possibilitar avaliações de usabilidade mais eficazes. O estudo de caso realizado nesta pesquisa avaliou a usabilidade de um portal *Web* e forneceu indícios de que este *framework* pode ser aplicado na avaliação de outras aplicações *Web*.

Conte (2009) apresenta uma técnica de inspeção de usabilidade baseada em perspectivas de projetos *Web*, denominada de WDP (*Web Design Perspectives-based Usability Evaluation*). Esta técnica de inspeção combina as dez Heurísticas de Usabilidade de Nielsen (1994b) com três perspectivas de projetos *Web*, a saber: apresentação, conceituação e navegação. A perspectiva de apresentação representa as características relativas à programação visual e ao *layout* da interface, definindo como as informações serão

apresentadas aos usuários. A perspectiva de conceituação diz respeito aos elementos conceituais que compõem o domínio da aplicação. A perspectiva de navegação, por sua vez, representa o espaço navegacional, definindo os elementos de acesso e suas associações na exploração das informações. O uso dessas perspectivas contribui para guiar a interpretação das Heurísticas de Usabilidade com foco em aplicações *Web*.

Hart e Portwood (2009) propõem uma metodologia para teste de usabilidade que combina o uso dos métodos de Avaliação Heurística e Teste Baseado em Tarefas. O estudo de caso realizado nesta pesquisa revelou que a metodologia proposta auxiliou, de forma organizada, a identificar os principais problemas de usabilidade do *site* avaliado. O trabalho de Ahmad, Sulaiman e Johari (2010) apresenta o Sistema de Gestão de Usabilidade (USEMATE) para auxiliar na realização e gestão de testes de usabilidade. Este sistema faz uso de Questionários como método de avaliação e emprega as medidas de usabilidade eficácia, eficiência e satisfação<sup>4</sup>. A pesquisa de Ferre e Bevan (2011), por sua vez, apresenta a ferramenta *Usability Planner* para apoiar o processo de seleção de métodos de *design* centrado no usuário. Esta ferramenta tem por objetivo fornecer uma lista de métodos de usabilidade recomendados para o projeto de *software*, a partir de informações coletadas sobre a natureza do projeto e restrições organizacionais.

O trabalho de Fernandes, Conte e Bonifácio (2012) apresenta a ferramenta WE-QT (*Web Evaluation – Question Technique*) de apoio à avaliação de usabilidade que faz uso do método Inspeção de Usabilidade. A avaliação, no entanto, é feita a partir de um conjunto de questões para reduzir as dificuldades e aumentar a eficiência da avaliação. Por fim, a pesquisa de Fischer (2012) discute a integração da Engenharia de Usabilidade no processo de desenvolvimento de *software*. Esta proposta baseia-se no uso de padrões para integrar a Engenharia de Usabilidade e a Engenharia de *Software*. Tais padrões podem ser ligados aos processos para apoiar papéis específicos de usabilidade na escolha de métodos e na comunicação de resultados.

Como visto, os trabalhos relacionados apresentam contribuições de suporte à usabilidade que versam desde a proposição de arcabouços de atividades e métodos até o desenvolvimento de ferramentas. Também foi possível observar que grande parte desses trabalhos são direcionados apenas ao processo de avaliação de usabilidade de interfaces. Esta dissertação faz uso de algumas dessas contribuições como, por exemplo, o uso dos métodos

---

<sup>4</sup> Segundo a ISO (2010), a medida de eficácia se refere à exatidão e completude com que objetivos específicos são atingidos pelos usuários; a medida de eficiência está associada aos recursos consumidos em relação à exatidão e completude com que os usuários alcançam os objetivos; a medida de satisfação mede a ausência de desconforto e a atitude positiva dos usuários quanto ao uso de um produto.

Avaliação Heurística, Inspeção de Usabilidade Baseada em Perspectivas, Teste de Usabilidade e Questionário de Satisfação. Entretanto, esta dissertação de mestrado diferencia-se dos trabalhos relacionados por fornecer, através de um *framework*, um arcabouço de fases, atividades, recursos e artefatos para auxiliar as organizações a sistematizar a adoção integrada de recursos de usabilidade durante o processo de planejamento, desenvolvimento e avaliação de interfaces de aplicações *Web*.

## 1.5 Justificativa

A usabilidade é considerada um dos mais importantes fatores de qualidade para aplicações *Web* juntamente com os fatores de confiabilidade e de segurança (OFFUTT, 2002). Dentre os seis atributos de qualidade de *software* listados na norma ABNT NBR 9126-1<sup>5</sup> (ABNT NBR, 2003), a usabilidade foi apontada como o atributo mais importante pelos usuários (DE LA VARA *et al.*, 2011). Essa importância mostra-se ainda maior no caso específico das aplicações *Web*, pois são naturalmente interativas, centradas no usuário e baseadas em hipermídia, nas quais a interface com o usuário representa o elemento central de comunicação e interação (OLSINA; COVELLA; ROSSI, 2006).

De acordo com Nielsen e Loranger (2007), um potencial cliente passa, em média, menos de 30 segundos na página inicial (*homepage*) e menos de dois minutos interagindo pela primeira vez com uma aplicação *Web*. E mais, este mesmo usuário acessará outra aplicação caso encontre dificuldade em interagir com as interfaces disponíveis. Esses dados reforçam, ainda mais, a importância da usabilidade no contexto das aplicações *Web* e a permanente necessidade de se desenvolver interfaces simples, intuitivas, eficientes e agradáveis, que possibilitem a realização de tarefas com sucesso e o cumprimento dos objetivos previamente estabelecidos pelos usuários.

As aplicações *Web*, na atualidade, são fundamentais não somente para os negócios, mas, também, para os governos, pois servem de meio para apresentar produtos e serviços a potenciais clientes e divulgar informações relevantes aos cidadãos, respectivamente. Desenvolver aplicações *Web* de melhor qualidade de uso é um desafio que tem motivado a criação de uma variedade de técnicas, métodos e ferramentas de apoio à usabilidade (INSFRAN; FERNANDEZ, 2008).

---

<sup>5</sup> A Associação Brasileira de Normas Técnicas, na Norma Brasileira 9126-1, apresenta os seguintes atributos de qualidade de *software*: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade (ABNT NBR, 2003).

Estas contribuições têm se tornado críticas para o processo de desenvolvimento de aplicações *Web*, tendo em vista que muitas dessas aplicações ainda não atendem às expectativas de usabilidade dos usuários (CYBIS; BETIOL; FAUST, 2010). Além disso, organizações têm falido por não considerar os problemas de usabilidade existentes nas interfaces de aplicações *Web* (NIELSEN; LORANGER, 2007). Apesar disso, há empresas que continuam sem utilizar recursos de suporte à usabilidade (INSFRAN; FERNANDEZ, 2008) seja por desconhecer, seja por não dispor de orçamento, seja por enfrentar dificuldades na seleção e implantação desses recursos (NIELSEN; LORANGER, 2007; CONTE, 2009; CYBIS; BETIOL; FAUST, 2010).

No estudo de mapeamento sistemático<sup>6</sup> realizado por Fernandez, Insfran e Abrahão (2011) acerca de abordagens de suporte à usabilidade para aplicações *Web*, foi possível observar que:

- Cerca de 90% dos estudos aplicam avaliações de usabilidade durante a fase de implementação das interfaces;
- Aproximadamente 68% dos estudos fornecem apenas relatórios sobre os problemas de usabilidade identificados e não apresentam orientações (sugestões de mudança) para as atividades de projeto subsequentes;
- Não há um único método adequado para todas as condições e tipos de aplicações *Web*, o que leva a sugerir o uso em conjunto de métodos para potencializar os resultados, a exemplo dos métodos de inspeção, teste de usabilidade com usuários e questionários de satisfação;
- Há uma preocupante escassez de propostas que aplicam recursos de usabilidade nas fases iniciais e intermediárias do processo de desenvolvimento de interfaces, a exemplo das fases de análise e de projeto;
- Há uma necessidade de aplicação integrada de métodos e recursos de usabilidade durante as diversas fases do processo de desenvolvimento de interfaces de aplicações *Web*.

---

<sup>6</sup> Estudo de mapeamento sistemático (também denominado de revisão sistemática) é útil para classificar e resumir, de maneira imparcial, informações relacionadas a uma determinada questão de pesquisa. Esses estudos são importantes para o progresso do conhecimento científico porque integram informações relativas a pesquisas realizadas separadamente e identificam temas que precisam de evidência, o que contribui para orientar novas investigações (SAMPAIO; MANCINI, 2007).

Na mesma direção do estudo de mapeamento sistemático realizado por Fernandez, Insfran e Abrahão (2011), foi possível perceber nos trabalhos relacionados a carência de soluções de suporte à usabilidade que: (i) contemple as diversas fases de um projeto de interfaces, a exemplo das fases de planejamento, desenvolvimento e avaliação; (ii) apresente um conjunto sucinto de atividades, recursos e artefatos para apoiar cada uma dessas fases. Estas lacunas e dificuldades contribuíram para motivar a realização desta pesquisa que consiste em propor um *framework* de apoio à Engenharia de Usabilidade para aplicações *Web*.

O *framework* em questão traz contribuições importantes para a Engenharia de *Software*, pois apresenta, por meio de um arcabouço de fases, atividades, recursos e artefatos, orientações metodológicas e práticas para a inserção facilitada e sistemática de recursos de suporte à usabilidade nas diversas etapas do processo de desenvolvimento de *software*, de modo a possibilitar a ampliação da qualidade de uso das interfaces desenvolvidas.

## 1.6 Metodologia

As pesquisas podem ser classificadas de diversas maneiras. Essa classificação, quando realizada de forma apropriada pelo pesquisador, traz benefícios à pesquisa porque confere maior racionalidade às etapas necessárias para a sua execução, o que vem resultar no desenvolvimento de trabalhos científicos mais rápidos, com maximização de recursos empregados e resultados mais satisfatórios (GIL, 2010). Na Figura 1.1, é apresentado o resumo do enquadramento metodológico desta pesquisa.

**Figura 1.1** – Enquadramento Metodológico da Pesquisa.

<b>Finalidade da Pesquisa</b>	• Aplicada
<b>Abordagem do Problema</b>	• Qualitativa
<b>Natureza do Objetivo</b>	• Exploratória
<b>Procedimentos Técnicos</b>	• Pesquisa Bibliográfica

Sob o ponto de vista de sua finalidade, esta pesquisa é classificada como pesquisa aplicada, pois objetiva gerar conhecimentos para aplicação prática dirigida à solução de problemas específicos (GIL, 2010). Quanto à forma de abordagem do problema, esta pesquisa é qualitativa, visto que se baseia na interpretação de resultados e na atribuição de significados descritivos (MIGUEL, 2007). Em relação à natureza do objetivo, esta pesquisa é classificada como exploratória, pois visa levantar informações sobre um determinado problema para melhor compreendê-lo, além de permitir a construção de hipóteses (GIL, 2010). Quanto aos procedimentos técnicos, esta pesquisa utiliza o método de pesquisa bibliográfica.

A pesquisa bibliográfica é realizada a partir da leitura de material publicado como livros, revistas, dissertações, teses, anais de eventos científicos, documentos disponíveis na Internet, entre outros. Basicamente, é realizada com o propósito de fundamentar teoricamente o trabalho e identificar o estágio atual de conhecimento acerca de um determinado tema em questão (SEVERINO, 2013). Esta pesquisa de mestrado foi desenvolvida de acordo com o roteiro de atividades apresentado a seguir:

- **Pesquisa Bibliográfica:** apresentação do referencial teórico relacionado à Engenharia de Usabilidade, modelo de processo, *framework*, princípios e diretrizes de projeto de interfaces para aplicações *Web*. Ademais, levantamento de trabalhos relacionados para observar o que já foi produzido e, com isso, identificar lacunas no suporte à usabilidade, no contexto de desenvolvimento de interfaces de aplicações *Web*;
- **Especificação do *Framework* UEF-WEB:** proposta de apoio à Engenharia de Usabilidade para auxiliar no processo de planejamento, desenvolvimento e avaliação de interfaces de aplicações *Web*. Este *framework* foi estruturado e desenvolvido a partir de contribuições e resultados obtidos na pesquisa bibliográfica conduzida acerca do objeto de estudo desta dissertação;
- **Geração do relatório:** escrita da dissertação com o propósito de descrever e documentar todas as etapas e resultados da pesquisa.

## 1.7 Organização da Pesquisa

Esta dissertação de mestrado está organizada em cinco capítulos, incluindo a introdução que apresenta a contextualização, o problema, a hipótese, os objetivos, os

trabalhos relacionados, a justificativa e a metodologia. Os tópicos a seguir apresentam o conteúdo dos demais capítulos, sendo eles:

- **Capítulo 2 – Engenharia de Usabilidade:** este capítulo apresenta os conceitos norteadores de usabilidade e de Interação Humano-Computador (IHC). Além disso, descreve os principais modelos de ciclos de vida, atividades e métodos da Engenharia de Usabilidade;
- **Capítulo 3 – Usabilidade em Aplicações *Web*:** neste capítulo, atributos, categorias, modelo de processo, *framework*, princípios e diretrizes para o desenvolvimento de interfaces de aplicações *Web* são apresentados. Além disso, é realizada uma discussão sobre a usabilidade em aplicações *Web* e sobre a importância da usabilidade para a qualidade das interfaces de aplicações *Web*;
- **Capítulo 4 – *Framework* UEF-WEB:** este capítulo discute e apresenta o *Framework* UEF-WEB, suas fases, atividades, recursos e artefatos de apoio à Engenharia de Usabilidade no processo de desenvolvimento de interfaces de aplicações *Web*;
- **Capítulo 5 – Considerações Finais:** neste capítulo, são apresentadas as considerações finais, as principais contribuições, limitações deste trabalho e as oportunidades de trabalhos futuros.

### ENGENHARIA DE USABILIDADE

Este capítulo apresenta a primeira parte do embasamento teórico da pesquisa. Muitos dos fundamentos discutidos aqui foram utilizados no desenvolvimento do *Framework* UEF-WEB proposto nesta dissertação, a exemplo dos métodos Avaliação Heurística, Inspeção de Usabilidade Baseada em Perspectivas, Teste de Usabilidade e Questionário de Satisfação. Inicialmente, são apresentados os conceitos de usabilidade, Interação Humano-Computador (IHC) e atributos de usabilidade. Em seguida, é realizada uma discussão acerca da Engenharia de Usabilidade, seus principais modelos de ciclo de vida, atividades e métodos.

#### 2.1 Usabilidade na Interação Humano-Computador (IHC)

A usabilidade representa um dos seis atributos de qualidade de *software* (ABNT NBR, 2003), sendo considerado um dos critérios mais importantes para aceitação de aplicações de *software* interativas, especialmente de aplicações *Web* (HITZ *et al.*, 2006). O termo usabilidade é definido pela norma ISO 9241-210 como a medida na qual um sistema, produto ou serviço pode ser utilizado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um dado contexto de uso (ISO, 2010).

A medida de eficácia está associada à exatidão e completude com que objetivos específicos são alcançados pelos usuários. A medida de eficiência diz respeito aos recursos consumidos em relação à exatidão e completude com que os usuários atingem os objetivos. A medida de satisfação mede a ausência de desconforto e a atitude positiva do usuário quanto ao uso de um produto ou serviço. O contexto de uso, por sua vez, envolve usuários, tarefas, equipamentos (*hardware*, *software* e materiais) e ambiente (ISO, 2010).

Para Cybis, Betiol e Faust (2010), a usabilidade é definida como um atributo de qualidade associado à facilidade e eficiência de uso, à capacidade de se lembrar e ao nível de satisfação em realizar algo. Uma definição alternativa para o termo usabilidade é a inexistência de obstáculos (problemas) que impeçam os usuários de realizar e completar com sucesso suas tarefas em um sistema (SKOV; STAGE, 2005).

Problemas de usabilidade são aspectos de um sistema que o torna ineficaz, ineficiente, desagradável, oneroso ou difícil de realizar tarefas e, com isso, permitir o alcance

dos objetivos definidos pelo usuário em uma situação típica de uso (MATERA *et al.*, 2006). Na prática, esses problemas se revelam na forma de interfaces altamente complexas, com inúmeros erros, sem consistência e padrões, com baixa flexibilidade e eficácia de uso (NIELSEN; LORANGER, 2007).

Quando as interfaces de um *software* ou de uma aplicação *Web* são mal projetadas e construídas, os usuários enfrentam dificuldades que impedem o uso correto de um sistema. Estas barreiras podem inclusive causar a rejeição a uma interface computacional (MENDES *et al.*, 2006). Entretanto, quando interfaces simples, eficientes e fáceis de usar são desenvolvidas, os usuários conseguem atingir seus objetivos com menos esforço, em menos tempo e com um número menor de erros. As organizações também são recompensadas quando investem recursos em usabilidade, pois conseguem aumentar a produtividade, diminuir o tempo de desenvolvimento, treinamento e suporte, reduzir custos e melhorar a sua imagem no mercado (CYBIS; BETIOL; FAUST, 2010).

De fato, não é uma tarefa fácil desenvolver interfaces ergonômicas, isto é, interfaces adaptadas e adequadas ao ser humano, pois as interfaces apresentam-se como sistemas abertos onde os usuários são agentes ativos e com comportamento não definido (CYBIS; BETIOL; FAUST, 2010). Diante dessas dificuldades, surgiu a área de conhecimento Interação Humano-Computador (IHC) que visa estudar, planejar e entender as formas de interação entre indivíduos e dispositivos computacionais, de modo que as necessidades dos indivíduos sejam atendidas em maior grau de efetividade (GALITZ, 2003). Em linhas gerais, essa efetividade é conseguida à medida que o usuário percebe o sistema e tem condições de se comunicar de maneira mais natural possível (NETO, 2013). A seguir, são enunciados os objetivos da IHC:

Produzir sistemas usáveis, seguros e funcionais. Esses objetivos podem ser resumidos como desenvolver ou melhorar a segurança, utilidade, efetividade e usabilidade de sistemas que incluem computadores. Nesse contexto, o termo sistemas se refere não somente ao hardware e ao software, mas a todo o ambiente que usa ou é afetado pelo uso da tecnologia computacional (ROCHA; BARANAUSKAS, 2003, p. 17).

A usabilidade na IHC pode ser entendida como o resultado de uma equação composta por elementos como a experiência do usuário com a tecnologia, a eficiência do *hardware*, a natureza da tarefa e o ambiente onde se desenvolve a tarefa (SCHNEIDER, 2008). Ainda, de acordo com Schneider (2008), a visão que o projetista de *software* possui sobre o problema a ser modelado muitas vezes não é a visão que o usuário tem para solucionar o mesmo problema, o que sem dúvida acaba dificultando o processo de interação entre o usuário e a interface do sistema. Esse descompasso de entendimento precisa ser

equacionado para que as concepções dos projetistas e dos usuários acerca da solução de um problema estejam devidamente alinhadas.

## 2.2 Atributos de Usabilidade

No decorrer dos anos, diversos atributos foram propostos por pesquisadores e organismos normativos como medidas de usabilidade. A norma ABNT NBR 9126-1 (ABNT NBR, 2003), por exemplo, apresenta os seguintes atributos de usabilidade:

- **Inteligibilidade:** a capacidade do produto de *software* de permitir ao usuário compreender se o *software* é adequado e como pode ser usado para realizar tarefas em determinadas condições de uso;
- **Apreensibilidade:** a capacidade do produto de *software* de possibilitar ao usuário aprender como utilizá-lo;
- **Operacionalidade:** a capacidade do produto de *software* de permitir ao usuário operá-lo e controlá-lo;
- **Atratividade:** a capacidade do produto de *software* de ser analisado como atrativo pelo usuário;
- **Conformidade:** a capacidade do produto de *software* de estar em conformidade com normas, convenções, guias de estilo ou regulamentações associadas à usabilidade.

Além dos atributos apresentados na Norma ABNT NBR 9126-1 (ABNT NBR, 2003), outros cinco componentes de qualidade de uso foram definidos por Nielsen (2003), a saber:

- **Facilidade de aprendizado:** o quão fácil é aprender a funcionalidade e o comportamento do sistema;
- **Eficiência de uso:** diz respeito ao nível de produtividade alcançado pelo usuário na realização de tarefas com o sistema;
- **Facilidade de memorização:** o quão fácil é lembrar a funcionalidade do sistema após algum tempo sem utilizá-lo;

- **Geração de poucos erros:** está relacionado à capacidade do sistema de apoiar os usuários nas tarefas de modo que poucos erros sejam cometidos e, quando cometidos, sejam brevemente corrigidos;
- **Satisfação dos usuários:** medida que avalia a percepção de satisfação dos usuários ao realizar tarefas em um sistema.

Esses são apenas alguns dos atributos de usabilidade existentes. Tais atributos são importantes porque representam um conjunto de qualidades necessárias às interfaces computacionais dos *softwares* (CYBIS; BETIOL; FAUST, 2010). Quando se avalia a usabilidade de uma aplicação *Web*, por exemplo, procura-se medir o valor alcançado pela aplicação *Web* em cada atributo definido como relevante para a aplicação em consideração.

Na próxima seção, será realizada uma discussão acerca da Engenharia de Usabilidade que tem fornecido modelos de ciclo de vida, atividades e métodos, no intuito de promover o atendimento aos atributos de usabilidade e, com isso, possibilitar o desenvolvimento de interfaces com melhor qualidade de uso.

## 2.3 Engenharia de Usabilidade

Diferentemente da Engenharia de *Software*<sup>7</sup> que direciona a atenção para a lógica de funcionamento do *software*, a Engenharia de Usabilidade está focada na interface do *software* com o usuário. Em suma, a Engenharia de Usabilidade é definida como o processo de construção de interfaces computacionais que visa promover facilidade de uso e de aprendizado e satisfação dos usuários ao interagir com as interfaces de um sistema (ROCHA; BARANAUSKAS, 2003).

O objetivo da Engenharia de Usabilidade é o de auxiliar as organizações na construção de *software* de maneira que o modelo mental do desenvolvedor ou *designer* esteja de acordo com o modelo mental do usuário (SCHNEIDER, 2008). Em outras palavras, as interfaces, isto é, meios de comunicação entre o usuário e o *software*, devem refletir o modo de pensar e agir dos usuários.

---

<sup>7</sup> Engenharia de *Software* é definida como uma abordagem sistemática e disciplinada para o desenvolvimento, operação e manutenção de *software*. A Engenharia de *Software* preocupa-se com o desenvolvimento do núcleo funcional de um *software*, que é constituído de algoritmos, estruturas dados e demais recursos computacionais, com o intuito de processar dados relacionados ao domínio de aplicação do *software*. Este núcleo é desenvolvido segundo uma lógica de funcionamento para que o *software* execute de forma correta, rápida e sem erros (PRESSMAN, 2011).

Estas interfaces, agora desenvolvidas sob a perspectiva de uma lógica de operação, são componentes de um sistema interativo de *software* constituído por estruturas de diálogos e apresentações, que fornecem informações e controles para o usuário realizar tarefas específicas com o sistema (ISO, 2010). É possível afirmar que um desenvolvedor tem mais chances de obter sucesso ao desenvolver o núcleo funcional de um *software* do que interfaces com o usuário. A explicação para isso reside no fato de que o núcleo funcional é um sistema fechado, com comportamento previamente estabelecido. As interfaces, por sua vez, correspondem a sistemas abertos utilizados por usuários ativos, com ações não determinísticas e em constante evolução (CYBIS; BETIOL; FAUST, 2010).

Diante das dificuldades que permeiam o processo de construção de interfaces, foram propostos nas últimas décadas modelos de ciclo de vida para a Engenharia de Usabilidade. Tais modelos possuem como característica básica a ênfase concedida ao usuário no processo de concepção e desenvolvimento das interfaces. Esses modelos fornecem atividades que ao serem realizadas contribuem para a criação de interfaces centradas no usuário e com maior qualidade de uso. Na próxima seção, serão apresentados os seguintes modelos de ciclo de vida: o Modelo Estrela, o Modelo de Ciclo de Vida da Engenharia de Usabilidade e o Modelo de Projeto Centrado no Usuário para Sistemas Interativos.

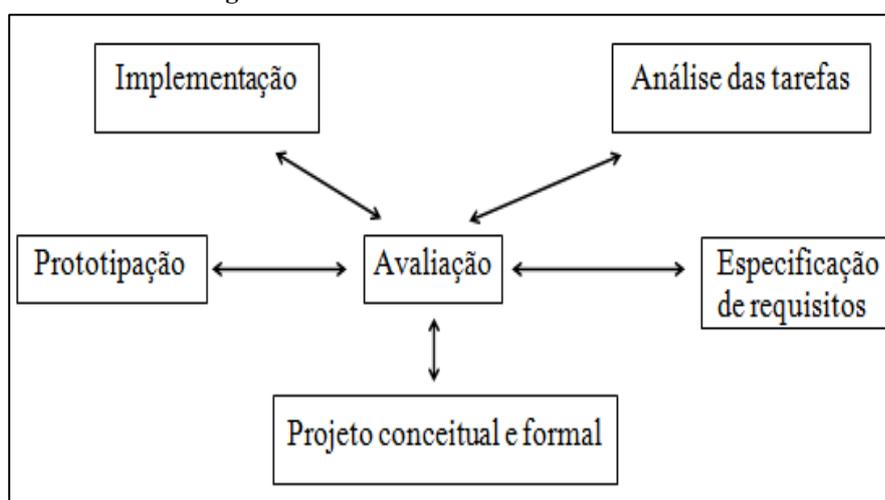
### **2.3.1 Modelos de Ciclo de Vida**

O modelo de Ciclo de Vida Estrela foi proposto em 1989 por Hartson e Hix, a partir de um trabalho empírico que estes autores desenvolveram através da observação de como *designers* de interfaces trabalhavam (PREECE; ROGERS; SHARP, 2005). O Modelo Estrela é composto basicamente por seis atividades, a saber: análise de tarefas, especificação de requisitos, projeto conceitual e formal, prototipação, implementação e avaliação.

A atividade de análise de tarefas é responsável pela análise da situação atual, pelo levantamento das necessidades e oportunidades de melhorias. A atividade de especificação de requisitos, além de coletar dados junto aos representantes dos usuários acerca das necessidades, define quais tarefas serão atendidas durante o projeto do sistema. Na atividade de projeto conceitual e formal, são exploradas alternativas de *design* para as interfaces do sistema a partir do envolvimento de usuários. Na atividade de prototipação, versões interativas de soluções de interfaces são elaboradas para serem avaliadas junto aos usuários. Os protótipos construídos nesta atividade auxiliam no entendimento do *design* das interfaces antes da implementação final do sistema. A atividade de implementação visa construir as

interfaces do sistema. A atividade de avaliação, por sua vez, objetiva analisar os resultados produzidos nas demais atividades desse modelo no intuito de identificar possíveis inconsistências (PREECE; ROGERS; SHARP, 2005). Na Figura 2.1, é apresentada a estrutura do Modelo de Ciclo de Vida Estrela e o relacionamento entre as suas atividades.

**Figura 2.1** – Modelo de Ciclo de Vida Estrela.



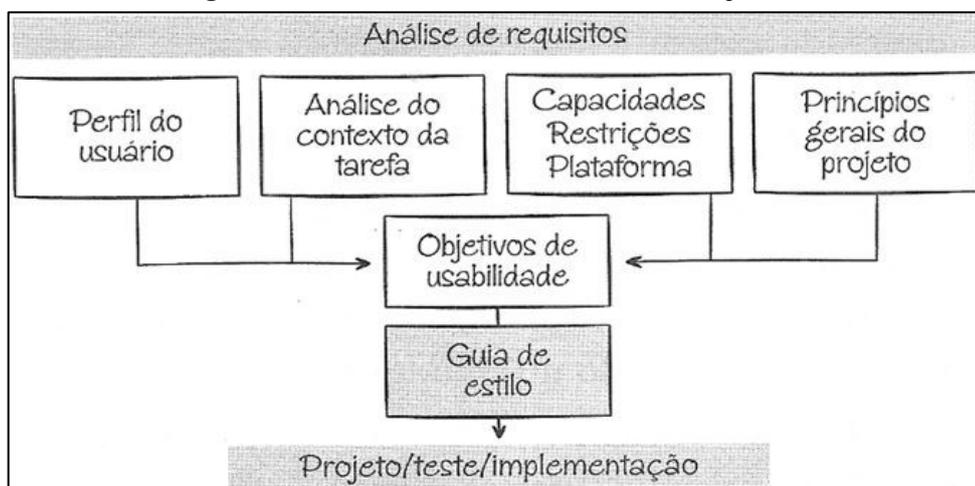
Fonte: (HARTSON; HIX, 1993).

Um fato importante a destacar é que este modelo não determina a ordem de execução das atividades, que são totalmente interconectadas. Isso significa que é possível ir de uma determinada atividade para qualquer outra. No entanto, para que isso aconteça deve-se passar primeiro pela atividade de avaliação. Esta atividade desempenha um papel central neste modelo, pois tem como objetivo avaliar os resultados produzidos nas outras atividades. Esta característica confere ao modelo flexibilidade na execução das atividades, mas apresenta algumas desvantagens como, por exemplo, dificuldades em rastrear o progresso do desenvolvimento do sistema e em identificar os recursos alocados (PREECE; ROGERS; SHARP, 2005).

O Modelo de Ciclo de Vida de Engenharia de Usabilidade proposto por Deborah Mayhew em 1999 apresenta uma abordagem que privilegia o papel do usuário na realização de tarefas, no intuito de construir sistemas adaptados às suas necessidades e objetivos. Esse modelo possui uma estrutura semelhante ao modelo proposto na norma ISO 13407, pois apresenta ciclos de atividades de análise, projeto, construção e testes. A diferença entre estas duas propostas é que o modelo de Mayhew fornece mais detalhes sobre o conteúdo das atividades em cada fase do processo de desenvolvimento, além de incluir a atividade de instalação do sistema (CYBIS; BETIOL; FAUST, 2010).

A fase de análise de requisitos é composta por quatro atividades cujos resultados são utilizados para especificar o contexto de uso e a usabilidade pretendida para o sistema. Estas atividades são apresentadas na Figura 2.2. Para cada tipo de usuário, é necessário coletar informações e conhecer suas habilidades e competências. Para cada tarefa, os projetistas precisam identificar, por exemplo, objetivos, resultados, estrutura, esforço e interrupções previstas. Restrições tecnológicas em termos de equipamentos, sistemas e recursos de rede devem ser observadas. E, para concluir, nesta fase são definidos princípios ergonômicos e de usabilidade a serem utilizados na concepção das interfaces do sistema (MAYHEW, 1999).

**Figura 2.2** – Atividades da Fase de Análise de Requisitos.

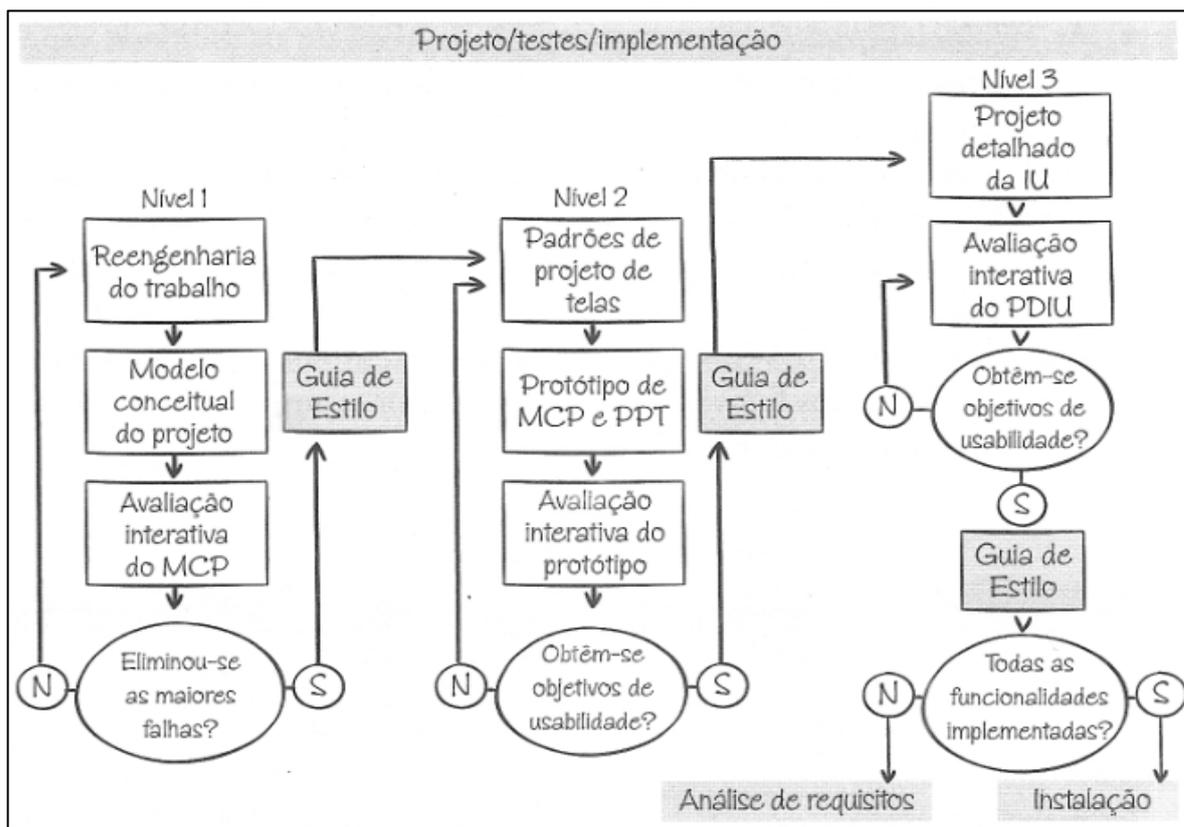


Fonte: Mayhew (1999).

Na fase de projeto, testes e implementação, ciclos de atividades são executados para tratar três níveis de aspectos das interfaces. No primeiro nível, é definido um Modelo Conceitual do Projeto (MCP) composto por telas e componentes associados. Uma avaliação interativa desse modelo também é realizada. No segundo nível, os ciclos de atividades se referem a definições em termos de estilo. Padrões de Projeto de Telas (PPT) são definidos e um protótipo de interface é construído a partir do MCP e do guia de estilo. Diversos testes são realizados nesse protótipo a fim de se obter uma versão de interface satisfatória. No terceiro nível, é realizado o Projeto Detalhado da Interface de Usuário (PDIU). Neste nível, uma versão da Interface de Usuário (IU) é construída e testada até ser considerada adequada nos testes realizados com os usuários (CYBIS; BETIOL; FAUST, 2010).

A depender do tamanho do projeto e da experiência da equipe de desenvolvimento, o resultado obtido no terceiro nível pode ser um protótipo detalhado ou uma versão final da interface desenvolvida para uma tarefa ou grupo de tarefas. Os três níveis de aspectos de interface discutidos nesse modelo são apresentados na Figura 2.3.

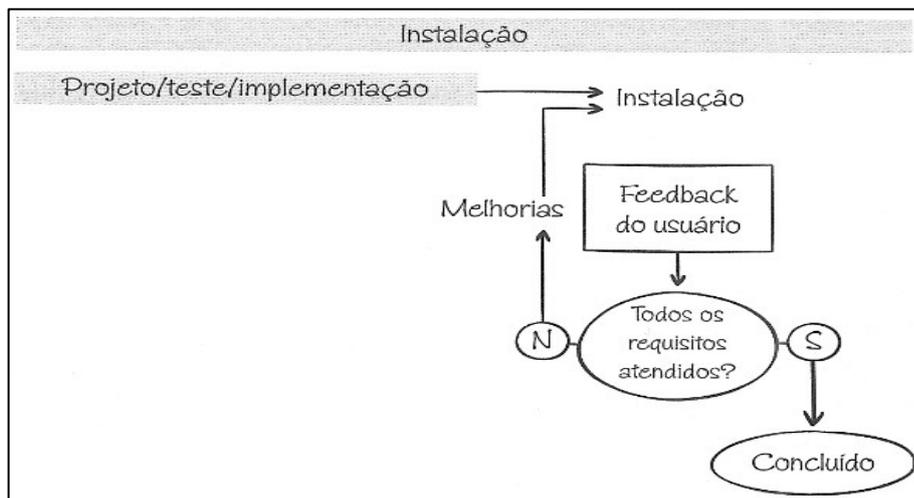
**Figura 2.3** – Atividades da Fase de Projeto, Testes e Implementação.



**Fonte:** Mayhew (1999).

A fase de instalação objetiva coletar *feedback* dos usuários sobre a usabilidade do sistema para: identificar e remover problemas do sistema (a partir de uma simples atualização); detectar e solucionar problemas mais abrangentes (através de uma nova versão); identificar oportunidades de melhorias a serem implantadas nas próximas versões do sistema. As atividades desta fase são apresentadas na Figura 2.4.

**Figura 2.4** – Atividades da Fase de Instalação.



**Fonte:** Mayhew (1999).

O Modelo de Ciclo de Vida de Engenharia de Usabilidade elaborado por Mayhew possui como características importantes: visão holística acerca da Engenharia de Usabilidade; envolvimento do usuário ao longo de todo o processo de desenvolvimento; ligação com abordagens de Engenharia de *Software*; etapas de identificação de requisitos, *design*, prototipação e avaliação; utilização de guias de estilo para capturar os objetivos de usabilidade do sistema. Este modelo, no entanto, não fornece maiores detalhes acerca de instrumentos e artefatos que podem ser empregados pelas organizações no processo de realização das atividades propostas.

O Modelo de Projeto Centrado no Ser Humano para Sistemas Interativos da norma ISO 9241-210 substitui o Modelo de Projeto Centrado no Usuário da norma ISO 13407 (ISO, 2010). O modelo da norma ISO 9241-210 fornece princípios e atividades de projeto centrado no ser humano para o ciclo de desenvolvimento de sistemas. Esta abordagem visa construir sistemas interativos mais utilizáveis pelo foco no uso do sistema e pela aplicação de conhecimentos e técnicas em fatores humanos e usabilidade. Independentemente do processo adotado, os seguintes princípios são recomendados (ISO, 2010):

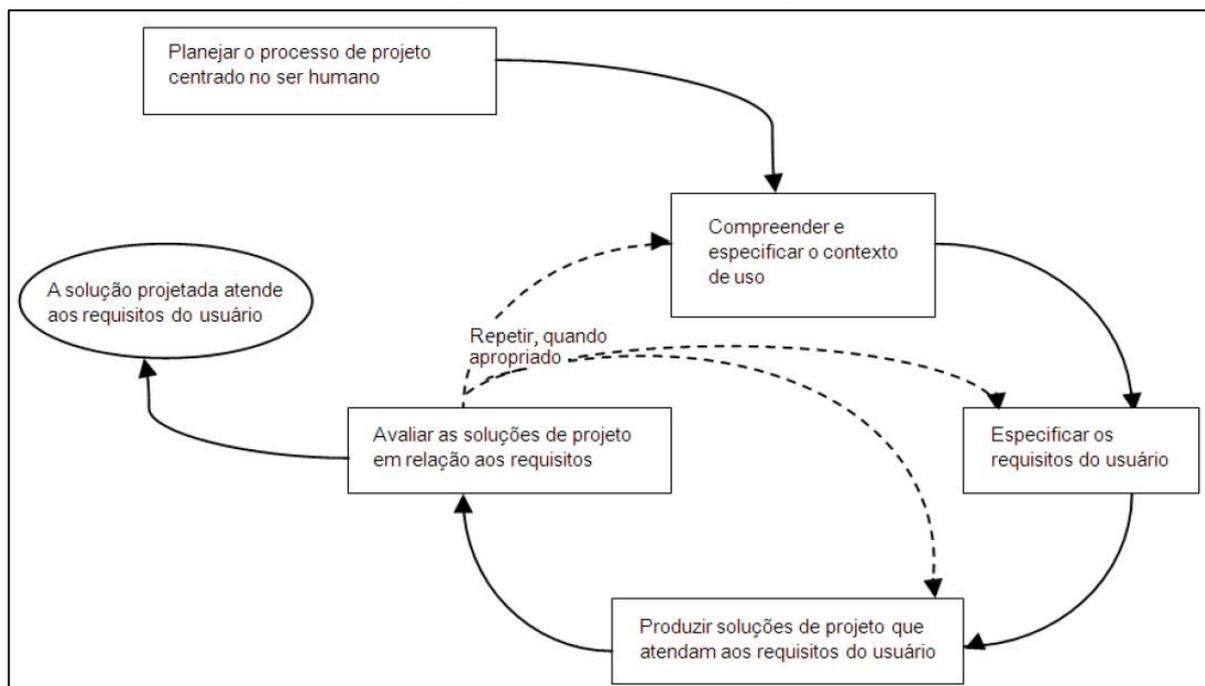
- **O projeto é baseado em um entendimento explícito de usuários, tarefas e ambientes:** produtos, sistemas e serviços devem ser projetados levando-se em consideração o conhecimento acerca dos usuários que deles farão uso, das tarefas que serão realizadas e do ambiente. Todo esse conhecimento constitui o contexto de uso, principal fonte de informação para especificar requisitos de interfaces e de usabilidade;
- **Os usuários são envolvidos em todo o projeto e desenvolvimento:** envolver os usuários no projeto e no desenvolvimento permite obter informações importantes sobre o contexto de uso. Essa participação deve ser ativa no projeto, ou seja, os usuários devem atuar fornecendo informações e também avaliando as soluções de interfaces desenvolvidas;
- **O projeto é conduzido e refinado por uma avaliação centrada no usuário:** o *feedback* do usuário constitui uma fonte de informação primordial no projeto centrado no ser humano. Avaliar projetos com o apoio do usuário e melhorá-los tendo por base esse *feedback* é um meio efetivo de se reduzir o risco de uma aplicação não atender as necessidades dos usuários e da organização;
- **O processo é iterativo:** a iteração nesse contexto significa repetir uma sequência de passos até que o resultado almejado seja alcançado. Em outras

palavras, as descrições, especificações e protótipos devem ser revisados e refinados assim que novas informações forem obtidas, no intuito de reduzir os riscos de o sistema falhar no atendimento aos requisitos dos usuários;

- **O projeto aborda a experiência do usuário como um todo:** o conceito de usabilidade não se resume à criação de produtos fáceis de usar. A experiência do usuário é mais ampla e decorre de fatores como, por exemplo, apresentação, funcionalidade, desempenho do sistema, comportamento da interação, experiências anteriores e habilidades. O projeto de um sistema deve considerar, quando apropriado, tais fatores entre outros;
- **A equipe de projeto inclui competências e perspectivas multidisciplinares:** recomenda-se que a expertise dos membros da equipe de projeto seja variada. Isso contribui para colaborar nas decisões e também na resolução de conflitos que surgem durante o projeto e implementação.

Além dos princípios apresentados, o modelo da norma ISO 9241-210 recomenda a realização de quatro atividades inter-relacionadas no decorrer do projeto de um sistema interativo, a saber: compreender e especificar o contexto de uso; especificar os requisitos do usuário; produzir soluções de projeto e avaliar o projeto. A estrutura desse modelo descrevendo o relacionamento entre as atividades é apresentada na Figura 2.5.

**Figura 2.5** – Modelo de Projeto Centrado no Ser Humano para Sistemas Interativos.



Fonte: (ISO, 2010).

As atividades do modelo da norma ISO 9241-210 consideram os seguintes desafios: a variedade de tipos de usuários e *stakeholders*<sup>8</sup> com necessidades peculiares; a diversidade de contexto de uso que pode variar entre os usuários e entre as tarefas; a dificuldade de definição dos requisitos e a dificuldade das soluções iniciais atenderem as necessidades dos usuários (ISO, 2010). A seguir, serão discutidas as quatro atividades desse modelo:

- **Compreender e especificar o contexto de uso:** as características dos usuários, das tarefas e dos ambientes organizacionais, técnico e físico estabelecem o contexto no qual o sistema é usado. É importante realizar a coleta e análise das informações sobre o contexto atual para compreender e especificar o contexto do futuro sistema. Observar sistemas existentes ou similares pode fornecer informações válidas de contexto como deficiências, níveis de referência de desempenho e satisfação. Em suma, a descrição do contexto de uso deve conter informações sobre os usuários e demais *stakeholders*, objetivos e tarefas dos usuários e ambiente do sistema;
- **Especificar os requisitos do usuário:** esta atividade objetiva estabelecer uma descrição explícita dos requisitos dos usuários acerca do contexto de uso pretendido e dos objetivos de negócio do sistema. Essa especificação inclui informações sobre as necessidades dos usuários e dos *stakeholders*, requisitos e objetivos de usabilidade. Potenciais conflitos entre os requisitos dos usuários devem ser resolvidos. Além disso, devem ser definidos meios para assegurar a qualidade das especificações realizadas;
- **Produzir soluções de projeto:** soluções de projeto possuem um impacto importante na experiência do usuário. Nesta atividade, recomenda-se a realização das seguintes subatividades: projetar tarefas do usuário, interação do usuário com o sistema e a interface com o usuário; tornar as soluções de projeto mais concretas (isso inclui o uso de protótipos); alterar as soluções de projeto em resposta a uma avaliação centrada no usuário e comunicá-las aos responsáveis por sua execução;
- **Avaliar o projeto:** a avaliação é uma atividade essencial no projeto centrado no ser humano, sendo utilizada para obter informações sobre novas

---

<sup>8</sup> *Stakeholders* são todos os indivíduos com algum interesse no sistema, afetando ou sendo afetados por seus resultados. Esse grupo de indivíduos vai além dos usuários de um sistema, pois envolve analistas, desenvolvedores, financiadores, entre outros (PRESSMAN, 2011). Os *stakeholders* normalmente possuem diferentes e, de forma frequente, inconsistentes necessidades em um sistema (SOMMERVILLE, 2007).

necessidades dos usuários, fornecer *feedback* sobre o projeto e avaliar se os requisitos foram atingidos. A avaliação é útil em todas as fases do projeto. Algumas abordagens podem ser utilizadas para isso, a exemplo de testes com usuários e avaliação baseada em inspeções.

Como visto na Figura 2.5, há uma interdependência entre as atividades deste modelo. Em outras palavras, cada atividade de projeto centrada no ser humano faz uso dos resultados produzidos nas atividades anteriores. Outro ponto positivo a destacar na norma ISO 9241-210 é que as atividades nela propostas podem ser incorporadas a diversas abordagens de projeto como, por exemplo, Orientação a Objetos, Cascata, HFI (Integração de Fatores Humanos) e Desenvolvimento Ágil (ISO, 2010).

Este modelo, em suma, traz importantes contribuições para a Engenharia de Usabilidade ao definir sólidos princípios e atividades de projeto centrado no ser humano para o processo de desenvolvimento. O objetivo principal deste modelo é fomentar a construção de sistemas interativos mais utilizáveis pelo foco no uso do sistema e pela aplicação de conhecimentos e técnicas em fatores humanos e usabilidade. Assim, como nos dois outros modelos apresentados, não são fornecidos maiores detalhes sobre instrumentos, recursos e artefatos a serem utilizados durante a execução das atividades estabelecidas. O modelo da norma ISO 9241-210, especificamente, sugere a adoção de mais normas no intuito de complementar as descrições fornecidas.

Os modelos de ciclo de vida apresentados fornecem orientações que colaboram para a obtenção de interfaces de melhor qualidade de uso, através da realização de um conjunto de atividades de projeto centradas no usuário, durante o ciclo de desenvolvimento de sistemas interativos de *software*. Tais modelos buscam, fundamentalmente, produzir soluções de interfaces de *software* mais consistentes e que respondam às expectativas dos usuários na realização de suas tarefas.

Além dos modelos de ciclo de vida, a Engenharia de Usabilidade conta com uma variedade de métodos, que apoiam a execução de atividades em diversas fases do processo de desenvolvimento de *software*. Os métodos de entrevista, reunião, prototipação, avaliação heurística, inspeção de usabilidade baseada em perspectivas, teste de usabilidade e questionário de satisfação são alguns desses métodos. Tais métodos são discutidos com mais detalhes na próxima seção, pois fazem parte da estrutura do *framework* desta dissertação.

## 2.3.2 Métodos

Os métodos desenvolvidos na Engenharia de Usabilidade podem auxiliar as organizações de forma efetiva na realização de variadas atividades como, por exemplo, levantamento e registro de informações, definição e especificação de requisitos, projeto e desenvolvimento de versões de interfaces, avaliação de soluções, entre outras.

No que diz respeito às avaliações de usabilidade, duas categorias de métodos são comumente empregadas para detectar problemas de usabilidade, a saber: (i) Inspeções de Usabilidade, onde inspetores examinam as interfaces a fim de identificar problemas e violações de princípios e diretrizes de usabilidade; (ii) Testes de Usabilidade, que correspondem a estratégias de avaliação baseadas na participação direta de usuários (NASCIMENTO; AMARAL, 2010).

Apesar de o teste de usabilidade ser considerado o método mais eficaz na avaliação de sistemas e protótipos sob a perspectiva dos usuários das aplicações, os custos envolvidos geralmente são elevados, pois envolve o tempo dos usuários e diversas vezes a utilização de laboratórios específicos de usabilidade. Outros aspectos dificultosos estão relacionados ao processo de seleção de uma amostra representativa de usuários e o treinamento destes para realizar os testes (MATERA *et al.*, 2006). Inspeções de usabilidade, por sua vez, são menos custosas do que os métodos de teste com usuários, tendo em vista que não são requeridos laboratórios e equipamentos especiais para a sua realização, apenas profissionais com um bom nível de conhecimento em usabilidade (VAZ *et al.*, 2008).

### 2.3.2.1 Entrevista e Reunião

As entrevistas são usadas pelos projetistas de *software* para coletar informações, preferências, experiências e motivações comportamentais dos atuais e também dos futuros usuários de sistemas que estão sendo concebidos ou em fase de avaliação. As entrevistas são mais flexíveis do que os questionários e são adequadas para estudos exploratórios. Podem ser estruturadas, semiestruturadas ou não estruturadas (GENA; WEIBELZAHN, 2007). O sucesso desse método depende de quão bem forem realizadas as etapas de planejamento, execução e geração de relatório (CYBIS; BETIOL; FAUST, 2010).

O planejamento das entrevistas inclui, em um primeiro momento, a identificação das necessidades das pessoas em termos de informações para o projeto. Após isso, deve ser preparado um roteiro contendo os tópicos e as questões a serem discutidas de maneira a obter

as informações requeridas. A estratégia de registro das respostas também precisa ser definida, podendo ser feita através da memória, anotações escritas ou gravações de áudio e vídeo. A estrutura do relatório e a forma de apresentação dos resultados da entrevista devem ser adequadas às necessidades das pessoas (NASCIMENTO; AMARAL, 2010).

As reuniões podem ser empregadas para obter informações acerca da finalidade de um sistema e seu contexto de uso, assim como sobre os objetivos de usabilidade. Esse método possibilita que os fatores relacionados ao uso de um sistema sejam identificados antes do início da fase de projeto. Outro benefício é a possibilidade de reunir os *stakeholders* a fim de criar uma visão comum para o projeto de desenvolvimento (USABILITYNET<sup>9</sup>, 2014).

### **2.3.2.2 Prototipação**

A Prototipação consiste na implementação de recursos de *software* que simulem as interfaces do sistema (GENA; WEIBELZAHN, 2007). Os protótipos podem ser de alta ou baixa fidelidade, a depender da similaridade com o sistema. De forma conceitual, os protótipos são utilizados pela equipe de desenvolvimento com o propósito de elucidar e especificar aspectos das interfaces, sendo descartados após serem avaliados pelos usuários.

Os protótipos possibilitam aos usuários não somente visualizar, mas também operar o futuro sistema no contexto de operação que foi especificado. Através desse método, os projetistas obtém *feedback* mais real sobre os recursos e interfaces de um *software*, o que contribui para identificar antecipadamente os problemas de usabilidade das interfaces em desenvolvimento (CYBIS; BETIOL; FAUST, 2010). Outro benefício conseguido a partir do uso de protótipos é a compreensão em detalhes das expectativas e das impressões dos usuários acerca do sistema (USABILITYNET, 2014).

### **2.3.2.3 Avaliação Heurística**

A avaliação de usabilidade é uma das atividades mais importantes no processo de desenvolvimento de sistemas interativos, pois possibilita que as interfaces do sistema sejam analisadas pelos *designers* quanto ao atendimento de requisitos dos usuários. Em linhas gerais, avaliações de usabilidade correspondem a processos que visam garantir, através da análise de interfaces, que um sistema ou aplicação disponha de desempenho adequado,

---

<sup>9</sup> USABILITYNET é um projeto da União Europeia (UE) que fornece recursos e *networking* para os profissionais de usabilidade (USABILITYNET, 2014).

satisfaça as expectativas dos usuários, bem como os requisitos levantados nas etapas iniciais do processo de desenvolvimento (NETO, 2013). Para Sharp, Rogers e Preece (2007), analisar a experiência dos usuários no processo de interação com as interfaces, avaliar a acessibilidade das funcionalidades disponíveis e encontrar problemas de *design* são objetivos básicos dessas avaliações.

Segundo Bertini, Gabrielli e Kimani (2006), problemas de *design* e de usabilidade identificados mais cedo são mais baratos e mais fáceis de serem solucionados do que os observados em uma etapa avançada do desenvolvimento de *software*. Diante disso, recomenda-se realizar avaliações periódicas de usabilidade no decorrer do processo de desenvolvimento cada vez que uma versão de interface estiver em construção, houver mudanças de requisitos do sistema ou for feita inclusão de funcionalidades.

Tipicamente, a maior parte das avaliações de usabilidade é realizada junto aos usuários finais da aplicação. Por meio deste modelo de avaliação, importantes parâmetros acerca da interação dos usuários com as interfaces são identificados, visto que experimentos são realizados no ambiente de uso das aplicações. Este modelo, no entanto, demanda mais tempo e recursos para sua execução (CYBIS; BETIOL; FAUST, 2010). Questionários também são utilizados para avaliar a usabilidade de interfaces, mas, neste caso, é importante prestar atenção na formulação das questões para que as respostas obtidas sejam relevantes ao objetivo da avaliação.

Embora os estudos sobre avaliações de interfaces continuem em aberto, há um consenso de que a avaliação heurística é um dos métodos mais barato, rápido e com obtenção de resultados relevantes (BONIFÁCIO *et al.*, 2010). Neste método, especialistas avaliam as interfaces e os diálogos do sistema a partir de um conjunto de regras gerais (padrões de qualidade) denominadas de heurísticas, que auxiliam na identificação de problemas de usabilidade (NASCIMENTO; AMARAL, 2010).

Uma avaliação heurística pode ser realizada em especificações de interfaces, protótipos ou em sistemas de *software* já desenvolvidos. Basicamente, um grupo de especialistas inspeciona as interfaces e analisa a sua adequação tendo por base um conjunto de heurísticas (NASCIMENTO; AMARAL, 2010). Dentre as heurísticas de usabilidade mais conhecidas, estão as dez heurísticas propostas por Nielsen (1994b), a partir de estudos sobre os problemas de usabilidade de interfaces (NIELSEN, 1994a). Tais heurísticas são apresentadas na Tabela 2.1.

**Tabela 2.1** – As Dez Heurísticas de Usabilidade de Nielsen.

Número	Heurística
1	<b>Visibilidade do estado do sistema:</b> o sistema deve sempre manter os usuários informados sobre o que está acontecendo, através de uma realimentação apropriada dentro de um tempo razoável.
2	<b>Concordância entre o sistema e o mundo real:</b> o sistema deve utilizar a linguagem do usuário, com palavras, frases e conceitos familiares ao usuário, mais do que termos orientados para o sistema. Seguir as convenções do mundo real, fazer a informação aparecer na ordem natural e lógica.
3	<b>Controle e liberdade ao usuário:</b> o sistema deve dar apoio a funções como <i>undo</i> e <i>redo</i> ou funções que permitam ao usuário utilizar “saídas de emergência” em caso de escolhas de funções erradas ou para sair de um estado não esperado.
4	<b>Consistência e padrões:</b> devem ser seguidas convenções da plataforma de desenvolvimento e padrões de interface normalmente aceitos. Usuários não devem ter que adivinhar se palavras, situações ou ações diferentes significam a mesma coisa.
5	<b>Prevenção de erros:</b> o sistema deve prevenir a ocorrência de erros na sua utilização. Melhor do que apresentar boas mensagens de erros, é ter um projeto cuidadoso que previne a ocorrência de um problema, em primeiro lugar.
6	<b>Reconhecer ao invés de lembrar:</b> tornar objetos, ações e opções visíveis, para que o usuário não tenha que lembrar de informações de uma parte do diálogo para outra. Instruções para uso do sistema devem estar visíveis ou facilmente recuperáveis, quando necessário.
7	<b>Flexibilidade e eficiência de uso:</b> aceleradores (abreviações, teclas de função...) podem tornar mais rápida a interação com o usuário. Permitir aos usuários customizar ações frequentes.
8	<b>Projeto minimalista e estético:</b> diálogos não devem conter informação irrelevante ou raramente necessária. Todas as unidades extras de informações em um diálogo competem com aquelas que são realmente relevantes e diminuem sua visibilidade relativa.
9	<b>Reconhecimento, diagnóstico e recuperação de erros:</b> mensagens de erros devem ser expressas em linguagem simples (sem códigos), indicando precisamente o problema e sugerindo construtivamente uma solução.
10	<b>Ajuda e Documentação:</b> as informações de ajuda e documentação devem ser fáceis de procurar, com foco na tarefa do usuário, listando passos concretos e sucintos que devem ser seguidos.

**Fonte:** Nielsen (1994b).

De acordo com Prates e Barbosa (2003), esse método de avaliação segue o seguinte roteiro de atividades: (i) realização de avaliações individuais (1 a 2 horas), nas quais os inspetores avaliam a conformidade das interfaces em relação a um conjunto de heurísticas, registram os problemas de usabilidade, analisam a gravidade destes e geram um relatório individual de avaliação; (ii) consolidação da avaliação dos especialistas, atividade desenvolvida através de uma reunião que busca gerar um relatório consolidado sobre os problemas encontrados; (iii) seleção dos problemas a serem corrigidos, que consiste na análise

de custo/benefício das correções para os problemas identificados. A severidade dos problemas de usabilidade encontrados é classificada por Nielsen (1994b) através da escala apresentada na Tabela 2.2.

**Tabela 2.2** – Escala de Severidade para os Problemas de Usabilidade.

Escala de Severidade	Descrição
0	Não considerado um problema de usabilidade.
1	Problema cosmético – consertar apenas se houver tempo disponível.
2	Problema leve – baixa prioridade para consertá-lo.
3	Problema grave – alta prioridade para consertá-lo.
4	Problema catastrófico – é imperativo consertá-lo.

**Fonte:** Nielsen (1994b).

A Avaliação Heurística é um dos principais métodos de inspeção de aplicações interativas. Este método é eficiente e possui uma boa relação custo/benefício (NASCIMENTO; AMARAL, 2010). No entanto, a sua eficiência é dependente da capacidade e da experiência dos avaliadores em identificar os problemas de usabilidade presentes nas interfaces (MATERA *et al.*, 2006).

#### **2.3.2.4 Inspeção de Usabilidade Baseada em Perspectivas**

Na inspeção de usabilidade, os avaliadores examinam aspectos de uma interface com o objetivo de encontrar problemas de usabilidade. Estes avaliadores podem ser consultores de desenvolvimento de *software*, especialistas em usabilidade, usuários finais, dentre outros (ROCHA; BARANAUSKAS, 2003). No método de Inspeção de Usabilidade Baseada em Perspectivas (*Perspective-based Usability Inspection*) proposto por Zhang *et al.* (1999), cada sessão de inspeção está focada em um conjunto de questões sob o ponto de vista de uma perspectiva de usabilidade (MAHRIN; STROOPER; CARRINGTON, 2009).

A ideia principal deste método é que, em decorrência do foco na perspectiva, cada sessão de inspeção pode identificar um maior número de problemas associados à perspectiva.

Além disso, a combinação de perspectivas pode ajudar na descoberta de mais problemas que a combinação do mesmo número de sessões em uma técnica de inspeção geral (ZHANG *et al.*, 1999). As três perspectivas definidas por Zhang *et al.* (1999) são: uso por novato, uso por especialista e tratamento de erros.

A perspectiva uso por novato diz que o conhecimento e a experiência dos usuários não indicam a forma de utilizar o sistema ou atingir sua meta. A perspectiva uso por especialista indica que os usuários sabem usar o sistema, mas preferem alcançar a meta de maneira eficiente e com facilidade, ou mesmo atingir metas maiores. A perspectiva tratamento de erros, por sua vez, traz a ideia de que os usuários se deparam com um problema gerado por uma ação anterior e precisam dirimir este problema (ZHANG *et al.*, 1999). Para cada uma dessas três perspectivas é apresentado um procedimento de inspeção com o propósito de ajudar os inspetores a organizar o processo de inspeção e, com isso, fazer com que as questões de usabilidade sejam corretamente analisadas e no momento apropriado.

O *Framework* UEF-WEB proposto nesta dissertação faz uso deste método através da técnica de inspeção desenvolvida por Conte (2009). Esta técnica, denominada de WDP (*Web Design Perspectives-Based Usability Evaluation*), combina três perspectivas de projeto *Web* (apresentação, conceituação e navegação) com as dez heurísticas de usabilidade definidas por Nielsen (1994b). A técnica WDP será apresentada com mais detalhes na seção 4.2.2.

### **2.3.2.5 Teste de Usabilidade**

Os testes de usabilidade possuem como foco de avaliação a qualidade das interações realizadas entre usuários e o sistema. Os objetivos deste método são identificar problemas de usabilidade, medir os impactos destes e descobrir suas causas na interface (NASCIMENTO; AMARAL, 2010). Um teste de usabilidade pode envolver usuários reais ou representativos do público alvo do sistema para realizar tarefas específicas em um contexto de utilização real ou simulado. Este tipo de teste se distingue de uma observação de interação, pois não possui como finalidade específica conhecer uma determinada situação ou contexto de trabalho. Diferentemente disso, visa testar um sistema em uma situação real ou com maior fidelidade possível.

O nível de dificuldade inerente à realização de um teste de usabilidade depende do grau de exigência necessário para os resultados, da amplitude do produto de *software* avaliado e da disponibilidade de recursos e de usuários. Geralmente, testes simples conduzidos para conhecer medidas de usabilidade de protótipos de produtos especializados (com tarefas

específicas) e com acesso facilitado aos usuários são realizados de forma rápida. No entanto, quando se objetiva ter resultados refletindo o comportamento geral de um público-alvo sobre um produto de *software* com funções gerais e abrangentes, os testes de usabilidade tornam-se mais custosos e difíceis de realizar. Mais recursos, usuários e procedimentos de análise também serão exigidos (CYBIS; BETIOL; FAUST, 2010).

Os testes de usabilidade podem ser realizados no próprio local de trabalho dos usuários ou em um laboratório (ambiente controlado). Independentemente do local de realização, é necessário definir um roteiro de tarefas. Este roteiro estabelece o conjunto de tarefas que os usuários irão realizar durante os testes. Na definição deste roteiro, é importante selecionar as tarefas relacionadas com (CYBIS; BETIOL; FAUST, 2010):

- Os objetivos fundamentais do *software* na visão dos projetistas;
- As funções do sistema entendidas como mais importantes pelos usuários, levando-se em consideração o impacto sobre o seu trabalho;
- As funções mais frequentemente utilizadas pelos usuários quando da interação com o *software*;
- As funções relacionadas com hipóteses acerca de problemas de ergonomia que causem problemas de usabilidade.

Basicamente, os roteiros de tarefas são desenvolvidos a partir da combinação desses parâmetros e através da análise de custo/benefício dos testes. Sendo assim, um teste de usabilidade deve focar principalmente os aspectos críticos de um *software*, sob o ponto de vista dos usuários e de suas tarefas.

### **2.3.2.6 Questionário de Satisfação**

Os questionários de satisfação são utilizados para coletar e avaliar a opinião dos usuários acerca da interação com as interfaces de um *software*. A associação deste método com testes de usabilidade é interessante porque permite relacionar as medidas de desempenho às medidas de satisfação dos usuários (CYBIS; BETIOL; FAUST, 2010). Este método pode ser aplicado nas fases finais ou após a conclusão do projeto de *software*. É um método simples de aplicar, eficiente, de baixo custo e, quando aplicado através da Internet (questionário *online*), possibilita o alcance simultâneo de diversos usuários dispersos geograficamente (DIAS, 2003). Ao longo dos anos, foram propostos diversos questionários

para avaliar a usabilidade e a satisfação dos usuários. Entre os mais conhecidos e utilizados estão os seguintes:

- **SUMI (*Software Usability Measurement Inventory*)**: questionário de medição da qualidade de um *software* sob o ponto de vista do usuário. Este questionário é mantido pelo Grupo de Pesquisa em Fatores Humanos da Universidade College Cork, sendo constituído por 5 aspectos principais (eficiência, empatia, controle, ajuda e aprendizagem) e 50 questões com três níveis de resposta (concordo, indeciso e não concordo) (SUMI, 2014);
- **WAMMI (*Web Analysis and Measurement Inventory*)**: desenvolvido por Jurek Kirakowski e Nigel Claridge (USABILITYNET, 2014), este questionário é composto por 20 questões sobre os seguintes fatores de usabilidade: atratividade, controlabilidade, eficiência, suporte ao usuário e facilidade de aprendizado. O questionário WAMMI faz uso de uma escala de cinco pontos, que varia de concordo completamente a discordo completamente, para avaliar cada uma das questões (WAMMI, 2014);
- **QUIS (*Questionnaire for User Interaction Satisfaction*)**: questionário desenvolvido por pesquisadores do Laboratório de Interação Humano-Computador (HCIL) da Universidade de Maryland em College Park, com o propósito de medir a satisfação dos usuários quanto à usabilidade. O questionário QUIS (versão 7.0) está organizado em 9 fatores específicos de interface (telas, terminologia e informações do sistema, aprendizagem do sistema, capacidades do sistema, manuais técnicos, tutoriais *online*, multimídia, teleconferência e instalação do *software*) e utiliza escala de nove pontos para avaliar as questões (QUIS, 2014);
- **ISONORM**: este questionário tem por objetivo avaliar a conformidade dos produtos de *software* em relação às recomendações presentes na parte 10 da norma ISO 9241 (CYBIS; BETIOL; FAUST, 2010). O questionário ISONORM está organizado em 7 seções que refletem os princípios de diálogo desta norma, a saber: adequação à tarefa, autodescrição, controlabilidade, conformidade com as expectativas dos usuários, tolerância a erros, adequação à individualização e adequação ao aprendizado;

- **ErgoList:** questionário de inspeção baseado em *checklist* desenvolvido pelo LabIUtil (Laboratório de Utilizabilidade da Informática da Universidade Federal de Santa Catarina). Este questionário de inspeção de usabilidade é composto por 194 questões e se baseia nos 18 critérios ergonômicos propostos por Bastien e Scapin (1993);
- **SUS (*System Usability Scale*):** desenvolvido por John Brooke em 1986 (USABILITYNET, 2014) para avaliar a usabilidade e a satisfação dos usuários em relação a sistemas e produtos. O questionário SUS é composto por 10 questões e utiliza a escala *Likert* de cinco pontos para medir a opinião dos usuários. Esta escala de avaliação está entre 1 (discordo completamente), 2 (discordo), 3 (neutro), 4 (concordo) e 5 (concordo completamente).

Os questionários SUMI, WAMMI e QUIS são recomendados por apresentar resultados confiáveis e por fornecer diversos recursos como, por exemplo, gráficos e relatórios. No entanto, estes questionários não são disponibilizados de forma gratuita. O questionário ISONORM é gratuito, mas o seu uso está mais relacionado ao processo de avaliação de usabilidade. O questionário ErgoList também é gratuito, mas o elevado número de critérios ergonômicos e de questões presentes (são 18 critérios e 194 questões) dificulta e encarece o processo de inspeção de usabilidade. O questionário SUS, além de ser gratuito e destinado a avaliar tanto a usabilidade quanto a satisfação dos usuários acerca de sistemas e produtos, é simples e sucinto (possui apenas 10 questões). Mais detalhes sobre o questionário SUS são fornecidos na subseção 4.2.3, pois este questionário faz parte da estrutura do *Framework* UEF-WEB.

## 2.4 Considerações Finais do Capítulo

Neste capítulo, foram apresentados os principais conceitos relacionados à usabilidade, considerada um dos critérios mais importantes para a aceitação de aplicações de *software* interativas (HITZ *et al.*, 2006) e um dos três critérios de qualidade que dirigem o processo de desenvolvimento em organizações produtoras de aplicações *Web* (OFFUTT, 2002). Este capítulo também apresentou uma discussão sobre a Engenharia de Usabilidade, seus modelos de ciclo de vida, atividades e alguns métodos existentes. Estes métodos foram apresentados por fazer parte da estrutura do *framework* proposto nesta dissertação.

Os modelos de ciclo de vida discutidos trazem, a partir de um conjunto de princípios e atividades de projeto centrado no usuário, orientações que ajudam na obtenção de interfaces de melhor qualidade de uso. Tais modelos, em suma, procuram suportar o desenvolvimento de soluções de interfaces de *software* mais consistentes, que respondam positivamente às expectativas dos usuários na execução de suas tarefas. No entanto, esses modelos não fornecem maiores informações sobre instrumentos e artefatos de apoio às atividades neles previstas o que, na prática, dificulta a realização dessas atividades e, por conseguinte, o alcance dos objetivos estabelecidos.

Os métodos de Entrevista e Reunião correspondem a técnicas de análise contextual simples, de fácil execução e que possibilitam a coleta rápida e eficiente de informações. A Prototipação possibilita a obtenção de *feedback* mais fidedigno sobre os recursos de um *software*, contribuindo para identificar de forma antecipada os problemas de usabilidade das interfaces em desenvolvimento. A Avaliação Heurística é um dos métodos de avaliação de usabilidade mais utilizados, eficiente e com boa relação custo/benefício. O método de Inspeção de Usabilidade Baseada em Perspectivas fornece perspectivas específicas de projeto *Web* que orientam o processo de inspeção de usabilidade. Os Testes de Usabilidade ajudam na avaliação da qualidade das interações entre usuários e o sistema. O método Questionário de Satisfação, por sua vez, auxilia no processo de avaliação da satisfação dos usuários acerca das interfaces e da interação com o sistema.

No próximo capítulo, será realizada uma discussão sobre a usabilidade em aplicações *Web*. Além de discutir conceitos, atributos, modelo de processo e *framework*, serão apresentados princípios e diretrizes de projeto de interfaces para aplicações *Web*.

### USABILIDADE EM APLICAÇÕES *WEB*

Neste capítulo, os atributos, categorias, modelo de processo, *framework*, princípios e diretrizes de projeto de interfaces para aplicações *Web* são apresentados. Este capítulo também realiza uma discussão sobre a usabilidade em aplicações *Web* e sobre a importância da usabilidade para a qualidade final dos *softwares* e das aplicações *Web*.

#### 3.1 Aplicações *Web*: conceito, atributos e categorias

Nos primeiros anos da *World Wide Web* (entre 1990 e 1995), os *sites* consistiam basicamente em um conjunto de páginas interligadas, que possibilitavam exibir informações textuais e alguns recursos gráficos de forma limitada. Com o passar do tempo e evolução das linguagens, ferramentas e tecnologias de desenvolvimento de *software*, foi possível agregar mais capacidade e diversidade de recursos computacionais aos *sites*, possibilitando o desenvolvimento de sistemas e aplicações baseadas na *Web*, também denominadas de *Webapps* (PRESSMAN, 2011).

Na última década, as aplicações *Web* evoluíram de tal maneira que passaram a fornecer, em um período de tempo cada vez menor, um conjunto diversificado de recursos e serviços, que vai desde funções de caráter mais geral como portais de notícias até funções especializadas como aplicações corporativas (SOMMERVILLE, 2007). Mais do que apoiar atividades, as aplicações *Web* estão colaborando para mudar a forma como as pessoas e organizações realizam compras (comércio eletrônico), buscam e fornecem informações (portais de notícias e governo eletrônico), comunicam-se (*email* e *chats*), emitem opiniões (*blogs* e redes sociais) e estudam (educação à distância) (PRESSMAN; LOWE, 2009).

O termo aplicação *Web* é definido por Kappel *et al.* (2006) como um sistema de *software* baseado em tecnologias e padrões do *World Wide Web Consortium* (W3C) que fornece recursos específicos da *Web* tais como conteúdo e serviços, por meio de uma interface de usuário denominada *browser Web*. Segundo Pressman (2011), as aplicações *Web* correspondem a uma das diferentes categorias de *software* listadas a seguir: *software* de sistema, *software* de aplicação, *software* científico/de engenharia, *software* embarcado, *software* para linha de produto, aplicações *Web* e *software* de inteligência artificial.

Na sua forma mais simples, as aplicações *Web* aparecem como pequenos *sites* que disponibilizam algumas páginas predominantemente informativas. Em outros casos, configuram-se como sofisticados ambientes de computação que integram muitos recursos computacionais e bancos de dados dispersos geograficamente (SOMMERVILLE, 2007). As aplicações *Web* reúnem atributos que vão além das características básicas de um *software*. Esses atributos são apresentados a seguir (PRESSMAN, 2011):

- **Uso intensivo de redes:** a rede, além fornecer a infraestrutura logística da aplicação *Web*, permite o acesso e comunicação dos usuários através de diversos meios (Internet, Extranet ou Intranet);
- **Simultaneidade:** uma grande quantidade de usuários pode utilizar uma aplicação *Web* em um determinado momento;
- **Carga não previsível:** a quantidade de usuários que acessa uma aplicação *Web* varia no decorrer do tempo. Em um dado momento, uma centena de usuários pode utilizar; em outro, algumas dezenas de milhares;
- **Desempenho:** as aplicações *Web* devem garantir um desempenho satisfatório. Caso contrário, os usuários poderão deixar de utilizar a aplicação com baixo desempenho;
- **Disponibilidade:** usuários de grandes aplicações populares da *Web*, por exemplo, contam em ter acesso a *sites* e aplicações 24 horas por dia, 7 dias por semana e 365 dias por ano (também conhecido por 24x7x365);
- **Orientada a dados:** aplicações *Web* têm como mídias principais texto, imagens, gráficos, áudio, vídeo e informações de bancos de dados;
- **Sensibilidade no conteúdo:** a qualidade e a natureza estética (de apresentação) do conteúdo são fatores relevantes para determinar o sucesso ou fracasso de uma aplicação *Web*;
- **Evolução contínua:** contrastando com os *softwares* convencionais (*software* de prateleira, por exemplo) que evoluem de forma planejada através de uma série de versões no decorrer dos anos, as aplicações *Web* evoluem rápida e constantemente;
- **Imediatismo:** a necessidade de colocar rapidamente no mercado um *software* é ainda maior quando se trata de aplicações *Web*. Estas podem, dependendo do contexto de negócio, ser disponibilizadas em semanas ou dias;

- **Segurança:** devido ao fato de estarem disponíveis através da Internet, na maioria das vezes, as aplicações *Web* (o que inclui a infraestrutura que as suporta) precisam ser projetadas e desenvolvidas sob fortes mecanismos de segurança;
- **Estética:** a aparência (forma de apresentação) de uma aplicação *Web* é parte tão importante quanto o projeto técnico quando, por exemplo, uma aplicação é voltada para o mercado de vendas de produtos e serviços.

Além desses atributos, as aplicações *Web* podem reunir características que as enquadram em diversas categorias (PRESSMAN; LOWE, 2009). Tais categorias são apresentadas no quadro 3.1.

**Quadro 3.1** – Categorias de Aplicações *Web*.

<b>Categoria</b>	<b>Descrição</b>
Informativa	Quando apresenta conteúdo somente de leitura, com navegação e <i>links</i> simples.
De Baixa ( <i>download</i> )	Quando incorpora a capacidade informativa e de <i>download</i> de conteúdos.
Personalizável	Quando o conteúdo disponibilizado é adaptado às necessidades específicas dos usuários.
Interação	Quando componentes de interação são fornecidos como meio de comunicação entre os usuários.
Entrada de Usuário	Quando mecanismos de coleta de dados como, por exemplo, formulários são disponibilizados para os usuários.
Orientada a Transação	Quando respostas automáticas para as entradas de dados dos usuários são fornecidas (a exemplo de uma transação).
Orientada a Serviço	Quando uma capacidade mais abrangente de assistência é apresentada.
Portais	Quando a aplicação <i>Web</i> recebe milhares de visitantes por dia e passa a dispor de uma grande variedade de fontes de informações.
Acesso a Banco de Dados	Quando bancos de dados são consultados para responder as entradas de dados fornecidas pelos usuários.
Armazém de Dados	Quando um componente de armazém de dados de grande escala é construído para armazenar e extrair informações.

**Fonte:** Pressman e Lowe (2009).

Além dos atributos e categorias apresentados, é importante acrescentar que três aspectos estão incorporados na maioria dos projetos de desenvolvimento de aplicações *Web* (PRESSMAN; LOWE, 2009), a saber:

- **Os requisitos evoluem com o tempo:** quando se inicia um projeto de aplicação *Web*, é normal existir incertezas sobre certos elementos da estratégia de negócios, do conteúdo, da funcionalidade, questões de interoperabilidade, entre outros aspectos do problema;
- **As mudanças ocorrem com frequência:** sabendo-se que a incerteza se faz presente na maior parte dos projetos de aplicações *Web*, as mudanças nos requisitos, como consequência, tornam-se comuns. Ademais, o *feedback* do usuário obtido através de avaliações de incrementos já entregues e alterações na lógica de negócio também podem suscitar mudanças;
- **As linhas de tempo são curtas:** esse aspecto contribui para aliviar a criação de documentação em grande volume. No entanto, isso não impede que a análise do problema, o projeto e os testes sejam documentados de alguma forma.

Diante dessa realidade, as aplicações *Web* normalmente são construídas e entregues de forma incremental, isto é, ao longo de sucessivos ciclos de atividades. Em cada um desses ciclos são desenvolvidos incrementos<sup>10</sup> da aplicação. O arcabouço de processo WebE discutido na próxima seção é um exemplo de modelo de processo para a Engenharia *Web* que utiliza essa estratégia.

### 3.2 WebE: Um Arcabouço de Processo para a Engenharia *Web*

Após discutir o conceito, atributos e categorias de aplicações *Web*, será apresentado, nesta seção, um arcabouço de processo para a Engenharia *Web* denominado WebE. Para Pressman e Lowe (2009, p. 11), “a Engenharia *Web* propõe um arcabouço ágil, porém disciplinado, para a montagem de Webapps de qualidade industrial”. Nesta perspectiva, um arcabouço define o alicerce para um processo completo de Engenharia *Web*, identificando um conjunto sucinto de atividades que são aplicadas a todos os projetos de Webapps,

---

<sup>10</sup> Incrementos: um incremento de uma aplicação *Web* entrega conteúdo e funcionalidade selecionados ao usuário final. Outros incrementos podem expandir o conteúdo e a funcionalidade já entregues até que a aplicação *Web* completa seja implantada. Por fim, o número de incrementos pode ser alterado à medida que o *feedback* dos incrementos entregues for sendo reportado pelos usuários (PRESSMAN; LOWE, 2009).

independente de seu tamanho e complexidade. Um arcabouço também engloba um grupo de atividades “guarda-chuva” aplicadas ao processo de Engenharia *Web* inteiro.

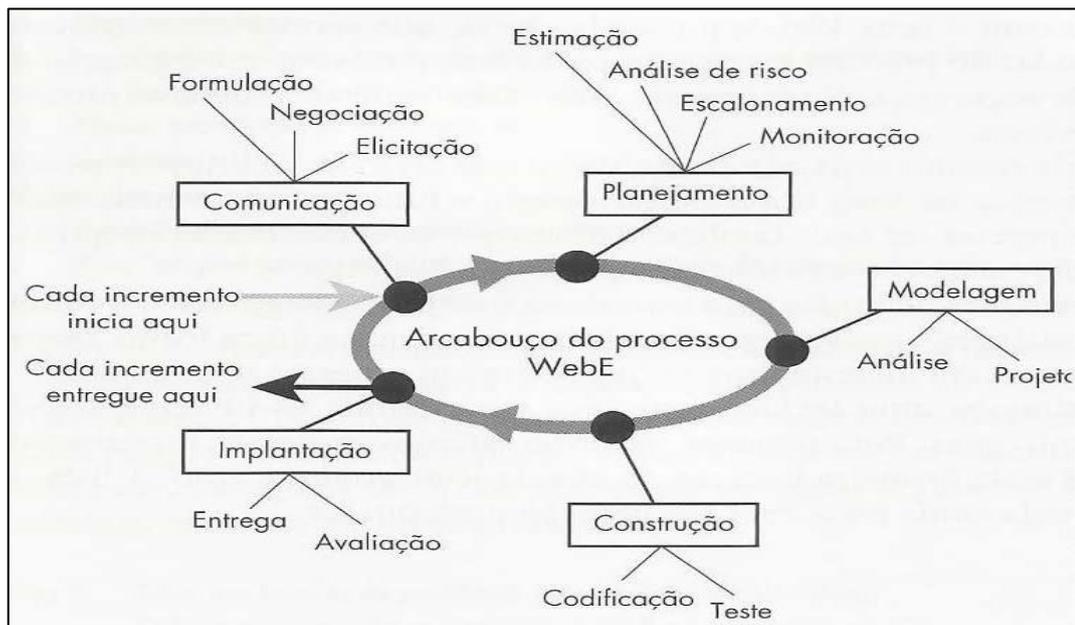
Sobre a importância da agilidade no processo de desenvolvimento de aplicações *Web*, é importante destacar que, nas organizações, as estratégias e regras de negócios mudam rapidamente. Os *stakeholders* também mudam de ideia mesmo quando solicitam entregas de incrementos de *software* em curtos prazos. Uma discussão útil acerca do conceito de agilidade é apresentada por Jacobson (2002, p. 18):

Uma equipe ágil é uma equipe leve, capaz de responder apropriadamente a mudanças. A mudança é tudo o que o desenvolvimento de software é. Mudanças no software sendo construído, mudanças nos membros da equipe, mudanças por causa de novas tecnologias, mudanças de todos os tipos que podem ter um impacto sobre o produto que eles constroem ou sobre o projeto que cria o produto. O apoio a mudanças deve estar embutido em tudo o que fazemos no software, algo que aceitamos porque isso é o coração do software. Uma equipe ágil reconhece que o software é desenvolvido por indivíduos trabalhando em equipes e que as habilidades dessas pessoas (e sua capacidade de colaborar) são essenciais para o sucesso do projeto.

No entendimento de Jacobson, a difusão da mudança é o elemento determinante da agilidade. Sendo assim, os engenheiros *Web* precisam ser ágeis para acomodar as rápidas mudanças inerentes ao desenvolvimento de aplicações *Web*. De acordo com Pressman e Lowe (2009), a implantação inteligente de qualquer arcabouço precisa reconhecer que a adaptação ao problema, ao projeto, à equipe e à cultura da organização, por exemplo, é fundamental para o sucesso. Essa adaptação influencia cada uma das seguintes características do arcabouço: fluxo geral de atividades, ações e tarefas e as interdependências entre elas; grau com o qual as tarefas de trabalho são definidas dentro de cada atividade do arcabouço; grau com o qual os produtos de trabalho são identificados e exigidos; maneira como as atividades de garantia da qualidade são aplicadas; forma como as atividades de acompanhamento e controle de projeto são estabelecidas; grau geral de detalhe e rigor com o qual o processo é descrito; grau com o qual os clientes e outros envolvidos estão comprometidos com o projeto; nível de autonomia dado à equipe de projeto de *software*; grau com o qual a organização e os papéis da equipe são prescritos.

O arcabouço de processo WebE proposto por Pressman e Lowe (2009) é composto por cinco atividades, a saber: Comunicação, Planejamento, Modelagem, Construção e Implantação. Cada uma dessas atividades é preenchida por um grupo de ações de Engenharia *Web*, isto é, um conjunto de tarefas relacionadas que produz um produto de trabalho. Por sua vez, cada ação é preenchida por tarefas de trabalho individuais que executam alguma parcela de trabalho relativa à ação. Na Figura 3.1, é apresentado o arcabouço de processo WebE.

**Figura 3.1** – Arcabouço de Processo WebE.



**Fonte:** Pressman e Lowe (2009).

A atividade de Comunicação é o ponto de partida para o fluxo de processo do arcabouço WebE. Nesta atividade, os Engenheiros *Web* e os *stakeholders* preocupam-se com três ações: (i) perguntam e respondem a questões relevantes acerca da aplicação *Web* e o seu contexto de negócio; (ii) elicitam requisitos elementares para as atividades posteriores; (iii) negociam as necessidades relativas a tempo, recursos e aspectos tecnológicos.

A ação de Formulacão tem por objetivos identificar as necessidades de negócio, descrever os objetivos e definir as principais características da aplicação *Web*. Com isso, espera-se que os *stakeholders* e a equipe de Engenharia *Web* consigam estabelecer um conjunto comum de metas e objetivos para o desenvolvimento de cada um dos incrementos da aplicação *Web*. A ação de Elicitacão é responsável por identificar um conjunto de requisitos que englobe não somente as funções da aplicação *Web*, como também conteúdo, interacão com usuários e interoperabilidade com sistemas de negócios e bancos de dados já existentes. A ação de Negociação, por outro lado, tenta resolver os conflitos que geralmente surgem, de modo a equilibrar funcionalidade, desempenho e demais características do *software* em relação a aspectos de custo e de tempo despendidos. O intuito principal desta ação é definir requisitos de incremento que satisfaçam às necessidades dos *stakeholders* e, ao mesmo tempo, reflitam as limitacões de tempo e de custo impelidas sobre a equipe de desenvolvimento.

Concluída a atividade de Comunicação, deve ser realizada a atividade de Planejamento que inicia considerando o escopo de projeto definido e conduz a um conhecimento do contexto de negócios, objetivos informativos, funcionalidades da aplicacão,

restrições e aspectos de desempenho. O plano geral de projeto contribui ao fornecer um guia para a entrega dos incrementos, porém esta atividade tem como foco as tarefas e produtos de trabalho que serão construídos para um incremento em questão. Quatro ações de WebE são previstas para a atividade de Planejamento, a saber: Estimação, Análise de Risco, Escalonamento e Monitoração.

A ação de Estimação tem por objetivos definir o número de incrementos da aplicação (esse número pode ser alterado no decorrer do projeto) e realizar a estimativa de recursos necessários para a construção do incremento. A ação de Análise de Risco é constituída por tarefas que auxiliam não só no entendimento, mas também na gestão dos riscos. Um risco, em suma, consiste em um problema em potencial que pode ou não ocorrer (PRESSMAN; LOWE, 2009). Independente da materialização ou não do risco, recomenda-se identificá-lo, analisar a probabilidade de ocorrência, estimar o impacto e definir um plano de contingência para resolvê-lo caso aconteça. A ação de Escalonamento estabelece o cronograma de projeto da aplicação *Web*. Esse cronograma aloca o esforço estimado para a realização das tarefas de WebE planejadas. É importante mencionar que, além do cronograma macro do projeto da aplicação *Web*, outros cronogramas específicos (alusivos à construção de cada um dos incrementos) também são definidos. A ação de Monitoração, por sua vez, consiste no acompanhamento do progresso do projeto, que pode ser feito a partir dos cronogramas de projeto e de incremento, bem como através de consultas à equipe de WebE para saber quais atividades foram realizadas e quantos cenários de usuários foram implementados.

A atividade de Modelagem tem por finalidade criar representações conceituais de aspectos da aplicação *Web* a ser desenvolvida. Algumas dessas representações conceituais incluem: esboços, documentos escritos, modelos gráficos, diagramas esquemáticos, cenários escritos, protótipos em papel ou executáveis. Duas ações de WebE são realizadas nesta atividade: Análise e Projeto.

A ação de Análise examina os requisitos dos *stakeholders* a partir de informações coletadas na atividade de Comunicação. Os modelos produzidos nesta ação geralmente estão focados no conteúdo, funcionalidade, modos de interação e configurações técnicas da aplicação. A ação de Projeto também objetiva criar representações da aplicação *Web*. Esses modelos, no entanto, consideram os seguintes aspectos: projeto de interface, projeto estético, projeto de conteúdo, projeto de navegação, projeto de arquitetura e projeto de componente.

Dando continuidade ao arcabouço de processo WebE, a próxima atividade a ser executada é a de Construção. Nesta atividade, são realizadas as ações de Codificação e de Teste. No contexto da Codificação, as seguintes tarefas e questões devem ser observadas:

criação do conteúdo e integração deste à arquitetura da aplicação *Web*; seleção de ferramentas para a geração de código; implementação de *layout* de página, função, formulário e mecanismo de navegação; implementação de todas as funções de computação; tratamento de aspectos relacionados à configuração. Para a ação de Teste, que consiste, basicamente, no processo de avaliação da aplicação *Web*, algumas tarefas e questões podem auxiliar no planejamento desta ação, entre elas: testar todos os componentes da aplicação *Web*, incluindo conteúdo e função; testar a navegação; testar a usabilidade; testar a segurança; testar os incrementos em diferentes configurações técnicas.

Para concluir, a atividade de Implantação tem por objetivos apresentar as funcionalidades da aplicação *Web* aos usuários, obter *feedback* destes no intuito de verificar se os requisitos inerentes aos incrementos foram satisfeitos e estabelecer uma base para a realização de ajustes que invariavelmente surgem como consequência da implantação. As seguintes tarefas e questionamentos podem ajudar no processo de implantação dos incrementos da aplicação *Web*: implantar o incremento da aplicação *Web* em um servidor predefinido; definir um mecanismo de *feedback online* para os usuários finais; avaliar a interação do usuário final; analisar as lições aprendidas e levar em consideração o *feedback* do usuário final; realizar ajustes no incremento da aplicação *Web*.

O modelo de processo WebE proposto por Pressman e Lowe (2009), ao definir um conjunto de atividades, ações e tarefas de trabalho individuais, estabelece um arcabouço de processo que une agilidade e disciplina, no intuito de produzir aplicações de qualidade industrial. Além da adoção de modelos de processo, a utilização de *frameworks* no processo de desenvolvimento de *software* também tem contribuído para ampliar a qualidade das aplicações construídas (SOMMERVILLE, 2007; PRESSMAN, 2011). Diante disso, será discutido, na próxima seção, o uso de *frameworks* como instrumento de apoio ao desenvolvimento e avaliação de aplicações *Web*.

### **3.3 *Framework* como Instrumento de Apoio ao Desenvolvimento e Avaliação de Aplicações *Web***

Um dos objetivos atuais da Engenharia de *Software* é o reuso. Por meio da reutilização de *software*, pode-se conseguir o aumento da qualidade e diminuição do esforço de desenvolvimento (GIMENES; HUZITA, 2005). A orientação a objetos, por exemplo, tem fornecido recursos que possibilitam não somente a reutilização de classes, mas também

métodos através dos mecanismos de herança<sup>11</sup> e polimorfismo<sup>12</sup> (SOMMERVILLE, 2007). O Desenvolvimento Baseado em Componentes (DBC), por sua vez, tem possibilitado a definição de unidades interoperáveis e reutilizáveis de *software* que proveem serviços a partir de interfaces bem definidas (GIMENES; HUZITA, 2005). Padrões de Projeto<sup>13</sup> (*Design Patterns*) é outra abordagem de reuso, mas com o foco no aproveitamento de projetos de soluções para problemas recorrentes (GAMMA *et al.*, 2007).

A demanda crescente de novos produtos de *software* e aplicações *Web*, observada principalmente na última década, vem requisitando das organizações o desenvolvimento de *software* com confiabilidade, de maneira rápida e competitiva, pois a procura por novos produtos está cada vez mais veloz e exigente. Uma das formas de se promover o reuso na Engenharia de *Software* e atender aos requisitos de produção e de qualidade exigidos no mercado de desenvolvimento de *software*, além dos recursos já discutidos como orientação a objetos, desenvolvimento baseado em componentes e *design patterns*, é a utilização de *frameworks* (SOMMERVILLE, 2007; PRESSMAN, 2011).

*Frameworks* têm por finalidade oferecer uma estrutura comum para um domínio de aplicações ao promover o reuso de conteúdo conceitual do domínio de um *software* ou da solução de um problema (GIMENES; HUZITA, 2005). O reuso proporcionado com *frameworks* não se restringe apenas à implementação, reflete-se também na análise e no projeto de um *software*. Na próxima seção, serão apresentados alguns conceitos de *framework*, classificação, tipos de processo de desenvolvimento, vantagens e desafios relativos à sua utilização.

### 3.3.1 O Conceito *Framework*

A tecnologia de *frameworks* permite que uma família de produtos seja construída a partir de uma única estrutura (PINTO, 2000). Essa estrutura captura os conceitos mais gerais da família de aplicações. Para Mattsson (1996, 2000), um *framework* é uma arquitetura

---

<sup>11</sup> Herança: o mecanismo de herança permite que uma subclasse (classe filha) herde todos os atributos e operações associadas a sua superclasse (classe pai). Novos atributos e operações podem ser adicionados às subclasses, operações também podem ser adaptadas (PRESSMAN, 2011).

<sup>12</sup> Polimorfismo: o polimorfismo possibilita que várias operações diferentes tenham o mesmo nome e possam ser desempenhadas por uma mesma chamada de função. Isso contribui para desacoplar os objetos uns dos outros, tornando-os mais independentes (PRESSMAN, 2011).

<sup>13</sup> Padrões de Projeto: apresentam um mecanismo codificado para descrever problemas e soluções relacionadas, de modo a capturar conhecimento de projeto a fim de permitir a reutilização (PRESSMAN, 2011). Em linhas gerais, um padrão descreve um problema, o contexto em que ocorre e apresenta uma solução para resolver o problema (GAMMA *et al.*, 2007).

construída com o intuito de alcançar o máximo reuso. Esta arquitetura é representada como um conjunto de classes abstratas e concretas com elevado potencial de especialização. De acordo com Gamma *et al.* (2007), um *framework* é um conjunto de classes que colaboram entre si provendo um projeto reutilizável para um domínio específico de classes de sistema. Tomhave (2005), por sua vez, define *framework* como um construtor fundamental que estabelece pressupostos, conceitos, valores e práticas e que inclui diretrizes para a execução propriamente dita.

O uso de *frameworks* também permite organizar e explicar, de forma gráfica e/ou textual, aspectos elementares a serem estudados como conceitos, constructos e suas relações. Alguns *frameworks* estão relacionados à solução de problemas ligados à tecnologia como, por exemplo, interface com o usuário e persistência de objetos. Outros, por seu turno, estão focados em um determinado domínio como aplicações bancárias e relacionamento com clientes. Os *frameworks* podem ser categorizados de diversas maneiras. Para começar, são classificados em dois grupos principais: *framework* de aplicação orientado a objetos (FAYAD *et al.*, 1999a; FAYAD *et al.*, 1999b; FAYAD; JOHNSON, 2000) e *framework* de componentes (SZYPERSKI, 1997).

De acordo com Fayad *et al.* (1999b), *frameworks* de aplicação são classificados quanto ao escopo em *frameworks* de infraestrutura de sistemas, *frameworks* de integração de *middleware* e *frameworks* de aplicações corporativas. *Frameworks* de infraestrutura de sistemas objetivam simplificar o desenvolvimento de sistemas de infraestrutura portáteis e eficientes, a exemplo de *frameworks* de comunicação, sistemas operacionais e ferramentas de processamento de linguagens. *Frameworks* de integração de *middleware* são empregados para integrar aplicações e componentes distribuídos, de modo a abstrair os detalhes de baixo nível existentes entre os componentes distribuídos. Por sua vez, os *frameworks* de aplicações corporativas são direcionados a um domínio de aplicação específico.

Ainda segundo Fayad *et al.* (1999b), os *frameworks* de aplicação podem ser classificados também quanto à forma utilizada para estendê-los. No *framework* Caixa Branca (*White Box*), o desenvolvimento das aplicações é baseado no mecanismo de herança. O usuário do *framework* realiza suas adaptações a partir da derivação de classes e reescrita de operações já definidas. *Frameworks* Caixa Preta (*Black Box*), por sua vez, são instanciados com base em configurações e composições, através da definição de classes que implementam uma dada interface ou contrato.

Para Szyperski (1997), um *framework* de componentes fornece apoio a componentes que seguem um determinado modelo e possibilita que instâncias desses componentes sejam

plugadas no *framework* de componentes. Além disso, estabelece condições necessárias para a execução de um componente e regula a interação entre as instâncias desses componentes. Uma das principais diferenças entre *frameworks* de aplicação e de componentes é que o primeiro define uma solução inacabada que possibilita a geração de uma família de aplicações e o segundo estabelece um contrato para plugar componentes.

Diversos atores estão envolvidos no processo de criação, uso e manutenção de *frameworks*, são eles (FROEHLICH *et al.*, 1997a; FROEHLICH *et al.*, 1997b; PINTO, 2000): projetista de *framework*, que é o responsável pelo desenvolvimento da estrutura elementar do *framework* (composta por classes e relacionamentos), incluindo pontos de extensão e levantamento de requisitos; usuário do *framework*, que faz uso da estrutura do *framework* para construir diferentes aplicações no domínio do *framework*; mantenedor de *framework*, que é o responsável pela alteração e/ou inclusão de novas funcionalidades ao projeto do *framework*. Cabe ressaltar que os papéis discutidos não são necessariamente exercidos por diferentes pessoas.

Para Pinto (2000), o processo de construção de um *framework* é dependente do nível de experiência da organização quanto ao domínio do problema. Isso significa que o nível de experiência da organização pode influenciar na adoção de um processo menos ou mais avançado de desenvolvimento. A seguir, são apresentados alguns processos de desenvolvimento de *framework* (MATTSSON, 2000; PINTO, 2000):

- **Processo convencional para o desenvolvimento de *frameworks*:** neste processo, a primeira atividade a ser realizada é a análise do domínio do problema. Em seguida, é construída a primeira versão do *framework* a partir das abstrações identificadas. Após isso, uma ou mais aplicações são desenvolvidas. O teste desempenha papel relevante, pois permite observar se *framework* é reutilizável;
- **Processo baseado em experiências de aplicações já desenvolvidas:** este processo tem início com o desenvolvimento de aplicações baseadas no domínio do problema (são sugeridas no mínimo duas). Feito isso, o *framework* será desenvolvido a partir das características em comum extraídas das aplicações. Em seguida, as aplicações são refeitas utilizando-se o *framework* no intuito de observar a correção das características extraídas;
- **Processo baseado na análise do domínio:** neste processo, a etapa inicial é a análise do domínio do problema com o propósito de identificar e compreender

as abstrações existentes. Essa análise demanda a observação de aplicações já construídas. Após isso, é realizada a construção do *framework* em conjunto com uma aplicação de teste. Modificações no *framework* são feitas quando necessário, assim como revisões nas aplicações a fim de observar se permanecem funcionando;

- **Processo de desenvolvimento utilizando *design patterns*:** neste processo, o primeiro passo consiste em construir uma aplicação e, na sequência, aplicar de forma sistemática *design patterns* para desenvolver o *framework*. Após isso, são realizadas as interações entre as aplicações e o *framework*;
- **Processo baseado em componentes:** neste processo, a reutilização de componentes é a palavra chave. Algumas formas de reutilização podem ser observadas como, por exemplo, a reutilização de um componente por várias estruturas abstratas. No processo de instanciação dessas estruturas, um dado componente é usado com o propósito de atender os requisitos selecionados.

A utilização de *frameworks* proporciona diversos benefícios como, por exemplo, modularidade, reusabilidade, extensibilidade e inversão de controle (FAYAD *et al.*, 1999b; PINTO, 2000). *Frameworks* encapsulam detalhes de implementação por meio de pontos de extensão, interfaces estáveis e bem definidas, o que contribui para aumentar a modularidade das aplicações. As interfaces existentes no *framework* definem componentes que podem ser redefinidos no processo de criação de novas aplicações. Isso colabora para aumentar não somente o reuso, mas também a produtividade, confiabilidade, qualidade e interoperabilidade das aplicações. Os pontos de extensão do *framework* possibilitam que funcionalidades sejam estendidas para criar uma aplicação. Concluindo, alguns *frameworks* também apresentam inversão de controle (*Inversion of Control*), isto é, transferência do controle da aplicação para o *framework*, que chama a aplicação em determinados momentos.

O desenvolvimento de um *framework* não é uma tarefa simples e precisa ser planejado para a reutilização (MATTSSON, 2000). Sendo assim, deve ser flexível (permitir o reuso de abstrações em diferentes contextos), extensível (possibilitar a introdução ou alteração de funcionalidades) e compreensível (com instruções de uso e documentação) (GIMENES; HUZITA, 2005). Na próxima seção, será apresentado, para fins de exemplo, um *framework* de modelagem para o processo de Engenharia *Web*.

### 3.3.2 Framework no Processo de Engenharia de *Software Web*

Ao longo dos anos, diversos arcabouços contendo sugestões de modelos a serem utilizados na Engenharia *Web* foram desenvolvidos. Esses arcabouços além de sugerir modelos têm por objetivo definir termos e conceitos relevantes (PRESSMAN; LOWE, 2009). No Quadro 3.2, é apresentado o Arcabouço de Arquitetura de Aplicação *Web* (AAAW) proposto por Kong, Liu e Lowe (2006).

**Quadro 3.2** – Arcabouço de Arquitetura de Aplicação *Web* (AAAW).

	<b>Estrutura (o que)</b>	<b>Comportamento (como)</b>	<b>Localização (onde)</b>	<b>Padrão</b>
<b>Arquitetura de Planejamento (Perspectiva do Planejador)</b>	Lista de coisas importantes para o negócio	Lista de processos que a empresa realiza	Lista de locais onde a empresa opera	Possíveis modelos e padrões de negócios
<b>Arquitetura de Negócios (Perspectiva do Proprietário da Empresa)</b>	Por exemplo, modelo entidade relacionamento de negócio	Por exemplo, modelo de processo de negócio	Por exemplo, modelo de localização da entidade de negócio	Por exemplo, padrões do modelo de negócio
<b>Arquitetura da Interface de Usuário (Perspectiva do Usuário)</b>	Por exemplo, modelo de estrutura da interface de usuário	Por exemplo, modelo de fluxo da interface do usuário	Por exemplo, modelo de mapa do <i>site</i> do usuário	Por exemplo, gabarito de interface, padrões de navegação
<b>Arquitetura da Informação (Perspectiva do Arquiteto de Informação)</b>	Por exemplo, dicionário de informações	Por exemplo, modelo de fluxo de informações	Por exemplo, modelo de localização do nó de informações	Por exemplo, padrões de esquema de informações
<b>Arquitetura do Sistema (Perspectiva do Arquiteto do Sistema)</b>	Por exemplo, estrutura do módulo, submódulo, página do servidor funcional do sistema	Por exemplo, modelo de fluxo de trabalho do módulo, submódulo, página do servidor	Por exemplo, modelo de mapeamento do <i>site</i> dos módulos, submódulos, páginas do servidor	Por exemplo, padrões de projeto, estilos de apresentação
<b>Arquitetura do Objeto <i>Web</i> (Perspectiva do Desenvolvedor)</b>	Por exemplo, relacionamento do objeto físico	Por exemplo, algoritmos no código fonte	Por exemplo, modelo de implantação da rede	Por exemplo, COTS, componentes, biblioteca de código
<b>Arquitetura de Teste (Perspectiva do Testador)</b>	Por exemplo, configuração de teste	Por exemplo, procedimento de teste	Por exemplo, implantação do teste	Por exemplo, gabaritos, padrões de documentação de teste

**Fonte:** Pressman e Lowe (2009).

O *framework* AAAW classifica os aspectos relacionados ao desenvolvimento de aplicações *Web* em duas dimensões. Como visto no Quadro 3.2, cada linha (dimensão horizontal) apresenta o ponto de vista de diferentes participantes no processo de Engenharia *Web*, a saber: proprietários de negócios, usuários da aplicação *Web*, arquitetos de informação, arquitetos de aplicação, desenvolvedores e testadores. Por sua vez, cada coluna (dimensão vertical) classifica as arquiteturas do arcabouço em quatro categorias com o objetivo de capturar os principais domínios de modelagem, tais como: estrutura (o que), comportamento (como), localização (onde) e padrão (KONG; LIU; LOWE, 2006).

As categorias de estrutura, comportamento e localização deste arcabouço refletem as três primeiras perguntas existentes no *Framework* de Zachman<sup>14</sup>. A última categoria (padrão) foi incluída dada a crescente importância dos padrões no contexto de desenvolvimento de *software* e aplicações *Web*. Para concluir, as interseções entre as dimensões (células do arcabouço) podem representar, a depender do caso, um modelo, uma descrição ou uma arquitetura, por exemplo (PRESSMAN, LOWE, 2009).

Após apresentar, de maneira geral, um arcabouço de arquitetura para Engenharia *Web*, serão discutidas, na próxima seção, algumas características relacionadas à usabilidade em aplicações *Web*.

### 3.4 Usabilidade em Aplicações *Web*

No que diz respeito às aplicações *Web*, observa-se que novos aspectos de interface e de interação precisam ser considerados, tendo em vista que essas aplicações são fundamentalmente multiusuário, podendo ser assíncronas ou síncronas. Além disso, certas características devem ser analisadas em relação à usabilidade de aplicações *Web*, são elas (HITZ *et al.*, 2006):

- **Inviabilidade (ou dificuldade) de treinamento dos usuários:** no decorrer do desenvolvimento ou mesmo atualização de aplicações de *software* complexas, é comum realizar treinamentos com o intuito de garantir a utilização plena, eficiente e satisfatória dessas aplicações. Esses treinamentos, algumas vezes, são empregados para compensar problemas de usabilidade e para adaptar os

---

<sup>14</sup> *Framework* de Zachman: *framework* de arquitetura composto por duas dimensões principais: a primeira dimensão apresenta seis perguntas (o que, como, onde, quem, quando e por quê) e a segunda dimensão representa as perspectivas dos interessados (planejador, proprietário, projetista, construtor, implementador e usuário) (ZACHMAN, 2015).

usuários às restrições de um *software*. Entretanto, nem sempre os programas de treinamentos são viáveis para as aplicações *Web*. Este fato enfatiza a necessidade de desenvolvimento de aplicações *Web* autoexplicativas;

- **Anonimidade dos usuários:** não é fácil fazer análises sobre os usuários de uma aplicação *Web*, visto que, não existe um ambiente de uso delimitado. Por esse motivo, é uma tarefa difícil adequar as aplicações *Web* às necessidades dos usuários.

Tendo por base os aspectos peculiares e as tarefas típicas encontradas nas aplicações *Web*, Matera *et al.* (2006) definem usabilidade de aplicações *Web* por meio da interpretação de cinco atributos apresentados por Nielsen (1993):

- **Facilidade de aprendizado da aplicação *Web*:** o quão fácil é para os usuários de uma aplicação *Web* entender o conteúdo e serviços disponíveis e como procurar informações específicas usando os *links* disponíveis para navegação no hipertexto. Facilidade de aprendizado também significa que cada página do hipertexto deve ser constituída de uma forma que o conteúdo seja fácil de entender e seus mecanismos de navegação sejam simples de identificar;
- **Eficiência da aplicação *Web*:** implica que qualquer conteúdo seja facilmente alcançado pelos usuários através dos *links* fornecidos. Além disso, quando os usuários acessam uma página, eles devem ser capazes de se orientar sozinhos e também compreender o significado da página a partir do ponto inicial de navegação;
- **Facilidade de memorização:** implica que, após um determinado período de tempo sem utilizar a aplicação, o usuário ainda consiga se localizar no hipertexto;
- **Geração de poucos erros:** devem ser reduzidas as chances de um usuário cometer erros típicos em uma aplicação *Web* como, por exemplo, seguir erradamente um *link*. Neste caso, os usuários devem ser capazes de retornar ao local de origem;
- **Satisfação do usuário:** diz respeito a situações nas quais os usuários sentem que estão no controle em relação ao hipertexto e que entendem o conteúdo disponível, assim como os comandos para navegação.

Diante da importância da usabilidade no contexto do processo de desenvolvimento de *software*, alguns princípios e diretrizes de projeto para a construção de interfaces foram propostos. Na próxima seção, tais princípios e diretrizes são discutidos.

### 3.5 Princípios e Diretrizes de Projeto de Interfaces para Aplicações *Web*

De acordo com Dix (1999), o projeto de uma interface deve ser realizado de modo a responder três perguntas principais para o usuário final:

- **A primeira pergunta é onde estou?** A interface deve prover uma indicação da página acessada e fornecer aos usuários sua localização na hierarquia de conteúdo;
- **A segunda pergunta é o que eu posso fazer agora?** A interface precisa auxiliar os usuários no entendimento das opções disponíveis como, por exemplo, que funções podem ser acessadas, que *links* estão ativos e que conteúdo é relevante;
- **A terceira pergunta é onde estive, para onde estou indo?** A interface deve facilitar a navegação. Isso inclui fornecer um mapa indicando onde os usuários estiveram e quais caminhos podem seguir para se mover até outro lugar na aplicação.

No intuito de ajudar a responder essas perguntas, alguns princípios de *design* de interfaces de sistemas interativos foram propostos. Dix *et al.* (2004), por exemplo, dividem os princípios de *design* em três categorias principais: facilidade de aprendizado, flexibilidade e robustez. O princípio facilidade de aprendizado está relacionado à capacidade do sistema interativo de ser aprendido e utilizado pelos usuários. O princípio de flexibilidade diz respeito à diversidade de formas de interações permitidas aos usuários junto ao sistema. O princípio de robustez refere-se ao nível de apoio fornecido aos usuários na interação com o sistema.

Para cada uma destas categorias, foram definidos outros princípios. Em relação à facilidade de aprendizado, tem-se: previsibilidade, isto é, obtenção de respostas do sistema condizentes com a ação realizada; capacidade de sintetização, através do fornecimento de *feedback* confiável e efetivo acerca de mudanças no sistema; familiaridade, por meio de ligações entre objetos e situações apresentadas aos usuários a situações vivenciadas por estes; consistência, a partir do uso dos mesmos elementos de entrada e saída para situações

semelhantes; generalização, que se refere à capacidade do usuário tomar porções reduzidas do sistema e generalizá-las, com o propósito de deduzir funcionalidades.

No que diz respeito à flexibilidade, foram apresentados os seguintes princípios: iniciativa de diálogo, ou seja, fornecimento de alguma forma de interação do sistema com os usuários; *multithreading*, capacidade de o usuário executar mais de uma tarefa ao mesmo tempo; migração de atividades, que se refere à capacidade do sistema realizar atividades rotineiras e de responsabilidade do usuário; substitutividade, isto é, possibilidade de se usar termos equivalentes para valores de entrada e saída; personalização, obtida por meio da adaptação de interfaces às necessidades dos usuários.

Quanto à robustez, quatro princípios de *design* de interfaces foram descritos por Dix *et al.* (2004): observabilidade, que está relacionada à capacidade do usuário identificar o estado atual da aplicação através de elementos dispostos na interface; recuperabilidade, isto é, capacidade do usuário se recuperar de um erro ou desfazê-lo; capacidade de resposta, conseguida por meio do fornecimento de respostas rápidas ao usuário no momento da interação e execução de atividades no sistema; conformidade de realização de atividades, referente ao nível de suporte oferecido pelo sistema para a realização de tarefas.

Tognozzi (2001) também propôs um conjunto de princípios para orientar o projeto de interfaces. Estes princípios, no entanto, foram adaptados e estendidos para o contexto de aplicações *Web* por Pressman e Lowe (2009). Tais princípios são apresentados no Quadro 3.3.

**Quadro 3.3** – Princípios de Projeto para Aplicações *Web*.

<b>Princípio</b>	<b>Descrição</b>
Antecipação	Uma aplicação <i>Web</i> deve ser projetada de maneira a antecipar o movimento (ação) seguinte do usuário.
Comunicação	A interface da aplicação <i>Web</i> precisa informar o estado das atividades realizadas pelos usuários.
Consistência	A utilização de controles de navegação, menus, ícones, botões, entre outros, deve ser consistente ao longo de toda a aplicação <i>Web</i> , isto é, um recurso deve ser usado com uma única finalidade.
Autonomia controlada	Os mecanismos de navegação existentes na interface devem facilitar a interação do usuário pela aplicação <i>Web</i> , mas devem assegurar os princípios de navegação estabelecidos na aplicação.
Eficiência	O projeto de interface de uma aplicação <i>Web</i> precisa otimizar a eficiência de realização do trabalho do usuário.

Princípio	Descrição
Flexibilidade	As interfaces de aplicações <i>Web</i> devem ser flexíveis de maneira a permitir que um grupo de usuários realize tarefas diferentemente, bem como explore a aplicação <i>Web</i> de modo mais ou menos aleatório.
Foco	A interface da aplicação <i>Web</i> deve permanecer focada na tarefa realizada pelo usuário em questão.
Lei de Fitt	Estabelece que o tempo gasto para se chegar a um determinado alvo é uma função da distância até o alvo e do seu tamanho.
Objetos de interface de usuário	Diversas bibliotecas de objetos (inclusive padrões) reutilizáveis de interface humana estão disponíveis para serem utilizadas nas interfaces de aplicações <i>Web</i> .
Redução de latência	A aplicação <i>Web</i> deve explorar a capacidade multitarefa da aplicação de maneira a possibilitar que o usuário continue trabalhando como se a operação em execução tivesse terminado. Além disso, os atrasos precisam ser reconhecidos, o que inclui o fornecimento de retroalimentação de áudio, relógio ou barra de progresso para indicar o processamento do trabalho.
Capacidade de aprendizado	Interfaces de aplicações <i>Web</i> devem ser projetadas de modo a minimizar o tempo de aprendizado, bem como reaprendizado quando a aplicação <i>Web</i> for visitada posteriormente.
Metáforas	Interfaces que usam metáforas de interação apropriadas são mais fáceis de aprender e de usar.
Mantenha a integridade do produto de trabalho	Uma aplicação <i>Web</i> precisa ser projetada para armazenar (salvar) automaticamente os dados fornecidos pelos usuários. Além disso, deve oferecer mecanismos de fácil acesso para recuperar informações perdidas.
Legibilidade	As informações apresentadas nas interfaces devem ser de fácil leitura tanto por pessoas jovens como idosas. Escolhas de <i>layout</i> , tamanho de fonte e cor do texto devem ser feitas de modo a melhorar o contraste.
Acompanhe o estado	O estado das interações realizadas pelos usuários durante a interação com as interfaces precisa ser acompanhado e armazenado, de modo a permitir continuar uma tarefa de onde parou.
Navegação visível	Quando os mecanismos de navegação são bem projetados, os usuários conseguem mais facilmente recuperar objetos e selecionar funções disponíveis nas interfaces.

**Fonte:** Tognozzi (2001) e Pressman e Lowe (2009).

Além dos princípios apresentados, algumas diretrizes de projeto pragmáticas para aplicações *Web* foram propostas por Nielsen e Loranger (2007), a saber: (i) não force o usuário a ler grandes quantidades de texto, pois a velocidade de leitura no monitor é aproximadamente vinte e cinco por cento mais lenta; (ii) evite sinais de “em construção”, visto que, estes geram expectativas e implicam em um *link* desnecessário; (iii) informações importantes devem ser apresentadas dentro das dimensões de uma janela, pois os usuários preferem não rolar a página; (iv) menus de navegação e barras no topo devem ser projetados

de modo consistente, devendo estar disponíveis em todas as páginas; (v) a estética não deve substituir a funcionalidade; (vi) as opções de navegação devem ser óbvias, mesmo para usuários casuais.

Shneiderman e Plaisant (2009) também elaboraram um conjunto de diretrizes com o objetivo de reduzir os problemas de usabilidade de interfaces computacionais. Estas recomendações foram organizadas segundo quatro interesses, reconhecidos como de grande importância para o processo de *design*, a saber:

### **Navegabilidade da Interface**

- Padronize a sequência de tarefas: funções semelhantes devem ser realizadas seguindo-se uma mesma sequência de atividades;
- Assegure que os *links* da página sejam descritivos: o texto associado a um *link* deve explicar claramente a ação executada caso este seja acionado;
- Use cabeçalhos distintos para funcionalidades distintas: cabeçalhos precisam estar associados ao conteúdo a ser apresentado;
- Use caixas de múltipla escolha (*checkboxes*) para escolhas que possuem mais de uma resposta: escolhas que permitem mais de uma resposta devem ser feitas através de *checkboxes*;
- Permita que o usuário imprima a página adequadamente: as interfaces precisam ser desenvolvidas em medidas adequadas para a impressão;
- Use imagens pequenas como forma de pré-visualizar as imagens originais: forneça imagens menores aos usuários quando as imagens de tamanho original não sejam necessárias.

### **Organização do Conteúdo Exibido**

- Consistência dos dados exibidos: abreviações, formatos e cores de termos precisam ser documentados na forma de um dicionário para criar uma identidade e facilitar o uso por parte dos *designers*;
- Facilidade de assimilação da informação pelo usuário: a linguagem utilizada deve ser familiar e facilmente assimilada pelos usuários;
- Minimização da carga de memória do usuário: deve-se evitar que o usuário tenha que se recordar de informações entre diferentes interfaces da aplicação;

- Compatibilidade entre a entrada de dados e o conteúdo exibido: elementos de informações apresentados aos usuários devem estar associados de forma precisa aos componentes de entrada de dados;
- Flexibilidade de acesso e uso de informações: os usuários devem ser capazes de compreender as informações disponíveis nas interfaces, assim como escolher de que maneira vão realizar certas atividades.

### **Captura da Atenção do Usuário**

- Intensidade dos elementos: diversifique a intensidade dos elementos na tela. Dois níveis de intensidade podem ser utilizados em uma mesma tela;
- Marcação: o uso do sublinhado, de bordas, setas ou demais marcadores chamam a atenção dos usuários;
- Tamanho: podem ser utilizados até quatro tamanhos diferentes em uma mesma tela;
- Escolha das fontes: os *designers* podem fazer uso de até três diferentes fontes para a escrita de mensagens;
- Efeito de vídeo inverso: refere-se à inversão de cor do fundo com a cor do componente;
- Cores: os *designers* podem fazer uso de até quatro cores diferentes;
- Áudio: diferentes níveis de sons podem ser empregados para indicar respostas positivas e negativas às ações dos usuários.

### **Facilitação de Entrada de Dados**

- Consistência das transações que exigem entrada de dados: as condições de entrada de dados podem variar nas interfaces, mas a sequência de ações requeridas para o usuário realizar a tarefa deve ser semelhante em termos de delimitadores, espaçamentos, abreviações, dentre outros;
- Minimização do número de ações a serem realizadas pelo usuário: interfaces devem ser desenvolvidas de modo a minimizar as ações e interações necessárias para se realizar uma tarefa.

Além das recomendações discutidas, um conjunto de diretrizes para a criação de interfaces de aplicações de dispositivos móveis também foi identificado. Estas diretrizes

específicas justificam-se dada a natureza de interação peculiar (ROBERTSON *et al.*, 2005), bem como características e limitações físicas existentes nestes dispositivos. Tais diretrizes foram desenvolvidas por fabricantes de dispositivos móveis e catalogadas por Neto (2013), a saber:

- Destaque a principal atividade da aplicação e garanta os subsídios necessários para que o usuário complete qualquer tarefa;
- Invista os maiores esforços nos fatores da aplicação mais importantes do ponto de vista do usuário;
- Pense no *design* da interface como uma atividade a ser preenchida de cima para baixo;
- Disponibilize um caminho lógico para o usuário;
- Torne a interação fácil e óbvia;
- Facilite a entrada de dados;
- Estimule a conectividade e o comportamento colaborativo;
- Torne a interface mais realista possível;
- Forneça suporte à mudança de orientação;
- Mantenha o usuário ciente de qualquer ação;
- Forneça controle ao usuário;
- Crie uma página de ajuda;
- Aposte em um *design* minimalista;
- Use imagens e gráficos em alta definição e editados profissionalmente;
- Use componentes na medida adequada.

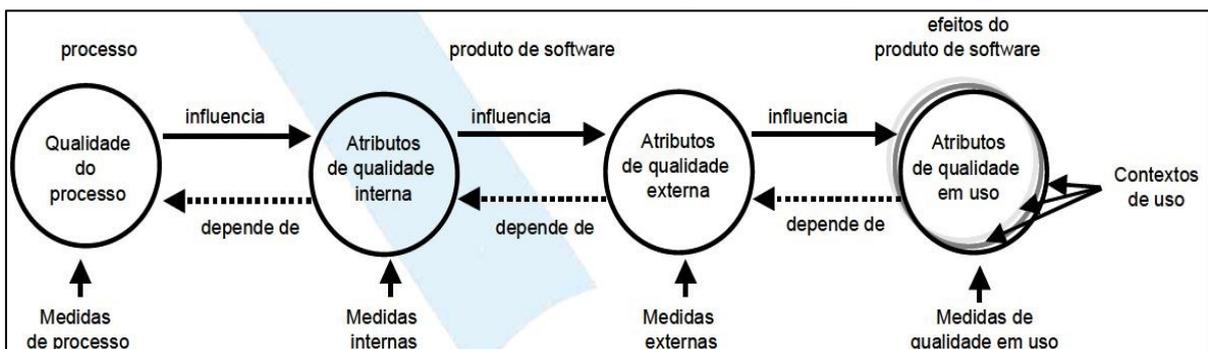
Os princípios e diretrizes de projeto de interface apresentados estabelecem um conjunto de características fundamentais e uma filosofia que os projetistas e desenvolvedores de interfaces de aplicações *Web* devem seguir. Em suma, a interface do usuário é a porta de entrada de uma aplicação *Web*. E, independentemente da importância de seu conteúdo, de sua capacidade de processamento e dos serviços oferecidos, uma interface mal projetada frustra e pode afastar os usuários, pois estes não conseguem realizar com sucesso e de maneira facilitada as tarefas desejadas.

Por sua vez, interfaces de aplicações *Web* eficazes e fáceis de usar são: tolerantes a erros; proporcionam aos usuários um sentido de controle; mostram claramente aos usuários as opções disponíveis, como realizar as tarefas e alcançar os objetivos; não incomodam os usuários; realizam um máximo de trabalho enquanto requerem um mínimo de esforço e informações dos usuários (TOGNOZZI, 2001). Na próxima seção, será discutida a importância da usabilidade para a qualidade final do *software* e das aplicações *Web*.

### 3.6 Importância da Usabilidade para a Qualidade das Aplicações *Web*

O modelo de qualidade proposto pela Associação Brasileira de Normas Técnicas (ABNT) na norma 9126-1 apresenta um *framework* composto por três visões de qualidade de *software*, a saber (ABNT NBR, 2003): qualidade interna, qualidade externa e qualidade em uso. Na Figura 3.2, é ilustrado o relacionamento entre estas três diferentes visões.

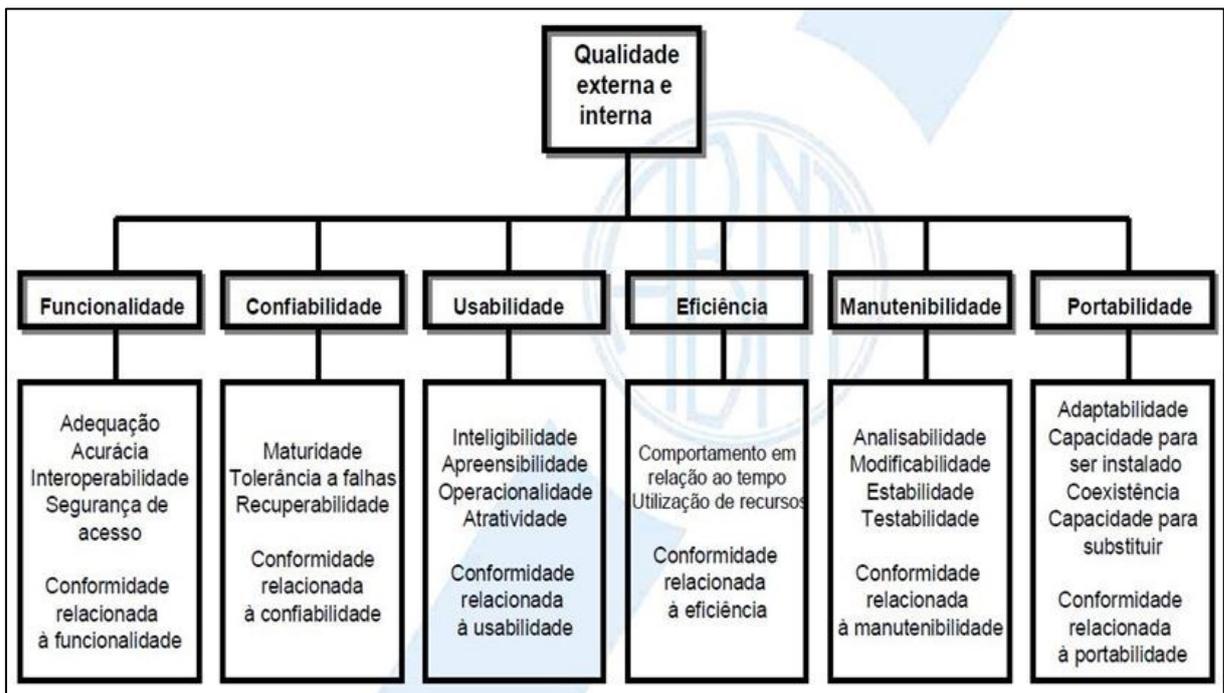
**Figura 3.2** – Diferentes Visões de Qualidade de *Software*.



Fonte: ABNT NBR (2003).

A qualidade interna corresponde à totalidade das características do produto de *software* do ponto de vista interno. Esta visão de qualidade é medida e avaliada em relação aos requisitos de qualidade interna. Detalhes da qualidade do *software* podem ser melhorados durante a implementação do código, revisão e teste. A qualidade externa diz respeito às características do *software* sob o ponto de vista externo. Esta visão pode ser medida e avaliada quando o *software* é executado e testado em um ambiente simulado. A qualidade em uso, por sua vez, representa a visão do ponto de vista do usuário quando o *software* é utilizado. Esta visão mede o quanto os usuários podem alcançar seus objetivos e não está relacionada com as propriedades intrínsecas do *software* (CHEIKHI; ABRAN; SURYN, 2006). Na Figura 3.3, é apresentado o modelo de qualidade externa e interna da norma ABNT NBR 9126-1.

Figura 3.3 – Modelo de Qualidade Externa e Interna.



Fonte: ABNT NBR (2003).

Como visto na Figura 3.3, o modelo de qualidade externa e interna da ABNT NBR 9126-1 apresenta seis atributos de qualidade de *software*, a saber:

- **Funcionalidade:** diz respeito à capacidade do produto de *software* de prover funções que atendam às necessidades explícitas e implícitas, quando o *software* estiver em uso sob condições especificadas;
- **Confiabilidade:** refere-se à capacidade do produto de *software* de manter um nível de desempenho especificado, quando utilizado em condições estabelecidas;
- **Usabilidade:** está relacionada à capacidade do produto de *software* de ser compreendido, aprendido, operado e agradável ao usuário quando usado em um contexto de uso específico;
- **Eficiência:** diz respeito à capacidade do produto de *software* de apresentar desempenho apropriado em relação aos recursos consumidos, sob condições especificadas;
- **Manutenibilidade:** está associada à capacidade do produto de *software* de ser modificado. Estas modificações podem incluir correções, melhorias ou

adaptações do *software* em virtude de mudanças no ambiente, nos requisitos ou especificações funcionais;

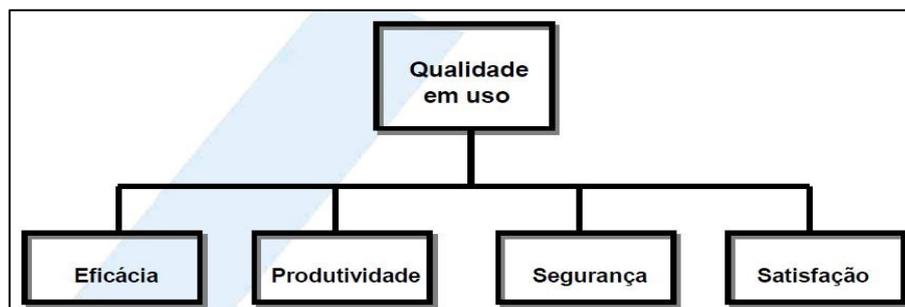
- **Portabilidade:** refere-se à capacidade do produto de *software* de ser transferido de um ambiente para outro.

Como visto, o atributo de usabilidade diz respeito à capacidade do produto de *software* ser entendido, aprendido, utilizado e agradável ao usuário, quando empregado em um contexto de uso específico (ABNT NBR, 2003). Para este atributo, foram definidas cinco subcaracterísticas:

- **Inteligibilidade:** corresponde à capacidade do *software* de permitir ao usuário entender se este é adequado e como pode ser utilizado para a realização de tarefas;
- **Apreensibilidade:** diz respeito à capacidade do *software* de possibilitar ao usuário aprender como utilizá-lo;
- **Operacionalidade:** está relacionada à capacidade do *software* de permitir ao usuário usá-lo e, também, controlá-lo;
- **Atratividade:** corresponde à capacidade do *software* de ser atrativo aos usuários;
- **Conformidade relacionada à usabilidade:** diz respeito à capacidade do *software* de atender a normas, convenções ou diretrizes associadas à usabilidade.

O modelo de qualidade em uso apresentado na Figura 3.4, por sua vez, divide os atributos de qualidade em uso em quatro características, são elas: eficácia, produtividade, segurança e satisfação (ABNT NBR, 2003).

Figura 3.4 – Modelo de Qualidade em Uso.



Fonte: ABNT NBR (2003).

A eficácia corresponde à capacidade do *software* de permitir ao usuário atingir seus objetivos com acurácia e completude. A produtividade está relacionada à capacidade do *software* de permitir ao usuário aplicar uma quantidade adequada de recursos em relação à eficácia obtida. A segurança corresponde à capacidade do *software* de apresentar níveis adequados e aceitáveis de riscos de danos a pessoas, negócios, outros *softwares* ou ao ambiente. A satisfação, por outro lado, refere-se ao grau de satisfação do usuário ao realizar uma tarefa em um dado contexto de uso (ABNT NBR, 2003).

Como visto, a usabilidade se faz presente nas mais variadas visões de qualidade de *software* (qualidade interna, qualidade externa e qualidade em uso). Este fato demonstra a importância da usabilidade para a qualidade final do *software*, ainda mais quando a usabilidade é apontada pelos usuários como o atributo de qualidade de *software* mais relevante (DE LA VARA *et al.*, 2011). Essa importância mostra-se ainda maior quando se coloca em perspectiva as aplicações *Web*, pois são naturalmente interativas, centradas no usuário e baseadas em hipermídia, onde a interface com o usuário representa o elemento central de interação (OLSINA; COVELLA; ROSSI, 2006).

### **3.7 Considerações Finais do Capítulo**

Neste capítulo, foi apresentada uma discussão sobre os principais aspectos e características associadas à usabilidade de aplicações *Web*. Inicialmente, foram discutidos os atributos e as categorias existentes. Além disso, foi apresentado um modelo de processo para a Engenharia *Web* que fornece um arcabouço de atividades para o desenvolvimento de aplicações *Web*. Em seguida, foi realizada uma discussão acerca do uso de *frameworks* no desenvolvimento de *software*, incluindo conceitos, tipos de processos existentes e benefícios decorrentes da utilização. Após isso, foram apresentados alguns princípios e diretrizes que orientam o projeto de interfaces com melhor qualidade de uso.

Este capítulo também discutiu a importância da usabilidade para a qualidade final do *software* e das aplicações *Web*. Para Offutt (2002), a usabilidade é um dos três critérios de qualidade que dirige o processo de desenvolvimento de aplicações *Web* nas organizações. Ainda nesta discussão, foi possível observar que o atributo de usabilidade se faz presente nas mais diversas visões de qualidade, a exemplo da qualidade interna, qualidade externa e qualidade em uso, apresentadas na norma ABNT NBR 9126-1 (ABNT NBR, 2003). Este fato vem reforçar a necessidade de se buscar e manter, ao longo do processo de desenvolvimento

de aplicações *Web*, o atendimento aos atributos e requisitos de qualidade de *software* relacionados à usabilidade. Para auxiliar no alcance deste importante objetivo, será apresentado no próximo capítulo o *Framework* UEF-WEB proposto nesta dissertação.

### **FRAMEWORK UEF-WEB**

Neste capítulo é apresentado o *Framework* UEF-WEB, um *Framework* de apoio à Engenharia de Usabilidade para aplicações *WEB*. Este *framework* fornece um arcabouço sucinto de fases, atividades, recursos e artefatos de suporte à Engenharia de Usabilidade para as fases de planejamento, desenvolvimento e avaliação de interfaces de aplicações *Web*.

#### **4.1 Introdução**

Nos últimos anos, a demanda por aplicações *Web* cresceu rapidamente para atender as mais variadas necessidades das pessoas e organizações. Mais do que isso, as aplicações *Web* assumiram papel de destaque ao colaborar para mudar a forma como as pessoas efetuam compras, buscam informações, comunicam-se, expressam opiniões, estudam e realizam atividades de entretenimento (PRESSMAN; LOWE, 2009).

O crescimento da demanda, importância e complexidade das aplicações *Web*, aliado às características inerentes dessas aplicações, contribuiu para motivar a utilização de boas práticas no desenvolvimento de aplicações *Web*, no intuito destas serem entregues dentro do prazo e orçamento previamente planejados, com alto grau de qualidade e facilidade de manutenção (PASTOR, 2004).

A pesquisa acerca de abordagens de suporte à Engenharia de Usabilidade para aplicações *Web* foi guiada através das seguintes diretrizes: (i) foco nos critérios de qualidade relevantes para a construção de aplicações *Web*; (ii) sistematização e facilidade de uso durante o processo de desenvolvimento. Para Offutt (2002), três critérios de qualidade orientam o processo de desenvolvimento de aplicações *Web*: confiabilidade, segurança e usabilidade. Esta pesquisa está focada fundamentalmente no critério de usabilidade.

Diversas técnicas, métodos e ferramentas têm sido desenvolvidos para apoiar a Engenharia de Usabilidade. Entretanto, apesar do número e da variedade de contribuições propostas, há organizações que ainda não estão fazendo uso dessas contribuições (INSFRAN; FERNANDEZ, 2008) por desconhecimento dos recursos disponíveis, limitação de orçamento

para investir em usabilidade e dificuldades na seleção e implantação desses recursos (NIELSEN; LORANGER, 2007; CONTE, 2009; CYBIS; BETIOL; FAUST, 2010).

Diante dessas dificuldades, esta pesquisa propõe uma abordagem metodológica e prática para auxiliar as organizações a adotar, de forma simples e sistemática, recursos da Engenharia de Usabilidade no desenvolvimento de aplicações *Web*, com vistas a garantir a qualidade de uso das interfaces produzidas. Esta foi a principal motivação que norteou o desenvolvimento desta dissertação. Na próxima seção, será apresentado o *Framework* UEF-WEB proposto para atender o objetivo supracitado.

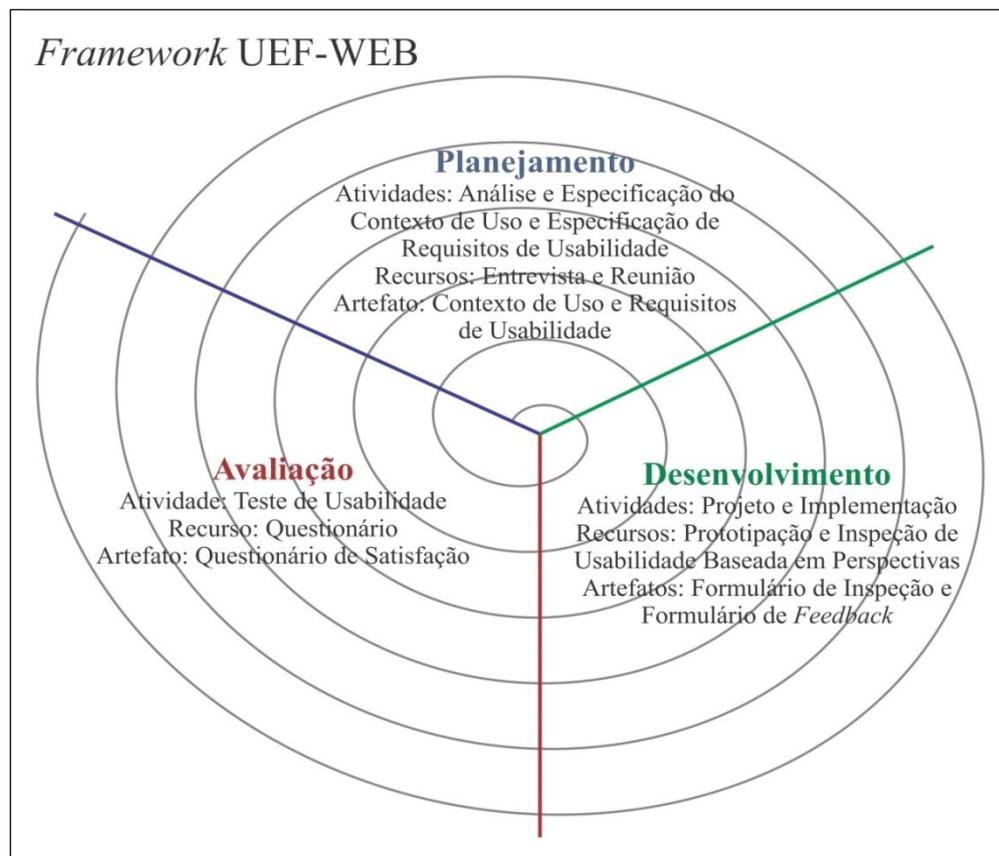
## **4.2 *Framework* UEF-WEB**

Conforme já mencionado, este trabalho propõe um *Framework* de apoio à Engenharia de Usabilidade para aplicações *WEB*, representado pelo acrônimo UEF-WEB. Este *framework* consiste em um arcabouço sucinto de fases, atividades, recursos e artefatos de apoio à Engenharia de Usabilidade para o processo de desenvolvimento de interfaces. A utilização deste *framework* em organizações busca, fundamentalmente, promover a adoção sistematizada de recursos de usabilidade e, como isso, elevar a qualidade de uso das interfaces de aplicações *Web* construídas.

A ideia central deste *framework* é a execução, de forma cíclica, evolutiva e com a participação dos usuários, de um conjunto de fases, atividades, recursos e artefatos de suporte à usabilidade. Tal estratégia, na qual o paradigma de desenvolvimento de interfaces com usuários parte de sucessivos ciclos de análise, concepção e testes, segue os preceitos da norma ISO 9241-210 (ISO, 2010). Ademais, busca reduzir os riscos de falhas conceituais de projeto e problemas na implementação de modo a garantir que, a cada ciclo de desenvolvimento, a aplicação *Web* corresponda cada vez melhor às expectativas e necessidades dos usuários. O desenvolvimento deste *framework* foi realizado a partir de pesquisa bibliográfica e, conseqüente, análise de propostas relacionadas à usabilidade identificadas na literatura e discutidas nos capítulos desta dissertação.

O *Framework* UEF-WEB é composto por três fases: Planejamento, Desenvolvimento e Avaliação. Cada uma destas fases tem objetivos específicos dentro do processo de desenvolvimento de interfaces e demanda a realização de uma série de atividades que, por sua vez, são suportadas por um conjunto de recursos e artefatos providos pela Engenharia de Usabilidade. A estrutura geral do *Framework* UEF-WEB é apresentada na Figura 4.1.

**Figura 4.1** – Framework UEF-WEB.



Fonte: Esta Pesquisa.

A Fase de Planejamento tem por objetivos compreender e especificar o contexto de uso e requisitos de usabilidade. Nesta fase, são coletadas informações sobre os usuários, tarefas, ambiente de uso e requisitos de usabilidade das interfaces. As atividades de Análise e Especificação do Contexto de Uso e de Especificação de Requisitos de Usabilidade são realizadas no intuito de atender estes objetivos. Os recursos de Entrevista e Reunião são empregados para dar suporte a estas atividades, assim como o artefato Contexto de Uso e Requisitos de Usabilidade.

A Fase de Desenvolvimento tem por finalidades projetar e construir as interfaces da aplicação *Web*. Para isso, são realizadas as atividades de Projeto e Implementação, atividades típicas da Engenharia de *Software*, que vão receber da Fase de Planejamento os insumos necessários, a saber: especificação do contexto de uso e especificação dos requisitos de usabilidade. Durante a atividade de Projeto, o recurso de Prototipação é utilizado. Este recurso conta com o suporte do artefato Formulário de *Feedback* e visa coletar e analisar as opiniões dos representantes dos usuários sobre as interfaces que estão sendo projetadas. A obtenção destas informações junto aos usuários é de grande importância para o prosseguimento do projeto, pois contribui para esclarecer dúvidas e resolver problemas de usabilidade em um

estágio precoce do processo de concepção de interfaces. Após a atividade de Implementação, é utilizado o recurso de Inspeção de Usabilidade Baseada em Perspectivas, no intuito de identificar problemas de usabilidade nas versões de interfaces desenvolvidas. O artefato Formulário de Inspeção fornece suporte a este recurso.

A Fase de Avaliação do *Framework* UEF-WEB tem por finalidade mensurar a satisfação dos usuários ao realizar tarefas e interagir com as interfaces de uma aplicação *Web*. Nesta fase, é realizada a atividade de Teste de Usabilidade como forma de avaliar, sob o ponto de vista dos usuários, a usabilidade das interfaces desenvolvidas. A coleta destas informações é feita com o auxílio do recurso de Questionário e por meio do artefato Questionário de Satisfação. Tais informações são de grande importância, pois revelam a percepção dos usuários sobre a facilidade e adequação de uso das interfaces desenvolvidas. No Quadro 4.1, é apresentada a distribuição das atividades, recursos e artefatos em cada uma das fases do *Framework* UEF-WEB.

**Quadro 4.1** – Fases, Atividades, Recursos e Artefatos do *Framework* UEF-WEB.

Fase	Atividade	Recurso	Artefato
Planejamento	Análise e Especificação do Contexto de Uso	Entrevista e Reunião	Contexto de Uso e Requisitos de Usabilidade
	Especificação de Requisitos de Usabilidade		
Desenvolvimento	Projeto	Prototipação	Formulário de <i>Feedback</i>
	Implementação	Inspeção de Usabilidade Baseada em Perspectivas	Formulário de Inspeção
Avaliação	Teste de Usabilidade	Questionário	Questionário de Satisfação

**Fonte:** Esta Pesquisa.

O *Framework* UEF-WEB foi pensado e projetado para ser utilizado de modo integrado a outros processos de desenvolvimento ou mesmo de maneira independente no desenvolvimento da interação humano-computador. Ao fazer uma análise comparativa entre o *Framework* UEF-WEB e o modelo de processo de Engenharia *Web* proposto por Pressman e Lowe (2009), é possível observar que:

- A fase de Planejamento se relaciona com a atividade de Comunicação, pois ambas se preocupam essencialmente com o levantamento de informações, definição de objetivos e especificação de requisitos;

- A fase de Desenvolvimento está relacionada com as atividades de Modelagem e de Construção, pois estão focadas no processo de análise, projeto e implementação da aplicação;
- A fase de Avaliação possui ligação com as atividades de Modelagem e de Implantação, tendo em vista que estas atividades estabelecem em seu escopo a realização de testes na aplicação.

Esta análise indica, de maneira sucinta, a relação entre as duas abordagens discutidas, mas é oportuno lembrar que diferentes objetivos específicos estão envolvidos, a saber: o arcabouço WebE propõe um modelo de processo geral para aplicações *Web* e o *Framework* UEF-WEB apresenta como contribuição um arcabouço de atividades, recursos e artefatos de suporte à usabilidade para o desenvolvimento de aplicações *Web*. Este *framework*, entretanto, não está restrito ao desenvolvimento, tendo em vista que pode ser utilizado para apoiar a usabilidade no processo de manutenção e/ou evolução de interfaces de aplicações *Web*.

É importante destacar que durante o processo de construção de um *software* ou aplicação *Web* podem ser realizados vários ciclos de atividades sendo que, em cada ciclo, um subconjunto de ações e tarefas é executado para atingir certo objetivo como, por exemplo, implementar um determinado requisito funcional<sup>15</sup>. Além disso, em alguns ciclos, algumas atividades podem ser executadas em maior ou menor grau de detalhes a depender do objetivo do ciclo em questão.

Diante disso, vários ciclos de atividades também são realizados no *Framework* UEF-WEB para desenvolver as interfaces da aplicação. Em cada ciclo, as atividades previstas nas três fases do *framework* são executadas com o propósito de atingir os objetivos definidos. Os resultados produzidos em cada fase servem de entrada para a fase seguinte. Exemplificando, o *feedback* obtido através dos testes com usuários (Teste de Usabilidade) constitui insumo de grande importância para a implementação de melhorias nas próximas versões de interfaces a serem desenvolvidas. As informações recolhidas a partir do recurso de Prototipação também contribuem ao fornecer diretrizes para a readequação das interfaces projetadas.

Nas próximas seções, serão apresentadas de forma detalhada as três fases do *Framework* UEF-WEB, assim como as atividades, recursos e artefatos estabelecidos para cada uma destas fases.

---

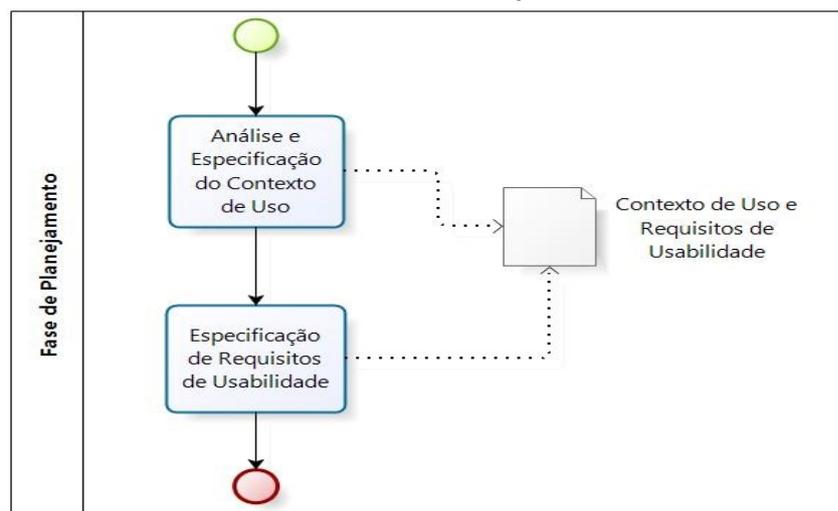
<sup>15</sup> Requisito funcional pode ser entendido como uma declaração que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em determinadas situações. Em resumo, requisitos funcionais descrevem as funcionalidades ou serviços de um sistema (SOMMERVILLE, 2007).

### 4.2.1 Fase de Planejamento

Para obter um produto de *software* ou aplicação *Web* de boa qualidade em termos de usabilidade, é importante seguir um processo de desenvolvimento que contemple atividades e recursos de apoio à usabilidade. Este processo deve ter como meta o desenvolvimento de interfaces humano-computador com qualidade de uso. Não é recomendado, por exemplo, iniciar o projeto de interfaces sem levar em consideração o conhecimento sobre o contexto de uso e sobre os requisitos de usabilidade (CYBIS; BETIOL; FAUST, 2010).

Diante disso, a Fase de Planejamento do *Framework* UEF-WEB estabelece, como objetivos principais, a compreensão e a especificação do contexto de uso e dos requisitos de usabilidade das interfaces da aplicação *Web*. Na Figura 4.2, são apresentadas as duas atividades, bem como o artefato utilizado nesta fase.

**Figura 4.2** – Atividades e Artefato da Fase de Planejamento do *Framework* UEF-WEB.



**Fonte:** Esta Pesquisa.

A atividade de Análise e Especificação do Contexto de Uso é a primeira atividade a ser realizada no *Framework* UEF-WEB e tem como objetivo levantar informações acerca dos usuários, das tarefas e do ambiente onde será utilizada a aplicação *Web* em questão. As informações produzidas nesta atividade constituem o contexto de uso e correspondem a insumos importantes para as demais atividades deste *framework*, pois são utilizadas no decorrer do projeto, implementação e avaliação das interfaces.

O termo Análise, aqui empregado, diz respeito à atividade de modelagem do contexto de uso diferentemente do termo Análise muito conhecido e utilizado na Engenharia de *Software*, o qual é usado para denotar as atividades do Modelo de Análise que, em linhas gerais, apresenta o *software* em nível de definição do problema (PRESSMAN, 2011). A

Análise e Especificação do Contexto de Uso não busca definir soluções de projeto. Pelo contrário, visa levantar informações a serem utilizadas durante o projeto de interfaces com usuário e demais atividades do processo de desenvolvimento de interfaces.

A atividade de Análise e Especificação do Contexto de Uso possui relativa interdependência com outras atividades da Engenharia de *Software* como, por exemplo, a atividade de levantamento de requisitos, pois as funções que deverão ser implementadas na aplicação *Web*, isto é, requisitos funcionais, vão dar suporte aos usuários nas tarefas que realizam. Esta atividade é constituída pelos seguintes tipos de análises: Análise de Usuários, Análise de Tarefas e Análise de Ambiente.

A Análise de Usuários visa obter informações sobre os usuários e demais grupos de *stakeholders* quanto a aspectos e necessidades que são importantes para o desenvolvimento de interfaces computacionais. Os grupos de usuários relevantes devem ser identificados, assim como a sua relação com o desenvolvimento da aplicação deve ser descrita em função dos objetivos e restrições mais importantes. A caracterização dos usuários é realizada a partir do levantamento de um conjunto de informações acerca de conhecimentos, habilidades, necessidades, expectativas, preferências e responsabilidades dos usuários sobre a aplicação a ser construída. Os seguintes objetivos específicos são estabelecidos para este tipo de análise:

- Identificar os tipos de usuários e *stakeholders* da aplicação *Web*;
- Caracterizar os usuários nos diversos aspectos do seu modo de trabalho ou comportamento relevante para a construção da aplicação;
- Fornecer insumos para a definição de requisitos e metas de usabilidade;
- Trazer contribuições para o projeto de interfaces e para o processo de avaliação de usabilidade de interfaces.

A Análise de Tarefas, por sua vez, tem por objetivos identificar e caracterizar as tarefas que serão realizadas pelos usuários através das interfaces da aplicação *Web*. Como resultado, são produzidas informações e recomendações para as atividades posteriores do processo de desenvolvimento sobretudo para o projeto de interface com o usuário. É importante destacar que este tipo de análise possui relação com a Análise de Usuários. Tipicamente, estas duas análises são executadas em um processo incremental e iterativo. Com isso, durante essas análises novos tipos de usuários podem ser identificados, assim como novas tarefas podem ser descritas. A Análise de Tarefas possui os objetivos específicos listados a seguir:

- Identificar as necessidades dos usuários em termos de tarefas que serão suportadas pela aplicação *Web* em questão;
- Caracterizar as tarefas nos diversos aspectos como, por exemplo, objetivos, resultados, estrutura, duração, frequência de realização, dependências;
- Trazer contribuições para a especificação de requisitos de usabilidade;
- Fornecer insumos para o projeto de interfaces e para o processo de avaliação de usabilidade de interfaces.

A Análise de Ambiente, em linhas gerais, tem o propósito de caracterizar o ambiente onde os usuários ou potenciais usuários desenvolvem suas tarefas relacionadas com a aplicação *Web*. Este tipo de análise também pode ser considerado tanto parte da Análise de Usuários quanto parte da Análise de Tarefas. É importante ressaltar que os usuários não executam suas tarefas de maneira isolada do ambiente. Diferentemente disso, o ambiente pode exercer muita influência no uso de uma aplicação, de modo que a incompatibilidade do ambiente com a aplicação pode inclusive ocasionar a rejeição desta.

Como exemplo, pode-se mencionar um local de trabalho onde o tempo de resposta deva ser muito rápido e a disponibilidade de informações deva ser imediata. Neste caso, a aplicação *Web* precisa ser projetada e desenvolvida para atender a estas exigências. Do contrário, a aplicação não estará alinhada com a realidade do ambiente e, conseqüentemente, não atenderá satisfatoriamente os usuários na realização de suas tarefas.

Diferentes recursos podem ser empregados no processo de coleta de informações sobre o contexto de uso e definição de requisitos de usabilidade de uma aplicação *Web*. O *Framework* UEF-WEB estabelece a utilização dos recursos de Entrevista e Reunião. É importante destacar que estes recursos, apesar de serem técnicas elementares de análise, fornecem meios que permitem de forma simples e rápida obter as informações requeridas na Fase de Planejamento. Além disso, promovem a interação entre a equipe de desenvolvimento e os representantes dos usuários, de modo a envolver a participação destes no processo de concepção das interfaces. Através de entrevistas junto aos usuários, por exemplo, é possível identificar as funcionalidades mais importantes de uma aplicação e, com isso, providenciar atalhos para a sua realização.

As entrevistas e reuniões devem ser realizadas pela equipe de desenvolvimento para obter informações dos representantes dos usuários e dos demais *stakeholders* acerca do contexto de uso e dos requisitos de usabilidade. Para auxiliar neste propósito, foi definido um

roteiro estruturado de questões. Tal roteiro, em suma, deve cobrir os objetivos das análises de usuários, de tarefas e de ambiente discutidos, bem como os requisitos de usabilidade da aplicação. Os recursos de entrevista e reunião também devem ser empregados para esclarecer informações já coletadas a fim de dirimir eventuais dúvidas e/ou inconsistências.

O sucesso destes recursos depende de quão bem forem realizadas as etapas de planejamento, de execução e de elaboração de relatório (CYBIS; BETIOL; FAUST, 2010). Para ajudar neste processo, foi desenvolvido um Roteiro Básico de Questões para Entrevista e Reunião. Este roteiro é constituído por quatro seções: a primeira seção apresenta questões sobre os usuários; a segunda seção aborda questões acerca das tarefas; a terceira seção discute o ambiente de uso; e, concluindo, a quarta seção apresenta questões acerca dos requisitos de usabilidade da aplicação. No Quadro 4.2, são apresentadas as questões deste roteiro.

**Quadro 4.2** – Roteiro Básico de Questões para Entrevista e Reunião.

Seção	Questão
Usuários	Quais são os tipos de usuários e demais <i>stakeholders</i> envolvidos de forma direta ou indireta com a aplicação <i>Web</i> em questão?
	Quais são as responsabilidades, expectativas, preferências e habilidades dos usuários em relação à aplicação <i>Web</i> ?
	Quais são as necessidades dos usuários quanto à aplicação <i>Web</i> ?
	Como a aplicação <i>Web</i> pode auxiliar os usuários no suporte às tarefas que realizam?
Tarefas	Que tarefas deverão ser realizadas através da aplicação <i>Web</i> ?
	Que tipo de usuário vai realizar qual ou quais tarefas na aplicação <i>Web</i> ?
	Quais são os objetivos e resultados das tarefas a serem suportadas pela aplicação <i>Web</i> ?
	Qual é a estrutura, dependência, duração estimada e frequência de realização das tarefas na aplicação <i>Web</i> ?
	Como a aplicação <i>Web</i> deve apoiar a realização das tarefas pelos usuários?
Ambiente	Quais são as características do ambiente de uso em torno das tarefas realizadas pelos usuários?
Requisitos de Usabilidade	Quais são as diretrizes gerais de usabilidade para as interfaces da aplicação <i>Web</i> ?
	Quais são os requisitos específicos de usabilidade para as interfaces da aplicação <i>Web</i> ?

**Fonte:** Esta Pesquisa.

Após a realização da atividade de Análise e Especificação do Contexto de Uso, deve ser executada a atividade de Especificação de Requisitos de Usabilidade. Esta atividade visa definir com mais detalhes os requisitos de usabilidade específicos da aplicação. O não estabelecimento desses requisitos, também considerados como parâmetros de qualidade em termos de usabilidade, dificulta o atendimento do nível de qualidade de uso pretendido para as interfaces da aplicação *Web*.

Os requisitos definidos durante a atividade de Especificação de Requisitos de Usabilidade correspondem a níveis de qualidade que devem ser alcançados pelos usuários quando da interação com as interfaces da aplicação *Web*. Essa especificação, quando feita no início do processo de desenvolvimento, torna possível o monitoramento dos requisitos de usabilidade, de modo a determinar se as interfaces construídas estão efetivamente atendendo os parâmetros de qualidade de uso exigidos.

Como forma de auxiliar esta atividade, foi definida a Tabela de Especificação de Requisitos de Usabilidade. Esta tabela é composta pelas seguintes informações: ID (identificador); atributo de usabilidade; tipo de usuário; instrumento de medida; valor a ser medido; nível de desempenho almejado. Os requisitos específicos definidos na Tabela 4.1 constituem informações importantes acerca da usabilidade da aplicação.

**Tabela 4.1** – Tabela de Especificação de Requisitos de Usabilidade.

<b>ID</b>	<b>Atributo de usabilidade</b>	<b>Tipo de usuário</b>	<b>Instrumento de medida</b>	<b>Valor a ser medido</b>	<b>Nível de desempenho almejado</b>

**Fonte:** Esta Pesquisa.

Os elementos de informações presentes na Tabela de Especificação de Requisitos de Usabilidade (Tabela 4.1) são discutidos, a seguir:

- **ID:** diz respeito a um identificador utilizado para referenciar cada requisito de usabilidade (linha da tabela);
- **Atributo de usabilidade:** corresponde a características de usabilidade definidas como parâmetros de qualidade de uso, a serem empregadas como critérios de avaliação de interfaces. Facilidade de aprendizado, desempenho e satisfação são alguns possíveis atributos. Outros atributos de usabilidade foram discutidos na seção 2.2;
- **Tipo de Usuário:** refere-se aos tipos ou perfis de usuários que vão realizar as tarefas através das interfaces e, para as quais, serão avaliados os atributos de usabilidade;
- **Instrumento de medida:** diz respeito ao mecanismo empregado para coletar os valores de um atributo de usabilidade específico. Este instrumento normalmente é quantitativo, isto é, medido numericamente. Entretanto, a medida pode ser objetiva e subjetiva. Os termos objetivo e subjetivo correspondem ao modo como os dados são coletados. As medidas são objetivas quando os dados são obtidos através do desempenho do usuário na realização de tarefas e subjetivas quando os dados são coletados por meio de questionários de preferências e de satisfação. Alguns exemplos de instrumento de medida incluem: a realização de alguma tarefa, a remoção de algum dado em uma dada tarefa, respostas a perguntas em um questionário, entre outros;
- **Valor a ser medido:** corresponde ao tipo de dado ou medida que deverá ser coletada sobre os atributos de usabilidade. Alguns possíveis tipos incluem: porcentagem de tarefas completadas com sucesso, número de vezes que o usuário solicitou ajuda e tempo de realização de tarefas. Outras medidas que podem ser utilizadas são (SHNEIDERMAN; PLAISANT, 2009): tempo de aprendizagem, isto é, tempo requerido para um usuário típico aprender a usar as funções do sistema; desempenho, que se refere ao tempo necessário para um usuário realizar uma tarefa; taxa de erros, ou seja, número de erros cometidos pelo usuário ao executar uma tarefa; satisfação subjetiva, isto é, grau de satisfação do usuário ao realizar tarefas e utilizar as interfaces do sistema;

- **Nível de desempenho almejado:** diz respeito a metas de desempenho entendidas como alvo para os atributos de usabilidade quando da realização de tarefas e interação com as interfaces da aplicação. Espera-se que este nível seja alcançado para todos os atributos de usabilidade específicos definidos na Tabela de Especificação de Requisitos de Usabilidade (Tabela 4.1).

As informações obtidas e produzidas durante as atividades de Análise e Especificação do Contexto de Uso e Especificação de Requisitos de Usabilidade devem ser documentadas no artefato Contexto de Uso e Requisitos de Usabilidade. Este artefato, que está disponível no APÊNDICE A desta dissertação, é constituído de três seções principais, são elas:

- **Primeira seção:** apresenta informações acerca do objetivo do artefato e sobre termos (siglas) empregados no projeto da aplicação *Web*;
- **Segunda seção:** reúne diferentes tipos de informações, a exemplo de visão (descrição dos objetivos da aplicação *Web*), metodologia (quais recursos serão utilizados para coletar as informações), programação de atividades (diz respeito ao cronograma de realização das atividades), usuários (quais são os tipos, expectativas e responsabilidades, etc.), tarefas (tipos e descrições detalhadas) e ambiente de uso;
- **Terceira seção:** apresenta informações sobre diretrizes gerais de usabilidade e requisitos de usabilidade específicos da aplicação *Web*.

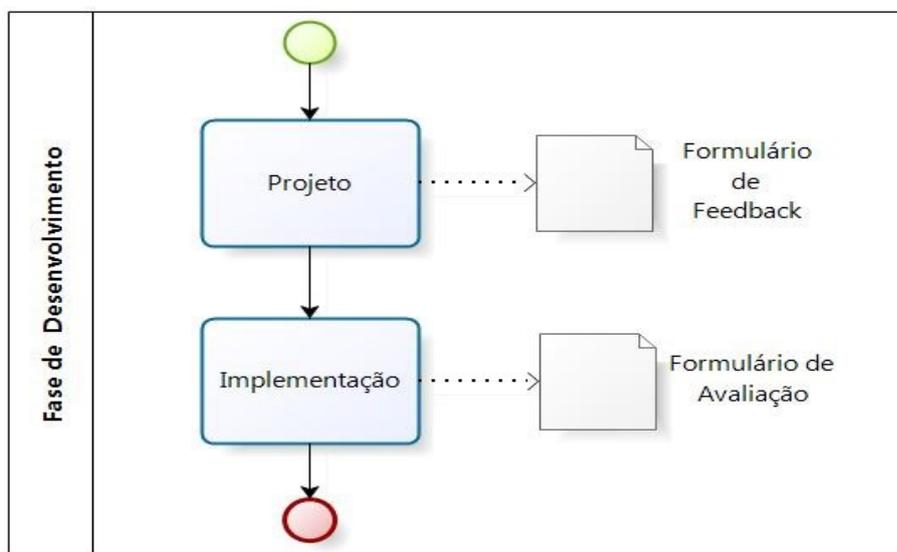
O artefato Contexto de Uso e Requisitos de Usabilidade fornece suporte às atividades da Fase de Planejamento, sendo de grande importância para o registro de informações sobre o contexto de uso e usabilidade pretendida para a aplicação *Web*. Tais informações são utilizadas pela equipe de desenvolvimento nas demais fases do *Framework* UEF-WEB. Na próxima seção, será apresentada a Fase de Desenvolvimento.

#### 4.2.2 Fase de Desenvolvimento

Após a Fase de Planejamento e consequente obtenção das informações acerca do contexto de uso e dos requisitos de usabilidade da aplicação *Web*, deve ser realizada a Fase de Desenvolvimento que tem por objetivos projetar e construir as interfaces da aplicação. As atividades de Projeto e de Implementação presentes nesta fase são atividades típicas da

Engenharia de *Software* relacionadas a aspectos de modelagem e de construção, mas que terão o apoio de recursos de usabilidade. Na Figura 4.3, são apresentadas as duas atividades, bem como os dois artefatos desta fase.

**Figura 4.3** – Atividades e Artefatos da Fase de Desenvolvimento do *Framework* UEF-WEB.



**Fonte:** Esta Pesquisa.

As soluções de projeto de interfaces humano-computador devem ser desenvolvidas com base na descrição do contexto de uso, no conhecimento acerca do domínio da aplicação, nas orientações e princípios de usabilidade. No contexto do *Framework* UEF-WEB, a produção de soluções de projeto envolve o projeto de tarefas do usuário, o projeto de interação do usuário com a aplicação e o projeto de interface de usuário.

Um projeto adequado de tarefas e de interação do usuário com a aplicação pode ser alcançado a partir da compreensão do contexto de uso, dos papéis dos usuários, de suas tarefas e de seus resultados. Basicamente, o projeto de interação envolve decidir como os usuários vão realizar as tarefas através das interfaces da aplicação. Em geral, este projeto inclui a atribuição de tarefas aos usuários e a outras partes da aplicação, a seleção de técnicas de diálogo apropriadas e o projeto da dinâmica de interação.

Para auxiliar no desenvolvimento de interfaces, foram propostos, nas últimas décadas, princípios, diretrizes e heurísticas de usabilidade para o projeto e avaliação de interfaces. Estas contribuições devem ser observadas com atenção pela equipe de desenvolvimento, pois fornecem uma configuração básica para o estabelecimento de interfaces com maior nível de usabilidade. Tais propostas já foram discutidas nos capítulos dois e três desta dissertação e encontram-se reunidas no Quadro 4.3.

**Quadro 4.3** – Princípios, Diretrizes e Heurísticas de Usabilidade para Interfaces.

Proposta	Descrição
<p>Heurísticas de Usabilidade (NIELSEN, 1994b)</p>	<p>Visibilidade do estado do sistema; concordância entre o sistema e o mundo real; controle e liberdade ao usuário; consistência e padrões; prevenção de erros; reconhecer ao invés de lembrar; flexibilidade e eficiência de uso; projeto minimalista e estético; reconhecimento, diagnóstico e recuperação de erros; ajuda e documentação.</p>
<p>Princípios de <i>Design</i> de Interfaces (DIX <i>et al.</i>, 2004)</p>	<p>Facilidade de aprendizado: previsibilidade; capacidade de sintetização; familiaridade; consistência; generalização. Flexibilidade: iniciativa de diálogo; <i>multithreading</i>; migração de atividades; substitutividade; personalização. Robustez: observabilidade; recuperabilidade; capacidade de resposta; conformidade de realização de atividades.</p>
<p>Diretrizes de Projeto Pragmáticas para Aplicações <i>Web</i> (NIELSEN; LORANGER, 2007)</p>	<p>Não force o usuário a ler grandes quantidades de texto; evite sinais de “em construção”; informações importantes devem ser apresentadas dentro das dimensões de uma janela; menus de navegação e barras no topo devem ser projetados de modo consistente; a estética não deve substituir a funcionalidade; as opções de navegação devem ser óbvias, mesmo para usuários casuais.</p>
<p>Diretrizes para Criação de Interfaces (SHNEIDERMAN; PLAISANT, 2009)</p>	<p>Navegabilidade da interface: padronize a sequência de tarefas; assegure que os <i>links</i> da página sejam descritivos; use cabeçalhos distintos para funcionalidades distintas; use caixas de múltipla escolha (<i>checkboxes</i>) para escolhas que possuem mais de uma resposta; permita que o usuário imprima a página adequadamente; use imagens pequenas para pré-visualizar as imagens originais. Organização do conteúdo exibido: consistência dos dados exibidos; facilidade de assimilação da informação pelo usuário; minimização da carga de memória do usuário; compatibilidade entre a entrada de dados e o conteúdo exibido; flexibilidade de acesso e uso de informações. Captura da atenção do usuário: intensidade dos elementos; marcação; tamanho; escolha das fontes; efeito de vídeo inverso; cores; áudio. Facilitação de entrada de dados: consistência das transações que exigem entrada de dados; minimização do número de ações a serem realizadas pelo usuário.</p>
<p>Princípios de Projeto de Sistemas Interativos (ISO, 2010)</p>	<p>Adequação à tarefa; autodescrição; conformidade com expectativas do usuário; adequação à aprendizagem; controlabilidade; tolerância a erro; adequação à individualização.</p>
<p>Diretrizes para Interfaces de Aplicações de Dispositivos Móveis (NETO, 2013)</p>	<p>Destaque a principal atividade da aplicação e garanta os subsídios necessários para que o usuário complete qualquer tarefa; invista os maiores esforços nos fatores da aplicação mais importantes do ponto de vista do usuário; pense no <i>design</i> da interface como uma atividade a ser preenchida de cima para baixo; disponibilize um caminho lógico para o usuário; torne a interação fácil e óbvia; facilite a entrada de dados; estimule a conectividade e o comportamento colaborativo; torne a interface mais realista possível; forneça suporte à mudança de orientação; mantenha o usuário ciente de qualquer ação; forneça controle ao usuário; crie uma página de ajuda; aposte em um <i>design</i> minimalista; use imagens e gráficos em alta definição e editados profissionalmente; use componentes na medida adequada.</p>

**Fonte:** Esta Pesquisa.

Durante a atividade de Projeto da aplicação *Web*, o *Framework* UEF-WEB estabelece a utilização do recurso de Prototipação. O uso de protótipos, em um estágio inicial do processo de desenvolvimento, é de grande importância, pois permite não somente explorar soluções alternativas junto aos representantes dos usuários, mas também avaliar o mais cedo possível estas soluções visando reduzir os riscos inerentes ao projeto e os encargos de mudanças futuras.

As avaliações, em suas diversas formas, constituem uma atividade relevante na Engenharia de Usabilidade, visto que, há a necessidade de se verificar a qualidade das soluções que envolvem a interação com o ser humano. Ciente de que o ciclo de desenvolvimento de *software* pode ser longo, recomenda-se realizar avaliações de usabilidade no decorrer deste processo para reduzir a ocorrência de problemas e diminuir custos de produção. Como não se consegue modelar de forma satisfatória o comportamento do ser humano, devido a sua complexidade, o uso de protótipos com a participação dos usuários pode auxiliar no processo de concepção e validação das soluções produzidas.

O recurso de Prototipação conta com o suporte do artefato Formulário de *Feedback*. Este artefato tem por objetivo auxiliar na coleta de informações e opiniões dos representantes dos usuários sobre as interfaces que estão sendo projetadas. A obtenção destas informações é importante para o prosseguimento do projeto, pois contribui para esclarecer dúvidas, identificar e resolver problemas de usabilidade em um estágio precoce do processo de criação de interfaces. No Quadro 4.4, é apresentado o artefato Formulário de *Feedback*.

**Quadro 4.4** – Artefato Formulário de *Feedback* do *Framework* UEF-WEB.

Tipo de usuário	Identificação da interface	Descrição do problema/necessidade/sugestão	Nível de prioridade (baixo, médio ou alto)

Fonte: Esta Pesquisa.

Como percebido no Quadro 4.4, o artefato Formulário de *Feedback* é constituído pelas seguintes informações:

- **Tipo de usuário:** diz respeito aos tipos ou perfis de usuários que realizam as tarefas através das interfaces e, sobre as quais, são coletadas diversas informações;
- **Identificação da interface:** corresponde a uma descrição da interface que está sendo analisada. Esta descrição deve incluir informações sobre a interface utilizada e também sobre a tarefa que está sendo realizada pelo usuário;
- **Descrição do problema/necessidade/sugestão:** nesta parte do formulário, os usuários devem relatar os problemas, necessidades ou sugestões de melhorias para as interfaces analisadas;
- **Nível de prioridade (baixo, médio ou alto):** diz respeito ao nível de prioridade atribuído pelo usuário para correção de um problema, atendimento de uma necessidade ou sugestão de melhoria na interface da aplicação.

As informações acerca de problemas, necessidades e sugestões de melhorias nas interfaces obtidas junto aos usuários, por meio do recurso de Prototipação e do artefato Formulário de *Feedback*, devem ser analisadas com atenção pela equipe de desenvolvimento no intuito de sanar dúvidas existentes, suprir novas necessidades, acolher sugestões de melhorias e proceder na correção dos problemas de usabilidade identificados. A realização da atividade de Projeto com o apoio dos recursos de usabilidade apresentados é de fundamental importância porque permite identificar e resolver de forma antecipada diversos problemas associados à qualidade de uso das interfaces.

Concluída a atividade de Projeto, deve ser realizada a atividade de Implementação que, basicamente, consiste na construção das interfaces da aplicação pela equipe de desenvolvimento. Esta atividade deve ser executada com base nas informações obtidas e produzidas durante as atividades de Análise e Especificação do Contexto de Uso, Especificação de Requisitos de Usabilidade e Projeto, pois fornecem orientações e subsídios visando o desenvolvimento de interfaces com maior qualidade de uso.

Para auxiliar na identificação de problemas de usabilidade nas versões de interfaces construídas na atividade de Implementação, o *Framework* UEF-WEB estabelece o uso do recurso de Inspeção de Usabilidade Baseada em Perspectivas. Como visto na subseção

2.3.2.4, esta técnica, em cada sessão de inspeção, foca em um conjunto de questões de usabilidade sob o ponto de vista de uma perspectiva de usabilidade.

No *Framework* UEF-WEB, a utilização deste recurso é feita através da técnica de inspeção desenvolvida por Conte (2009). Esta técnica, denominada de WDP (*Web Design Perspectives-Based Usability Evaluation*), combina três perspectivas de projeto *Web* (apresentação, conceituação e navegação) com as dez heurísticas de usabilidade definidas por Nielsen (1994b). A escolha da técnica WDP foi feita pelos seguintes motivos: a avaliação heurística é um dos principais métodos de inspeção de aplicações interativas, é eficiente e possui uma boa relação custo/benefício (NASCIMENTO; AMARAL, 2010); o uso de perspectivas de projeto *Web* serve de guia para interpretar as heurísticas com foco em aplicações *Web* (CONTE, 2009). Além disso, estudos de caso realizados por Conte (2009) forneceram indícios de eficácia, de eficiência e de viabilidade de uso desta técnica em ambiente industrial. No Quadro 4.5, são descritas as três perspectivas de projeto *Web* da técnica WDP e, na Tabela 4.2, são apresentadas as relações entre estas três perspectivas e as dez heurísticas de usabilidade.

**Quadro 4.5** – Perspectivas de Projeto *Web* e o Foco na Usabilidade.

Perspectiva de Projeto <i>Web</i>		Foco na Usabilidade
<b>Apresentação (A)</b>	Representa as características relativas à programação visual e ao <i>layout</i> da interface definindo, por exemplo, como as informações serão apresentadas aos usuários.	Foco na consistência da apresentação de informações ao usuário. Sob essa perspectiva, a usabilidade é satisfatória caso a programação visual e o <i>layout</i> da interface permitam ao usuário realizar suas tarefas de maneira eficaz, eficiente e agradável.
<b>Conceituação (C)</b>	Representa os elementos conceituais (negócio, problema, etc.) que compõem o domínio da aplicação.	Foco na clareza e concisão dos elementos do domínio do problema. Sob essa perspectiva, a usabilidade é satisfatória se a representação é facilmente compreendida pelo usuário e não o leve a cometer erros por causa de termos ambíguos, inconsistentes ou desconhecidos.
<b>Navegação (N)</b>	Representa o espaço navegacional, definindo os elementos de acesso e suas associações usados na exploração das informações.	Foco na acessibilidade das funcionalidades do sistema pelos diferentes tipos de usuários. Sob essa perspectiva, a usabilidade é satisfatória se as opções de navegação do sistema permitem aos usuários realizar suas tarefas de maneira eficaz, eficiente e agradável.

**Fonte:** Conte (2009).

**Tabela 4.2** – Relação entre Heurísticas e Perspectivas na Técnica WDP.

Heurísticas (NIELSEN, 1994b)		Relacionadas com as Perspectivas:		
		Apresentação (A)	Conceituação (C)	Navegação (N)
1	Visibilidade do estado do sistema	A.1	C.1	
2	Concordância entre o sistema e o mundo real	A.2	C.2	
3	Controle e liberdade ao usuário			N.3
4	Consistência e padrões	A.4	C.4	
5	Prevenção de erros	A.5		N.5
6	Reconhecer ao invés de lembrar	A.6	C.6	
7	Flexibilidade e eficiência de uso	A.7		N.7
8	Projeto minimalista e estético	A.8		
9	Reconhecimento, diagnóstico e recuperação de erros	A.9	C.9	N.9
10	Ajuda e documentação	A.10	C.10	N.10

**Fonte:** Conte (2009).

No que concerne à perspectiva de apresentação, o inspetor deve estar com a atenção focada no que ele está vendo e como isso é apresentado. As questões que orientam o inspetor a avaliar a usabilidade da aplicação a partir desta perspectiva são discutidas a seguir (CONTE, 2009):

#### **A.1 – Visibilidade do estado do sistema**

- Avalie se o estado do sistema está sempre visível para o usuário.

#### **A.2 – Concordância entre o sistema e o mundo real**

- Avalie se a informação e as opções do sistema estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema.

#### **A.4 – Consistência e padrões**

- Avalie se a terminologia, gráficos e símbolos da interface estão consistentes.
- Avalie se há aderência a convenções de plataforma e padrões de interface adotados em relação a *layout*, formatação e controles.
- Avalie se há consistência de interfaces para tarefas que são equivalentes.

- Avalie se as mensagens de erro são apresentadas de forma consistente com os padrões de apresentação utilizados.

#### **A.5 – Prevenção de erros**

- Avalie se os dados obrigatórios na entrada de dados estão claramente definidos.
- Avalie se a interface indica o formato correto para uma entrada de dados específica.
- Avalie se as informações são apresentadas de forma balanceada e na ordem natural do domínio do problema.
- Avalie se a interface facilita a distinção entre tarefas e dados diferentes.

#### **A.6 – Reconhecer ao invés de lembrar**

- Avalie se é fácil reconhecer/visualizar a opção que deve ser usada para atingir o objetivo desejado.
- Avalie se a interface do sistema permite ao usuário visualizar informações chave durante a realização de uma tarefa.
- Avalie se é fácil reconhecer/visualizar dados já fornecidos.

#### **A.7 – Flexibilidade e eficiência de uso**

- Avalie se a disposição dos elementos da interface do sistema aumenta a eficiência de uso, minimizando o esforço de ações físicas.
- Avalie se a disposição dos elementos da interface minimiza o esforço em buscas visuais.
- Avalie se a interface apoia tarefas específicas frequentemente repetidas.
- Avalie se a interface facilita a entrada de dados, sejam eles simples ou complexos estruturalmente.
- Avalie se a interface permite a utilização de mecanismos de busca de informação que auxiliem a entrada de dados obrigatória (por exemplo, busca de CEP).

#### **A.8 – Projeto minimalista e estético**

- Avalie se a interface apresenta em destaque as informações relevantes para a tarefa que está sendo executada.
- Avalie se a interface não apresenta em destaque informações irrelevantes.

- Avalie se a interface agride visualmente o usuário (por exemplo, através do uso de cores que provocam desconforto físico).
- Avalie se a interface possui informação ilegível ou que cause dificuldade para a sua leitura (por exemplo, devido ao uso de fontes muito pequenas ou devido à falta de contraste).

#### **A.9 – Reconhecimento, diagnóstico e recuperação de erros**

- Avalie se as mensagens de erros estão claramente visíveis para o usuário.

#### **A.10 – Ajuda e documentação**

- Avalie se as informações e opções de ajuda e documentação estão claramente visíveis para o usuário.
- Avalie se as informações de ajuda e documentação estão apresentadas de uma forma que auxilie a realização de tarefas de forma eficaz, como em uma lista de passos concretos.

Em relação à perspectiva conceitual, o inspetor deve observar se compreende o que está sendo apresentado. Para avaliar a usabilidade da aplicação através desta perspectiva, as seguintes questões devem ser utilizadas (CONTE, 2009):

#### **C.1 – Visibilidade do estado do sistema**

- Avalie se o estado do sistema está expresso em uma representação facilmente compreendida pelo usuário – utilizando símbolos, palavras ou frases que façam parte do domínio do problema.

#### **C.2 – Concordância entre o sistema e o mundo real**

- Avalie se a interface do sistema utiliza a linguagem do domínio do problema.
- Avalie se os termos (palavras ou símbolos) utilizados pelo sistema seguem as convenções do mundo real.

#### **C.4 – Consistência e padrões**

- Avalie se os termos do domínio do problema são apresentados de forma consistente ao usuário.

#### **C.6 – Reconhecer ao invés de lembrar**

- Avalie se é possível reconhecer o significado dos termos (palavras ou símbolos) utilizados.

- Avalie se a definição de termos (palavras ou símbolos) específicos do domínio do problema está acessível aos usuários.

### **C.9 – Reconhecimento, diagnóstico e recuperação de erros**

- Avalie se as mensagens de erros utilizam linguagem simples de ser entendida pelos usuários, com conceitos do domínio do problema e de acordo com o perfil do usuário.
- Avalie se as mensagens de erros contém uma indicação de recuperação ou procedimento a ser executado que possa ser compreendido pelo usuário.
- Avalie se as soluções alternativas apresentadas nas mensagens de erro estão claras para o usuário.

### **C.10 – Ajuda e documentação**

- Avalie se o usuário compreende a ajuda fornecida pelo sistema.
- Avalie se o sistema provê ajuda ou direcionamento para execução de tarefas complexas ou entendimento de conceitos não triviais.

No que diz respeito à perspectiva navegação, o inspetor deve estar com a atenção focada em saber se ele consegue acessar as diferentes opções do sistema. As questões que orientam o inspetor a avaliar a usabilidade da aplicação a partir desta perspectiva são apresentadas a seguir (CONTE, 2009):

### **N.3 – Controle e liberdade ao usuário**

- Avalie se a interface permite ao usuário funções de *undo* (desfazer) e *redo* (refazer) ou funções similares que permitam ao usuário utilizar saídas de emergência em caso de escolhas erradas ou para sair de um estado (ou local) não esperado.
- Avalie se a interface facilita o retorno para o fluxo principal de uma tarefa após a execução de um desvio ou tarefa secundária.

### **N.5 – Prevenção de erros**

- Avalie se a interface previne a ocorrência de erros de navegação, ou seja, se as opções de navegação disponíveis definem claramente que resultados ou estados serão atingidos.

#### **N.7 – Flexibilidade e eficiência de uso**

- Avalie se a interface permite ao usuário navegar com facilidade pelos diferentes passos de uma tarefa.
- Avalie se a interface permite ao usuário diferentes formas de acesso às tarefas principais.
- Avalie se a interface permite ao usuário a utilização de aceleradores ou atalhos na interação com as tarefas principais.
- Avalie se as formas de acesso oferecidas pelo sistema minimizam o esforço de ações físicas.

#### **N.9 – Reconhecimento, diagnóstico e recuperação de erros**

- Avalie se o sistema mostra como acessar as soluções alternativas apresentadas nas mensagens de erro.

#### **N.10 – Ajuda e documentação**

- Avalie se o usuário é capaz de acessar facilmente a informação de ajuda e documentação relevante para a tarefa desejada.

Para apoiar o recurso de Inspeção de Usabilidade Baseada em Perspectivas, especificamente a aplicação da técnica WDP pela equipe de desenvolvimento, o *Framework* UEF-WEB fornece o artefato Formulário de Inspeção apresentado na Tabela 4.3.

**Tabela 4.3** – Artefato Formulário de Inspeção do *Framework* UEF-WEB.

<b>Avaliador</b>	<b>Tarefa</b>	<b>Classificação do problema (Perspectiva.Heurística)</b>	<b>Descrição do problema</b>	<b>Severidade (cosmético, leve, grave ou catastrófico)</b>

**Fonte:** Esta Pesquisa.

A seguir, são discutidos os elementos de informações presentes no artefato Formulário de Inspeção (Tabela 4.3):

- **Avaliador:** integrante da equipe de desenvolvimento que está realizando a inspeção de usabilidade na aplicação;
- **Tarefa:** descrição da tarefa e da interface da aplicação que está sendo inspecionada;
- **Classificação do problema (Perspectiva.Heurística):** classificação do problema de usabilidade através da relação Perspectiva.Heurística (ver Tabela 4.2 para maiores detalhes);
- **Descrição do problema:** descrição do problema/violação de usabilidade identificado na aplicação;
- **Severidade (cosmético, leve, grave ou catastrófico):** atribuição do nível de severidade (gravidade) do problema de usabilidade identificado. Estes níveis de severidade foram propostos por Nielsen (1994b) e apresentados na subseção 2.3.2.3 deste trabalho.

A realização de inspeções de usabilidade no *Framework* UEF-WEB deve ser feita por meio deste roteiro de atividades: **(i) sessões de inspeção individuais** (geralmente de 1 a 2 horas), nas quais os inspetores avaliam a conformidade das interfaces em relação às perspectivas de projeto *Web* e heurísticas discutidas, registram os problemas de usabilidade, analisam a gravidade destes e geram um relatório individual de inspeção (artefato Formulário de Inspeção); **(ii) consolidação das inspeções**, atividade desenvolvida através de uma reunião entre os inspetores que busca apresentar um relatório consolidado sobre os problemas de usabilidade encontrados; **(iii) seleção dos problemas a serem corrigidos**, que consiste na análise de custo/benefício das correções para os problemas de usabilidade identificados.

No *Framework* UEF-WEB, os próprios integrantes da equipe de desenvolvimento realizam as inspeções na aplicação após receber treinamento sobre a técnica de inspeção de usabilidade baseada em perspectivas WDP. Este fato reduz a dependência de especialistas em usabilidade no processo de avaliação das interfaces da aplicação. É importante destacar que os requisitos específicos de usabilidade definidos na Fase de Planejamento também devem ser analisados durante as inspeções de usabilidade para verificar se estão sendo atendidos.

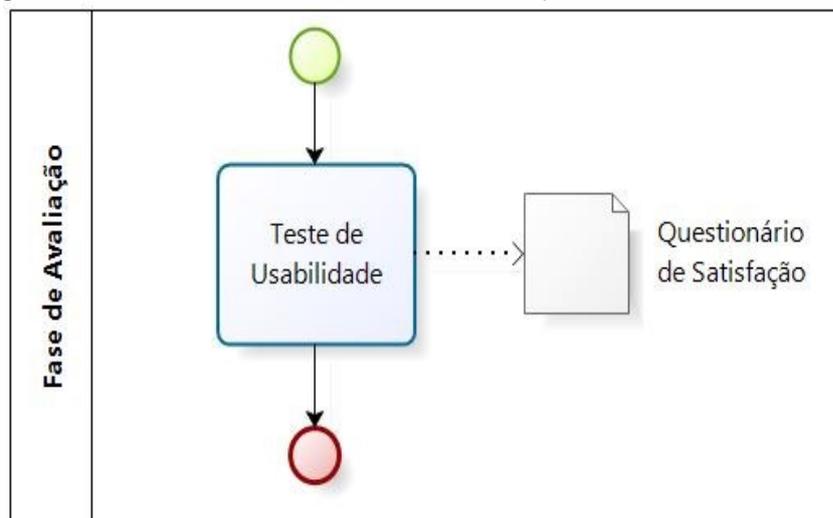
Como visto, a Fase de Desenvolvimento tem por objetivo auxiliar as atividades de Projeto e de Implementação das interfaces de uma aplicação *Web* através da utilização de um

conjunto de recursos de usabilidade. Estes recursos, em suma, são empregados visando elevar a qualidade de uso das interfaces que estão sendo desenvolvidas. Na próxima seção, será apresentada a Fase de Avaliação do *Framework* UEF-WEB.

### 4.2.3 Fase de Avaliação

A Fase de Avaliação do *Framework* UEF-WEB tem por objetivo mensurar a satisfação dos usuários ao realizar tarefas e interagir com as interfaces da aplicação *Web*. Nesta fase, é realizada a atividade de Teste de Usabilidade como forma de avaliar, sob o ponto de vista dos usuários, as interfaces desenvolvidas. A coleta destas informações é realizada com o apoio do recurso de Questionário e através do artefato Questionário de Satisfação. Na Figura 4.4, é apresentada a atividade, bem como o artefato utilizado nesta fase.

**Figura 4.4** – Atividade e Artefato da Fase de Avaliação do *Framework* UEF-WEB.



**Fonte:** Esta Pesquisa.

A atividade de Teste de Usabilidade consiste em convidar um grupo de usuários para realizar as tarefas que foram projetadas e implementadas na aplicação *Web*. O objetivo desta atividade é colocar os usuários em contato direto com a aplicação desenvolvida para que possam interagir e, depois, relatar suas percepções, bem como nível de satisfação acerca das interfaces construídas. No contexto do *Framework* UEF-WEB, esta atividade se apresenta de forma simplificada, pois não demanda a realização de inúmeras atividades como, por exemplo, planejamento, observação e registro da interação dos usuários com a aplicação, normalmente requeridas em um típico teste de usabilidade.

A coleta de informações dos usuários sobre a sua satisfação em relação às interfaces da aplicação deve ser feita após a atividade de Teste de Usabilidade e com o auxílio do recurso de Questionário. Na Engenharia de Usabilidade, os questionários podem ser utilizados tanto para reunir informações acerca do perfil dos usuários quanto para mensurar o grau de satisfação destes ao interagir com as interfaces de um produto de *software*. Como visto na subseção 2.3.2.6, diversos questionários de satisfação foram propostos ao longo dos anos, a exemplo dos questionários SUMI, WAMMI, QUIS, ISONORM, ErgoList e SUS.

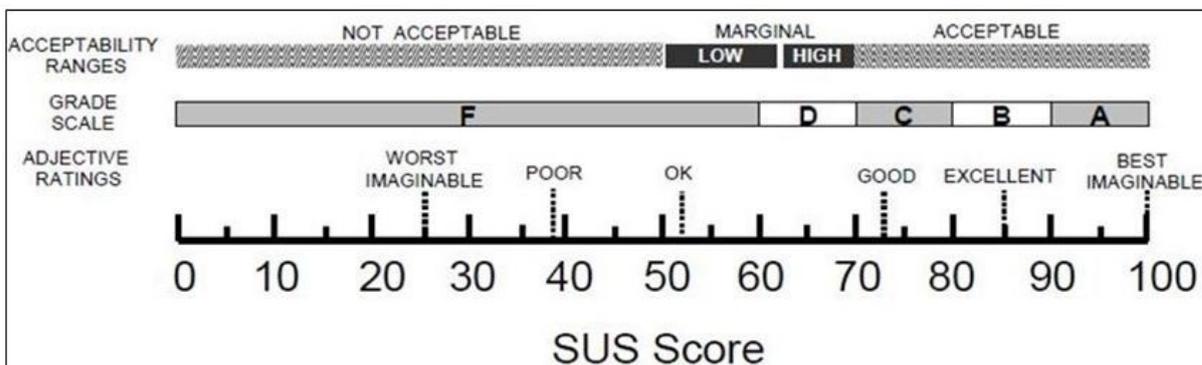
O questionário SUS (*System Usability Scale*) proposto por Brooke (1986) foi escolhido para ser utilizado no *Framework* UEF-WEB pelas seguintes razões: (i) dentre os questionários gratuitos disponíveis, é o mais recomendado (USABILITYNET, 2014); (ii) é um dos questionários mais aceitos pela confiabilidade e validade (TULLIS; STETSON, 2004; BANGOR; KORTUM; MILLER, 2008); (iii) é destinado a avaliar a usabilidade e a satisfação dos usuários em relação a sistemas e produtos (BANGOR; KORTUM; MILLER, 2009); (iv) é rápido e simples de aplicar, pois é composto por apenas 10 questões de fácil entendimento; (v) as suas questões refletem grande parte dos princípios, diretrizes e heurísticas de usabilidade apresentados, a exemplo dos cinco componentes de qualidade de uso (facilidade de aprendizado, eficiência de uso, facilidade de memorização, geração de poucos erros e satisfação dos usuários) definidos por Nielsen (2003).

O artefato Questionário de Satisfação disponibilizado no *Framework* UEF-WEB é o questionário SUS mais algumas questões fechadas e abertas para coletar informações acerca do perfil e da percepção dos usuários, respectivamente. O questionário SUS é composto por 10 questões e utiliza escala *Likert* de cinco pontos com valores entre 1 (discordo completamente), 2 (discordo), 3 (neutro), 4 (concordo) e 5 (concordo completamente). A pontuação do questionário SUS é calculada da seguinte forma (BROOKE, 1986):

- Cada questão possui valores de 0 a 4;
- A contribuição das questões ímpares (1, 3, 5, 7 e 9) é calculada pela posição na escala *Likert* menos 1;
- A contribuição das questões pares (2, 4, 6, 8 e 10), por sua vez, é de 5 menos a posição na escala *Likert*;
- A soma das contribuições das 10 questões é multiplicada por 2,5 para obter a pontuação final. No questionário SUS, a pontuação final (*score*) de usabilidade conferida a um sistema ou produto varia de 0 a 100.

A pesquisa realizada por Bangor, Kortum e Miller (2009) apresenta uma comparação entre três escalas de classificação (*Acceptability Ranges*, *Grade Scale* e *Adjective Ratings*) e a pontuação do questionário SUS. Esta comparação entre escalas é ilustrada na Figura 4.5.

**Figura 4.5** – Escalas de Classificação e a Pontuação do SUS.



**Fonte:** Bangor, Kortum e Miller (2009).

Com base na Figura 4.5, observa-se que na escala Intervalo de Aceitabilidade (*Acceptability Ranges*) um sistema ou produto com pontuação a partir de 70 pontos é aceitável e abaixo de 50 não é aceitável. Na classificação tradicional Escala de Notas (*Grade Scale*), um sistema ou produto é aceito quando recebe as notas A (de 90 a 100), B (de 80 a 89) ou C (de 70 a 79). Na Classificação Adjetiva (*Adjective Ratings*), um sistema ou produto avaliado com os adjetivos Melhor Imaginável (*Best Imaginable*), Excelente (*Excellent*) e Bom (*Good*) é aceito. A avaliação através do adjetivo OK, no entanto, pode suscitar dúvidas quanto à aceitação e necessidade de ajustes nas interfaces do sistema ou produto em análise. Ainda de acordo com a pesquisa de Bangor, Kortum e Miller (2009), a média de 70 pontos<sup>16</sup> tem se mantido em diferentes aplicações do SUS. Diante disso, esta média de pontuação do questionário SUS será empregada como medida de usabilidade para avaliar a satisfação dos usuários em relação à aplicação *Web*.

No *Framework* UEF-WEB, será utilizada a versão traduzida para o português do questionário SUS. A versão original no idioma inglês deste questionário pode ser consultada no ANEXO A desta dissertação. No Quadro 4.6, é apresentado o artefato Questionário de Satisfação do *Framework* UEF-WEB.

<sup>16</sup> A média de 70 pontos no questionário SUS foi obtida a partir da aplicação de aproximadamente 3.500 *surveys* (questionários) em 273 estudos, que avaliaram a usabilidade e satisfação dos usuários em relação a uma ampla variedade de interfaces como, por exemplo, *sites*, telefones celulares, GUI, *hardware* e interfaces de usuários de TV. Em todos os estudos, os participantes realizaram uma amostra representativa de tarefas com o produto em questão (normalmente em um teste de usabilidade) e, após isso, responderam o questionário SUS (BANGOR; KORTUM; MILLER, 2009).

**Quadro 4.6** – Artefato Questionário de Satisfação do *Framework* UEF-WEB – Baseado no Questionário SUS (*System Usability Scale*) de Brooke (1986).

<p>Prezados, este questionário tem por objetivo avaliar a usabilidade e a satisfação dos usuários em relação a uma aplicação <i>Web</i>. Na Parte I do questionário, são coletadas informações básicas acerca do perfil dos usuários. Na Parte II do questionário, é realizada a avaliação da aplicação <i>Web</i> através das respostas fornecidas pelos usuários. Na Parte III do questionário, são apresentadas duas questões abertas para coletar informações dos usuários sobre características positivas e negativas da aplicação <i>Web</i>. Para cada questão da Parte I e Parte II do questionário, marque apenas uma resposta. Para as questões da Parte III, dê a sua opinião sobre características positivas e negativas observadas na aplicação <i>Web</i>.</p> <p>Desde já, agradecemos a participação dos usuários e afirmamos que as informações fornecidas a este questionário são completamente confidenciais. Informamos também que os usuários não são obrigados a responder este questionário e podem retirar-se livremente a qualquer momento.</p>				
<b>Parte I – Perfil dos Usuários</b>				
<p>1. Sexo</p> <p><input type="radio"/> Masculino</p> <p><input type="radio"/> Feminino</p>	<p>2. Qual é a sua Idade?</p> <p><input type="radio"/> Abaixo de 18</p> <p><input type="radio"/> 18-29</p> <p><input type="radio"/> 30-39</p> <p><input type="radio"/> 40-49</p> <p><input type="radio"/> 50-59</p> <p><input type="radio"/> 60 ou mais</p>	<p>3. Como você avalia o seu nível de habilidade na Internet?</p> <p><input type="radio"/> Baixo</p> <p><input type="radio"/> Médio</p> <p><input type="radio"/> Alto</p> <p><input type="radio"/> Muito alto</p>		
<b>Parte II – Avaliação da Aplicação <i>Web</i> pelos Usuários</b>				
	Discordo Completamente		Concordo Completamente	
1. Eu acho que gostaria de usar esta aplicação <i>Web</i> frequentemente.	1	2	3	4 5
2. Eu achei a aplicação <i>Web</i> desnecessariamente complexa.	1	2	3	4 5
3. Eu achei a aplicação <i>Web</i> fácil de usar.	1	2	3	4 5
4. Eu acho que precisaria da ajuda de um técnico para poder usar esta aplicação <i>Web</i> .	1	2	3	4 5
5. Eu achei que as várias funções nesta aplicação <i>Web</i> estavam bem integradas.	1	2	3	4 5
6. Eu acho que houve muita inconsistência na aplicação <i>Web</i> .	1	2	3	4 5
7. Eu acho que a maioria das pessoas acharia esta aplicação <i>Web</i> fácil de usar.	1	2	3	4 5
8. Achei a aplicação <i>Web</i> muito complicada de usar.	1	2	3	4 5
9. Eu senti muita segurança usando a aplicação <i>Web</i> .	1	2	3	4 5
10. Eu precisava aprender muitas coisas antes de poder utilizar esta aplicação <i>Web</i> .	1	2	3	4 5
<b>Parte III – Percepção dos Usuários sobre a Aplicação <i>Web</i></b>				
1. Que características da aplicação <i>Web</i> você avalia como positivas e por quê?				
2. Que características da aplicação <i>Web</i> precisam ser melhoradas e por quê?				

Como visto no Quadro 4.6, o artefato Questionário de Satisfação do *Framework* UEF-WEB é constituído por três partes principais, sendo que:

- Na primeira parte do artefato (Parte I – Perfil dos Usuários), são apresentadas três questões fechadas com o intuito de coletar informações básicas sobre o perfil dos usuários. Estas informações são importantes, pois permitem observar a evolução do nível de satisfação dos usuários sob os critérios de orientação sexual, idade e nível de habilidade na Internet;
- Na segunda parte do artefato (Parte II – Avaliação da Aplicação *Web* pelos Usuários), são apresentadas as 10 questões fechadas do questionário SUS traduzidas para o português. Como já foi dito, estas questões foram propostas para avaliar a usabilidade e a satisfação dos usuários quanto ao uso de sistemas e produtos. No entanto, há uma peculiaridade na disposição das perguntas no questionário SUS. As questões de número ímpar são declarações positivas e as questões de número par são declarações negativas. Isso foi feito para prevenir viés de resposta decorrente da desatenção dos participantes ao responder cada uma das questões. Ao alternar questões positivas e negativas, os participantes precisam ler cada enunciado com maior atenção para determinar se concordam ou não com a declaração da questão (BROOKE, 1986);
- Na terceira parte do artefato (Parte III – Percepção dos Usuários sobre a Aplicação *Web*), são apresentadas duas questões abertas com o propósito de coletar informações dos usuários sobre aspectos e características positivas e negativas da aplicação *Web* em avaliação. Na primeira questão, os usuários são questionados a respeito de características na aplicação *Web* que julgam ser positivas. Na segunda questão, os usuários são perguntados sobre características negativas, isto é, aspectos que precisam ser melhorados na aplicação *Web*. No final de cada uma destas questões, os usuários também são convidados a fornecer mais detalhes sobre as respostas apresentadas (através da interrogação por quê?). Estas questões são importantes, pois registram de maneira subjetiva a percepção de satisfação ou insatisfação dos usuários sobre a aplicação *Web* avaliada.

Ainda sobre o questionário SUS, é possível observar a seguinte correlação entre questão e objetivo de análise: 1ª questão – Frequência de uso da aplicação; 2ª questão –

Complexidade da aplicação; 3ª questão – Facilidade de uso; 4ª questão – Assistência para usar a aplicação; 5ª questão – Integração de funções na aplicação; 6ª questão – Inconsistência da aplicação; 7ª questão – Rápida aprendizagem; 8ª questão – Aplicação incômoda e complicada de usar; 9ª questão – Segurança e confiança ao usar a aplicação; 10ª questão – Aprendizagem de outras informações para usar a aplicação.

Além da correlação entre questão e objetivo de análise, os componentes de qualidade de uso definidos por Nielsen (2003) são reconhecidos nas perguntas do questionário SUS, a saber: facilidade de aprendizado (questões 3, 4, 7 e 10), eficiência de uso (questões 5, 6 e 8), facilidade de memorização (questão 2), geração de poucos erros (questão 6) e satisfação dos usuários (questões 1, 4 e 9). Este fato mostra que as 10 questões presentes no SUS refletem grande parte dos princípios, recomendações, diretrizes e heurísticas de usabilidade discutidos nesta dissertação. Nesse sentido, cabe destacar que, além da pontuação final fornecida pelo questionário SUS, outra análise de usabilidade e satisfação dos usuários pode ser feita através da média das questões relacionadas aos componentes de qualidade de uso de Nielsen (2003), possibilitando observar também, por componente de qualidade de uso, o nível de satisfação dos usuários.

Desse modo, a Fase de Avaliação tem por objetivo mensurar a satisfação dos usuários com as interfaces desenvolvidas. Para atingir este objetivo, os usuários precisam, antes, realizar tarefas e interagir com a aplicação *Web* por meio da atividade de Teste de Usabilidade. A coleta de informações acerca da usabilidade e satisfação dos usuários é feita, por sua vez, através do artefato Questionário de Satisfação. Na próxima seção, serão apresentadas as considerações finais deste capítulo.

### **4.3 Considerações Finais do Capítulo**

Neste capítulo, foi apresentado o *Framework* UEF-WEB de apoio à Engenharia de Usabilidade para aplicações *Web* com o objetivo de auxiliar as organizações a introduzir, de maneira sistematizada e facilitada, recursos de usabilidade no processo de desenvolvimento de interfaces de aplicações *Web*. A ideia central deste *framework* é a execução evolutiva e com a participação dos usuários de um conjunto de fases, atividades, recursos e artefatos de suporte à Engenharia de Usabilidade. Tal estratégia, na qual o paradigma de desenvolvimento de interfaces parte de sucessivos ciclos de análise, construção e teste, busca reduzir os riscos de falhas conceituais de projeto e problemas durante o desenvolvimento, de modo a garantir

que, a cada ciclo, a aplicação *Web* corresponda cada vez melhor às expectativas e necessidades dos usuários em termos de qualidade de uso.

O *Framework* UEF-WEB é composto por três fases: Planejamento, Desenvolvimento e Avaliação. Cada uma destas fases possui objetivos específicos dentro do processo de desenvolvimento de interfaces e demanda a realização de atividades específicas. Estas atividades, por sua vez, são suportadas por recursos e artefatos providos pela Engenharia de Usabilidade. É importante destacar que, durante o processo de construção de um *software* ou aplicação *Web*, vários ciclos de atividades podem ser necessários sendo que, em cada ciclo, um subconjunto de atividades é realizado para atingir certo objetivo. Além disso, em alguns ciclos, algumas atividades podem ser executadas em maior ou menor grau de detalhes a depender do objetivo do ciclo de desenvolvimento em questão.

### CONSIDERAÇÕES FINAIS

Como se afirmou, a demanda por aplicações *Web* cresceu rapidamente para atender às variadas necessidades das pessoas e organizações na contemporaneidade. Esse crescimento tem sido observado não somente em relação à quantidade, mas, também, na diversidade e nível de complexidade das aplicações comumente utilizadas. As aplicações *Web*, na atualidade, tornaram-se fundamentais para empresas e governos ao fornecer meios efetivos de apresentar produtos e serviços a potenciais clientes e informações de utilidade pública aos cidadãos, respectivamente.

Assim, a crescente importância das aplicações *Web* para a sociedade como um todo tem requisitado, cada vez mais, o emprego de boas práticas no processo de desenvolvimento de *softwares*, no intuito destes serem entregues dentro do prazo e orçamento previamente definidos, com elevado grau de qualidade e facilidade de manutenção. À luz do discutido, o atendimento aos requisitos de usabilidade, em especial, constitui um desafio a mais e crítico para as organizações, tendo em vista os inúmeros problemas relativos à ergonomia das interfaces encontrados frequentemente nas aplicações *Web*. Estes problemas têm dificultado e, em muitos casos, inviabilizado o uso destas aplicações. Este fato é deveras preocupante, pois a usabilidade é considerada um dos critérios mais importantes para aceitação de aplicações interativas, especialmente de aplicações *Web*, sendo ainda apontada pelos usuários como o atributo de qualidade de *software* mais relevante.

Como se pôde constatar neste trabalho, diversas abordagens foram desenvolvidas para apoiar a Engenharia de Usabilidade, a exemplo de modelos de ciclo de vida, técnicas, métodos e ferramentas. Tais contribuições têm por finalidade principal auxiliar no desenvolvimento de soluções de interfaces de *software* mais consistentes e que respondam às expectativas dos usuários na realização de suas tarefas. No entanto, conforme se apresentou, apesar das inúmeras propostas, existem organizações que ainda não fazem uso de recursos de suporte à usabilidade em seus processos de desenvolvimento seja por desconhecimento, restrição de orçamento para investir em usabilidade e/ou dificuldades na seleção e utilização efetiva destes recursos.

Diante dessas dificuldades, esta dissertação de mestrado acadêmico propôs o *Framework* UEF-WEB para auxiliar as organizações no desenvolvimento de interfaces de aplicações *Web*. O *framework* em questão traz contribuições importantes para a Engenharia de *Software*, pois apresenta, através de um arcabouço sucinto de fases, atividades, recursos e artefatos, orientações metodológicas e práticas para a inserção facilitada e sistemática de recursos de suporte à usabilidade nas diversas etapas do processo de desenvolvimento de *software*, possibilitando a ampliação da qualidade de uso das interfaces computacionais desenvolvidas.

O *Framework* UEF-WEB foi pensado e projetado para ser utilizado de modo integrado a outros processos de desenvolvimento ou mesmo de maneira independente na construção da interação humano-computador. Este *framework*, entretanto, também pode ser utilizado pelas organizações para apoiar a usabilidade no processo de manutenção e/ou evolução de interfaces de aplicações *Web*. O desenvolvimento do *Framework* UEF-WEB, em suma, foi pautado pela pesquisa bibliográfica e análise crítica de contribuições relacionadas à Engenharia de Usabilidade oferecidas pelos teóricos visitados.

Diante do exposto, é possível concluir que o objetivo geral da pesquisa, referente à proposição de um *framework* de suporte à Engenharia de Usabilidade para as fases de planejamento, desenvolvimento e avaliação de interfaces de aplicações *Web*, foi atingido, orientado pelos objetivos específicos definidos para a pesquisa. O primeiro objetivo específico – conhecer os princípios, atividades e recursos norteadores da Engenharia de Usabilidade, assim como modelo de processo, *framework*, princípios e diretrizes de projeto de interfaces empregados no desenvolvimento de aplicações *Web* – foi contemplado no desenvolvimento dos capítulos 2 e 3; o segundo objetivo específico – especificar o *Framework* UEF-WEB de apoio à Engenharia de Usabilidade para as fases de planejamento, desenvolvimento e avaliação de interfaces de aplicações *Web* – foi realizado no capítulo 4.

A hipótese de trabalho desta dissertação, baseada no argumento de que a utilização de um *framework* de apoio à Engenharia de Usabilidade pode auxiliar as organizações a introduzir sistematicamente recursos de usabilidade no processo de desenvolvimento de aplicações *Web* e promover maior qualidade de uso para as interfaces construídas, foi validada através do suporte teórico da pesquisa, demonstrando que a adoção sistemática de princípios, atividades e recursos relativos à usabilidade contribui efetivamente para o desenvolvimento de interfaces computacionais com maior qualidade de uso.

Como possibilidade de trabalhos futuros, tem-se a realização de estudos de caso para verificar a aplicabilidade de utilização do *Framework* UEF-WEB em projetos reais de

desenvolvimento de interfaces, com o propósito de analisar a eficácia deste *framework* no suporte aos recursos de usabilidade e para a qualidade de uso das interfaces construídas. Dois indicadores principais podem ser utilizados nestes estudos: (i) a percepção de utilidade deste *framework* sob a perspectiva da equipe de desenvolvimento; (ii) a percepção dos usuários acerca da usabilidade e satisfação de uso com as interfaces de aplicações *Web* desenvolvidas. Outro possível trabalho é a construção da ferramenta “UEF-WEB Tool” para apoiar as fases, atividades, recursos e artefatos previstos no *Framework* UEF-WEB. Além de facilitar o emprego deste *framework* nas organizações, esta ferramenta constitui um instrumento para operacionalização do *Framework* UEF-WEB.

## REFERÊNCIAS

---

ABRAHÃO, S; CONDORI-FERNANDEZ, N; OLSINA, L; PASTOR, O. Defining and Validating Metrics for Navigational Models. In: **Proceedings of the 9th International Symposium on Software Metrics**, pp. 200–210, Australia, September, 2003.

AHMAD, W. F. W; SULAIMAN, S; JOHARI, F. S. Usability Management System (USEMATE): A Web-Based Automated System for Managing Usability Testing Systematically. In: **International Conference on User Science Engineering (i-USER)**, 2010.

AL-WABIL, A; AL-KHALIFA, H. A Framework for Integrating Usability Evaluation Methods: The Mawhiba Web Portal Case Study. In: **International Conference on Current Trends in Information Technology (CTIT)**, 2009.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (ABNT). NBR 9126-1 – Engenharia de Software – Qualidade de Produto – Modelo de Qualidade. Rio de Janeiro: ABNT, 2003.

BANGOR, A; KORTUM, P; MILLER, J. Determining what individual SUS score mean: adding an adjective rating scale. In: **Journal of Usability Studies**, v. 4, issue 3, pp. 114-123, 2009.

\_\_\_\_\_. An empirical evaluation of the System Usability Scale. In: **International Journal of Human-Computer Interaction**, v. 24 issue 6, pp. 574-594, 2008.

BASTIEN, C; SCAPIN, D. Critérios ergonômicos para a avaliação de interfaces homem-computador. 1993. Disponível em: <<http://www.labiutil.inf.ufsc.br/CriteriosErgonomicos/Abertura.html>>. Acesso em: 01 dez. 2013.

BERTINI, E; GABRIELLI, S; KIMANI, S. Appropriating and Assessing Heuristics for Mobile Computing. In: **Proceedings of the Working Conference on Advanced Visual Interfaces**, AVI 06, New York, 2006.

BONIFÁCIO, B; VIANA, D; VIEIRA, S; ARAÚJO, C; CONTE, T. Aplicando técnicas de inspeção de usabilidade para avaliar aplicações móveis. In: **Proceedings of the IX Symposium on Human Factors in Computing Systems**, IHC 10, Porto Alegre, Brasil, 2010.

BROOKE, J. SUS – A quick and dirty usability scale. 1986. Disponível em: <[http://www.usabilitynet.org/tools/r\\_questionnaire.htm](http://www.usabilitynet.org/tools/r_questionnaire.htm)>. Acesso em: 25 ago. 2014.

CHEIKHI, Laila; ABRAN, Alain; SURYN, Witold. Harmonization of usability measurements in ISO 9126 software engineering standards. In: **International Symposium on Industrial Electronics (ISIE)**, IEEE, 2006.

CONTE, T. U. **Técnica de inspeção de usabilidade baseada em perspectivas de projeto Web**. Rio de Janeiro: UFRJ/COPPE. Tese de Doutorado. Universidade Federal do Rio de Janeiro, 2009.

COULOURIS, G; DOLLIMORE, J; KINDBERG, T. **Sistemas Distribuídos: conceitos e projeto**. 4. ed. Porto Alegre: Bookman, 2005.

CYBIS, Walter; BETIOL, Adriana Holtz; FAUST, Richard. **Ergonomia e usabilidade: conhecimentos, métodos e aplicações**. 2. ed. São Paulo: Novatec, 2010.

DE LA VARA, J; WNUK, K; BERNTSSON-SVENSSON, R; SÁNCHEZ, J; REGNELL, B. An Empirical Study on the Importance of Quality Requirements in Industry. In: **23rd International Conference on Software Engineering and Knowledge Engineering (SEKE)**, 2011.

DIAS, Claudia. **Usabilidade na Web: Criando Portais mais Acessíveis**. Rio de Janeiro: AltaBooks, 2003.

DIX, A. Design of User Interfaces for the Web. In: **User Interfaces to Data Intensive Systems (UIDIS 99)**, Edinburgh, September, 1999.

DIX, A; FINLAY, J. E; ABOWD, G. D; BEALE, R. **Human-Computer Interaction**. 3 ed. Prentice Hall: New York, 2004.

FAYAD, M. E; JOHNSON, R. E. **Domain-specific application frameworks: frameworks experience by industry**. New York: J. Wiley, 2000.

FAYAD, M. E; SCHIMIDT, D. C; JOHNSON, R. E. (1999a). **Implementing application frameworks: object-oriented frameworks at work**. New York: J. Wiley, 1999.

\_\_\_\_\_. (1999b). **Building application frameworks: object-oriented foundations of frameworks design**. New York: J. Wiley, 1999.

FERNANDES, P; CONTE, T; BONIFÁCIO, B. WE-QT: A Web Inspection Technique to Support Novice Inspectors. In: **Brazilian Symposium on Software Engineering**, 2012.

FERNANDEZ, A; INSFRAN, E; ABRAHÃO, S. Usability evaluation methods for the web: a systematic mapping study. **Information and Software Technology**. v. 53, pp. 789-817, 2011.

FERRE, Xavier; BEVAN, Nigel. **Usability Planner: A Tool to Support the Process of Selecting Usability Methods**. In: INTERACT 2011, Part IV, LNCS 6949, pp. 652-655. Springer-Verlag, 2011.

FISCHER, Holger. Integrating Usability Engineering in the Software Development Lifecycle Based on International Standards. In: **EICS'12: Proceedings of the 4<sup>th</sup> ACM SIGCHI Symposium on Engineering Interactive Computing Systems**, 2012.

FROEHLICH, G; HOOVER, H. J; LIU, L; SORENSON, P. (1997a). Hooking into Object-Oriented Application Frameworks. In: **Proceedings of the International Conference on Software Engineering**. Boston, 1997.

\_\_\_\_\_. (1997b). Reusing Application Frameworks through Hooks. In: **Communications of the ACM special issue on object-oriented frameworks**, 1997.

GALITZ, W. O. (2003). **The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques**. 2 ed. John Wiley & Sons: New York, 2003.

GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. **Padrões de Projeto: soluções reutilizáveis de software orientado a objetos**. Porto Alegre: Bookman, 2007.

GENA, Cristina; WEIBELZAHN, Stephan. Usability Engineering for the Adaptive Web. In: BRUSILOVSKI, Peter; KOBSA, Alfred; NEJDL, Wolfgang. **The Adaptive Web: Methods and Strategies of Web Personalization**. Springer-Verlag, pp. 720-762, 2007.

GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social**. 6. ed. São Paulo: Atlas, 2010.

GIMENES, I. M. S; HUZITA, E. H. M. **Desenvolvimento Baseado em Componentes**. Rio de Janeiro: Ciência Moderna, 2005.

HART, D; PORTWOOD, D. M. Usability Testing of Web Sites Designed for Communities of Practice: Tests of the IEEE Professional Communication Society (PCS) Web Site Combining Specialized Heuristic Evaluation and Task-Based User Testing. In: **IEEE International Professional Communication Conference (IPCC)**, 2009.

HARTSON, H. R; HIX, D. **Developing User Interfaces: Ensuring Usability Through Product and Process**. New York: John Wiley and Sons, 1993.

HITZ, M; LEITNER, G; MELCHER, R. Usability of Web Applications. In: Kappel, G; Proll, B; Reich, S; Retschitzegger, W. (Eds.). **Web Engineering: The Discipline of systematic Development of Web Applications**, Chapter 11, John Wiley & Sons, 2006.

INSFRAN, E; FERNANDEZ, A. A Systematic Review of Usability Evaluation in Web Development. In: **Proceedings of Second International Workshop on Web Usability and Accessibility (IWWUA 2008)**, v. LNCS 5176 – Advances in Web Information Systems Engineering - WISE 2008 Workshops, pp. 81-91, New Zealand, 2008.

International Standards Organization (ISO). ISO 9241-210. **Human-centred design for interactive systems**. Geneva: International Standards Organization, 2010.

JACOBSON, I. A Resounding Yes to Agile Processes – But Also More. In: **Cutter IT Journal**, v. 15, n. 1, pp. 18-24, 2002.

KAPPEL, G; PROLL, B; REICH, S; RETSCHITZEGGER, W. An Introduction to Web Engineering. In: Kappel, G; Proll, B; Reich, S; Retschitzegger, W. (Eds.). **Web Engineering: The Discipline of Systematic Development of Web Applications**, John Wiley & Sons, 2006.

KONG, X; LIU, L; LOWE, D. Separation of Concerns: a Web Application Architecture Framework. In: **Journal of Digital Information**, v. 6, n. 2, 2006.

MAHRIN, Mohd; STROOPER, Paul; CARRINGTON, David. Selecting Usability Evaluation Methods for Software Process Descriptions. In: **16th Asia-Pacific Software Engineering Conference**, IEEE, 2009.

MATERA, M; RIZZO, F; CARUGHI, G. Web Usability: Principles and Evaluation Methods. In: Mendes, E; Mosley, N. (Eds.). **Web Engineering**, Chapter 5, New York, Spring Verlag, 2006.

MATTSSON, M. **Evolution and Composition of Object-Oriented Frameworks**. PhD Thesis. Department of Software Engineering and Computer Science, University of Karlskrona/Ronneby, 2000.

\_\_\_\_\_. **Object-oriented Frameworks**: A survey of methodological issues. Licentiate Thesis. Department of Computer Science, Lund University, also as Technical Report, LU-CS-TR: 96-167, 1996.

MENDES, E; MOSLEY, N; COUNSELL, S. The Need for Web Engineering: An Introduction. In: Mendes, E; Mosley, N. (Eds.). **Web Engineering**, Chapter 1, New York, Spring Verlag, 2006.

MAYHEW, Deborah J. **The Usability Engineering Lifecycle**. San Diego: Academic Press, 1999.

MIGUEL, Paulo. Estudo de caso na Engenharia de Produção: estruturação e recomendações para sua condução. São Paulo: Produção, 2007. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0103-65132007000100015&lng=pt&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-65132007000100015&lng=pt&nrm=iso)>. Acesso em: 23 abr. 2014.

NASCIMENTO, José Antonio Machado do; AMARAL, Sueli Angélica do. **Avaliação de usabilidade na Internet**. Brasília: Thesaurus, 2010.

NETO, Olibário José Machado. **Usabilidade da interface de dispositivos móveis**: heurísticas e diretrizes para o design. Dissertação de Mestrado. Universidade de São Paulo (ICMC-USP), 2013.

NIELSEN, J; LORANGER, H. **Usabilidade na Web**. 2. ed. Rio de Janeiro: Elsevier, 2007.

NIELSEN, J. Usability 101: Introduction to Usability. (2003). Disponível em: <<http://nngroup.com/articles/usability-101-introduction-to-usability>>. Acesso em: 25 set 2014.

\_\_\_\_\_. (1994a). Guerrilla HCI: using discount usability engineering to penetrate the intimidation barrier (eds). Cost-justifying usability, USA: Academic Press.

\_\_\_\_\_. (1994b). Heuristic evaluation. In: NIELSEN, J; MACK, R (Eds.). **Usability inspection methods, Heuristic Evaluation**. New York: John Wiley & Sons.

\_\_\_\_\_. **Usability Engineering**. Morgan Kaufmann Publishers Inc, 1993.

OFFUTT, J. Quality Attributes of Web Software Application. In: **IEEE Software: Special Issue on Software Engineering of Internet Software**, v. 19, n. 2, pp. 25-32, 2002.

OLSINA, L; COVELLA, G; ROSSI, G. Web Quality. In: MENDES, E., MOSLEY, N. (Eds.). **Web Engineering**, Springer-Verlag, 2006.

PASTOR, O. Fitting the Pieces of the Web Engineering Puzzle. In: **Proceedings of 18<sup>th</sup> Brazilian Symposium on Software Engineering (SBES 2004)**, v. 1, pp 10 - 22, Brasília: 2004.

PINTO, Sérgio Crespo Coelho da Silva. **Composição em WebFrameworks**. Tese de Doutorado. Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), 2000.

PRATES, R; BARBOSA, S. Avaliação de Interfaces de Usuário: Conceitos e Métodos. In: COELHO, J; FABBRI, S (Eds.). **Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação**, capítulo 6, Campinas, 2003.

PREECE J; ROGERS Y; SHARP. **Design de Interação**: além da interação homem-computador. Tradução de Viviane Possamai. Porto Alegre: Bookman, 2005.

PRESSMAN, Roger. S. **Engenharia de Software**. Tradução de Ariovaldo Griesi e Mario Moro Fecchio. 7. ed. Porto Alegre: AMGH, 2011.

PRESSMAN, Roger S; LOWE, David. **Engenharia Web**. Tradução de Daniel Vieira. Rio de Janeiro: LTC, 2009.

QUIS. Questionnaire for User Interaction Satisfaction. About the QUIS, version 7.0. Disponível em: <<http://www.lap.umd.edu/quis/index.html>>. Acesso em: 25 ago. 2014.

ROBERTSON, G; CZERWINSKI, M; BAUDISCH, P; MEYERS, B; ROBBINS, D; SMITH, G; TAN, D. The large-display user experience. In: **IEEE Computer Graphics and Applications**, v. 25, issue 4, 2005.

ROCHA, H. V; BARANAUSKAS, M. C. C. **Design e Avaliação de Interfaces Humano-Computador**. Campinas: NIED/UNICAMP, 2003.

SAMPAIO, R. F; MANCINI, M. C. Estudos de revisão sistemática: um guia para síntese criteriosa da evidência científica. **Revista Brasileira de Fisioterapia**, v. 11, n. 1, pag. 83-89, 2007.

SCHNEIDER, H. N. Interface de Software Educacional: a questão da usabilidade. In: CRUZ, M. H. S. **Pluralidade dos saberes e territórios de pesquisa em educação sob múltiplos olhares dos sujeitos investigadores**. Aracaju: Editora UFS, 2008, p. 199-231.

SEVERINO, Antônio Joaquim. **Metodologia do Trabalho Científico**. 23. ed. São Paulo: Cortez, 2013.

SHARP, H; ROGERS, Y; PREECE, J. **Interaction Design: Beyond Human-Computer Interaction**. 2 ed. Wiley, 2007.

SHNEIDERMAN, B; PLAISANT, C. **Designing the User Interface: Strategies for Effective Human-Computer Interaction**. 5. ed. Addison-Wesley: Boston, 2004.

SILVA, J. L. M. da. **Uma Abordagem para Especificação de Requisitos Dirigida por Modelos Integrada ao Controle de Qualidade de Aplicações Web**. Tese de Doutorado. UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação. Rio de Janeiro: UFRJ/COPPE, 2011.

SKOV, M; SATAGE, J. Supporting problem identification in usability evaluations. In: **Proceedings of the 17th Australia conference on Computer Human Interaction**: Citizens Online: Considerations for Today and the Future, v. 122, pp. 1-9, Austrália, 2005.

SOMMERVILLE, Ian. **Engenharia de software**. Tradução de Selma Shin Shimizu Melnikoff, Reginaldo Arakaki e Edilson de Andrade Barbosa. 8. ed. São Paulo: Pearson, 2007.

SUMI. Software Usability Measurement Inventory. Disponível em: <<http://www.sumi.ucc.ie>>. Acesso em: 25 ago. 2014.

SZYPERSKI, C. **Component Software: Beyond Object-Oriented Programming**. Addison-Wesley, 1997.

TOGNOZZI, Bruce. First Principles. 2001. Disponível em: <<http://www.asktog.com/basics/firstprinciples.htm>>. Acesso em: 21 abr. 2014.

TOMHAVE, B. L. Alphabet Soup: making sense of models, frameworks and methodologies. 2005. Disponível em: <[http://www.secureconsulting.net/Papers/Alphabet\\_Soup.pdf](http://www.secureconsulting.net/Papers/Alphabet_Soup.pdf)>. Acesso em: 05 jan. 2015.

TULLIS, T; STETSON, J. 2004. A Comparison of Questionnaires for Assessing Website Usability. Disponível em: <<http://www.home.comcast.net/~tomutullis/publications/UPA2004TullisStetson.pdf>>. Acesso em: 25 ago. 2014.

USABILITYNET. Requirements meeting. Disponível em: <<http://www.usabilitynet.org/tools/requirements>>. Acesso em: 20 maio 2014.

\_\_\_\_\_. Questionnaire resources. Disponível em: <[http://www.usabilitynet.org/tools/r\\_questionnaire.htm](http://www.usabilitynet.org/tools/r_questionnaire.htm)>. Acesso em 25 ago. 2014.

VAZ, V; CONTE, T; BOTT, A; MENDES, E; TRAVASSOS, G. H. Inspeção de Usabilidade em Organizações de Desenvolvimento de Software: uma Experiência Prática. In: **Simpósio Brasileiro de Qualidade de Software (SBQS)**, Florianópolis-SC, pp. 369-378, 2008.

WAMMI. Web Analysis and Measurement Inventory. What is WAMMI. Disponível em: <<http://www.wammi.com/whatis.html>>. Acesso em: 25 ago. 2014.

W3C. World Wide Web Consortium. About W3C. Disponível em: <<http://www.w3.org/Consortium/>>. Acesso em: 25 de jul. 2014.

ZACHMAN, J. About The Zachman Framework. Disponível em: <<http://www.zachman.com/about-the-zachman-framework>>. Acesso em: 15 jan. 2015.

ZAHLER, T; BARDACH, H. **Usability engineering as a part of the software design process**. Elektrotechnik und Informationstechnik, Springer-Verlag, 2004.

ZHANG, Z; BASILI, V; SHNEIDERMAN, B. Perspective-based Usability Inspection: An Empirical Validation of Efficacy. In: **Empirical Software Engineering**, v. 4, n.1, pp. 43-69, 1999.

## ANEXO A – Questionário SUS (*System Usability Scale*)

---

Questionário SUS (*System Usability Scale*) (BROOKE, 1986).

	Strongly disagree					Strongly agree
1. I think that I would like to use this system frequently.	<input type="checkbox"/>					
	1	2	3	4	5	
2. I found the system unnecessarily complex.	<input type="checkbox"/>					
	1	2	3	4	5	
3. I thought the system was easy to use.	<input type="checkbox"/>					
	1	2	3	4	5	
4. I think that I would need the support of a technical person to be able use this system.	<input type="checkbox"/>					
	1	2	3	4	5	
5. I found the various functions in this system were well integrated.	<input type="checkbox"/>					
	1	2	3	4	5	
6. I thought there was too much inconsistency in this system.	<input type="checkbox"/>					
	1	2	3	4	5	
7. I would imagine that most people would learn to use this system very quickly.	<input type="checkbox"/>					
	1	2	3	4	5	
8. I found the system very cumbersome to use.	<input type="checkbox"/>					
	1	2	3	4	5	
9. I felt very confident using the system.	<input type="checkbox"/>					
	1	2	3	4	5	
10. I needed to learn a lot of things before I could get going with this system.	<input type="checkbox"/>					
	1	2	3	4	5	

## APÊNDICE A – Contexto de Uso e Requisitos de Usabilidade

---

**Fase:** Planejamento

**Atividades:** Análise e Especificação do Contexto de Uso  
Especificação de Requisitos de Usabilidade

**Artefato:** Contexto de Uso e Requisitos de Usabilidade

**Nome da Organização**

**Contexto de Uso e Requisitos de Usabilidade**

**Nome do Projeto**

**Versão do Artefato**

**Local e Data**

### Versões do Artefato

Número da Versão	Comentário	Data

UEF-MEB

# 1. Introdução

## 1.1 Objetivos do Artefato

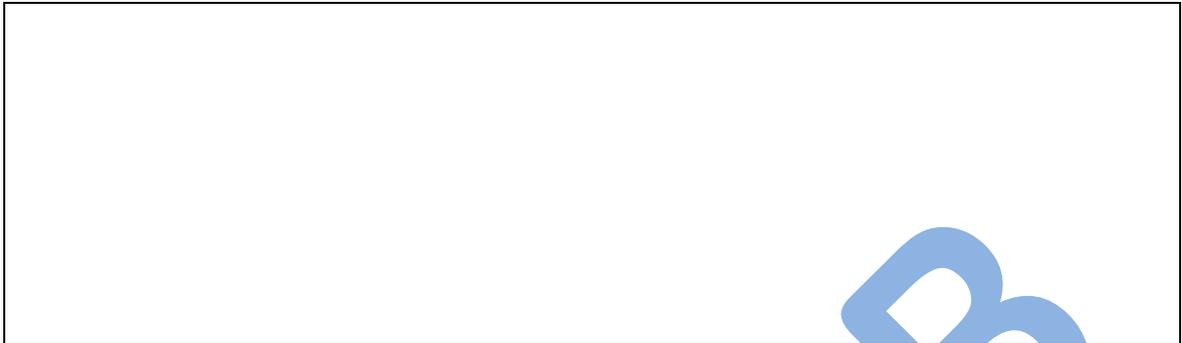
--

## 1.2 Lista de Siglas

Sigla	Definição

## 2. Contexto de Uso

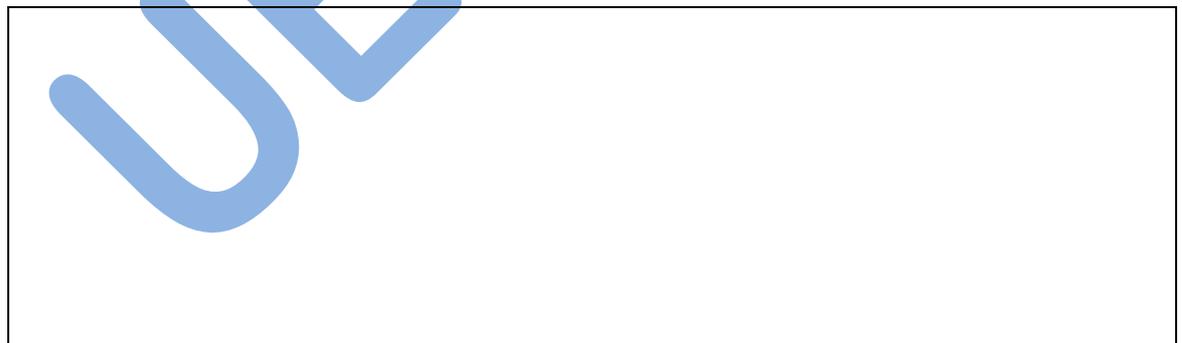
### 2.1 Declaração da Visão<sup>1</sup>



### 2.2 Metodologia<sup>2</sup>



### 2.3 Programação de Atividades<sup>3</sup>



---

<sup>1</sup> Corresponde à descrição dos objetivos da aplicação *Web*.

<sup>2</sup> Diz respeito ao modo e aos recursos utilizados na coleta de informações acerca do Contexto de Uso e dos Requisitos de Usabilidade.

<sup>3</sup> Refere-se ao cronograma de execução de atividades de coleta de informações do Contexto de Uso e dos Requisitos de Usabilidade.

## 2.4 Análise de Usuários<sup>4</sup>

### 2.4.1 Lista de Usuários

Tipo de Usuário	Descrição do Usuário

### 2.4.2 Descrição Detalhada dos Usuários

Tipo de Usuário	Descrição Detalhada do Usuário

<sup>4</sup> Visa obter informações sobre os usuários e demais *stakeholders* da aplicação *Web*. Os tipos de usuários devem ser identificados. Uma descrição detalhada acerca das habilidades, necessidades, expectativas, preferências e responsabilidades dos usuários também deve ser fornecida.

## 2.5 Análise de Tarefas<sup>5</sup>

### 2.5.1 Lista de Tarefas

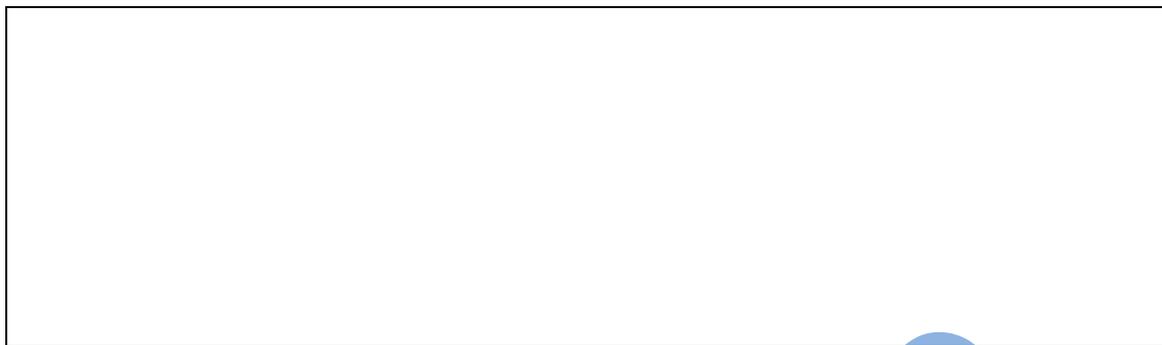
ID	Tarefa

### 2.5.2 Descrição Detalhada das Tarefas

ID	Tipo de Usuário	Descrição Detalhada da Tarefa

<sup>5</sup> Objetiva identificar e caracterizar as tarefas que serão realizadas pelos usuários na aplicação *Web*. Devem ser fornecidas informações detalhadas das tarefas, tais como: objetivos, resultados, estrutura, duração, frequência de realização e dependências. ID corresponde a um identificador utilizado para referenciar cada tarefa a ser realizada na aplicação.

## 2.6 Análise de Ambiente<sup>6</sup>



UEF-WEB

---

<sup>6</sup> Tem o propósito de caracterizar o ambiente de uso onde os usuários ou potenciais usuários desenvolvem suas tarefas com a aplicação *Web*.

### 3. Requisitos de Usabilidade<sup>7</sup>

#### 3.1 Diretrizes Gerais de Usabilidade

--

#### 3.2 Tabela de Especificação de Requisitos de Usabilidade

ID	Atributo de usabilidade	Tipo de usuário	Instrumento de medida	Valor a ser medido	Nível de desempenho almejado

<sup>7</sup> Visa levantar informações sobre diretrizes gerais e requisitos específicos de usabilidade da aplicação *Web*.

- ID: corresponde a um identificador utilizado para referenciar cada requisito de usabilidade.
- Atributo de usabilidade: corresponde a características de usabilidade definidas como parâmetros de qualidade de uso, a serem empregadas como critérios de avaliação de interfaces.
- Tipo de usuário: refere-se aos tipos ou perfis de usuários que vão realizar as tarefas através das interfaces da aplicação *Web*.
- Instrumento de medida: diz respeito ao mecanismo empregado para coletar os valores de um atributo de usabilidade específico. Como exemplos, tem-se: a realização de alguma tarefa, a remoção de algum dado em uma dada tarefa, respostas a perguntas em um questionário, entre outros.
- Valor a ser medido: corresponde ao tipo de dado que será coletado sobre os atributos de usabilidade. Alguns possíveis tipos incluem: desempenho ao realizar uma tarefa, taxa de erros e satisfação subjetiva (grau de satisfação do usuário ao realizar tarefas e utilizar as interfaces da aplicação).
- Nível de desempenho almejado: diz respeito a metas de desempenho entendidas como alvo para os atributos de usabilidade quando da realização de tarefas e interação com as interfaces da aplicação.

## APÊNDICE B – Formulário de *Feedback*

---

**Fase:** Desenvolvimento

**Atividade:** Projeto

**Artefato:** Formulário de *Feedback*<sup>8</sup>

Tipo de usuário	Identificação da interface	Descrição do problema/necessidade/sugestão	Nível de prioridade (baixo, médio ou alto)

---

<sup>8</sup> Tem por objetivo auxiliar a coleta de informações e opiniões dos representantes dos usuários sobre as interfaces que estão sendo projetadas.

- Tipo de usuário: diz respeito aos tipos ou perfis de usuários que realizam as tarefas através das interfaces da aplicação.
- Identificação da interface: corresponde a uma descrição da interface utilizada e tarefa realizada pelo usuário.
- Descrição do problema/necessidade/sugestão: os usuários devem relatar os problemas, necessidades ou sugestões de melhorias para as interfaces analisadas.
- Nível de prioridade: diz respeito ao nível de prioridade atribuído pelo usuário para correção de um problema, atendimento de uma necessidade ou sugestão de melhoria na interface da aplicação.

## APÊNDICE C – Formulário de Inspeção

---

**Fase:** Desenvolvimento

**Atividade:** Implementação

**Artefato:** Formulário de Inspeção<sup>9</sup>

Avaliador	Tarefa	Classificação do problema (Perspectiva.Heurística)	Descrição do problema	Severidade (cosmético, leve, grave ou catastrófico)

---

<sup>9</sup> Visa identificar problemas de usabilidade nas versões de interfaces desenvolvidas.

- Avaliador: integrante da equipe de desenvolvimento que está realizando a inspeção de usabilidade na aplicação.
- Tarefa: descrição da tarefa e da interface da aplicação que está sendo inspecionada.
- Classificação do problema (Perspectiva.Heurística): classificação do problema de usabilidade através da relação Perspectiva.Heurística.
- Descrição do problema: descrição do problema/violação de usabilidade identificado na aplicação.
- Severidade (cosmético, leve, grave ou catastrófico): atribuição do nível de severidade (gravidade) do problema de usabilidade identificado. Problema cosmético (consertar apenas se houver tempo disponível), problema leve (baixa prioridade para consertá-lo), problema grave (alta prioridade para consertá-lo) e problema catastrófico (é imperativo consertá-lo).

## APÊNDICE D – Questionário de Satisfação

Prezados, este questionário tem por objetivo avaliar a usabilidade e a satisfação dos usuários em relação a uma aplicação *Web*. Na Parte I do questionário, são coletadas informações básicas acerca do perfil dos usuários. Na Parte II do questionário, é realizada a avaliação da aplicação *Web* através das respostas fornecidas pelos usuários. Na Parte III do questionário, são apresentadas duas questões abertas para coletar informações dos usuários sobre características positivas e negativas da aplicação *Web*. Para cada questão da Parte I e Parte II do questionário, marque apenas uma resposta. Para as questões da Parte III, dê a sua opinião sobre características positivas e negativas observadas na aplicação *Web*.

Desde já, agradecemos a participação dos usuários e afirmamos que as informações fornecidas a este questionário são completamente confidenciais. Informamos também que os usuários não são obrigados a responder este questionário e podem retirar-se livremente a qualquer momento.

### Parte I – Perfil dos Usuários

<p>1. Sexo</p> <p><input type="radio"/> Masculino</p> <p><input type="radio"/> Feminino</p>	<p>2. Qual é a sua Idade?</p> <p><input type="radio"/> Abaixo de 18</p> <p><input type="radio"/> 18-29</p> <p><input type="radio"/> 30-39</p> <p><input type="radio"/> 40-49</p> <p><input type="radio"/> 50-59</p> <p><input type="radio"/> 60 ou mais</p>	<p>3. Como você avalia o seu nível de habilidade na Internet?</p> <p><input type="radio"/> Baixo</p> <p><input type="radio"/> Médio</p> <p><input type="radio"/> Alto</p> <p><input type="radio"/> Muito alto</p>
---	---	---

### Parte II – Avaliação da Aplicação *Web* pelos Usuários

	Discordo Completamente				Concordo Completamente
1. Eu acho que gostaria de usar esta aplicação <i>Web</i> frequentemente.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
2. Eu achei a aplicação <i>Web</i> desnecessariamente complexa.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
3. Eu achei a aplicação <i>Web</i> fácil de usar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
4. Eu acho que precisaria da ajuda de um técnico para poder usar esta aplicação <i>Web</i> .	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
5. Eu achei que as várias funções nesta aplicação <i>Web</i> estavam bem integradas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
6. Eu acho que houve muita inconsistência na aplicação <i>Web</i> .	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
7. Eu acho que a maioria das pessoas acharia esta aplicação <i>Web</i> fácil de usar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
8. Achei a aplicação <i>Web</i> muito complicada de usar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
9. Eu senti muita segurança usando a aplicação <i>Web</i> .	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
10. Eu precisava aprender muitas coisas antes de poder utilizar esta aplicação <i>Web</i> .	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5

### Parte III – Percepção dos Usuários sobre a Aplicação *Web*

1. Que características da aplicação <i>Web</i> você avalia como positivas e por quê?
2. Que características da aplicação <i>Web</i> precisam ser melhoradas e por quê?