



UNISUL

UNIVERSIDADE DO SUL DE SANTA CATARINA

CLEIDER RAMOS DA ROSA JUNIOR

RODRIGO SOUZA ALMEIDA

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA AVALIAÇÃO
ERGONÔMICA:
A BUSCA POR APLICAÇÕES COM MAIOR USABILIDADE**

Palhoça
2016

**CLEIDER RAMOS DA ROSA JUNIOR
RODRIGO SOUZA ALMEIDA**

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA AVALIAÇÃO
ERGONÔMICA:
A BUSCA POR APLICAÇÕES COM MAIOR USABILIDADE**

Trabalho de Conclusão de Curso submetido a
Universidade do Sul de Santa Catarina como
requisito parcial à obtenção do título de
Bacharel em Ciências da Computação.

Orientadora: Profa. Flavia Lumi Matuzawa M. Sc.

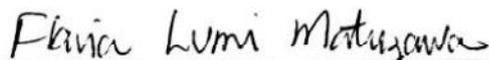
Palhoça
2016

CLEIDER RAMOS DA ROSA JUNIOR
RODRIGO SOUZA ALMEIDA

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA AVALIAÇÃO
ERGONÔMICA:
A BUSCA POR APLICAÇÕES COM MAIOR USABILIDADE**

Trabalho de Conclusão de Curso submetido a
Universidade do Sul de Santa Catarina como
requisito parcial à obtenção do título de
Bacharel em Ciências da Computação.

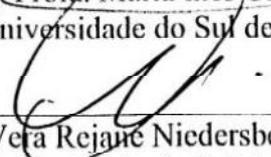
Palhoça, 16 de junho de 2016.



Professora e orientadora Flavia Lumi Matuzawa, Ms.
Universidade do Sul de Santa Catarina



Profa. Maria Inés Castiñeira, Dr.
Universidade do Sul de Santa Catarina


Profra. Vera Rejane Niedersberg Schuhmacher, Dr.
Universidade do Sul de Santa Catarina

Dedicamos este trabalho, a nossos familiares, amigos, colegas e professores, pessoas as quais temos orgulho de incluir como fundamentais para a construção deste momento.

AGRADECIMENTOS

A conciliação da vida acadêmica e profissional exige uma grande dedicação, pois nesta etapa diversas responsabilidades são designadas, e muitas vezes os momentos de lazer com as pessoas a quem estimamos, são deixados de lado. Por este motivo, gostaríamos de gratular primeiramente a família, a qual sempre nos manteve a prumo de nossos objetivos, e em situações difíceis, soube nos fazer renascer.

Apesar das dificuldades, todos os obstáculos aos quais fomos submetidos foram ultrapassados, grande parte desta vitória devemos aos ensinamentos e ao apoio do grupo de docentes, sendo assim não poderíamos deixar de reconhecer os professores e mestres por todo o conhecimento desprendido.

É importante também enaltecer as parcerias estabelecidas ao longo de nossa graduação, em especial a união que resultou na elaboração deste trabalho, ligação que já transcende para amizade.

E por fim agradecemos aos interessados, que dedicaram seu tempo a leitura de nosso trabalho.

“A construção de um sistema com usabilidade depende da análise cuidadosa dos diversos componentes de seu contexto de uso e da participação ativa do usuário nas decisões de projeto da interface[...]” (CYBIS, 2010).

RESUMO

A cada dia que passa, uma infinidade de sistemas, são idealizados e produzidos, com a intenção, de atender os mais variados objetivos e ramificações de mercado. A evolução tecnológica, é responsável por uma grande parcela desta eclosão, na disponibilidade de sistemas. Porém, este aumento de oferta, não trouxe apenas o atributo quantidade ao mercado de softwares, mas também configurou uma tendência de qualidade aos produtos oferecidos. Envolvido neste contexto, o presente trabalho dedica-se a projeção, de uma solução protótipo, de baixo custo, que possibilite disseminar os princípios ergonômicos. O sistema desenvolvido busca fornecer, um ambiente favorável e disponibilizar um conjunto de recursos, para avaliações de usabilidade. Os usuários cadastrados no sistema, poderão também, cadastrar recomendações personalizadas, com a finalidade de analisar a interface dos mais diversos produtos, nas mais distintas fases de desenvolvimento. Para idealização deste software, os autores tiveram que se aprofundar nas concepções de interação homem-computador, ergonomia e usabilidade, bem como analisar alguns sistemas CAUSE existentes. Após a fase de definição de funcionamento do sistema, os conceitos estudados através da pesquisa foram colocados em prática, por meio de ferramentas como Enterprise Architect, Bootstrap, JAVA e PostgreSQL.

Palavras-chave: interface homem-computador. Inspeção por lista de verificação. Princípios ergonômicos, Recomendações ergonômicas. Usabilidade

ABSTRACT

Everyday, a multitude of systems are conceived and produced with the intention to meet the various objectives and market ramifications. Technological development, is responsible for a large portion of this outbreak in the availability of systems. However, this increase in supply, not only brought the quantity attribute to the software market, but also set a trend, on the quality of offered products. Involved in this context, the present work is dedicated to the projection of a low cost prototype solution, that enables disseminate ergonomic principles. This tool will provide a favorable environment and a set of resources for usability evaluations, and users registered in the system will be able to save their own custom recommendations, in order to analyse the interface of various products, in distinct stages of development. For this software idealization, the authors had to learn about conceptions of human-computer interaction, ergonomics and usability, as well as analyse some existing CAUSE systems. After the system operation definition phase, the concepts studied through the research were put into practice, with the help of tools like Enterprise Architect, Bootstrap, PostgreSQL and Java.

Keywords: human-computer interface, checklist inspection, ergonomic principles, ergonomic recommendations, usability.

SUMÁRIO

1	INTRODUÇÃO	15
1.1	PROBLEMÁTICA	16
1.2	OBJETIVOS	16
1.2.1	Objetivo Geral	16
1.2.2	Objetivos Específicos	16
1.3	JUSTIFICATIVA	17
1.4	ESTRUTURA DA MONOGRAFIA	17
2	REVISÃO BIBLIOGRÁFICA	19
2.1	INTERAÇÃO HOMEM COMPUTADOR	19
2.1.1	Interação	20
2.1.1.1	Perspectiva de Sistema	21
2.1.1.2	Perspectiva de Parceiro de Discurso	22
2.1.1.3	Perspectiva de Ferramenta	23
2.1.1.4	Perspectiva de Mídia	23
2.1.2	Interface	24
2.1.3	Affordance	25
2.2	USABILIDADE E ERGONOMIA	26
2.2.1	Princípios Ergonômicos	27
2.2.1.1	Heurísticas de Usabilidade	27
2.2.1.2	Regras de Ouro	28
2.2.1.3	Princípios de Dialogo	28
2.2.1.4	Critérios Ergonômicos	29
2.3	TÉCNICAS DE AVALIAÇÃO	33
2.3.1	Avaliação Analítica	33
2.3.2	Avaliação Heurística	34
2.3.3	Inspeção por Lista de Verificação (<i>Checklist</i>)	36
2.3.4	Percurso Cognitivo	38
2.3.5	Inspeções Preventivas de Erros	39
2.4	FERRAMENTAS CAUSE	40
2.4.1	Ergolist	41
2.4.2	Glist	42
2.4.3	QUIS	42

2.4.4 TOWABE	42
2.5 CONCLUSÃO DA ETAPA DE FUDAMENTAÇÃO TEÓRICA	43
3 METODOLOGIA	44
3.1 CARACTERIZAÇÃO DO TIPO DE PESQUISA	44
3.2 ETAPAS METODOLÓGICAS	44
3.3 PROPOSTA DE SOLUÇÃO	45
3.4 DELIMITAÇÕES	46
3.5 RECURSOS NECESSÁRIOS	46
3.6 CONCLUSÃO DA ETAPA METODOLÓGICA	46
4 MODELAGEM	47
4.1 UML.....	47
4.1.1 Atores.....	47
4.1.2 Casos de Uso	48
4.1.3 Diagrama de Classes	49
4.1.4 Requisitos Funcionais	50
4.1.5 Requisitos Não Funcionais	51
4.1.6 Regras de Negócio	51
4.1.7 Diagrama de Banco de Dados	52
4.1.8 Diagrama de Atividades – Gerência do Sistema	52
4.1.9 Diagrama de Atividades – Cadastro Recomendações e Perguntas.....	53
4.1.10 Diagrama de Atividades – Execução do Checklist	54
4.1.11 Diagrama de Robustez – Usuário Comum.....	56
4.1.12 Diagrama de Atividades – Desenvolvedor	56
4.1.13 Diagrama de Atividades – Administrador	57
4.2 CONCLUSÃO DA ETAPA DE MODELAGEM	57
5 DESENVOLVIMENTO	58
5.1 TECNOLOGIAS E FERRAMENTAS UTILIZADAS	58
5.1.1 Enterprise Architect.....	58
5.1.2 PostgreSQL	59
5.1.3 Bootstrap	60
5.1.4 Linguagem Java.....	60
5.2 PROCESSO DE DESENVOLVIMENTO	61
5.2.1 Problemas Identificados Durante a Etapa de Desenvolvimento	61
5.2.2 Perspectivas Inicial Sobre o Sistema	61

5.2.3	Teste do Sistema	62
5.2.3.1	Validação com Usuário Experiente	62
5.2.3.2	Validação com Usuários Comuns	62
5.3	APRESENTAÇÃO DO SISTEMA	63
5.3.1	Principais Interfaces e Funcionalidades.....	63
5.3.1.1	Interfaces do Sistema.....	63
5.4	CONCLUSÃO DA ETAPA DE DESENVOLVIMENTO.....	70
6	CONCLUSÃO.....	72
	REFERÊNCIAS	74
	APÊNDICE A – CRONOGRAMA DE ATIVIDADES DO TCC.....	76

LISTA DE ILUSTRAÇÕES

Figura 1 – Esquema do Sistema Homem-Computador	19
Figura 2 – Perspectivas de Interação Humano Computador	21
Figura 3 – Exemplificando a Visão de Sistema: Terminal de Comando	22
Figura 4 – Exemplificando o Sistema Parceiro de Discurso: Tradutor Automático	22
Figura 5 – Exemplificando a Perspectiva de Ferramenta: Aplicações de Escritório.....	23
Figura 6 – Exemplificando o Ponto de Vista de Mídia: Redes Sociais.....	24
Figura 7 – Atores do Sistema	47
Figura 8 – Modelo de Casos de Uso.....	48
Figura 9 – Diagrama de Classes	50
Figura 10 – Requisitos Funcionais	51
Figura 11 – Requisitos Não Funcionais.....	51
Figura 12 – Regras de Negócio	52
Figura 13 – Diagrama de Banco de Dados	52
Figura 14 – Diagrama de atividades - Gerencia do Sistema	53
Figura 15 – Diagrama de atividades - Cadastro de Recomendações.....	54
Figura 16 – Diagrama de atividades - Execução do Checklist.....	55
Figura 17 – Diagrama de Robustez – Usuário Comum.....	56
Figura 18 – Diagrama de Robustez – Desenvolvedor	56
Figura 19 – Diagrama de Robustez – Administrador	57
Figura 20 – Interface - Inicial.....	63
Figura 21 – Interface - Recomendações Padrões.....	64
Figura 22 – Interface - Responder Checklist Padrão.....	64
Figura 23 – Interface - Relatório das Recomendações Ergonômicas	65
Figura 24 – Interface - <i>Login</i> do sistema.....	65
Figura 25 – Interface - Menu Cadastro Recomendações Ergonômicas e Perguntas	66
Figura 26 – Interface - Cadastro de Recomendações Ergonômicas	66
Figura 27 – Interface - Cadastro de perguntas das Recomendações Ergonômicas	66
Figura 28 – Interface - Menu Consulta e Exclusão Recomendações e Perguntas.....	67
Figura 29 – Interface - Consulta e exclusão de Recomendações	67
Figura 30 – Interface - Consulta e exclusão de Recomendações	67
Figura 31 – Interface - Ferramenta de Avaliação	68
Figura 32 – Interface - Recomendações Ergonômicas cadastradas manualmente	68

Figura 33 – Interface - Cadastro de Perfil	69
Figura 34 – Interface - Cadastro de Administrador	69
Figura 35 – Interface - Cadastro de Desenvolvedor	69
Figura 36 – Interface - Consulta de Perfil	70
Figura 37 – Interface - Consulta de Administrador	70
Figura 38 – Interface - Consulta de Desenvolvedor	70

LISTA DE QUADROS

Quadro 1 – Cronograma de atividades TCC 1	76
Quadro 2 – Cronograma de atividades TCC 2	76
Quadro 3 – Cronograma TCC	77

LISTA DE SIGLAS

BPMN – *Business Process Model and Notation*
CASE – *Computer-Aided Software Engineering*
CAUSE – *Computer-Aided Usability Engineering*
CSS – *Cascading Style Sheets*
CVS – *Concurrent Version System*
HTML – *HyperText Markup Language*
IHC – **Interação Homem-Computador**
ISO – *International Organization for Standardization*
JSF – *JavaServer Faces*
LABIUTIL – **Laboratório de Utilizabilidade da Informática**
MVC – *Model-View-Controller*
QUIS – *Questionnaire for User Interaction Satisfaction*
RTF – *Rich Text Format*
SGBD – **Sistema Gerenciador de Bancos de Dados**
SOAML – *Service Oriented Architecture Modeling Language*
SysML – *Systems Modeling Language*
TOWABE – *Tool for Web Application Usability Evaluation*
UC – *Use Case*
UFRGS – **Universidade Federal do Rio Grande do Sul**
UFSC – **Universidade Federal de Santa Catarina**
UNISUL – **Universidade do Sul de Santa Catarina**
UML – *Unified Modeling Language*
VDT – *Visual Display Terminals*
XML – *eXtensible Markup Language*

1 INTRODUÇÃO

Com a revolução dos computadores e a subsequente diminuição do preço aplicado sobre os dispositivos tecnológicos, o produto sistema passou a ser acessível a uma parcela significativa da população. Consecutivamente, com o aumento de usuários, o mercado de softwares foi inundado com uma infinidade de programas, destinados a realizar as mais variadas atividades. Em um ambiente de vasta opção de sistemas, as empresas desenvolvedoras começaram a buscar diferenciais, para se tornarem competitivas, e passaram a considerar a interface o principal componente de seu projeto. (NIELSEN, 1994).

Investir em ergonomia e usabilidade, durante o processo de fabricação de conteúdo computacional, consiste em considerar os diversos tipos de usuário e suas respectivas limitações, buscando sempre que possível, confeccionar ambientes amigáveis, de fácil manipulação e interpretação. (CYBIS, 2010)

Apesar da maior parte destas concepções já terem sido elaborados a bastante tempo, seus fundamentos permanecem em evidencia, e se consolidam dia a dia. Atualmente as empresas que produzem softwares sem se preocupar com o usuário estão fadadas a perder espaço.

Contudo, alguns princípios ergonômicos sugerem testes, nas mais diversas etapas de desenvolvimento, fato que pode aumentar consideravelmente o tempo para a conclusão de um projeto, porem assegura uma maior eficácia e aceitabilidade aos produtos construídos com estas metodologias.

A utilização da técnica de inspeção por lista de verificação, possibilita o emprego dos critérios de usabilidade, para avaliação de interfaces. Ao aplicar este método, o produto em fase de desenvolvimento, pode ser analisado por pessoas comuns, que através de um questionário, formulado com base nos conceitos ergonômicos, relatam suas experiências como usuários, em relação a interface apreciada. Este mecanismo possibilita encontrar problemas, que dependem de ajuste, simplificando o processo de criação e diminuindo o custo de implementação.

Através desta perspectiva, o presente trabalho propõe apresentar uma síntese sobre os conhecimentos ergonômicos associados a Interação Homem Computador (IHC), bem como desenvolver uma ferramenta protótipo, que terá a finalidade de auxiliar avaliações de usabilidade em interfaces.

1.1 PROBLEMÁTICA

O problema que motivou esta pesquisa está diretamente relacionado as consequências de não se utilizar os conceitos de usabilidade no processo de desenvolvimento de software. Este mal andamento, comumente adotado, afeta todos os envolvidos. Para os usuários, este tipo de interface, de trabalhosa compreensão, dificulta ou até mesmo impede o uso do sistema, causando frustrações e perda de autoestima. Já para quem compra, uma ideia não projetada de forma concisa, o prejuízo está vinculado a questões financeiras, visto que, sem uma interação adequada, os erros e perdas se multiplicam a cada tarefa. Por fim, temos o responsável pela criação deste software, que também é prejudicado, por não ter seguido as diretrizes de usabilidade, e acaba envolvido a uma constante perda de credibilidade e mercado (CYBIS, 2010).

A partir desta abordagem inicial, surge a questão que fundamenta este trabalho. Como atender as mais diversas necessidades, otimizar o processo de desenvolvimento e garantir uma interface simples, intuitiva e de fácil utilização?

1.2 OBJETIVOS

Os objetivos deste trabalho serão especificados a seguir.

1.2.1 Objetivo Geral

Identificar os conceitos de ergonomia desejáveis no processo de interação homem computador, e aplicar estes princípios no processo de desenvolvimento de uma ferramenta protótipo, que irá auxiliar na tarefa de avaliação de usabilidade.

1.2.2 Objetivos Específicos

Os objetivos específicos deste trabalho compreendem:

1. Disseminar os conceitos de interação, ergonomia e usabilidade.
2. Identificar as técnicas de avaliação existentes e desenvolver um protótipo, para a realização de testes de usabilidade.
3. Promover a utilização dos princípios de qualidade para a construção de interfaces amigáveis.

4. Aplicar em um projeto prático as teorias aprendidas em diversas disciplinas do curso de Ciências da Computação.

1.3 JUSTIFICATIVA

Este trabalho foi impulsionado pelo desejo dos autores de propiciar aos interessados um ambiente favorável para a realização de avaliações de usabilidade. Apresenta conceitos de interação, ergonomia e usabilidade, concepções que fundamentam o desenvolvimento da aplicação proposta, bem como servem de base para a resolução de outros problemas na área de IHC.

A ferramenta idealizada, utiliza-se de técnicas de inspeção por lista de verificação, para avaliar sistemas e o correto emprego dos critérios de usabilidade, em seu desenvolvimento. Esta técnica foi adotada por, comprovadamente, proporcionar um rápido resultado e possuir baixo custo de manutenção. A utilização desta prática visa atender as necessidades intrínsecas ao processo de desenvolvimento de software, garantindo interfaces amigáveis e sistemas de bom desempenho.

Apesar de já existirem ferramentas com estas características, a ideia se mantém justificável, pela dificuldade de acesso a estes mecanismos, seja por serem pagos, ou por questões de indisponibilidade.

1.4 ESTRUTURA DA MONOGRAFIA

Este trabalho de conclusão de curso, atende as especificações da instituição, e está dividido nas seguintes partes predefinidas:

Capítulo 1 – Introdução: Etapa do trabalho no qual o assunto é contextualizado, sendo exposto, aos leitores, os problemas, objetivos e motivação para realização da proposta.

Capítulo 2 – Revisão Bibliográfica: Fase do projeto na qual as fundamentações teóricas, utilizadas para o desenvolvimento da atividade, são apresentadas. Neste capítulo, conceitos de Interação, Usabilidade e Ergonomia são evidenciados.

Capítulo 3 – Metodologia: Parcela do trabalho direcionada ao detalhamento das metodologias utilizadas para construção da pesquisa.

Capítulo 4 – Modelagem: Momento destinado a apresentação dos modelos e diagramas, que representam o funcionamento do sistema.

Capítulo 5 – Desenvolvimento: Parte da monografia destinada a documentação da solução proposta. Neste ato, o passo a passo para atingir o objetivo do trabalho é trazido à tona, informações como modelagem, prototipação e construção do código fonte, são profundamente detalhados.

Capítulo 6 – Conclusão: Capítulo destinado as considerações finais do trabalho. Neste ponto são mencionadas informações sobre as expectativas do projeto e os resultados alcançados ao longo de seu desenvolvimento.

2 REVISÃO BIBLIOGRÁFICA

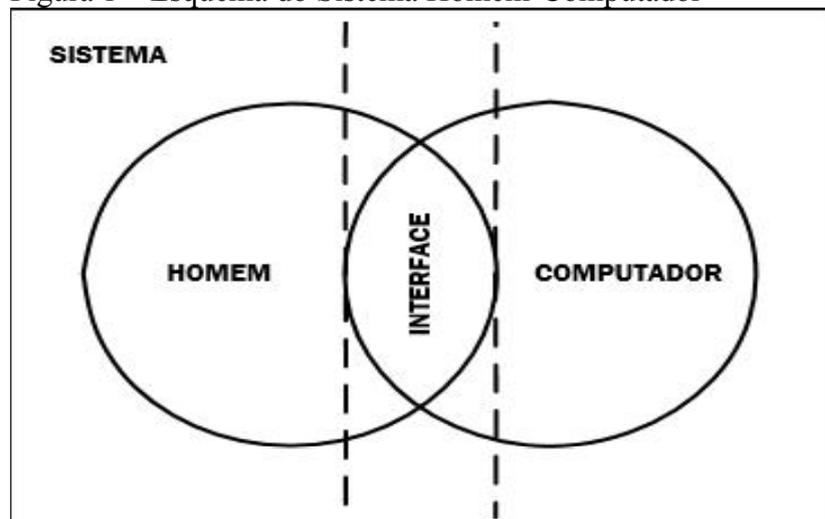
Este capítulo é destinado a integração das concepções, que fundamentam o trabalho, com o objetivo proposto. Para atender este propósito, esta etapa irá propiciar aos leitores a capacidade de reconhecimento de um sistema de qualidade, partindo das definições, de renomados estudiosos da área de IHC e Ergonomia.

Nesta literatura utiliza-se uma ordem sequencial e complementar para apresentação das referências documentais. Inicialmente são expostas as definições de Interação Homem-Computador, conseguindo evidenciar-se os conceitos relativos a Ergonomia e Usabilidade. Após esta iniciação teórica, são expostas algumas Técnicas de Avaliação e soluções CAUSE existentes.

2.1 INTERAÇÃO HOMEM COMPUTADOR

Para Másculo e Vidal (2011), entende-se como um sistema Homem-Computador a relação operativa entre pessoas e computadores, que se comunicam a fim de executar uma determinada tarefa. A resultante dessa interação é conhecida como a interface do sistema Homem-Computador (Figura 1).

Figura 1 – Esquema do Sistema Homem-Computador



Fonte: Adaptado de Másculo e Vidal (2011, p. 460).

Após análise do esquema Homem-Computador proposto por Másculo e Vidal, uma construção computacional de qualidade pode parecer simples, porém este processo depende de fatores implícitos, que elevam a complexidade de criação, conforme especificado no texto a seguir:

A dificuldade no desenvolvimento de interfaces ergonômicas se deve ao fato de elas constituírem, fundamentalmente, sistemas abertos dos quais os usuários são agentes ativos, atores de comportamento não determinístico, cujas mudanças na maneira de pensar e de se comportar são tanto consequência como causa de um ambiente tecnológico sempre em evolução. As mesmas entradas e saídas podem significar coisas diferentes para diferentes pessoas, em função do momento e dos contextos em que elas se encontram. Assim, pode-se afirmar que a experiência do usuário é individual e única, na medida que cada pessoa é única em sua bagagem de conhecimento e expectativas. Dificilmente uma mesma interface significará exatamente a mesma coisa para dois usuários distintos. Menor ainda é a chance de ela ter um significado integralmente compartilhado entre usuários e projetistas (CYBIS, 2010, p.16).

Segundo Barbosa e Silva (2010), uma forma de aumentarmos a qualidade no desenvolvimento de sistemas informatizados, pode ser obtida através da compreensão dos elementos envolvidos no processo de Interação Homem-Computador. Desta forma, as seções a seguir apresentam os conceitos de interação, interface e *affordance*.

2.1.1 Interação

Quando o conceito de interação surgiu, estava geralmente associado a um processo de operação de máquinas. Porém com o avanço das pesquisas na área de IHC, os aspectos cognitivos e perceptivos dos usuários, passaram a ser considerados, evidenciando-se uma concepção de comunicação, para este tipo de contato, homem-computador. (ROCHA e BARANAUSKAS, 2003).

A partir do ponto de vista de Norman (1986 apud BARBOSA e SILVA, 2010), o conceito de interação se aproxima de uma definição mais harmoniosa com a realidade, quando o percebemos como um processo através do qual um usuário projeta uma intenção, arquiteta suas ações, atua sobre a interface, percebe a resposta do sistema, interpreta os resultados e julga se os objetivos foram atingidos.

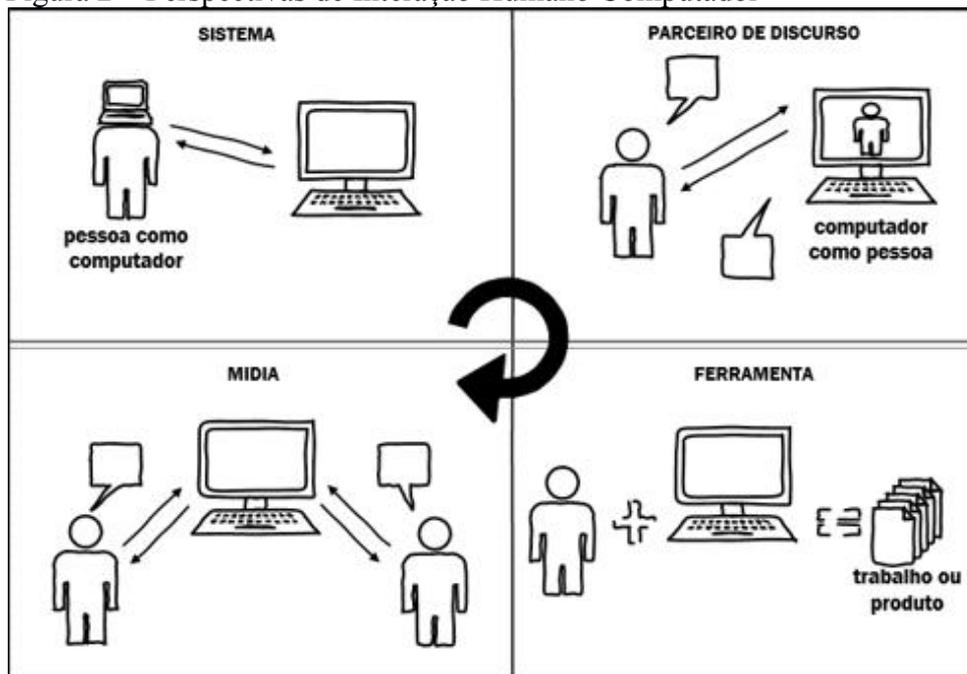
“Em geral, a interação usuário-sistema pode ser considerada como tudo que acontece quando uma pessoa e um sistema computacional se unem para realizar tarefas, visando a um objetivo” (HIX e HARTSON, 1993 apud BARBOSA e SILVA, 2010, p. 20).

Atualmente, evidencia-se a interação como processo de comunicação entre pessoas, intercedida por sistemas computacionais (SOUZA, 2005a apud BARBOSA e SILVA, 2010).

Ao combinar e interpretar estas concepções, Barbosa e Silva (2010) propuseram de forma generalista, que o sistema de interação usuário-computador, poderia ser entendido como um processo de manipulação, comunicação, conversa, troca e influencia.

Kammersgaard (1985) apresentou quatro diferentes definições comumente adotadas por investigadores e designers, para identificação de sistemas com qualidade. Em seu artigo foram expostos os seguintes prismas, referentes ao processo de interação humano computador: visão de sistema, de parceiro de discurso, de ferramenta e de mídia. Cada uma dessas perspectivas atribui ao usuário e ao sistema determinado papel, e caracteriza a interação sob diferente ponto de vista (Figura 2).

Figura 2 – Perspectivas de Interação Humano Computador

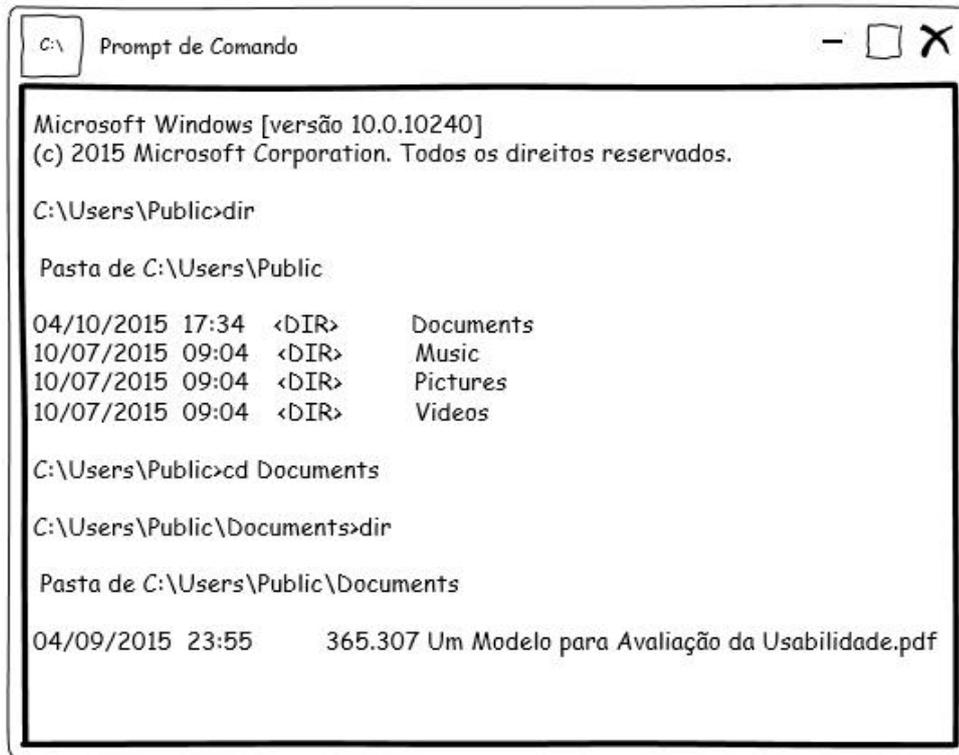


Fonte: Adaptado de Barbosa e Silva (2010, p 21).

2.1.1.1 Perspectiva de Sistema

Na visão do sistema, o usuário é considerado um mecanismo computacional, e nesta perspectiva a interação humano-computador assemelha-se a interação entre sistemas computacionais, ou seja, é vista como uma mera troca de dados entre pessoas e sistemas computacionais, análogo a uma transmissão de dados entre sistemas. Dessa maneira, o usuário precisa se portar como um computador, aprendendo a comunicar-se de forma disciplinada e contida, sendo limitado por padrões de entrada rígidos. Sistemas que operam sobre este ponto de vista, tem como propósito central a preocupação com o aumento de eficiência, o qual é obtido através da garantia de troca de dados com a mínima possibilidade de erros (BARBOSA e SILVA, 2010) (Figura 3).

Figura 3 – Exemplificando a Visão de Sistema: Terminal de Comando

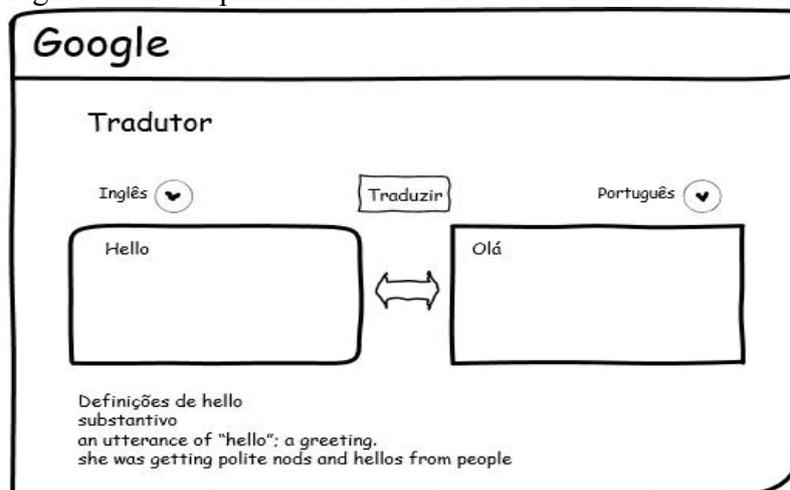


Fonte: Ilustrado pelos próprios autores (2015).

2.1.1.2 Perspectiva de Parceiro de Discurso

Contrapondo à perspectiva de sistema, surgiu no campo de Inteligência Artificial a ideia de transformar um sistema interativo em um parceiro de discurso. Nessa concepção, o sistema computacional assume a função de um ser humano, exercendo um comportamento muito próximo a de seus usuários, chegando a proporcionar uma consciência de interação entre pessoas (BARBOSA e SILVA, 2010) (Figura 4).

Figura 4 – Exemplificando o Sistema Parceiro de Discurso: Tradutor Automático

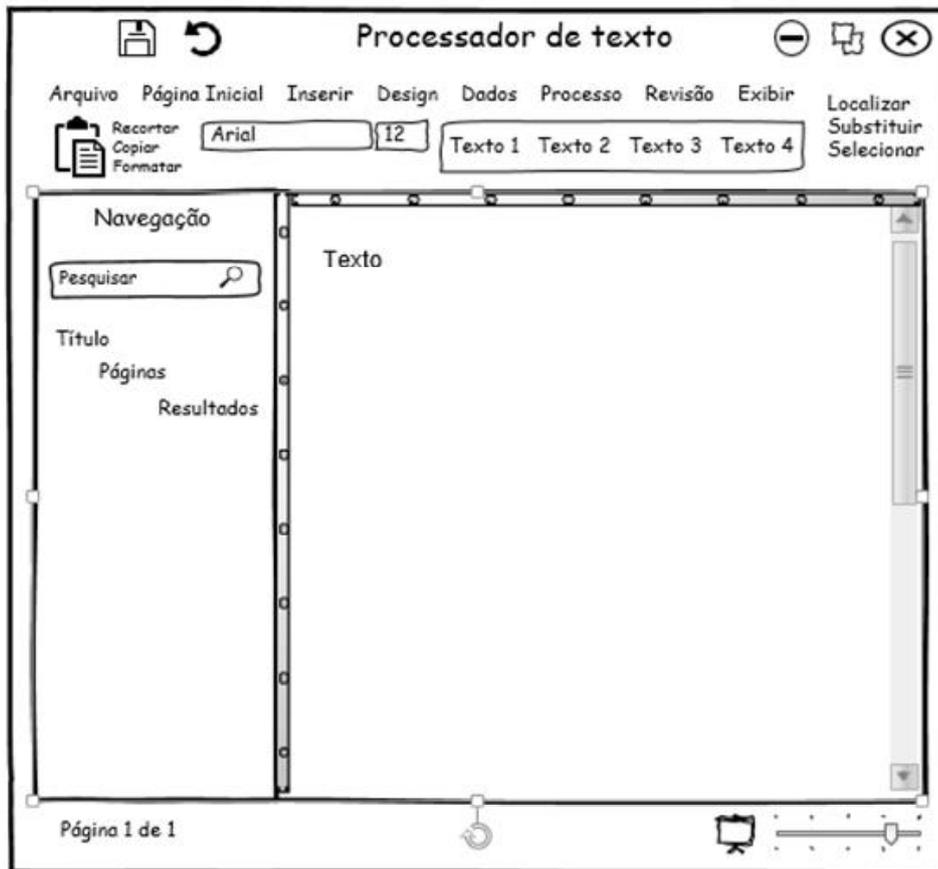


Fonte: Ilustrado pelos próprios autores (2015).

2.1.1.3 Perspectiva de Ferramenta

Sob o prisma de ferramenta, o sistema interativo é visto com um instrumento que ajuda o usuário a executar suas tarefas. Dessa maneira, a interação pode ser definida como a aplicação de uma ferramenta à um objeto, com a finalidade de executar um determinado trabalho. Seu sucesso depende do entendimento do usuário a respeito da ferramenta, bem como sua capacidade de operá-la com facilidade. Esse processo de interação, é caracterizado pela execução da atividade, de forma automática, sem a necessidade de raciocínio sobre o sistema (BARBOSA e SILVA, 2010) (Figura 5).

Figura 5 – Exemplificando a Perspectiva de Ferramenta: Aplicações de Escritório

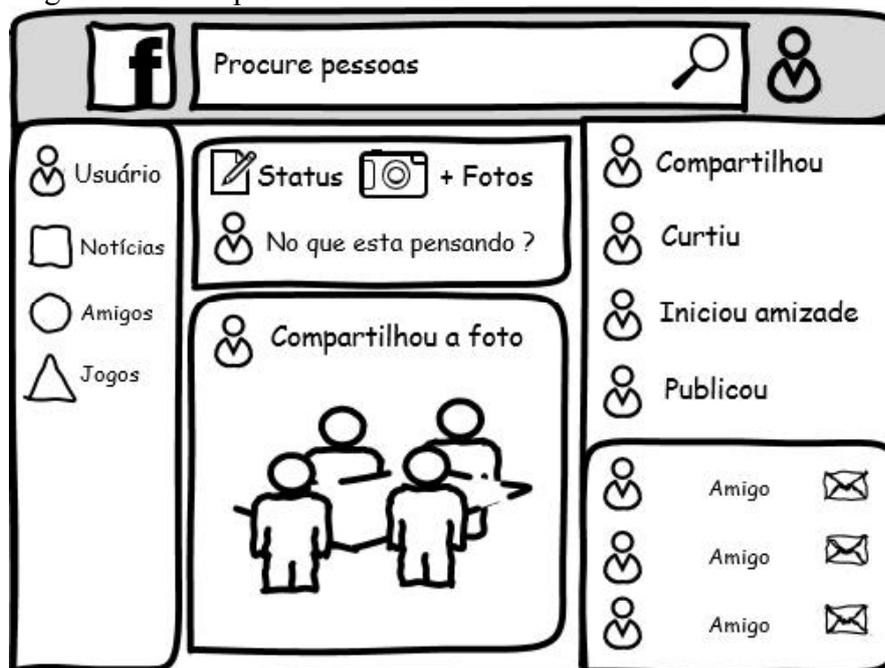


Fonte: Ilustrado pelos próprios autores (2015).

2.1.1.4 Perspectiva de Mídia

O sentido de mídia vem ganhando espaço nos sistemas interativos da atualidade, em particular aplicações computacionais que conectam pessoas através da internet. Nestes sistemas o processo interativo é visto como uma mídia pela qual as pessoas se comunicam umas com as outras (BARBOSA e SILVA, 2010) (Figura 6).

Figura 6 – Exemplificando o Ponto de Vista de Mídia: Redes Sociais



Fonte: Ilustrado pelos próprios autores (2015).

2.1.2 Interface

A noção de interface está diretamente relacionada a toda a porção do sistema com a qual o usuário mantém algum tipo de contato, durante a interação, seja ele físico, motor, perceptivo ou conceitual (MORAN, 1981 apud BARBOSA e SILVA, 2010).

Por ser considerada a única forma de contato entre o usuário e o sistema, grande parte dos usuários acreditam que o sistema é a interface com a qual interagem (Hix e Hartson, 1993 apud BARBOSA e SILVA, 2010).

Na interface, o processo de interação física, obrigatoriamente é mediado pelo hardware e software. Mecanismos de entrada, como teclado e mouse, possibilitam aos usuários operar sobre uma determinada interface, fato que os torna participantes ativos do sistema interativo. Em contrapartida, os dispositivos de saída, como monitor e alto-falantes, permitem aos usuários identificar as reações do sistema e participar passivamente da interação. Já o contato conceitual depende de uma interpretação do usuário em relação ao que foi percebido através do contato físico. Ao compreender as respostas do sistema, o usuário dá seguimento a operação, arquitetando e definindo os próximos passos a serem seguidos no processo de interação (BARBOSA e SILVA, 2010).

A camada de interface é responsável pela definição dos processos de interação existentes no sistema, é ela que indica ao usuário de que forma ele pode participar, como deve

agir e em que ordem suas ações devem ser executadas. Desta maneira, quando o modo de interação é definido, algumas características da interface podem ser restringidas ou limitadas.

Para Barbosa e Silva (2010), todos os elementos envolvidos no processo de interação estão fortemente ligados, e por este motivo, a formação, o conhecimento e as experiências do usuário não podem ser ignorados na definição da interface. Para exemplificar este pensamento, consideremos primeiramente a circunstância de uso, realidade que influencia a forma como os usuários compreendem e interpretam a interface, respondendo por diferentes formas de percepção. Outro exemplo importante, está relacionado a condição física e cognitiva dos usuários, fato que geralmente estabelece uma necessidade específica para um determinado sistema.

2.1.3 *Affordance*

As características físicas de um dispositivo, tendem a revelar sua aplicabilidade e forma de uso. Esse entendimento também é extensivo ao processo de interação usuário-sistema, onde um conjunto de propriedades de hardware e software são percebidas pelo usuário, interpretadas e relacionadas a uma possível operação e forma de manuseio. *Affordance* é o termo técnico que define esse conjunto de características (BARBOSA e SILVA, 2010).

“Em IHC, a *affordance* de um objeto corresponde ao conjunto das características de um objeto capaz de revelar aos seus usuários as operações e manipulações que eles podem fazer com ele”. (NORMAN, 1988 apud BARBOSA e SILVA 2010).

“Em uma interface gráfica, por exemplo, a *affordance* de um botão de comando diz respeito a possibilidade de pressioná-lo usando o mouse ou o teclado e, assim, acionar uma operação do sistema”. (BARBOSA e SILVA 2010).

A utilização de *affordances* coerentes, em sistemas interativos, é imprescindível, visto que através deste conjunto de informações o usuário pode operar o sistema de forma fácil e intuitiva, atingindo os objetivos propostos. Os designers devem ter cautela nesta etapa de desenvolvimento, evitando erros, pois má elaboração de uma *affordance* pode dar a impressão de que a interface funciona de determinada maneira, quando na verdade funcionam de outra forma (BARBOSA e SILVA, 2010).

2.2 USABILIDADE E ERGONOMIA

A usabilidade é uma qualidade desejável nos programas e aplicações, porém por não se tratar de uma característica natural do sistema, depende da combinação entre os atributos da interface e as particularidades de seus usuários, sempre considerando os objetivos e situações de uso. Concomitantemente uma determinada interface pode conceder interações adequadas a usuários experientes e desfavorecer utilizadores novatos. Da mesma maneira, o emprego de diferentes rotinas de tarefa e capacidades computacionais, podem construir variadas percepções de uso do sistema. A existência de aplicações que funcionam constantemente e softwares que trabalham de forma eventual, bem como sistemas operados através dispositivos rápidos e programas manipulados por mecanismos lentos, são características inerentes ao processo de interação e podem influenciar os usuários quanto a consciência de qualidade do sistema (CYBIS, 2010).

Uma das principais preocupações da área de ergonomia está relacionada a necessidade de aperfeiçoamento nos processos interativos. De maneira geral, para se atingir este propósito, o projeto necessita de uma combinação de design de produto e software, que inclua monitores, dispositivos de entrada, interface, local de trabalho, capacidades e limitações dos utilizadores. Comprovadamente ao se utilizar as propriedades ergonômicas para aprimoramento dos sistemas, o produto passa a apresentar uma redução de erros e desconforto, fato que minimiza os riscos de saúde e segurança para os usuários, conseqüentemente um maior desempenho é obtido (ISO 9241-1, 1997).

A norma ISO 9241 (1997) define a usabilidade como “a capacidade que um sistema interativo oferece a seu usuário, em um determinado contexto de operação, para a realização de tarefas, de maneira eficaz, eficiente e agradável”.

O uso de conceitos ergonômicos na elaboração de processos interativos dá origem a usabilidade (CYBIS, 2010).

Para Bank (2015), a preocupação com usabilidade é imprescindível para quem desenvolve sistemas com algum tipo de interatividade. Pois com um pouco mais de tempo na fase de concepção, pode-se evitar grandes perdas ao final do projeto.

“Os programas de software e suas interfaces com o usuário constituem ferramentas cognitivas, capazes de modelar representações, abstrair dados e produzir informações. Elas facilitaram a percepção, o raciocínio, a memorização e a tomada de decisão” (CYBIS, 2010, p 17).

Para gerar tais interfaces, os designers devem conhecer as condições intelectuais dos seus usuários, bem como ter ciência de suas possíveis diferenças em termos de inteligência e personalidade.

Segundo CYBIS (2010), a interação homem-computador deve ser entendida como um processo em constante evolução, considerando que as pessoas em sua diversidade natural elaboram estratégias e situações de operação muito diferentes, e essas estratégias e situações de uso evoluem com o tempo e com o uso do sistema.

No entanto, como conceber algo fácil de usar em tantas situações diferentes? Em poucas palavras, serão necessários recursos especializados e o envolvimento com os usuários. Será necessária uma equipe que reúna conhecimento, competência técnica e metodológica para trabalhar com os usuários, além do apoio de ferramentas especializadas. É evidente que a construção de interfaces com usabilidade vai exigir mais tempo e dinheiro, afinal as empresas precisam implementar um esforço sistemático e continuado para garantir esse desenvolvimento. A ideia da otimização: maximizar os resultados e minimizar os recursos necessários (CYBIS, 2010, p 17).

2.2.1 Princípios Ergonômicos

O desenvolvimento de um sistema de qualidade é resultado de um estudo minucioso dos diversos componentes do contexto de uso. Porém, de forma a facilitar o processo de confecção de interfaces, foi proposto uma estrutura de base, a partir da qual uma interface pode proporcionar o estabelecimento da usabilidade na relação usuário-sistema. Essa configuração se faz seguindo critérios, princípios ou heurísticas de usabilidade propostos por diversos autores e instituições nas últimas décadas (CYBIS, 2010).

A seguir serão apontados alguns sistemas e princípios, que segundo Cybis (2010), foram concebidos para coordenar a etapa de desenvolvimento, assegurando a produção de interfaces melhores.

Como a ferramenta proposta, visa utilizar os critérios de Bastien e Scapin, para formulação de sua lista de verificação, este princípio apresentará maior detalhamento.

2.2.1.1 Heurísticas de Usabilidade

Jacob Nielsen (1994), propôs um conjunto de dez recomendações base para qualquer interface, qualidades estas que ele intitulou de heurísticas de usabilidade. São elas:

1. Visibilidade do estado do sistema;
2. Mapeamento entre o sistema e o mundo real;
3. Liberdade e controle ao usuário;

4. Consistência e padrões;
5. Prevenção de erros;
6. Reconhecer em vez de relembrar;
7. Flexibilidade e eficiência de uso;
8. Design estético e minimalista;
9. Suporte para o usuário reconhecer, diagnosticar e recuperar erros;
10. Ajudar e documentação.

2.2.1.2 Regras de Ouro

Ben Shneiderman (2004), propôs oito regras de ouro, para o desenvolvimento de projetos homem computador. São elas:

1. Perseguir a consistência;
2. Fornecer atalhos;
3. Fornecer feedback informativo;
4. Marcar o final dos diálogos;
5. Fornecer prevenção e manipulação simples de erros;
6. Permitir o cancelamento das ações;
7. Fornecer controle e iniciativa ao usuário;
8. Reduzir a carga de memória de trabalho.

2.2.1.3 Princípios de Dialogo

A norma ISO 9241:110 propõe sete princípios de diálogo para o desenvolvimento de um projeto ou avaliação de uma interface humano-computador (ISO 9241:110, 2006). São eles:

1. Adaptação à tarefa;
2. Autodescrição (*feedback*);
3. Controle ao usuário;
4. Conformidade às expectativas do usuário;
5. Tolerância aos erros;
6. Facilidade de individualização;
7. Facilidade de aprendizagem.

2.2.1.4 Critérios Ergonômicos

Dominique Scapin e Christian Bastien, propuseram, em 1993, um conjunto de oito critérios ergonômicos principais que se subdividem em dezoito subcritérios e critérios elementares. O propósito deste sistema era diminuir a ambiguidade no reconhecimento e classificação das qualidades e problemas ergonômicos de softwares interativos, ou seja, quando diferentes especialistas utilizavam esses critérios como ferramenta de avaliação, obtinham resultados mais parecidos, diminuindo, assim um dos inconvenientes da falta de sistematização nos resultados (CYBIS, 2010).

Abaixo são evidenciadas as interpretações de CARNIEL e CYBIS (2003), para o critério Condução, de Bastien e Scapin, bem como expostas suas compreensões relativas aos subcritérios correspondentes a este princípio.

- **Condução:** este critério está relacionado, aos meios disponíveis para aconselhar, orientar, informar, e conduzir o usuário na interação com o computador (mensagens, alarmes, rótulos, etc.). É constituído pelos subcritérios:
 - **Presteza:** este subcritério engloba os meios utilizados, para levar o usuário a realizar determinadas ações, como por exemplo entrada de dados. Responde também por todos os mecanismos ou meios que permitem ao usuário, conhecer as alternativas, em termos de ações, conforme o estado ou contexto nos quais ele se encontra. A presteza diz respeito as informações que permitem ao usuário, identificar o estado ou contexto, no qual ele se encontra, de forma a exemplificar, podemos citar as ferramentas de ajuda em seu modo de acesso.
 - **Agrupamento e distinção entre itens:** este subcritério diz respeito à organização visual dos itens de informação, relacionados uns com os outros, de alguma maneira. O Agrupamento e distinção entre itens, leva em conta a topologia (localização), e algumas características gráficas (formato), para indicar as relações entre os vários itens mostrados, indicando se eles pertencem ou não a uma dada classe, ou ainda determinando as diferenças entre essas classes. Este princípio também diz respeito à organização dos itens de uma classe. Esta subdividido em:
 - **Agrupamento / Distinção por localização:** O critério em questão, diz respeito ao posicionamento relativo dos itens,

indicar se eles pertencem ou não a uma dada classe. Este critério também diz respeito ao posicionamento relativo dos itens dentro de uma classe.

- **Agrupamento/Distinção por formato:** diz respeito às características gráficas (formato, cor, etc.) que indicam se itens pertencem ou não a uma dada classe, ou que indicam diferenças entre classes distintas, ou ainda diferenças entre itens de uma dada classe
- **Legibilidade:** diz respeito às características lexicais das informações apresentadas na tela, que possam dificultar ou facilitar a leitura desta informação (brilho do caractere, contraste letra/fundo, tamanho da fonte, espaçamento entre palavras, espaçamento entre linhas, espaçamento de parágrafos, comprimento da linha, etc.). Por definição, o critério Legibilidade não engloba mensagens de erro ou de feedback.
- **Feedback imediato:** diz respeito às respostas do sistema, às ações do usuário. Estas entradas podem ir de um simples pressionar de tecla, até uma lista de comandos. Em todos os casos, respostas do computador devem ser fornecidas, de forma rápida, com passo (*timing*) apropriado e consistente para cada tipo de transação. Em todos os casos, uma resposta rápida deve ser fornecida com informação sobre a transação solicitada e seu resultado.

Dando continuidade, e permanecendo com as perspectivas de CARNIEL e CYBIS (2003), sobre o conjunto de princípios ergonômicos propostos por Bastien e Scapin, serão apresentados, de forma condensada, os critérios Carga de Trabalho, Controle explícito, Adaptabilidade e seus respectivos subcritérios.

- **Carga de trabalho:** diz respeito a todos elementos da interface que têm um papel importante na redução da carga cognitiva e perceptiva do usuário, e no aumento da eficiência do diálogo. Este critério é subdividido em:
 - **Brevidade:** diz respeito à carga de trabalho perceptiva e cognitiva, tanto para entradas e saídas individuais, quanto para conjuntos de entradas (conjuntos de ações necessárias para se alcançar uma meta). Brevidade

corresponde ao objetivo de limitar a carga de trabalho de leitura e entradas, e o número de passos. Este subcritério está dividido em:

- **Concisão:** diz respeito à carga perceptiva e cognitiva de saídas e entradas individuais. Por definição, Concisão não engloba às mensagens de erro e de feedback.
 - **Ações mínimas:** diz respeito à carga de trabalho em relação ao número de ações necessárias à realização de uma tarefa. O que temos aqui é uma questão de limitar, tanto quanto possível, o número de passos pelos quais o usuário deve passar.
- **Densidade informacional:** diz respeito à carga de trabalho do usuário, do um ponto de vista perceptivo e cognitivo, com relação ao conjunto total de itens e informação apresentados ao mesmo, e não a cada elemento ou item individual.
- **Controle explícito:** diz respeito tanto ao processamento explícito, gerado pelo sistema em resposta às ações do usuário, quanto do controle que os usuários têm sobre o processamento de suas ações para com o sistema. Esta subdividido em:
 - **Ações explícitas do Usuário:** estão relacionados, às relações entre o processamento do sistema, ocasionado pelas ações do usuário. Esta relação deve ser explícita, o computador deve processar somente aquelas ações solicitadas pelo usuário e somente quando for solicitado.
 - **Controle de usuário:** está relacionado ao fato de que, os usuários deveriam estar sempre no controle do processamento do sistema (interromper, cancelar, suspender e continuar). Cada ação possível do usuário deve ser antecipada e opções apropriadas devem ser oferecidas.
- **Adaptabilidade:** diz respeito a sua capacidade de reagir conforme o contexto, e conforme as necessidades e preferências do usuário. Subdividido em:
 - **Flexibilidade:** se refere aos meios colocados à disposição do usuário, que permite a personificação da interface, levando em conta as exigências da tarefa, de suas estratégias e hábitos de trabalho. Ela corresponde também ao número das diferentes maneiras à disposição do usuário para alcançar um certo objetivo. Trata-se em outros termos, da capacidade da interface de se adaptar as variadas ações do usuário.

- **Consideração da experiência do usuário:** diz respeito aos meios implementados, que permitem que, o sistema respeite o nível de experiência do usuário.

Por fim, ainda sob o ponto de vista de CARNIEL e CYBIS (2003), serão apresentados os conceitos de Bastien e Scapin restantes, através da exposição dos critérios Gestão erros, Homogeneidade / Consistência, Significado de Códigos e Denominações, Compatibilidade e os subcritérios correspondentes

- **Gestão de erros:** diz respeito a todos os mecanismos que permitem evitar ou reduzir a ocorrência de erros, e quando eles ocorrem, tem apresenta as formas de correção. Os erros, aqui considerados, são ocasionados através da entrada de dados incorretos, entradas com formatos inadequados, entradas de comandos com sintaxes incorretas. Dividido nos seguintes subcritérios:
 - **Proteção contra os erros:** diz respeito aos mecanismos empregados, para detectar e prevenir os erros de entradas de dados, comandos, possíveis ações de consequências desastrosas e/ou não recuperáveis.
 - **Qualidade das mensagens de erros:** está relacionado à pertinência, legibilidade e exatidão da informação, dada ao usuário, sobre a natureza do erro cometido (sintaxe, formato), e sobre as ações a serem executadas para corrigir tal erro.
 - **Correção de erros:** diz respeito aos recursos disponibilizados ao usuário, com o objetivo de permitir a correção de eventuais erros.
- **Homogeneidade / consistência:** diz respeito à forma na qual as escolhas, na concepção da interface (códigos, denominações, formatos, procedimentos) são conservadas idênticas, em contextos idênticos, e diferentes para contextos diferentes.
- **Significado de códigos e denominações:** diz respeito a adequação, entre o objeto ou informação apresentada, e sua referência. Códigos e denominações possuem uma forte relação semântica com seu referente. Termos pouco expressivos para o usuário podem ocasionar problemas de condução onde ele pode ser levado a selecionar uma opção errada.

- **Compatibilidade:** refere-se ao acordo entre as características do usuário (memória, percepção, hábitos, competências, idade, expectativas, etc.) e das tarefas, de uma determinada aplicação. Este critério, diz respeito também, ao grau de similaridade entre diferentes ambientes e aplicações.

2.3 TÉCNICAS DE AVALIAÇÃO

Para Cybis e outros (2010, p. 209), “As técnicas de avaliação de ergonomia são diagnósticas e se baseiam em verificações e inspeções de aspectos ergonômicos das interfaces que possam colocar-se como um problema ao usuário durante sua interação com o sistema.”. De acordo com autores, essas avaliações podem ser classificadas em:

- Analíticas: caracterizadas por verificar se complexidade ou tempo de interação das tarefas realizadas estão de acordo com as interações iniciais definidas, com o uso de dispositivo informatizado.
- Heurísticas: devem realizadas por avaliadores que possuam experiência no ramo ergonômico, para que sejam identificados mais facilmente os aspectos responsáveis por causar dificuldade aos usuários, tendo como foco as interfaces de relação dos usuários com o sistema.
- Inspeções por listas de verificação: como as avaliações heurísticas, as avaliações por inspeção também focam a análise das interfaces de relacionamento usuário e sistema, porém podem ser realizadas por pessoas sem conhecimento aprofundado em ergonomia, sendo necessário que possuam um conhecimento da ferramenta que será utilizada para realizar tais inspeções.

2.3.1 Avaliação Analítica

Cybis e outros (2010) mencionam que essa técnica de avaliação permite que seja identificado logo no início de um projeto de desenvolvimento de uma Interface Humano-Computador, se a organização proposta para as tarefas interativas será consistente, se oferecerá um índice aceitável de carga de trabalho para o usuário que utilizar a interface e se o usuário conseguirá controlar de forma eficiente as tarefas que necessita fazer.

Dessa forma o analista deverá utilizar um modelo de estrutura de tarefas para definir os objetivos do usuário, as operações que serão realizadas na interface, os métodos utilizados e as regras de seleção. Após definidas essas premissas, deve-se associar tempos para cada operação física e cognitiva relacionada ao usuário. (CARD et al., 1983 apud CYBIS, 2010, p.210).

A aplicação dessa técnica tem como proposta o uso da interface por um usuário, no melhor tempo possível, sem levar em consideração erros ou incidentes de interação que poderiam ocorrer em um ambiente comum de uso. (CYBIS et al., 2010, p. 210).

2.3.2 Avaliação Heurística

É mencionado também por Nielsen e Molich (1990 apud BARBOSA e SILVA, 2010, p. 315), que a avaliação heurística é utilizada durante um processo de design iterativo, para encontrar problemas de usabilidade através da inspeção sistemática das interfaces de maneira rápida e de baixo custo, quando comparada com outros métodos de avaliação empíricos.

Segundo Cybis e outros (2010, p. 212), para que essa técnica seja aplicada com eficiência, é necessário que seus avaliadores possuam experiência no assunto, fazendo com que a taxa de sucesso desses tipos de avaliação dependa muito de quem está aplicando essas técnicas.

Nielsen (1994 apud CYBIS et al., 2010, p. 213), menciona através de dados fundamentados, que é possível se obter um melhor custo-benefício na formação das equipes de avaliação, desde que cada equipe possua entre quatro a cinco avaliadores, e que estes sejam especialistas em usabilidade.

Cybis e outros (2010, p. 212), escrevem: “os avaliadores baseiam-se em heurísticas ou em padrões de usabilidade gerais, próprios ou desenvolvidos por especialistas na área, como Jakob Nielsen, Ben Schneiderman, Dominique Scapin e Christian Bastien”. Os autores, também afirmam que durante uma avaliação, os avaliadores frequentemente trocam, de acordo com a experiência e conhecimento, de perspectivas, sendo estas:

- Abordagem por objetivos dos usuários: são avaliadas as principais tarefas que são realizadas pelos usuários;
- Abordagem pela estrutura de interface: avaliação dos menus da interface em níveis hierárquicos, tanto por profundidade como por largura;

- Abordagem pelos níveis de abstração: a interface pode ser avaliada em dois sentidos, de cima para baixo (*top-down*) ou de baixo para cima (*bottom-up*), segundo um modelo linguístico que pode ser dividido em 4 níveis de abstração;
- Semântico: mostra o significado das ações que são realizadas na interface pelo usuário;
- Sintático: sequência das ações de comandos e das apresentações nas telas;
- Léxico: significado das unidades de apresentação;
- Físico: características perceptíveis das apresentações e dos dispositivos de entrada;
- Abordagem pelos objetos das interfaces: avaliação a partir de um modelo de estrutura de objetos de interação;
- Abordagem pelas qualidades esperadas das interfaces: a interface é avaliada de acordo com qualidades, princípios, heurísticas ou critérios de usabilidade que deveriam estar presentes;

Como mencionado anteriormente, dependendo dos níveis de conhecimento e experiência dos avaliadores, é possível que haja muita discrepância nos resultados obtidos de avaliador para avaliador. Dessa forma, é necessário que exista um plano de trabalho, que guiará os diagnósticos resultantes das avaliações. Descrito por Cybis e outros (2010, p. 214), esses planos de trabalho são divididos em:

- Análise do contexto da avaliação: análise pelo contexto da avaliação, pelo responsável pela avaliação em conjunto com os responsáveis pelo software, para verificar os recursos disponíveis os objetivos da avaliação;
- Montagem da equipe de avaliadores: os membros da equipe devem ser escolhidos de acordo com experiência e competência na avaliação de sistemas similares ao que será avaliado, e de acordo com os recursos disponíveis, tendo como objetivo o responsável pela avaliação, de disponibilizar informações, organizar reuniões, apoiar execução e redigir o relatório final da avaliação;
- Análise do contexto de operação do sistema: utiliza-se um documento de especificação do contexto de uso do sistema em processo de avaliação. Se não existir o documento, é necessário a aplicação e técnicas de análise por

questionários ou entrevistas para se descobrir quais são as informações de contexto necessárias a avaliação;

- Análise de conhecimento disponível: é necessário que o gerente de avaliação possua conhecimento das qualidades esperadas para a interface e usabilidade do software;
- Reunião de preparativos para a avaliação: são divididos os conhecimentos sobre os contextos da avaliação e da operação do sistema que será avaliado. Serão definidos os critérios considerados como prioritários e os cenários de uso que serão explorados, de acordo com os perfis dos avaliadores, softwares utilizados, interface avaliada, etc.;
- Execução da avaliação: avaliadores trabalharão em paralelo utilizando as informações recolhidas da etapa anterior, até recolherem informações o suficiente para realizar outra reunião e discutir os resultados objetivos das avaliações;
- Redação do relatório: essa etapa é considerada muito importante pois é nela onde serão registrados os problemas identificados e as soluções propostas pela equipe de avaliação. É de responsabilidade do chefe da equipe que faça a redação do relatório;
- Reunião de apresentação do relatório: os avaliadores e os envolvidos com o sistema reúnem-se para discutir os problemas encontrados e as melhorias propostas pela equipe dos avaliadores.

2.3.3 Inspeção por Lista de Verificação (*Checklist*)

É comentado por Barbosa e Silva (2010, p. 316), que métodos de inspeção atuam de forma a tentar prever possíveis problemas de ergonomia gerados por certas decisões de design. Por esse método não possuir interação direta com os usuários, esses problemas que podem ser encontrados são considerados problemas potenciais, e não reais.

Os avaliadores nesse caso, não precisam ser especialistas em ergonomia pois serão as qualidades explícitas da ferramenta utilizada (lista de verificação) que determinarão as possibilidades de avaliação, sendo necessário identificar na maioria das vezes problemas pequenos (chamados de ruídos) e que não impedem a interação do usuário com uma interface, mas apenas atrapalham, diminuindo a eficiência das interações realizadas e podendo

possivelmente causar uma má impressão do sistema a quem o utilizar. (CYBIS et al., 2010, p. 216).

Dessa forma, Cybis e outros (2010, p.216) apresentam algumas vantagens em relação a essa avaliação:

- Fornece conhecimento ergonômico (nas questões e notas explicativas) sobre os aspectos a avaliar;
- Sistematizar as avaliações em se tratando de qualidade a inspecionar;
- Sistematizar as avaliações em se tratando de abrangência de componentes a inspecionar;
- Reduzir a subjetividade normalmente associada a processos de avaliação;
- Reduzir o custo de avaliação, pois avaliadores especializados não são necessários.

Da mesma forma que as avaliações heurísticas, também é utilizado um planejamento de trabalho para garantir o planejamento adequado das atividades. Cybis e outros (2010, p. 214), o descrevem da seguinte forma.

- Análise do contexto da avaliação: segue o mesmo procedimento realizado na avaliação heurística;
- Montagem da equipe de avaliadores: nessa etapa os inspetores serão profissionais de outros setores da própria empresa, e que não possuam nenhum vínculo inicial ao projeto do sistema em avaliação;
- Análise do contexto de operação do sistema: segue o mesmo procedimento realizado na avaliação heurística;
- Definição e configuração da lista de verificação a aplicar: etapa onde é utilizada uma lista já existente ou não, e que pode ser especializada, ou até mesmo criada do zero, especificamente para o sistema a ser avaliado. Deve possuir uma escala que indique o nível de importância (nível 1, 2 e 3) das suas questões;
- Reunião de preparativos para a avaliação: a lista do passo anterior é analisada pelos avaliadores para definir a sua configuração atual e como a interface será percorrida durante as avaliações;

- Execução da avaliação: os avaliadores trabalharão em paralelo aplicando as listas de verificação definidas na etapa anterior. Devem se reunir para que os resultados obtidos individualmente sejam compartilhados com o resto da equipe;
- Redação do relatório: diferentemente da avaliação, esse procedimento não é tão crítico, e acaba sendo mais objetivo e simples, com a apresentação das questões não atendidas pelo software;
- Reunião de apresentação do relatório: mesmo procedimento executado na avaliação heurística.

2.3.4 Percurso Cognitivo

Wharton e outros (1994 apud BARBOSA e SILVA, 2010, p. 322), mencionam que este tipo de técnica tem como objetivo avaliar o nível de facilidade fornecido por um sistema para que um usuário aprenda a usá-lo.

Segundo Kieras e Polson (1991 apud CYBIS et al. 2010, p.218), essa técnica avalia a interação de um usuário com uma interface e os processos cognitivos que surgem através dessa interação, quando o usuário a executa pela primeira vez, estando de acordo com a lista de verificação relacionada às tarefas interativas, que é aplicada pelos avaliadores.

Barbosa e Silva (2010, p. 322) dizem que essa técnica está baseada no fato de que muitas pessoas preferem aprender a utilizar ou descobrir as funções presentes em um sistema na tentativa e erro do que utilizando um manual ou guia de usuário.

De acordo com Cybis e outros (2010, p.218), essa lista leva em consideração como um usuário é capaz de realizar certas tarefas de acordo com as funções que são oferecidas pelo software utilizado em questão, analisando a cognição envolvida entre usuário e a tarefa realizada.

É necessário que quem aplique essa técnica, tenha um conhecimento prévio do sistema, principalmente em relação aos caminhos necessários para a realização das principais tarefas e também como o usuário reagirá caso possua algum conhecimento sobre alguma tarefa ou operação disponível no sistema. Assim, o avaliador pode verificar cada caminho existente utilizando as questões disponíveis na lista de verificação, conforme mencionado por Cybis e outros (2010, p.218), que seguem a seguir:

- O usuário tentará realizar a tarefa certa? Ao iniciar uma tarefa, o usuário irá se propor à realização do que foi previsto pelo projetista?
- É possível visualizar o objeto associado à tarefa? É de fácil visualização pelo usuário?
- O usuário reconhece que o objeto possui relação com a tarefa? As denominações ou representações gráficas representam o que significam e são de fácil entendimento do usuário?
- O usuário saberá operar o objeto? O nível de conhecimento necessário para operar o sistema é compatível com a forma de interação proposta?
- A resposta a cada ação do usuário é bem representada no sistema, não o deixando em dúvida se as ações estão sendo completadas?

E mais tarde essa técnica acabou sendo aperfeiçoada por pesquisadores brasileiros, para auxiliar na recuperação e normalização das ações do usuário em situações em que erros ou incidentes ocorram. Dessa forma, Filgueiras (2009 apud CYBIS et al. 2010, p. 219) comenta sobre as duas regras outras regras adicionadas a lista de verificação inicial:

- Se alguma ação do usuário não for a certa, ele saberá que não está utilizando o sistema de forma eficiente, a lhe garantir o melhor caminho?
- Caso o usuário note que não está no caminho correto, o sistema permite o retrocesso do usuário para uma etapa anterior?

2.3.5 Inspeções Preventivas de Erros

Paternò (2002 apud CYBIS, 2010, p.219) menciona que essa técnica é aplicada com um conjunto de questões (*guidewords*) focadas em sistemas de alta responsabilidade e que procura por falhas de projeto que possivelmente levarão um usuário a cometer erros.

Isso faz com que o avaliador necessite ter um conhecimento prévio de qual é o contexto da operação e a estrutura da tarefa a ser avaliada. Serão avaliadas apenas as interfaces relacionadas às tarefas críticas, utilizando as questões anteriormente citadas, de acordo com três componentes básicos: entradas, realização e resultados. De acordo com cada resposta estabelecida, é possível levantar hipóteses de erros gerados no sistema e as suas consequências.

Abaixo, algumas dessas perguntas mencionadas por Cybis e outros (2010, p. 220).

- E se nada acontecer?

- E se algo diferente acontecer?
- E se algo acontecer a mais?
- E se algo acontecer a menos?
- E se acontecer outro tipo de entrada?
- E se algo acontecer fora de tempo?
- E se algo acontecer antes?
- E se algo acontecer depois?

O resultado dessas inspeções trará:

- Guideword de desvio possível;
- Explicações sobre os desvios;
- Causas dos desvios;
- Consequências dos desvios;
- Recomendações de re-projeto.

Assim, de acordo com os resultados, os erros considerados mais críticos ao sistema, e com maior probabilidade de acontecimentos, farão com que medidas sejam tomadas para uma reorganização das interfaces avaliadas (CYBIS e outros, 2010, p. 220).

2.4 FERRAMENTAS CAUSE

A sigla CAUSE é um acrônimo da expressão *Computer-Aided Usability Engineering* (Engenharia de Usabilidade Assistida por Computador). Através do uso deste tipo de ferramenta computadorizada, é possível antever de forma muito eficiente, múltiplas tarefas do ciclo de vida do projeto. Os atributos que caracterizam um instrumento CAUSE, estão relacionadas a possibilidade desta ferramenta realizar as seguintes tarefas:

- Prototipação de modelos de interfaces de usuário para testes pelo próprio usuário;
- Construção interativa de esboços de telas, caixas de diálogo, ícones, botões, etc, através de manipulação direta;
- Manipulação interativa e fácil utilização de notações formais, especificações, modelos e técnicas de análise de tarefas, de maneira a

reduzir as barreiras para seus usos (JOHNSON AND JOHNSON, 1990 apud NIELSEN, 1994);

- Representação de hipermídia dos padrões de interface de usuário e suas diretrizes, permitindo que exemplos animados de técnicas de interação possam ser vistos pelos designers e que pulem entre assuntos relacionados;
- Design de representações racionais;
- Simulação de interfaces avançadas através de uma pessoa, para construir mais facilmente respostas que são restringidas de acordo com as regras do experimento (conhecida como técnica do Mago de OZ, “Wizard of OZ”);
- Registro de logs para posteriormente ser identificado o tempo decorrido das ações ou eventos de um usuário, podendo informá-los com um conjunto pré-especificado de códigos ou comentários de forma livre;
- Suporte à tradução e localização para interfaces de usuários internacionais;
- Logs de eventos ou de número de teclas pressionadas para realizar certas tarefas, tanto para o uso em testes de usuários ou instrumentação dos sistemas instalados;
- Uso de um banco de dados para registro das reclamações e suporte oferecido, como ferramentas de análise para possivelmente extrair mais informações armazenadas no banco de dados. (NIELSEN, 1994, p. 264).

2.4.1 Ergolist

Desenvolvido em 1997 pelo LabIutil - Laboratório de Utilizabilidade - da Universidade Federal de Santa Catarina – UFSC, é uma ferramenta para avaliar a ergonomia de interfaces humano-computador, que se dá através de três módulos. O módulo de *Checklist*, que inspeciona a qualidade ergonômica da interface entre usuário e sistema. O módulo das Questões, tendo como objetivo o fornecimento das informações das questões relacionadas ao módulo de *Checklist*. E o módulo das recomendações, representando as recomendações ergonômicas que podem ter influência nas decisões de projeto de interface entre usuário e máquina. Por se tratar de uma ferramenta *web*, é possível ser acessada pelo endereço: (LABORATÓRIO DE UTILIZABILIDADE DA INFORMÁTICA, 2011).

2.4.2 Glist

Este sistema propôs abastecer um sistema, para avaliação de usabilidade, com a elaboração de um *checklist* próprio. Foi desenvolvido e objetivado a partir de estudos aprofundados sobre o tema.

O GLIST foi disponibilizado na *Web*, com a intenção de possibilitar aos pesquisadores e especialistas da área, uma maneira de criar uma base de recomendações apropriada para as características específicas do produto a ser avaliado. Através dessa ferramenta desenvolvedores poderiam acessar o sistema através da *Web*, e por meio do *checklist*, realizar uma avaliação eficiente e segura de seu produto (LESSA, 2006, p 15).

2.4.3 QUIS

Essa ferramenta foi desenvolvida por uma equipe de pesquisadores do Laboratório de Interação Humano-Computador, da Universidade de Maryland, na cidade de College Park. Com o objetivo de avaliar o nível de satisfação de usuários, de acordo com aspectos específicos da interface humano-computador, encontra-se na ferramenta um questionário demográfico, uma medida em seis níveis da satisfação de uso dos sistemas, e nove medidas organizadas hierarquicamente de fatores específicos da interface, como: fatores de tela, terminologia e feedback do sistema, fatores de aprendizado, recursos do sistema, manuais técnicos, tutoriais online, multimídia, teleconferência e instalação de softwares. Cada área é responsável por medir a satisfação geral dos usuários de acordo com a parte da interface utilizada (*THE LABORATORY FOR AUTOMATION PSYCHOLOGY AND DECISION PROCESSES*, s. d.).

2.4.4 TOWABE

A TOWABE - *Tool for Web Application Usability Evaluation* (Ferramenta para Avaliação de Usabilidade em Aplicações para *Web*), foi desenvolvida com o objetivo de automatizar a avaliação de usabilidade em aplicações para *web*, sendo também uma aplicação *web*, excluindo a necessidade de migrações de sistemas operacionais para uso da ferramenta ou incompatibilidades provindas do ambiente em que for utilizada. Ela integra mais de uma técnica de avaliação de usabilidade, questionário de satisfação do usuário, inspeções de usabilidade utilizando *checklist* e card sorting. Possui também nativamente um mecanismo responsável por armazenar os dados coletados pelo *checklist*, custo zero para a aplicação dos métodos

implementados na ferramenta, que são utilizados pela própria ferramenta e a emissão dos relatórios através da própria ferramenta, sem ser necessário o uso de softwares de terceiros. Dessa forma, ao mesmo tempo que é mais flexível, permite que possa ser utilizada por um maior número de pessoas, independentemente de suas localizações.

Ela possui 3 módulos principais, o TCheck, responsável por inspeções de usabilidade utilizando *checklist*, o Tquest, que contém os questionários de satisfação para os usuários, o Tcat, utilizado para card sorting e mais oito módulos de suporte que incluem: Manutenção, Ajuda, Busca, Cadastro, Notícias, Relatório, Convite de Usuários e Aplicação. (ITAKURA, 2001, p. 29-30)

2.5 CONCLUSÃO DA ETAPA DE FUNDAMENTAÇÃO TEÓRICA

O trabalho de pesquisa, nos permitiu construir o embasamento teórico necessário para o desenvolvimento da solução proposta. Porém, apesar de haver uma diversidade significativa de fontes, esta etapa exigiu uma constante conferência das informações, visto que muitos trabalhos presentes na internet, não disponibilizam de forma correta, a autoria das concepções, o que desqualifica boa parte dos dados. Outra questão percebida, está atrelada a restrição de direitos autorais existentes para o compartilhamento de alguns artigos, sendo que boa parte desses documentos, exige uma contribuição financeira, para acesso. Para nós, esse tipo de barreira dificulta a disseminação do conhecimento, bem como retardam o desenvolvimento de novas tecnologias.

3 METODOLOGIA

A ciência destinada à solução de problemas, sempre buscou através da utilização de métodos, uma maneira de organizar ideias, de forma coerente e concisa, determinando sentido e facilitando o encontro de possíveis respostas. Seguindo esta lógica, os caminhos e passos praticados para idealização e construção deste projeto, serão detalhados ao decorrer do capítulo.

3.1 CARACTERIZAÇÃO DO TIPO DE PESQUISA

Segundo Fonseca (2002 apud GERHARDT e SILVEIRA, 2009), qualquer trabalho científico deve ser iniciado por meio de uma pesquisa bibliográfica, pois através desse levantamento, o pesquisador poderá tomar conhecimento sobre os avanços a respeito do assunto de interesse.

Para Gerhard e Silveira (2009), é sabido que ao se realizar uma pesquisa científica, uma abordagem qualitativa, permite aos pesquisadores explicar o porquê das coisas.

De acordo com estas definições, introdutoriamente este trabalho pode ser considerado uma pesquisa bibliográfica qualitativa, pois são utilizados conceitos já existentes, de autores conhecidos, para fundamentação da proposta. Porém, ao decorrer do trabalho, o sentido de pesquisa científica experimental se torna evidente, visto que um problema foi identificado e uma hipótese definida.

Após o desenvolvimento da proposta, o projeto é submetido a uma experimentação, no qual a solução elaborada é validada de maneira particular. Em caso de retorno satisfatório, o software entra em fase de conclusão.

Atendendo os objetivos, a ferramenta desenvolvida, pode ser utilizada, preferencialmente por pequenos grupos, e os dados obtidos podem ser quantificados. Por isso de uma maneira global, a pesquisa pode ser caracterizada como experimental qualitativa com dados quantitativos.

3.2 ETAPAS METODOLÓGICAS

As principais etapas metodológicas deste projeto são:

1. Escolha do tema a ser pesquisado e definição do orientador.

2. Formulação do problema, especificação dos objetivos e elaboração da justificativa.
3. Estruturação da pesquisa teórica e seleção de obras relevantes.
4. Definição da abordagem metodológica a ser utilizada na produção do trabalho.
5. Desenvolvimento da ferramenta solução; Modelagem, Codificação e Implementação.
6. Experimentação do sistema desenvolvido (Testes).
7. Análise dos resultados alcançados após a conclusão das etapas anteriores.
8. Entrega da ferramenta produzida.
9. Realização de testes de usabilidade, mediado pelo software elaborado.
10. Interpretação dos dados obtidos através do sistema desenvolvido.
11. Análise sobre os objetivos definidos e os resultados alcançados.
12. Conclusão e apresentação da relevância do trabalho.

A distribuição destas etapas ao longo do tempo, são apresentadas nos cronogramas, que se encontram em Apêndice A.

3.3 PROPOSTA DE SOLUÇÃO

O software proposto tem o objetivo de auxiliar os desenvolvedores, no processo de criação de sistemas automatizados de qualidade. Após o término deste trabalho, a ferramenta elaborada deverá atuar da seguinte forma:

- Um pequeno grupo deverá ser formado, de acordo com suas características de usuário. Este grupo será submetido a ferramenta desenvolvida.
- Esta ferramenta consiste em um questionário, coletor de informações, o qual irá realizar um levantamento sobre a percepção de uso de um determinado produto ou sistema.
- Após a realização deste checklist, as informações obtidas passam a ser interpretados e quantificados pelo sistema, de acordo com as heurísticas definidas.

- Através deste processo, relatórios e laudos poderão ser gerados, e como em um feedback, a equipe de criação poderá acompanhar os itens dependentes de melhoria.

3.4 DELIMITAÇÕES

Este trabalho apresenta as seguintes delimitações:

1. Para garantir a menor necessidade de captação de recursos, o desenvolvimento e estruturação da proposta deve ser baseada em sistemas de código aberto.
2. Os questionários elaborados, devem poder ser aplicados aos mais diferentes tipos de sistema, de maneira que, sua abstração possa fazer sentido ao maior número de aplicações possíveis.
3. A solução desenvolvida deverá ser disponibilizada gratuitamente pela internet, podendo assim ser utilizada pelos interessados no auxílio de ferramentas CAUSE.

3.5 RECURSOS NECESSÁRIOS

Como este trabalho é essencialmente intelectual, não é previsto a necessidade de captação de recursos explícita, para o seu desenvolvimento, sendo que as fontes de pesquisa foram obtidas através de bibliotecas físicas e digitais e repositórios, e a construção da solução foi elaborada pelos próprios pesquisadores.

Quanto a disponibilização do protótipo, estuda-se a possibilidade de hospedagem do sistema, em uma pequena estrutura provisória, havendo apenas a necessidade de aquisição de um computador pessoal, que irá operar como servidor, e acesso à internet para veiculação da ferramenta aos interessados.

3.6 CONCLUSÃO DA ETAPA METODOLÓGICA

Esta etapa permitiu direcionar nossa pesquisa, para o emprego da metodologia mais condizente com as características do projeto, bem como programar atividades, dimensionando a força de trabalho. Porém, apesar de toda essa sistematização, como não foi previsto o nível de dificuldade de cada tarefa, algumas projeções não puderam ser atendidas, mas seguirão no projeto, e servirão de sugestão para implementação em trabalhos futuros.

4 MODELAGEM

A modelagem de sistemas, é a etapa do desenvolvimento, onde os modelos abstratos são elaborados, de forma que, cada modelo represente uma percepção sobre o sistema, fato que ajuda o analista a entender o sistema (BOOCH, RUMBAUGH e JACOBSON, 2012). Este capítulo destina-se a apresentação genérica, através de modelos e diagramas, das características e funcionalidades da solução proposta.

4.1 UML

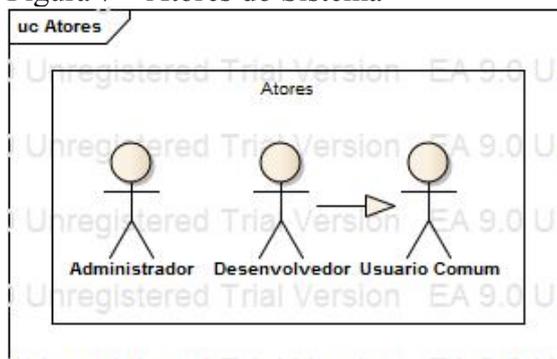
Segundo Booch, Rumbaugh e Jacobson (2012, p. 14), "a UML (*Unified Modeling Language*) é uma linguagem-padrão para elaboração da estrutura de projetos de software. Ela poderá ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software".

Apesar de ser uma linguagem muito expressiva, não é difícil de compreendê-la, pois ela pode ser utilizada independente do processo. (BOOCH, RUMBAUGH e JACOBSON, 2012).

4.1.1 Atores

Como mencionado por Booch, Rumbaugh e Jacobson (2012), atores de um sistema podem ser representados por humanos ou autômatos (dispositivos de hardware ou outros sistemas) que possuem uma interação individual com o sistema de forma específica e que dependendo da situação (um sistema em execução, por exemplo), um objeto pode representar o papel de mais de um ator, sendo considerado nesse caso a representação pelo autor, de um aspecto de objeto (Figura 7).

Figura 7 – Atores do Sistema



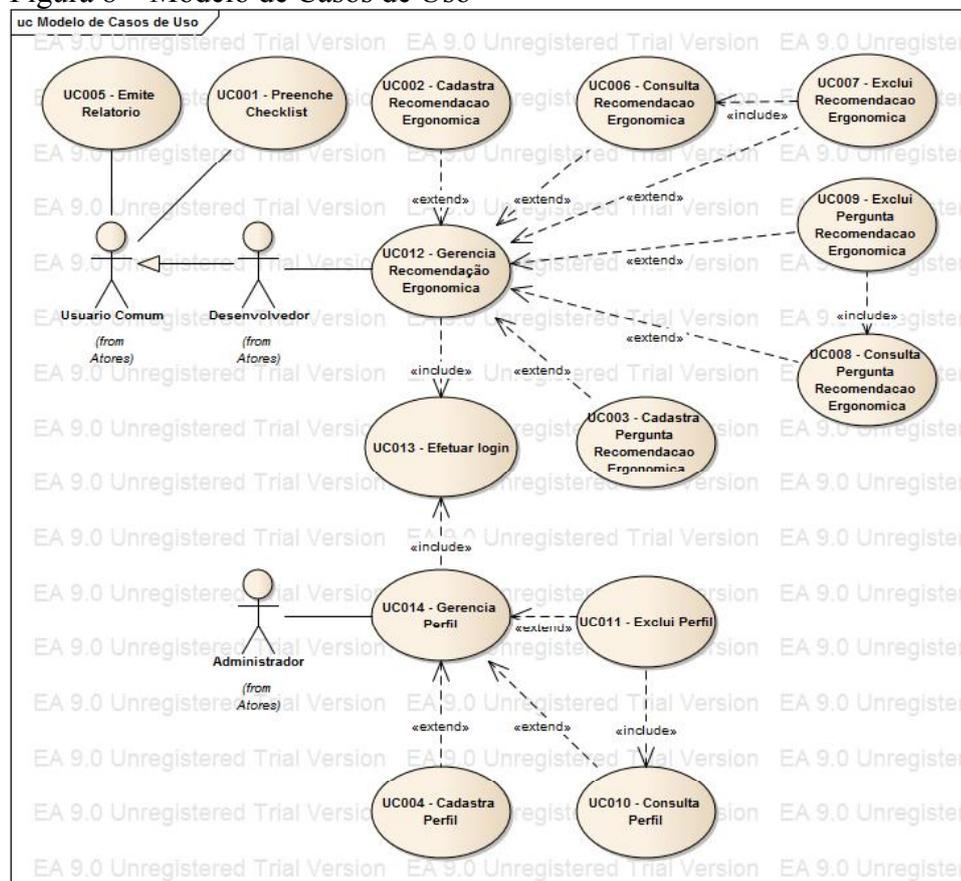
Fonte: Produzido pelos próprios autores (2016).

- Administrador: Único usuário com permissão para cadastro dos usuários do sistema de acordo com seus perfis de acesso.
- Desenvolvedor: Possui permissão de acesso para cadastrar, consultar e excluir as recomendações ergonômicas e perguntas vinculadas à sua conta e de utilizar o módulo de *checklist* com as recomendações e perguntas cadastradas manualmente.
- Usuário comum: Possui permissão de acesso apenas ao módulo de checklist padrão do sistema.

4.1.2 Casos de Uso

Os casos de uso podem ser aplicados de forma que ao longo do desenvolvimento de um sistema, desenvolvedores tenham a possibilidade de chegar a uma compreensão comum entre usuários finais e especialistas mais facilmente, já que é possível identificar o comportamento planejado para o sistema sem ser necessário descrever como esse seria implementado (BOOCH et al, 2012, p. 246) (Figura 8).

Figura 8 – Modelo de Casos de Uso



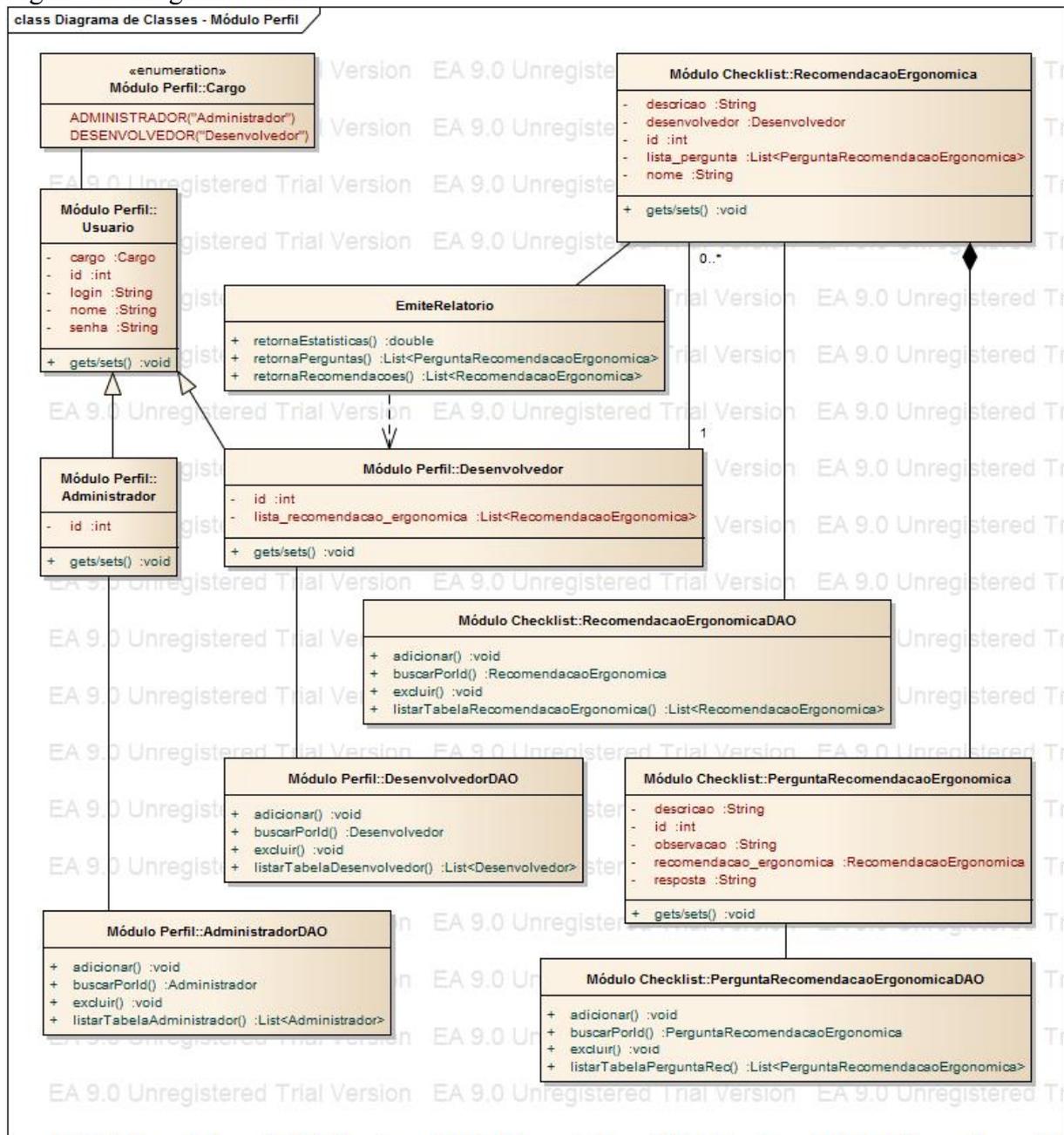
Fonte: Produzido pelos próprios autores (2016).

- UC001: onde a avaliação de determinado sistema de acordo com as recomendações ergonômicas definidas será realizada.
- UC002: parte responsável por permitir o cadastro das recomendações ergonômicas no sistema.
- UC003: permite o cadastro das perguntas relacionadas a cada recomendação ergonômica.
- UC004: onde os usuários do sistema são cadastrados pelo administrador de acordo com seus perfis de acesso ao sistema.
- UC005: emite o relatório contendo as informações das respostas de acordo com as heurísticas definidas para a realização do teste de avaliação das normas ergonômicas.
- UC006: permite a consulta das recomendações ergonômicas.
- UC007: permite a exclusão das recomendações ergonômicas.
- UC008: permite a consulta das perguntas das recomendações ergonômicas.
- UC009: permite a exclusão das perguntas das recomendações ergonômicas.
- UC010: permite a consulta dos perfis cadastrados no sistema.
- UC011: permite a exclusão dos perfis cadastrados no sistema.
- UC012: gerencia o cadastro, consulta e exclusão das recomendações ergonômicas e perguntas personalizadas.
- UC013: gerencia os acessos do sistema.
- UC014: gerencia o cadastro, consulta e exclusão dos perfis do sistema.

4.1.3 Diagrama de Classes

Booch et al (2012, p. 115) diz que, "os diagramas de classes são encontrados com maior frequência na modelagem de sistemas orientados a objetos. Um diagrama de classes mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos" (Figura 9).

Figura 9 – Diagrama de Classes



Fonte: Produzido pelos próprios autores (2016).

4.1.4 Requisitos Funcionais

Para os requisitos funcionais do sistema, que são responsáveis por indicar as funções que o software deve ter, segue relação ilustrada. (Figura 10)

Figura 10 – Requisitos Funcionais

req Requisitos Funcionais
RF001 - O sistema deve permitir o cadastro e exclusão de perfis de usuário com diferentes níveis de acesso
RF002 - O sistema deve permitir o cadastro, consulta e exclusão de recomendações ergonômicas
RF003 - O sistema deve permitir o cadastro, consulta e exclusão das perguntas vinculadas às recomendações ergonômicas
RF004 - O sistema deve possuir um módulo em que o usuário possa responder as perguntas das recomendações ergonômicas
RF005 - O sistema deve permitir que o usuário escolha quando deseja gerar o relatório

Fonte: Produzido pelos próprios autores (2016).

4.1.5 Requisitos Não Funcionais

Os requisitos não funcionais a seguir, mostram algumas funções com o objetivo de aumentar a facilidade, disponibilidade e usabilidade em relação ao sistema (Figura 11).

Figura 11 – Requisitos Não Funcionais

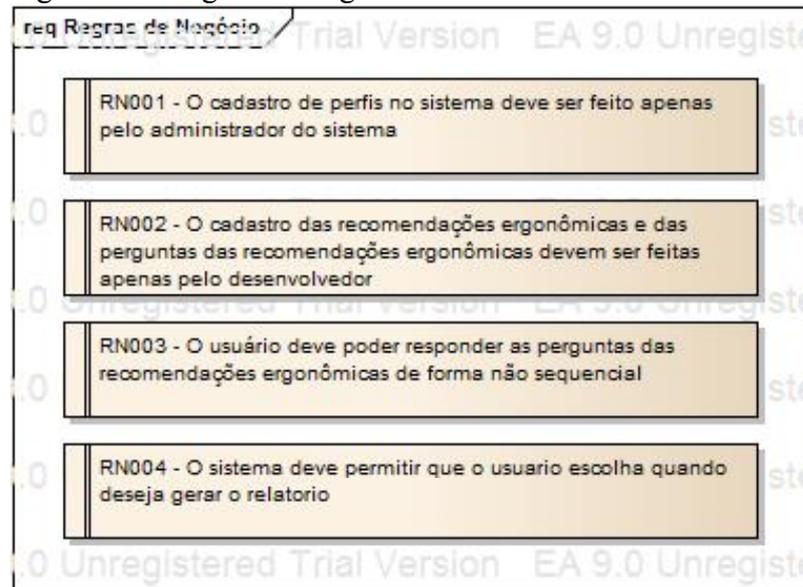
req Requisitos Não Funcionais
RNF001 - O sistema deve estar no ar 24 horas por dia
RNF002 - Deve ser possível acessar o sistema pela internet
RNF003 - O sistema deve estar sempre atualizado para garantir a compatibilidade com versões mais novas de navegadores

Fonte: Produzido pelos próprios autores (2016).

4.1.6 Regras de Negócio

As regras de negócio possuem o propósito de informar como as funções que são necessárias ao sistema, devem ser executadas (Figura 12).

Figura 12 – Regras de Negócio

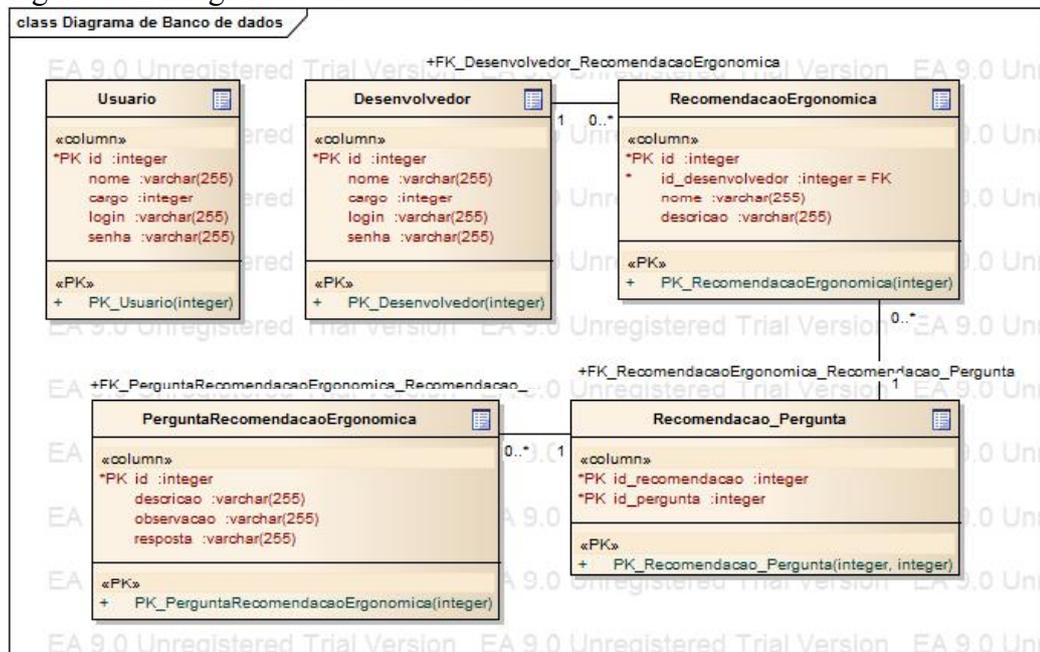


Fonte: Produzido pelos próprios autores (2016).

4.1.7 Diagrama de Banco de Dados

No contexto de base de dados, é apresentado o diagrama (Figura 13):

Figura 13 – Diagrama de Banco de Dados



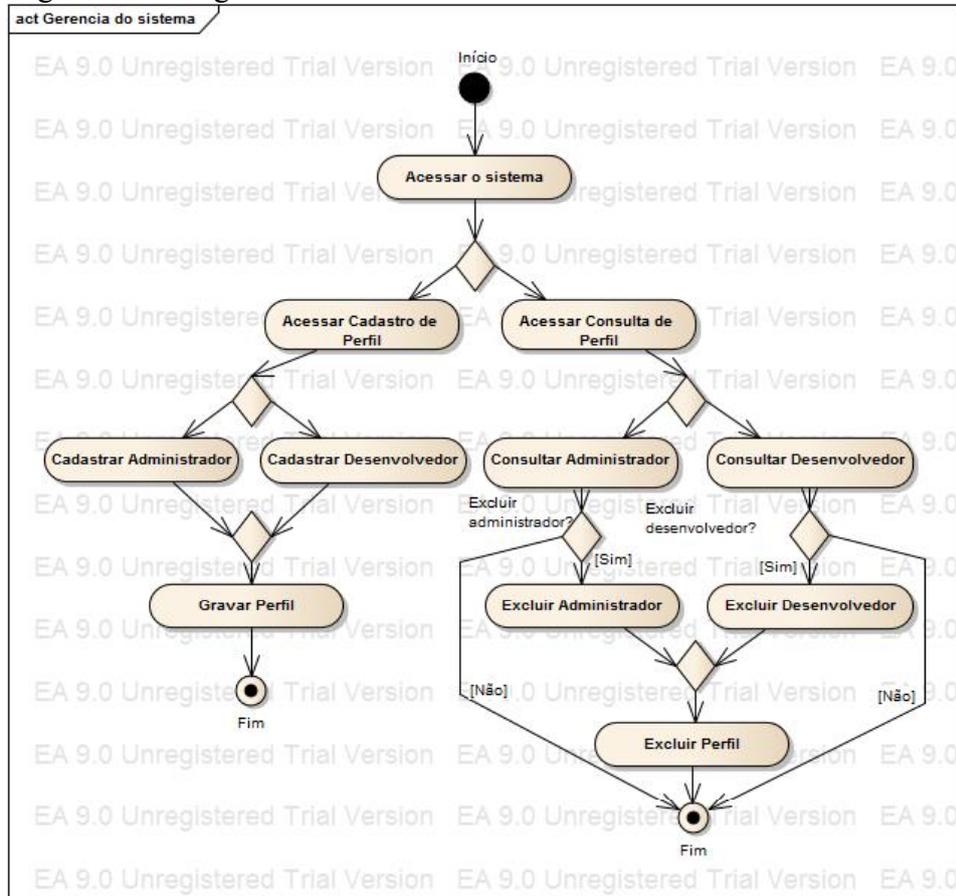
Fonte: Produzido pelos próprios autores (2016).

4.1.8 Diagrama de Atividades – Gerência do Sistema

Após o administrador acessar o sistema, é possível cadastrar administradores ou desenvolvedores no sistema, ou consultar os perfis já cadastrados. Caso opte-se pelo cadastro,

as informações entradas são salvas. Caso seja escolhido verificar os perfis já existentes, uma opção de exclusão para os perfis também é oferecida para o administrador utilizando o sistema (Figura 14).

Figura 14 – Diagrama de atividades - Gerencia do Sistema

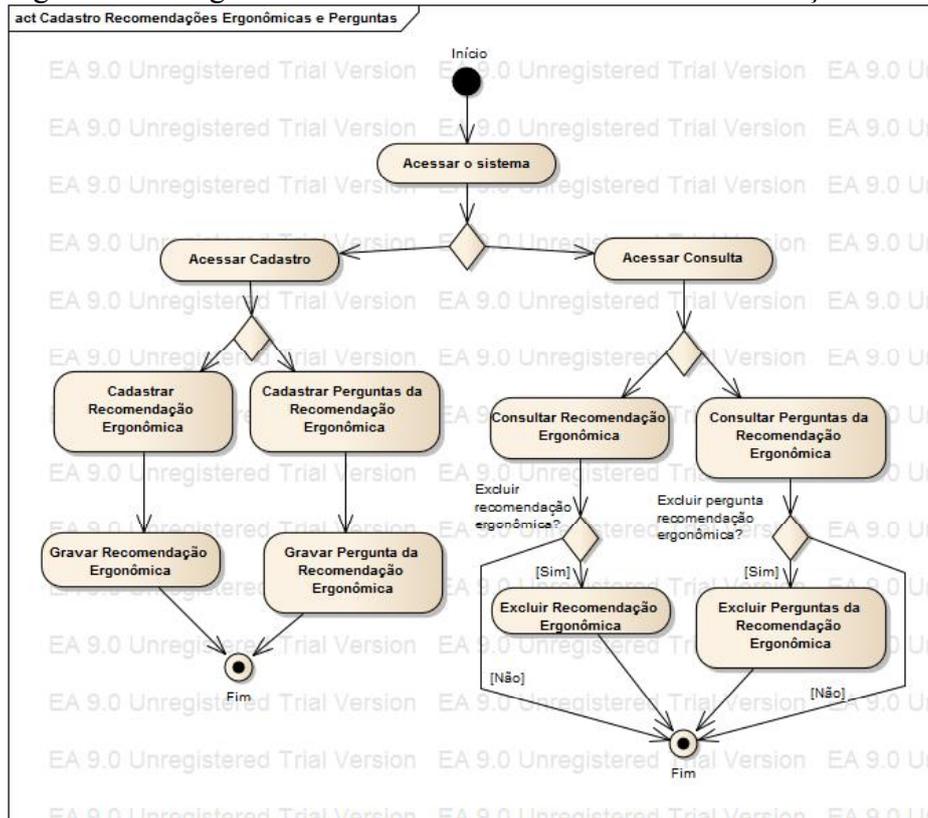


Fonte: Produzido pelos próprios autores (2016).

4.1.9 Diagrama de Atividades – Cadastro Recomendações e Perguntas

Após o acesso ao sistema com o perfil de desenvolvedor, o menu de cadastro de recomendações ergonômicas e perguntas das recomendações ergonômicas é habilitado. Se o menu de cadastro for escolhido, o desenvolvedor pode optar por cadastrar uma nova recomendação ergonômica ou uma pergunta à uma recomendação já existente. Se a opção inicial escolhida o menu de consulta, é possível escolher entre visualizar as recomendações ergonômicas ou as suas perguntas, que estão vinculadas à conta conectada ao sistema. Na tela de visualização, também é permitido que o desenvolvedor escolha quais recomendações ergonômicas ou perguntas possam ser excluídas (Figura 15).

Figura 15 – Diagrama de atividades - Cadastro de Recomendações



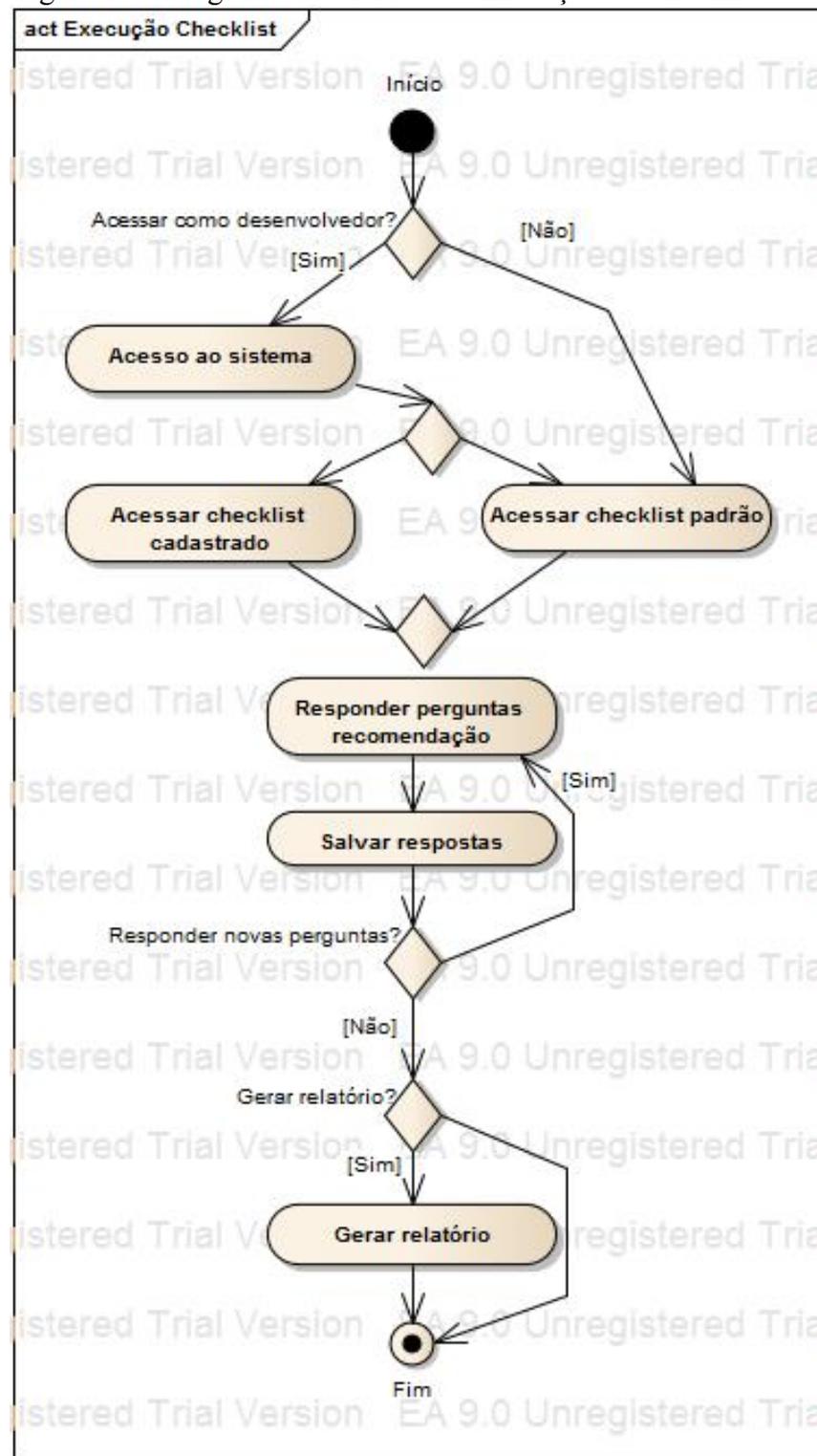
Fonte: Produzido pelos próprios autores (2016).

4.1.10 Diagrama de Atividades – Execução do Checklist

No módulo de avaliação o sistema permite que usuários comuns (sem conta no sistema) possam utilizar o checklist padrão do sistema e que também possam gerar um relatório ao final do processo.

Caso um desenvolvedor acesse o sistema, uma opção é renderizada para que seja possível utilizar o checklist personalizado com as recomendações ergonômicas e suas perguntas cadastradas manualmente, podendo ao final do processo gerar também um relatório (Figura 16).

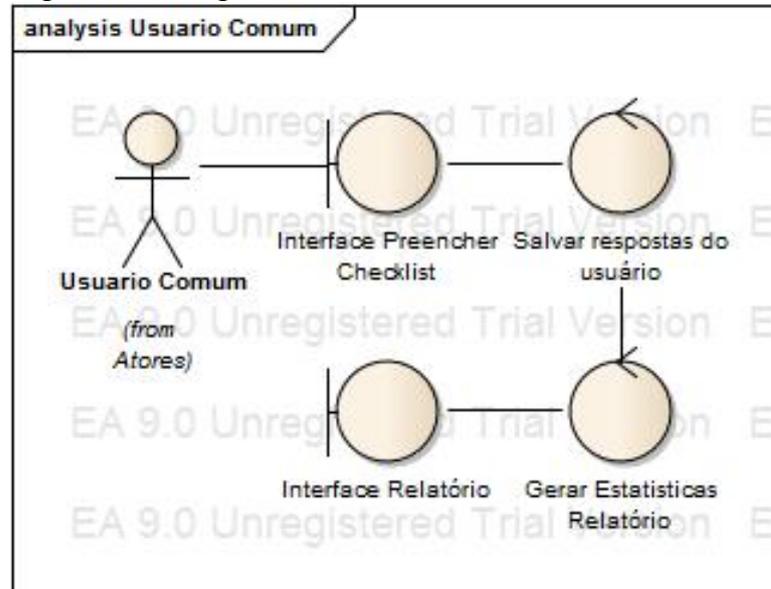
Figura 16 – Diagrama de atividades - Execução do Checklist



Fonte: Produzido pelos próprios autores (2016).

4.1.11 Diagrama de Robustez – Usuário Comum

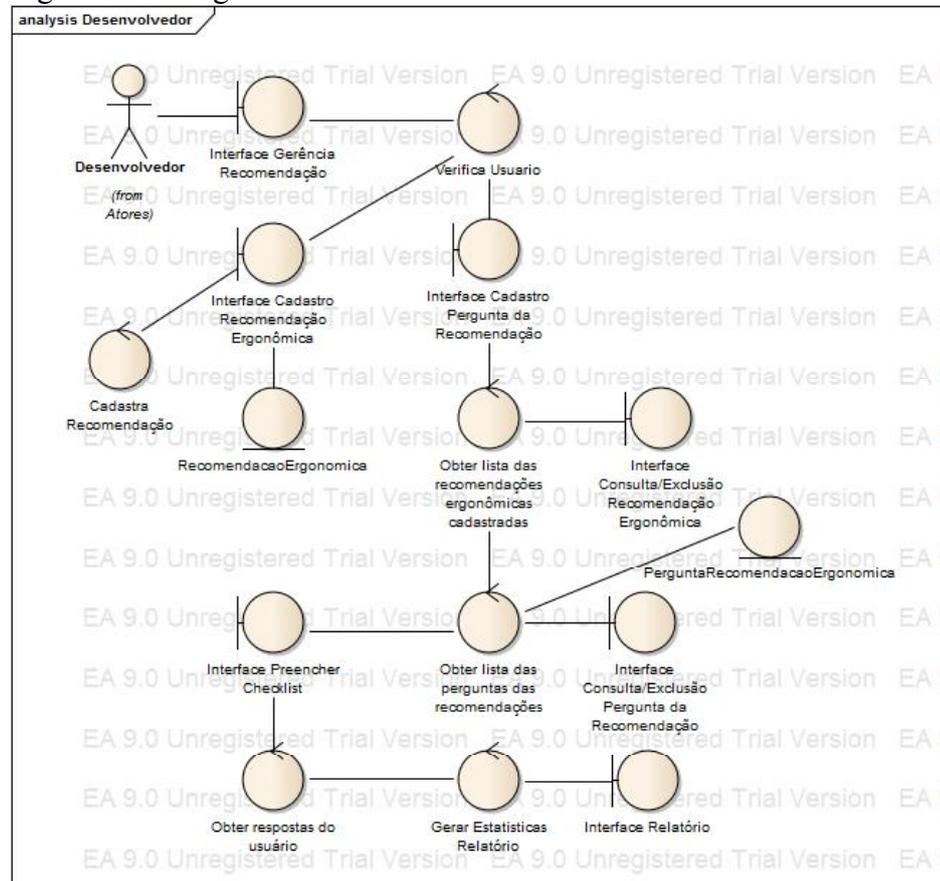
Figura 17 – Diagrama de Robustez – Usuário Comum



Fonte: Produzido pelos próprios autores (2016).

4.1.12 Diagrama de Atividades – Desenvolvedor

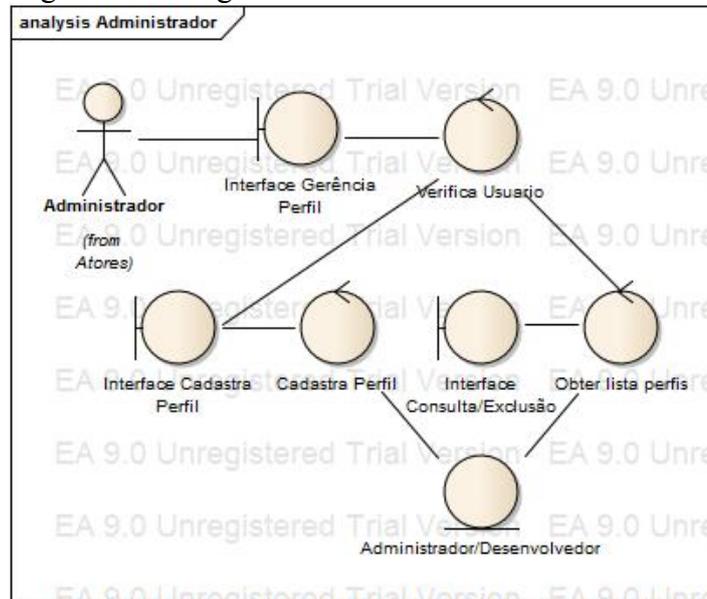
Figura 18 – Diagrama de Robustez – Desenvolvedor



Fonte: Produzido pelos próprios autores (2016).

4.1.13 Diagrama de Atividades – Administrador

Figura 19 – Diagrama de Robustez – Administrador



Fonte: Produzido pelos próprios autores (2016).

4.2 CONCLUSÃO DA ETAPA DE MODELAGEM

Booch et al (2012) menciona que um bom modelo é aquele que além de incluir componentes que possuem ampla repercussão, também exclui componentes que não são pertinentes dependendo o nível de abstração necessário para um sistema em questão.

Seguindo esta concepção, optamos por produzir modelos simplificados, que permitissem aos interessados, um melhor entendimento, do protótipo a ser desenvolvido, proporcionando de forma coerente, a difusão do conhecimento necessário para operar, manter e aperfeiçoar o sistema, abstraindo assim relações desnecessárias, que poluem a contextualização da ferramenta.

5 DESENVOLVIMENTO

Este capítulo destina-se a apresentação do passo a passo percorrido, durante o processo de concepção e desenvolvimento do protótipo idealizado.

5.1 TECNOLOGIAS E FERRAMENTAS UTILIZADAS

Para construção do sistema proposto, foram utilizadas competências adquiridas ao longo da graduação. A seguir serão detalhadas as principais tecnologias utilizadas para confecção da solução.

5.1.1 Enterprise Architect

Este sistema foi utilizado por ser familiar aos autores deste trabalho, sendo que o primeiro contato com essa ferramenta, ocorreu por intermédio da graduação, na instituição UNISUL. A seguir algumas características da ferramenta são evidenciadas.

O Enterprise Architect é uma ferramenta CASE que possui um ambiente de modelagem baseado em equipes, que abrange todos os aspectos de desenvolvimento de software. Foi concebido para oferecer meios para modelagem de negócios, engenharia de sistemas, arquitetura-corporativa, gerenciamento de requisitos, geração de códigos, dentre outros recursos (SBROCCO, 2014).

A seguir algumas de suas principais características do Enterprise Architect:

- Suporte ao ciclo de vida de modelagem de processos, dados e sistemas.
- Permite a utilização de técnicas de levantamento e documentação de requisitos.
- Contempla inúmeras notações e técnicas (BPMN, UML, Modelagem de Dados, SOAML, SysML, etc.).
- Possui abordagem completa em análise e projetos de sistemas conforme a UML.
- Oferece recurso de automatização na engenharia de código (geração, reversa e sincronização), contemplando múltiplas linguagens de programação.
- Interação nativa com o Visual Studio .NET e Eclipse.

- Geração de documentação de apoio a relatórios personalizados (em HTML e RTF).
- Permite rastreabilidade entre todos os elementos de modelagem (processos, regras, requisitos, casos de uso, classes, componentes, tabelas, etc.).
- Possui integração com ferramentas de gerenciamento e configuração e versionamento (como CVS, Subversion, SourceSafe etc).
- Workgroup, possibilitando uso compartilhado e seguro pelos usuários.
- Recursos de exportação e troca de informação com outras ferramentas, via XML.
- Permite criar protótipos de interfaces para validação e rastreabilidade de documentação.

5.1.2 PostgreSQL

Assim como a ferramenta Enterprise Architect, este sistema foi escolhido por ser de domínio dos desenvolvedores, sendo que o PostgreSQL, foi inserido ao cotidiano pelos docentes, ao longo da graduação. A seguir, alguns atributos da ferramenta serão relatados.

O PostgreSQL é um sistema gerenciador de banco de dados (SGBD) Open Source, sua continuação depende do grupo de desenvolvedores, em sua maioria voluntários, espalhados pelo mundo, que compartilham ideais e soluções, através da internet, mantendo o projeto atualizado (THE POSTGRES GLOBAL DEVELOPMENT GROUP, 2007).

Possui recursos como:

- Consultas complexas
- Chaves estrangeiras
- Integridade transacional
- Controle de concorrência multiversão
- Suporte ao modelo híbrido objeto-relacional
- Ele tem facilidade de Acesso
- Gatilhos
- Visões
- Linguagem Procedural em várias linguagens
- Indexação por texto
- Estrutura para guardar dados Georeferenciados

5.1.3 Bootstrap

Esta ferramenta foi escolhida por apresentar melhores resultados que o framework Primefaces, quanto o quesito, design de interface. A seguir iremos citar algumas características do Bootstrap.

O Twitter Bootstrap, possui uma base completa de estilos e design de interfaces pré-confeccionados, dos quais pode-se iniciar o desenvolvimento de uma aplicação, sem a necessidade de começar o projeto do zero. Este framework foi criado por Mark Otto e Jacob Thornton, funcionários do Twitter, a partir do código que eles utilizavam internamente, na manutenção da rede social. Após ter sido liberado como Open Source, ganhou muitos adeptos, esta ascensão trouxe a maturidade e importância no mercado necessária para desvincular o Twitter de seu nome, passando a ser conhecido apenas por Bootstrap (SPURLOCK, 2013).

O Bootstrap possui recursos como:

- Reset CSS
- Estilo Visual Base para a maioria das *Tags*
- Ícones
- Grids prontos para uso
- Componentes CSS
- Plugins JavaScript
- Tudo responsivo e *mobile-first*

5.1.4 Linguagem Java

Por possuir uma vasta documentação e sintaxe de alto nível, a linguagem Java comumente é utilizado por instituições educacionais, para promover o conhecimento de programação, o que também ocorre na Universidade do Sul de Santa Catarina, e motivou a escolha dos autores. Em seguida, algumas propriedades da linguagem serão anunciadas.

Em 1991, uma equipe liderada por James Gosling e Patrick Naughton, da Sun Microsystems, criou uma linguagem que eles chamaram de “*Green*”, esta foi projetada para ser simples e neutra, em relação à arquitetura, de modo que pudesse ser executada em uma variedade de hardwares. Java cresceu de maneira vertiginosa, pois os programadores abraçaram a linguagem, por ser mais simples do que as linguagens da época (HORSTMANN, 2004).

As principais características do Java são:

- Linguagem Concisa e Simples

- Suporte à Programação Orientada a Objetos
- Linguagem Robusta – Fortemente Tipada
- Comunidade de Usuários Ativa
- Portável
- Independente de plataforma
- Interpretada
- Compilada
- Vasta biblioteca

5.2 PROCESSO DE DESENVOLVIMENTO

Para os autores o processo de desenvolvimento teve início em sua idealização, através da construção de conhecimento, em seguida foram realizadas análises de sistemas existentes, com o objetivo de margear o projeto. Posterior a estas etapas, unindo os conceitos e perícia adquiridos, as ideias foram colocadas em prática, por meio de modelos e diagramas, por último o sistema começou a adquirir forma através do design de interfaces e codificação.

Após o desenvolvimento o sistema entrou em fase de teste e os ajustes necessários para entrega do protótipo foram realizados.

5.2.1 Problemas Identificados Durante a Etapa de Desenvolvimento

Durante o processo de desenvolvimento, os autores se deparam com o nível elevado de dificuldade para abstração das heurísticas e construção do questionário, por este motivo optaram por garantir a entrega do protótipo, para, pelo menos, um dos critérios de qualidade.

Em contrapartida, uma nova funcionalidade foi agregada ao sistema, que permite aos desenvolvedores, cadastrarem suas próprias heurísticas.

5.2.2 Perspectivas Inicial Sobre o Sistema

A intensão inicial do projeto consistia em elaborar um sistema que auxiliasse os desenvolvedores, na aplicação dos critérios de usabilidade, de Bastien e Scapin, aos mais variados produtos.

5.2.3 Teste do Sistema

Para a realização da validação da ferramenta, optou-se por realizar os testes com dois grupos distintos de usuários.

- Com uma professora, que possui conhecimentos na área de usabilidade.
- Com usuários comuns, que não possuíam conhecimento no assunto e estavam utilizando o sistema pela primeira vez.

Durante os testes, os objetivos procurados estavam associados à identificação da facilidade que um usuário comum utilizando o sistema pela primeira vez, encontraria ao navegar pelas interfaces e ao utilizar seus menus e funcionalidades. Também, com a ajuda da orientadora, quais das funcionalidades presentes ou não, seriam interessantes, úteis ou viáveis para uma ferramenta de avaliação ergonômica.

5.2.3.1 Validação com Usuário Experiente

Ao utilizar a ferramenta, a professora com conhecimentos em usabilidade, enumerou pontos referentes à organização e consistência das interfaces, dos nomes dados aos menus e botões, o nível de dificuldade em seguir o fluxo de uso da ferramenta, quantidade de cliques necessários para realizar uma tarefa, entre outros. A adição de novas funcionalidades que poderiam ser interessantes para trabalhos futuros também foi discutida.

5.2.3.2 Validação com Usuários Comuns

Com a ajuda de alguns alunos da orientadora, foi possível ter uma visão de um usuário comum tendo contato pela primeira vez com o sistema e quais dificuldades eram as mais comuns em utilizá-lo.

Logo no início dos testes, notou-se que os usuários tinham dificuldade em encontrar o início do fluxo de uso da ferramenta, já que da maneira inicial, no menu existente, dois dos botões principais, esses por sua vez possuíam botões tanto de cadastros como de consultas. Então optou-se por uma separação melhor desses botões, de forma que fosse criado um botão principal apenas para os cadastros, outro apenas para as consultas e por final um para a utilização da ferramenta de avaliação.

Esses testes ajudaram a confirmar alguns dos pontos descritos pelo usuário experiente nos testes anteriores e que levou à alteração de algumas partes do sistema, de como esses componentes estavam dispostos nas interfaces e também a maneira de como estavam sendo rotulados.

5.3 APRESENTAÇÃO DO SISTEMA

Nesse subcapítulo serão apresentadas as interfaces do sistema desenvolvido, e suas funcionalidades, de acordo com as interações disponíveis entre o sistema e usuário.

5.3.1 Principais Interfaces e Funcionalidades

A seguir serão ilustradas as principais interfaces do sistema.

5.3.1.1 Interfaces do Sistema

Na interface inicial do sistema, é disponibilizado para o usuário comum, apenas a ferramenta de avaliação com o checklist padrão do sistema (Figura 17).

Figura 20 – Interface - Inicial



Fonte: Produzido pelos próprios autores (2016).

Ao clicar no menu “Ferramenta de Avaliação” o usuário é levado à interface do “Checklist Padrão” contendo as recomendações padrões do sistema. Essas recomendações

ergonômicas e suas perguntas não podem ser excluídas ou alteradas e servem caso um usuário queira apenas fazer um teste rápido e genérico (Figuras 18 e 19).

Figura 21 – Interface - Recomendações Padrões



Fonte: Produzido pelos próprios autores (2016).

Figura 22 – Interface - Responder Checklist Padrão



Fonte: Produzido pelos próprios autores (2016).

Após clicar em “Responder Checklist” é possível responder as perguntas e clicar em salvar, o usuário é levado à interface anterior, onde pode escolher novamente mais recomendações ergonômicas e suas perguntas ou pode clicar no botão Gerar Relatório. Caso “Gerar Relatório” seja escolhido, as informações e estatísticas apenas das recomendações ergonômicas e perguntas respondidas serão mostradas na interface para que se possa analisar o

nível de critérios seguidos ou não, e serem devidamente corrigidos de acordo com a necessidade de quem os está aplicando (Figura 20).

Figura 23 – Interface - Relatório das Recomendações Ergonômicas

Informações Gerais			
Total de perguntas:	4		
Total de perguntas respondidas:	4		
Nº de perguntas respondidas <u>sim</u> :	3	Porcentagem do total de questões <u>sim</u> :	75.0%
Nº de perguntas respondidas <u>não</u> :	1	Porcentagem do total de questões <u>não</u> :	25.0%
Nº de perguntas <u>ignoradas</u> :	0	Porcentagem do total de questões <u>ignoradas</u> :	0.0%

Recomendação1
Perguntas: Pergunta1 Rec1 Pergunta2 Rec1 Pergunta3 Rec1
Respostas: Sim Sim Sim
Observações: obs1 obs2

Fonte: Produzido pelos próprios autores (2016).

Para conectar ao sistema como outro usuário (desenvolvedor ou administrador) é necessário clicar em “Sair”. A interface de *login* será apresentada (Figura 21).

Figura 24 – Interface - *Login* do sistema

Acesso ao Sistema

Entrar

Fonte: Produzido pelos próprios autores (2016).

De acordo com o perfil registrado no sistema, novos menus serão renderizados. Caso um desenvolvedor, será possível cadastrar, consultar e excluir suas próprias recomendações ergonômicas e suas respectivas perguntas (Figuras 22 até 27).

Figura 25 – Interface - Menu Cadastro Recomendações Ergonômicas e Perguntas



Fonte: Produzido pelos próprios autores (2016).

Figura 26 – Interface - Cadastro de Recomendações Ergonômicas

The screenshot shows the 'Cadastro de Recomendações Ergonômicas' form. It includes a navigation bar with the same menu items as Figure 25. The form has two input fields: 'Nome da Recomendação Ergonômica:' and 'Descrição da Recomendação Ergonômica:'. Below the description field is a blue 'Salvar' button.

Fonte: Produzido pelos próprios autores (2016).

Figura 27 – Interface - Cadastro de perguntas das Recomendações Ergonômicas

The screenshot shows the 'Cadastro de Perguntas para Recomendações Ergonômicas' form. It includes a navigation bar with the same menu items as Figure 25. The form has a dropdown menu for 'Selecione a Recomendação Ergonômica:' with 'Recomendação 1' selected. Below it is a text input field for 'Descrição da pergunta:' with a question mark icon to its right. A blue 'Salvar' button is located below the input fields.

Fonte: Produzido pelos próprios autores (2016).

Figura 28 – Interface - Menu Consulta e Exclusão Recomendações e Perguntas



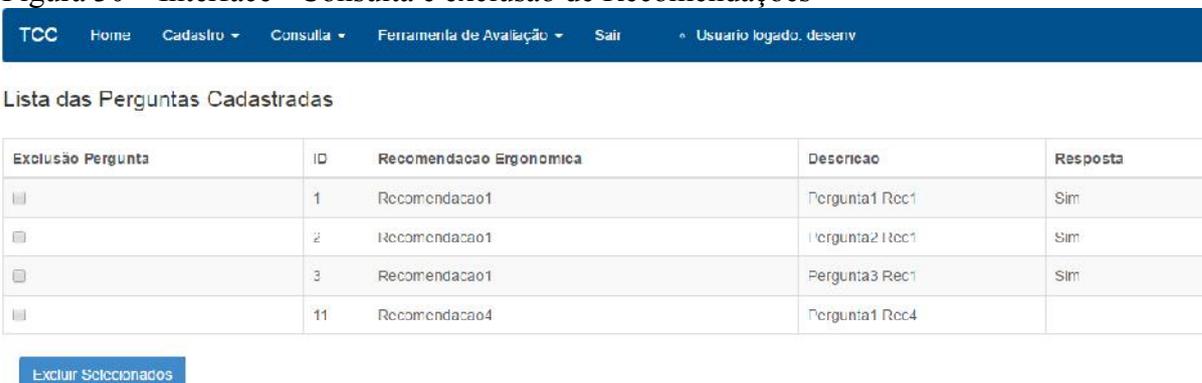
Fonte: Produzido pelos próprios autores (2016).

Figura 29 – Interface - Consulta e exclusão de Recomendações



Fonte: Produzido pelos próprios autores (2016).

Figura 30 – Interface - Consulta e exclusão de Perguntas



Fonte: Produzido pelos próprios autores (2016).

Para o desenvolvedor, o menu “Ferramenta de Avaliação”, apresentará um botão extra, o “Checklist Cadastrado”. A interface será semelhante à do “Checklist Padrão”, porém

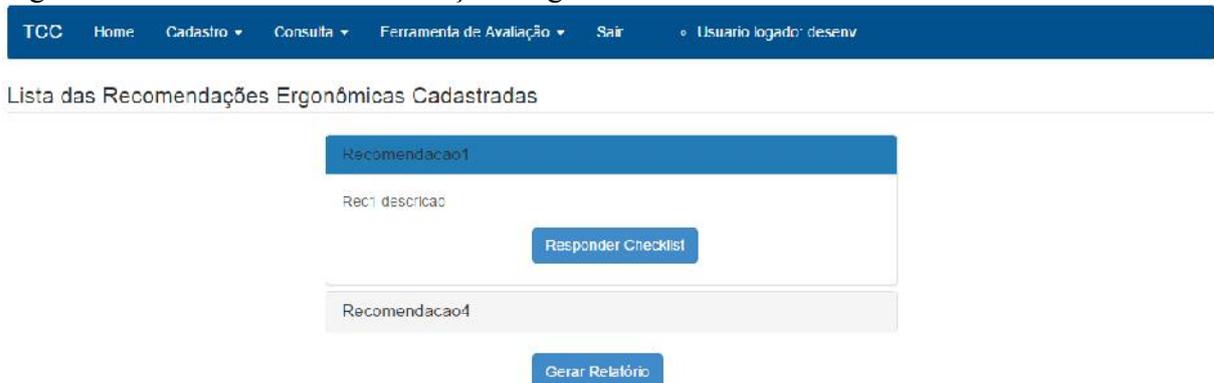
exibirá as recomendações ergonômicas e suas perguntas cadastradas manualmente pelo usuário conectado, e que estão vinculadas à sua conta (Figuras 28 e 29). O relatório é gerado da mesma forma que o checklist padrão.

Figura 31 – Interface - Ferramenta de Avaliação



Fonte: Produzido pelos próprios autores (2016).

Figura 32 – Interface - Recomendações Ergonômicas cadastradas manualmente



Fonte: Produzido pelos próprios autores (2016).

Os menus cadastro e consulta de perfis é renderizado apenas para os administradores do sistema (Figuras 30 até 35). Nele é possível cadastrar novos administradores ou desenvolvedores para o sistema.

Figura 33 – Interface - Cadastro de Perfil



Fonte: Produzido pelos próprios autores (2016).

Figura 34 – Interface - Cadastro de Administrador

Fonte: Produzido pelos próprios autores (2016).

Figura 35 – Interface - Cadastro de Desenvolvedor

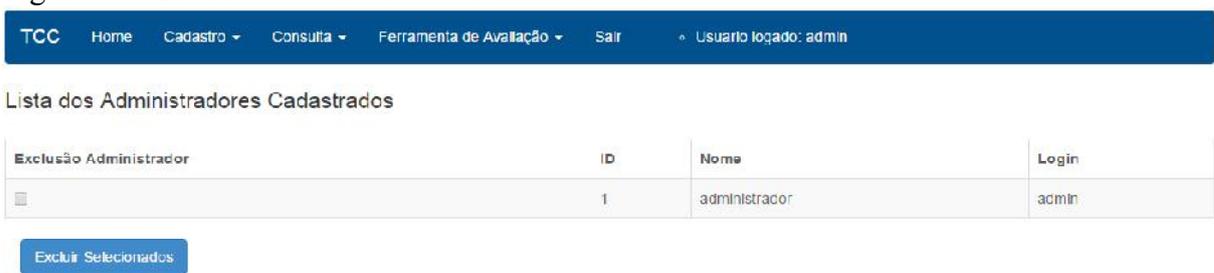
Fonte: Produzido pelos próprios autores (2016).

Figura 36 – Interface - Consulta de Perfil



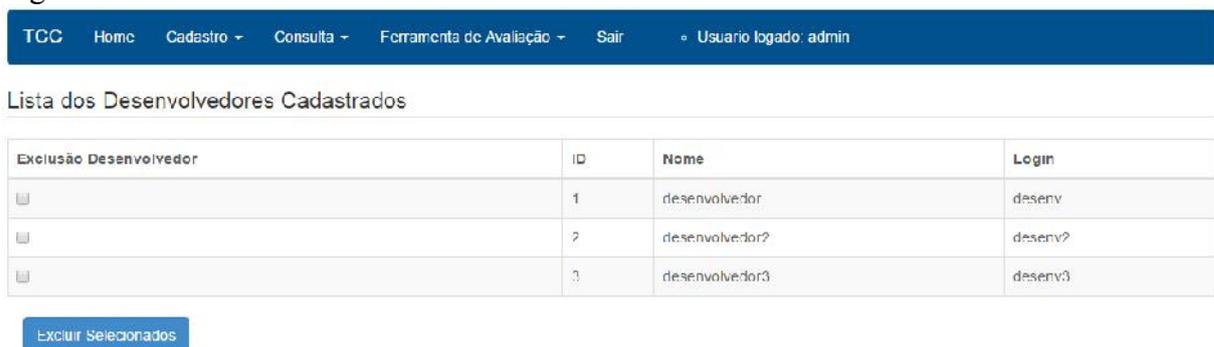
Fonte: Produzido pelos próprios autores (2016).

Figura 37 – Interface - Consulta de Administrador



Fonte: Produzido pelos próprios autores (2016).

Figura 38 – Interface - Consulta de Desenvolvedor



Fonte: Produzido pelos próprios autores (2016).

5.4 CONCLUSÃO DA ETAPA DE DESENVOLVIMENTO

O desenvolvimento do protótipo, foi moldado, de acordo com as dificuldades com as quais os autores se depararam ao decorrer do processo de elaboração, visto que dispunham

de um tempo determinado, para entrega do projeto. Para os autores, a maior dificuldade desta etapa, está relacionada, a necessidade de elaboração, de uma lista de verificação, abstrata o suficiente, para possibilitar o emprego dessas perguntas, ao processo de avaliação ergonômica, das mais variadas interfaces e produtos. Com esse problema evidentes, passou-se a idealizar e empregar novas funcionalidades ao sistema, o qual passou a dispor de dois módulos, sendo eles descritos a seguir:

- Primeiro módulo, ainda em produção, haverá um checklist, com perguntas baseadas nos critérios de Bastien e Scapin;
- Segundo módulo, os interessados em utilizar a ferramenta, geralmente desenvolvedores, poderão criar grupos de avaliação, e suas próprias heurísticas, para submeter sistemas particulares a avaliação de pessoas comuns.

6 CONCLUSÃO

Ao decorrer do desenvolvimento do projeto, os autores foram se aprofundando, cada vez mais, no tema usabilidade, e nesta busca por conhecimento, algumas de suas concepções foram divergindo, e novos entendimentos foram sendo moldados. Como experiência, os elaboradores da pesquisa podem afirmar que o desenvolvimento de uma lista de verificação, baseada em critérios ergonômicos, depende de muito estudo, e como desenvolvedores, os autores admitem a necessidade de uma característica intrínseca, de abstração elevada. Talvez esse seja o motivo, de existirem poucas ferramentas, Open Source, estáveis, com estas características.

Apesar dos obstáculos encontrados, ao decorrer da caminhada, estes fatores não resultaram na desistência da pesquisa, visto que os autores já haviam obtido um vasto conteúdo, de grande importância na área de usabilidade, material este, provido em sua maioria, por cientistas conhecidos, da área de IHC, ou seja, o encontro destes documentos, tem sua importância, e pode auxiliar outros desenvolvimentos no campo de usabilidade. Outro fator importante, que manteve os autores crentes, da relevância deste trabalho, foi que, apesar de não terem conseguido, concluir o protótipo de maneira integral, com a lista de verificação, atendendo detalhadamente os critérios de Bastien e Scapin, garantiram a entrega parcial. Como contrapartida, foi adicionado ao projeto um novo módulo, o qual permite os desenvolvedores, alimentarem uma base do sistema com recomendações personalizadas, fator que amplia o sistema a um patamar mais abrangente.

Concluimos que apesar de evidente a necessidade de sistemas que respeitem os critérios de ergonomia, ainda não há um movimento comunitário, a favor do avanço e divulgação destas técnicas. Sendo assim é proposto pelos autores, para trabalhos futuros, a complementação deste sistema, através da concretização das seguintes etapas:

1. Aumento da base de recomendações ergonômicas e perguntas para o checklist padrão do sistema, a fim de abranger uma maior quantidade de possibilidades de avaliações, para diferentes interfaces e sistemas em geral.
2. Inclusão de funcionalidades que facilitem o uso do sistema tanto para os administradores, como um painel de controle com mais opções de gerenciamento do sistema e de seus usuários, como para os desenvolvedores e usuários comuns, proporcionando novas alternativas para organização de suas recomendações ergonômicas, perguntas, relatórios salvos no sistema.

3. Deixar as interfaces do sistema dentro dos padrões de recomendações ergonômicas.
4. Confeção de um módulo responsável por gerenciar e oferecer testes relacionados à acessibilidade de sistemas.

REFERÊNCIAS

- BARBOSA, S.; SILVA, B. S. **Interação Humano-Computador**. Rio de Janeiro: Elsevier, 2010. p. 384.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: Guia do Usuário**. 2. ed. Rio de Janeiro: Campus Elsevier, 2012. p. 500.
- CARNIEL, A; CYBIS, W. **Utilizabilidade - Critérios Ergonômicos**. Disponível em <[Http://www.labiutil.inf.ufsc.br/CriteriosErgonomicos/Abertura.html](http://www.labiutil.inf.ufsc.br/CriteriosErgonomicos/Abertura.html)>. Acesso em: 22 fev. 2016
- CYBIS, W.; BETIOL, A.; FAUST, R. **Ergonomia e Usabilidade: Conhecimento, Métodos e Aplicações**. 2. ed. São Paulo: Novatec, 2010. p. 422.
- GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. Porto Alegre: Editora UFRGS, 2009. p. 120. Disponível em <<http://www.ufrgs.br/cursopgdr/downloadsSerie/derad005.pdf>>. Acesso em: 22 fev. 2016.
- HORSTMANN, C. *Big Java*. Porto Alegre: Bookman, 2004. p. 1125.
- ITAKURA, F. **TOWABE: Uma Ferramenta para Avaliação de Usabilidade de Aplicações para WEB**. Curitiba: 2001, p. 104. Disponível em <<http://dspace.c3sl.ufpr.br:8080/dspace/bitstream/handle/1884/24734/D%20-%20ITAKURA,%20FERNANDO%20TAKASHI.pdf?sequence=1>>. Acesso em: 20 set. 2015.
- ISO. **9241-1: Ergonomic requirements for office work with visual display terminals (VDTs) Part 1: General introduction**. ISO, 1997.
- ISO. **9241-110: Ergonomics of human-system interaction - Part 110: Dialogue principles**. ISO, 2006.
- SPURLOCK, J. *Bootstrap: Responsive Web Development*. New York: O'Reilly, 2013, p. 128.
- KAMMERGAARD, J. *Four Different Perspectives on Human-Computer Interaction*. Aarhus – Dinamarca: Matematik Institut - Aarhus University, 1985, p. 36. Disponível em <ojs.statsbiblioteket.dk/index.php/daimipb/article/download/7554/6401>. Acesso em: 10 set. 2015.
- LABORATÓRIO DE UTILIZABILIDADE DA INFORMÁTICA. **Ergolist**. Disponível em <<http://www.labiutil.inf.ufsc.br/ergolist/>>. Acesso em: 24 out. 2015
- LESSA, R. O. **Glist – Um Checklist Automatizado para Usabilidade**. Palhoça: Trabalho de Conclusão de Curso da Universidade do Sul de Santa Catarina, 2006, p. 103. Disponível em <http://busca.unisul.br/pdf/94741_Rafael.pdf>. Acesso em: 15 ago. 2015
- MASCULO, F.; VIDAL, M. C. **Ergonomia: Trabalho Adequado e Eficiente**. Rio de Janeiro: Elsevier, 2011. p. 606.

NIELSEN, J. *Usability Engineering*. Cambridge - Massachusetts: Academic Press, 1994. p. 264.

ROCHA, H.; BARANAUSKAS, M. **Design e Avaliação de Interfaces Humano-Computador**. Campinas - São Paulo: Instituto de Computação da Universidade Estadual de Campinas, 2003. p. 241.

SBROCCO, J.H.T.C. **UML 2.5 Com Enterprise Architect 10 - Modelagem Visual de Projeto Orientada A Objetos**. Editora Érica, 2014. p. 320.

SHNEIDERMAN, B.; PLAISANT, C. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 4. ed. Reading - Massachusetts: Addison-Wesley, 2010. p. 606.

THE LABORATORY FOR AUTOMATION PSYCHOLOGY AND DECISION PROCESSES.

QUIS - Questionnaire for user interaction satisfaction. Disponível em:

<<http://www.lap.umd.edu/quis/>>. Acesso em: 24 out. 2015

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL Documentation**.

2007. p. 1310. Disponível em:

<<https://sourceforge.net/projects/pgdocptbr/files/latest/download>>. Acesso em: 12 mai. 2016.

APÊNDICE A – Cronograma de atividades do TCC

O presente cronograma destina-se ao detalhamento das atividades da pesquisa, e sua distribuição no tempo, fato que permite uma análise e controle geral sobre execução do trabalho, garantindo um planejamento otimizado e possibilitando a obtenção dos resultados pretendidos.

Quadro 1 – Cronograma de atividades TCC 1

Nome da Tarefa	ago/15	set/15	out/15	nov/15	dez/15
Escolha do tema e orientador					
Encontros com o orientador - TCC 1					
Elaboração do primeiro capítulo					
Revisão e entrega do primeiro capítulo					
Pesquisa bibliográfica preliminar					
Leituras e elaboração de resumos					
Revisão e entrega do segundo capítulo					
Definição da metodologia					
Revisão e entrega do terceiro capítulo					
Revisão bibliográfica complementar					

Fonte: Produzido pelos próprios autores.

Quadro 2 – Cronograma de atividades TCC 2

Nome da Tarefa	jan/16	fev/16	mar/16	abr/16	mai/16	jun/16
Encontros com o orientador - TCC 2						
Revisão e entrega correções do Avaliador - TCC 1						
Estudo das ferramentas de desenvolvimento						
Modelagem Banco de dados e Sistema						
Revisão e entrega do quarto capítulo						
Concepção do layout						
Desenvolvimento da solução proposta						
Testes do sistema						
Correção de possíveis erros no sistema						
Redação do trabalho						
Revisão e entrega oficial do trabalho						
Apresentação do trabalho em banca						

Fonte: Produzido pelos próprios autores.

Quadro 3 – Cronograma TCC

Nome da Tarefa	Duração	Início	Término
Escolha do tema e orientador	11 dias	Qui 06/08/15	Qui 20/08/15
Encontros com o orientador - TCC 1	75 dias	Qua 12/08/15	Ter 24/11/15
Elaboração do primeiro capítulo	18 dias	Ter 25/08/15	Qui 17/09/15
Revisão e entrega do primeiro capítulo	18 dias	Ter 25/08/15	Qui 17/09/15
Pesquisa bibliográfica preliminar	46 dias	Qui 20/08/15	Qui 22/10/15
Leituras e elaboração de resumos	46 dias	Qui 20/08/15	Qui 22/10/15
Revisão e entrega do segundo capítulo	46 dias	Qui 20/08/15	Qui 22/10/15
Definição da metodologia	11 dias	Qui 22/10/15	Qui 05/11/15
Revisão e entrega do terceiro capítulo	11 dias	Qui 22/10/15	Qui 05/11/15
Revisão bibliográfica complementar	26 dias	Ter 20/10/15	Ter 24/11/15
Encontros com o orientador TCC 2	53 dias	Ter 15/03/16	Qui 26/05/16
Revisão e entrega correções do Avaliador - TCC 1	31 dias	Qui 18/02/16	Qui 31/03/16
Estudo das ferramentas de desenvolvimento	26 dias	Qui 10/03/16	Qui 14/04/16
Modelagem Banco de dados e Sistema	16 dias	Qui 24/03/16	Qui 14/04/16
Revisão e entrega do quarto capítulo	16 dias	Qui 24/03/16	Qui 14/04/16
Concepção do layout	6 dias	Qui 14/04/16	Qui 21/04/16
Desenvolvimento da solução proposta	16 dias	Qui 21/04/16	Qui 12/05/16
Testes do sistema	6 dias	Qui 12/05/16	Qui 19/05/16
Correção de possíveis erros no sistema	11 dias	Qui 19/05/16	Qui 02/06/16
Redação do trabalho	41 dias	Qui 24/03/16	Qui 19/05/16
Revisão e entrega oficial do trabalho	6 dias	Qui 19/05/16	Qui 26/05/16
Apresentação do trabalho em banca	4 dias	Seg 13/06/16	Qui 16/06/16

Fonte: Produzido pelos próprios autores (2016).