



UNIVERSIDADE FEDERAL DE PELOTAS
INSTITUTO DE FÍSICA E MATEMÁTICA
DEPARTAMENTO DE INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**UTILIZANDO MÉTODOS NÃO FORMAIS E FORMAIS NA AVALIAÇÃO E
MODELAGEM DE INTERFACES COM O USUÁRIO**

VILMAR PETER

Pelotas, 2008

VILMAR PETER

**UTILIZANDO MÉTODOS NÃO FORMAIS E FORMAIS NA AVALIAÇÃO E
MODELAGEM DE INTERFACES COM O USUÁRIO**

Monografia apresentada ao Curso de
Ciência da Computação do Instituto de Física e
Matemática da Universidade Federal de Pelotas,
como requisito parcial à obtenção do título de
Bacharel em Ciência da Computação.

Orientadora: Prof^a. Eliane da Silva
Alcoforado Diniz, M.Sc.

Pelotas - 2008

Dados de catalogação na fonte:
Maria Beatriz Vaghetti Vieira – CRB-10/1032
Biblioteca de Ciência & Tecnologia - UFPel

P478u Peter, Vilmar

Utilizando métodos não formais e formais na avaliação e modelagem de interfaces com o usuário / Vilmar Peter ; orientador Eliane da Silva Alcoforado Diniz. – Pelotas, 2008. – 89f. - Monografia (Conclusão de curso). Curso de Bacharelado em Ciência da Computação. Departamento de Informática. Instituto de Física e Matemática. Universidade Federal de Pelotas. Pelotas, 2008.

1.Informática. 2.Métodos não formais. 3. StateChart. 4. Métodos formais. 5. Avaliação. 6. Modelagem. 7. Interface com o usuário I.Diniz, Eliane da Silva Alcoforado. II.Título.

CDD: 004.6

Agradecimentos

Primeiramente gostaria de agradecer à minha família e esposa, que me apoiaram, incentivaram e deram todo o suporte durante minha trajetória nesta conquista. Sem eles, eu jamais teria chegado até aqui.

Agradeço também a minha mais que Orientadora, e sim amiga Eliane da Silva Alcoforado Diniz, que esteve sempre disposta a me ajudar, e por me proporcionar crescimento tanto pessoal quanto profissional.

Agradeço também a todos os meus colegas e professores, pela contribuição que de alguma forma me deixaram, e que ao longo destes cinco anos e meio de curso trouxeram muitos momentos de diversão, os quais com certeza deixarão saudades.

A todos vocês meu muito obrigado.

RESUMO

PETER, VILMAR. **Utilizando Métodos Não Formais E Formais Na Avaliação E Modelagem De Interfaces Com O Usuário**. 2008. 89f. Trabalho acadêmico – Curso de Bacharelado em Ciência da Computação. Universidade Federal de Pelotas, Pelotas.

Um dos grandes problemas que se observa no desenvolvimento da interface com o usuário - IU é o pouco uso de técnicas apropriadas para avaliação. Com isso, tem-se muitas vezes, interfaces de difícil aprendizagem e de pouca legibilidade, fazendo com que o usuário tenha uma perda de tempo considerável; tanto no aprendizado do mesmo, quanto na utilização posterior do software. Decorrente desse fato, existe a necessidade de se realizar uma avaliação que possibilite encontrar os problemas e, subseqüentemente, fazer as devidas alterações visando a melhoria da interface e uma adequação da mesma às características do seu usuário e das tarefas a serem realizadas. Atualmente, os pesquisadores têm desenvolvido uma série de métodos para essa finalidade. Esses métodos estão classificados em dois grandes grupos: Métodos Formais e Métodos Não Formais. Nesse trabalho procurou-se utilizar ambas as classificações, uma vez que foram utilizadas as técnicas de questionário e observação, pertencentes aos Métodos Não Formais de Inspeção, para junto aos usuários coletar informações sobre os problemas que vem diminuindo o seu desempenho. Após essa coleta, empregou-se a Técnica Formal Orientada a Modelo (StateChart), visando realizar alterações na interface proposta. O StateChart é um formalismo utilizado, geralmente, para especificar sistemas complexos com concisão e em diferentes níveis de abstração, tendo a possibilidade de especificar um sistema completo mais rápido, através de suas representações hierárquicas. O uso do formalismo para a nova versão da Interface com o Usuário se justifica pelas características inerentes a esse tipo de método, as quais permitem que a funcionalidade e o modelo de interação possam ser avaliados por desenvolvedores e usuários, antes mesmo de sua implementação. A utilização de formalismo na especificação do projeto de IU nos traz algumas vantagens, tanto na concisão, como na coerência e na não ambigüidade das tarefas previstas. As características observadas nos Métodos Formais e as vantagens apresentadas com a utilização dos mesmos vêm trazendo resultados satisfatórios na fase de modelagem da nova IU, cujos resultados já indicam uma melhora significativa durante o processo iterativo.

Palavras- chaves: Métodos Não Formais, Métodos Formais, Avaliação, Modelagem, Interface com o Usuário, StateChart.

ABSTRACT

PETER, VILMAR. **Utilizando Métodos Não Formais E Formais Na Avaliação E Modelagem De Interfaces Com O Usuário**. 2008. 89f. Trabalho acadêmico – Curso de Bacharelado em Ciência da Computação. Universidade Federal de Pelotas, Pelotas.

One of the great problems observed in the development of the user interface - UI is the lack of use of appropriate techniques for evaluation. With that, interfaces of difficult learning and little legibility are made, making the user have a considerable waste of time; both in learning as in the subsequent use of the software. Considering that fact, there is a need to accomplish an evaluation to find the problems and, subsequently, to make the due alterations seeking the improvement and adaptation of the interface to the user's characteristics and of his tasks to be accomplished. Nowadays, researchers have been developing methods for that purpose. Those methods are classified in two groups: Methods don't Formal and Formal. This work tries to use both classifications, using the questioning and observation techniques, both belonging to the don't Formal Methods of Inspection, collecting information closely with the users about the problems that reduce their performance. After the information collection, the Model-Oriented Formal Technique (StateChart) was applied, seeking to accomplish changes in the proposed interface. StateChart is a formalism usually employed to specify complex systems with conciseness and in different abstraction levels, with the possibility to specify faster a complete system, through their hierarchical representations. The use of formalism in the new version of the User interface is justified for the inherent characteristics to this method type, which allow the functionality and the interaction model to be evaluated by developers and users, even before its implementation. The use of formalism in the specification of the UI project brings us some advantages, both in the conciseness as in the coherence and unambiguous foreseen tasks. The observed characteristics in the Formal Methods and the presented advantages with their use are bringing satisfactory results in the modeling phase of new UI, whose results already show a significant improvement during the interactive process.

Keywords: Methods don't formal, Formal Methods, Evaluation, Modelling, User Interface, StateChart.

LISTA DE FIGURAS

Figura 1: O processo de interação humano-computador.....	14
Figura 2 – Elementos Básicos de um StateChart	32
Figura 3 – Exemplo de Diálogo de <i>Login</i>	33
Figura 4 – Caixa Principal – Menu de opções	39
Figura 5 - Caixa do Dia – Tela de operações principais do Caixa	39
Figura 6 - Cadastro Clientes - Dados Pessoais	40
Figura 7 - Cadastro Clientes - Dados Profissionais	40
Figura 8 - Cadastro Clientes - Cônjuge e referências	41
Figura 9 - Cadastro da Dioptria (grau) do Cliente	41
Figura 10 - Venda ao Cliente – Formas de Pagamento	42
Figura 11 - Entrada de Dinheiro	42
Figura 12 - Recebimento Carnê	43
Figura 13 - Retirada de Dinheiro	43
Figura 14 - Modelagem StateChart correspondente as janelas apresentadas nas Fig. 4 e 5.	44
Figura 15 - Modelagem StateChart sobre as avaliações da Fig. 5.	45
Figura 16 - Modelagem StateChart sobre as avaliações das Fig. 7,8 e 9.	46
Figura 17 - Modelagem StateChart sobre as avaliações da Fig. 9.	47
Figura 18 - Modelagem StateChart sobre as avaliações da Fig. 10.	48
Figura 19 - Modelagem StateChart sobre as avaliações da Fig. 11	49
Figura 20 - Modelagem StateChart sobre as avaliações da Fig. 12	49
Figura 21 - Modelagem StateChart sobre as avaliações da Fig. 13.	50
Figura 22 - Nova IU do Caixa Principal - Menu de Opções	51
Figura 23 - Caixa do dia – Tela de operações principais do Caixa	52
Figura 24 - Caixa do dia – Tela de operações principais do Caixa	52

Figura 25 - Cadastro de Clientes – Dados Pessoais	53
Figura 26 - Cadastro de Clientes – Cônjuge	54
Figura 27 - Cadastro de Clientes – Endereço	54
Figura 28 - Cadastro de Clientes – Dados Profissionais	55
Figura 29 - Cadastro de Clientes – Referências	55
Figura 30 - Cadastro da Dioptria (grau) de Clientes	56
Figura 31 - Venda ao Cliente - Forma de pagamento	57
Figura 32 - Entrada de Dinheiro	58
Figura 33 - Recebimento Carnê	59
Figura 34 - Retirada de Dinheiro	60
Figura 35 - Janelas de confirmação e Ajuda	61
Figura 36 - Janelas de confirmação	61
Figura 37 - Janela de Login do Caixa Dia e Janelas Informativas	62

LISTA DE TABELAS

Tabela 1 - Tempo gasto para utilização das Interfaces.	65
Tabela 2- Tempo gasto para utilização das Interfaces.	65
Tabela 3- Tempo gasto para utilização das Interfaces.	66
Tabela 4- Tempo gasto para utilização das Interfaces.	66

LISTA DE ABREVIATURAS E SIGLAS

IU - interface com o usuário

IHC - Interação Humano- Computador

SC - StateChart

SUMÁRIO

1	Introdução	11
1.1	Motivação	11
1.2	Objetivos	12
1.3	Organização do trabalho	12
2	Interação e Usabilidade	14
2.1	Interação	14
2.2	Objetivos e Importância da Avaliação de Interação	15
2.3	Usabilidade	16
3.	Métodos de Construção e Avaliação de Interface	18
3.1	Métodos não Formais	18
3.1.1	Design Participativo	19
3.1.2	Análise das Tarefas	19
3.1.3	Diretrizes de Projeto	19
3.1.4	Prototipagem Rápida.....	20
3.2	Métodos Formais	20
3.2.1	Métodos Formais e IHC	22
3.2.2	Abordagem Orientada a Propriedade	23
3.2.3	Abordagem Orientada a Modelo	24
3.3	A Escolha de uma Abordagem	25
4.	Avaliação Heurística	26
4.1	Procedimentos da Avaliação Heurística	26
4.1.1	Avaliação Individual	26
4.1.2	Consolidação da avaliação dos usuários	27
4.1.3	Seleção dos problemas que devem ser corrigidos	27
4.1.4	Heurísticas	27

5. Formalismo StateChart	31
5.1 Elementos Básicos de StateChart	31
6. Estudo de Caso	35
6.1 Resultados Obtidos com a avaliação Heurística sobre o sistema ABBA.	35
6.1.1 Figuras demonstrando problemas da interface atual	39
6.2 Modelando a nova interface com o StateChart	43
6.3 Novas Interfaces projetadas decorrentes da aplicação do StateChart	50
6.4 Resultados Obtidos com a nova avaliação sobre a interface modificada	62
6.5 Análise de tempo em virtude da utilização da antiga (ABBA) e nova IU.	64
6.6 Tabelas de comparação de tempos entre a antiga (ABBA) e nova IU.	64
Tabela 1 - Tempo gasto para utilização das Interfaces.	65
Usuário 1, experiência com a Interface antiga: 8 meses	65
6.7 Considerações finais sobre os tempos	67
7. Conclusão	68
7.1 Trabalhos Futuros	69
Apêndice	74

1 Introdução

Com o decorrer dos anos a popularização dos computadores cresceu muito, tanto na área pessoal, como na comercial. Desse modo, cada vez mais esses aparelhos são incorporados ao cotidiano das pessoas, sendo utilizados não só como ferramentas de trabalho, mas também como meio de comunicação, de inserção social e de planejamento familiar. Nesta infinidade de sistemas deve-se priorizar a qualidade da interface com o usuário desse sistema, já que este ponto é fundamental na hora em que o usuário vai decidir em usar um determinado sistema ou não (HIX, 1993). Decorrente desse fato, as preocupações dos projetistas de sistemas aumentaram consideravelmente, já que além dos aspectos funcionais, os aspectos operacionais passaram a exigir esforços adicionais. Dispensando-se considerável quantidade de tempo e esforço na elaboração e construção da interface com o usuário - IU.

Porém, o emprego correto de métodos de construção de IU não garante, por si só, que se obtenha um resultado satisfatório. As IU freqüentemente nem sempre são boas como poderiam ser: na maioria das vezes elas são construídas tardiamente, tornando-se difíceis de entender, levando o usuário a cometer erros desnecessários (THIMBLEBY, 1991). Desse modo, faz-se necessário o emprego de uma avaliação que tenha como objetivo checar se o produto final está em conformidade com as necessidades e expectativas dos seus usuários (DINIZ, 1996).

1.1 Motivação

A melhoria da qualidade de um produto de software é a meta principal no seu processo de desenvolvimento. Todavia, apesar de aplicações criadas a partir de princípios e processos ergonômicos surgiram em número cada vez maior, inovando o mercado de consumo com recursos muito mais eficientes do que as aplicações convencionais (SHNEIDERMAN, 1998), esses recursos constituem apenas suportes físicos para os projetos de ambientes interativos. Pois, os sistemas computacionais

com estas características não apresentam, por si só, uma solução para a especificação e a implementação de interfaces eficientes (DINIZ, 1996).

Esse problema começou a ser mais visível principalmente quando os “usuários domésticos” começaram a utilizar sistemas diariamente, como por exemplo a internet que se popularizou muito. Com isso, observou-se que muitos usuários tinham problemas para realizar tarefas simples, já que os programas não apresentavam uma boa interface com usuário - IU, pois essa área é pouco explorada e não se tem um tempo destinado a cuidados especiais ao seu desenvolvimento e a avaliação. Vale salientar que a criação de uma IU que permita uma fácil interação do usuário com o sistema não é uma tarefa fácil. A sua construção implica em um aumento da complexidade no gerenciamento da comunicação do usuário com a informação (WINOGRAD, 1995). Daí, a necessidade de se utilizar métodos que auxiliem os projetistas, visando permitir um melhor gerenciamento do processo de criação após a fase de avaliação. Nas últimas décadas, tem sido desenvolvida uma variedade de métodos com essa finalidade, os quais diferem não somente nas suas estruturas sintáticas, como também na quantidade de suporte que oferecem para diferentes aspectos de desenvolvimento (JOHNSON, 1988). As características observadas nesses métodos e as vantagens apresentadas com a utilização dos mesmos que são citados na literatura (BRANDÃO, 2005) foram um dos principais agentes motivadores a realização desse trabalho.

1.2 Objetivos

O propósito deste trabalho é realizar uma avaliação de IU. Procurando identificar as necessidades dos usuários, que podem ser: verificar o entendimento dos projetistas sobre estas necessidades, identificar problemas de interação ou de interface, investigar como uma interface afeta a forma de trabalhar dos usuários e conferir a inteligibilidade, e a operacionalidade entre usuário e computador. A partir dessas informações, comparar alternativas de projeto de interface, visando construir uma nova IU, de modo a alcançar objetivos quantificáveis em métricas de usabilidade. Fornecendo também ao projetista uma interface de fácil manutenção posterior, reduzindo assim tempo e custos.

1.3 Organização do trabalho

Para entendimento do trabalho, o texto será apresentado da seguinte maneira:

O segundo capítulo faz um estudo sobre a interação e usabilidade; quais os objetivos e a importância da avaliação, entre o usuário e o sistema

O terceiro capítulo traz os tipos de métodos e seus principais aspectos tanto para avaliações como para modelagens de Interfaces.

No quarto capítulo são demonstrados quais os procedimentos a serem realizados através da utilização da avaliação Heurística.

No quinto faz-se uma abordagem ao Formalismo StateChart e os elementos que o constituem.

No sexto capítulo são expostos os resultados obtidos com a avaliação e a modelagem sobre uma interface específica, e posteriormente uma comparação entre a antiga interface e a nova.

Por fim, o capítulo sete apresenta as conclusões e trabalhos futuros.

2 Interação e Usabilidade

Neste capítulo será feita uma breve apresentação sobre os temas, interação e Usabilidade. O conhecimento desses temas é essencial para auxiliar no desenvolvimento de IU, evidenciando os pontos fundamentais para se ter uma melhor Interação Humano- Computador- IHC.

2.1 Interação

Interação é o processo de comunicação entre pessoas e sistemas interativos (PREECE, 1994). Neste processo, usuário e sistema trocam informações onde um espera e o outro envia algum dado, interpreta-se e realiza-se uma ação e assim vice versa constantemente. A área de IHC estuda este processo, principalmente do ponto de vista do usuário: quais as ações que ele realiza usando a IU, e suas interpretações das respostas transmitidas pelo sistema através da mesma (Fig. 1).

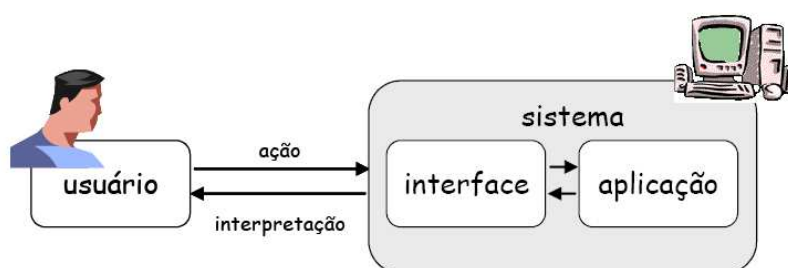


Figura 1: O processo de interação humano-computador

Fonte: PRATES, 2002.

O processo de interação se dá através da Interface, sendo essa a porção de um sistema com a qual um usuário mantém contato ao utilizá-lo, tanto ativa quanto passivamente. A interface engloba tanto software quanto hardware (dispositivos de entrada e saída, tais como: teclados, mouse, tablets, monitores, impressoras e etc.). Considerando a interação como um processo de comunicação, a interface pode ser

vista como o sistema de comunicação utilizado neste processo. Uma definição de interface utilizada com frequência foi proposta por Moran (1981): “a interface de usuário deve ser entendida como sendo a parte de um sistema computacional com a qual uma pessoa entra em contato — física, perceptiva ou conceitualmente”. Entende-se por dimensão física todos os elementos de interface que podem ser manipulados pelo usuário, enquanto que a dimensão perceptiva engloba aqueles que o usuário pode perceber. Por sua vez, os processos de interpretação e raciocínio do usuário são resultados da dimensão conceitual, desencadeados pela sua interação com o sistema, com base em suas características físicas e cognitivas, seus objetivos e seu ambiente de trabalho.

Atualmente, para melhorar o processo interativo, as interfaces procuram envolver elementos visuais e sonoros, com entrada de dados via teclado e mouse. Outros tipos de interfaces, como interface via voz e entrada de dados através de canetas, devido à disseminação de dispositivos móveis, estão se tornando freqüentes nesse processo.

2.2 Objetivos e Importância da Avaliação de Interação

Antes de declarar um software pronto para uso, é importante saber se ele apóia adequadamente os usuários nas suas tarefas e no ambiente em que será utilizado. Assim como testes de funcionalidade são necessários para se verificar a robustez da implementação, a avaliação da IU é necessária para se analisar a qualidade de uso de um software. Quanto mais cedo detectados os problemas de interação ou de IU, menor o custo de se consertá-los (KARAT, 1993).

Um projetista não deve supor que basta seguir métodos e princípios de projeto de interfaces para garantir uma alta qualidade de uso de seu software. Além disto, também não deve presumir que os usuários são como ele próprio, portanto bastaria sua avaliação individual para atestar essa qualidade (PREECE et al., 2002). Alguns dos principais objetivos de se realizar avaliação de sistemas interativos são (HARTSON, 1998; PREECE et al., 2002):

- Identificar as necessidades de usuários ou verificar o entendimento dos projetistas sobre estas necessidades.
- Identificar problemas de interação ou de interface.
- Investigar como uma interface afeta a forma de trabalhar dos usuários.

- Comparar alternativas de projeto de interface.
- Alcançar objetivos quantificáveis em métricas de usabilidade.
- Verificar conformidade com um padrão ou conjunto de heurísticas.

Infelizmente, é freqüente encontrar gerentes de projeto que pensam apenas nos custos envolvidos na realização de avaliações de seus sistemas. Isto se deve geralmente pela pouca informação que eles têm sobre os benefícios que existem com a utilização dos processos. Dependendo do momento em que for realizada a avaliação, estes benefícios podem ter efeitos imediatamente. Caso tenha-se uma interface de pouca interação, isso logo acarretará em um rendimento abaixo do esperado dos usuários e assim numa diminuição na produção total, gerando talvez um custo de maior valor do que executar uma avaliação no sistema.

2.3 Usabilidade

O conceito de usabilidade permite avaliar a qualidade de um sistema com relação a fatores que podem influenciar na aceitação do mesmo. Em outras palavras, a meta que se pretende atingir quando se constrói uma IU é a facilidade de aquisição da competência necessária pelo usuário, para interagir com o sistema. Segundo Moran (1981) e Norman (1986), “a competência do usuário é o conhecimento a respeito do sistema necessário para ele desempenhar a interação com o sistema é denominado de modelo conceitual do usuário”. Alguns fatores típicos envolvidos no conceito de usabilidade são (NIELSEN, 1993; PREECE, 2002):

- Facilidade de aprendizado - se refere ao tempo e esforço necessários para que os usuários aprendam a utilizar uma determinada parte do sistema com determinado nível de desempenho.
- Facilidade de uso - está relacionado não apenas com o esforço para interagir com o sistema, mas também com o número de erros cometidos durante esta interação. É importante observar que um IU fácil de aprender não é necessariamente fácil de utilizar.
- Eficiência de uso e produtividade - Sistemas fáceis de utilizar podem ser ineficientes de duas formas: com relação ao que permite o usuário fazer (eficiência de uso), e a como o usuário deve fazê-lo (produtividade). O fator eficiência de uso serve para analisar se o sistema faz bem aquilo que se propõe. Já o fator produtividade serve para avaliar se o usuário

consegue fazer o que precisa de forma rápida e eficaz. Este fator é geralmente avaliado pelo tempo decorrido desde o início até a conclusão de uma tarefa e pelo número de passos que o usuário precisou realizar.

- Satisfação do usuário - enfatiza a avaliação subjetiva do sistema feita por seus usuários, incluindo emoções que possam surgir durante a interação, sejam elas positivas ou negativas, como tédio ou frustração. Pessoas diferentes podem seguir caminhos distintos para atingir um mesmo objetivo.
- Utilidade – refere-se ao conjunto de funcionalidades necessárias para os usuários realizarem suas tarefas. Esta dimensão está intimamente relacionada ao conceito de aplicabilidade proposto por Fischer (1998).
- Segurança no uso – refere-se ao grau de proteção de um sistema contra condições desfavoráveis ou até mesmo perigosas para os usuários. Trata-se principalmente de como permitir que usuário se recupere de condições de erro, que poderão ter conseqüências sérias para seu trabalho ou para sua saúde.

Esses fatores se forem observados durante a fase de construção, poderão minimizar as possíveis falhas e se observados na avaliação, poderão servir de base para checar, se a IU, está em conformidade com as expectativas de seus usuários. Haja vista, que a não observância desses fatores poderá vir a torná-lo um sistema inaceitável por seus usuários, uma vez que a falta de uma boa interação entre o usuário e o sistema pode gerar a insatisfação do mesmo e, conseqüentemente, fazer com que esse empregue mais tempo na execução das suas tarefas, ocasionando desse modo uma perda considerável na produtividade.

Preocupados com os elementos vistos acima, a comunidade de IHC tem procurado encontrar maneiras de auxiliar o projetista/ avaliador na fase de construção e avaliação. Esses mecanismos aparecem na forma de métodos, os quais, bem aplicados poderão mostrar os problemas de usabilidade que ocorrem ou que poderão ocorrer durante a interação do usuário com a IU.

3. Métodos de Construção e Avaliação de Interface

Existe uma considerável quantidade de métodos disponíveis para a construção e avaliação de IU. Nesse capítulo serão apresentados os principais métodos que possibilitam a construção/avaliação de interfaces. A aplicação correta desses métodos poderá evitar que a construção ocorra sem levar em conta a fase de especificação, trazendo a tona os vícios de programação. Por outro lado, quando aplicados na fase de avaliação, poderão auxiliar os avaliadores a encontrarem os problemas que afetam a qualidade do produto final. Esses métodos estão incluídos em duas principais categorias: Formais e não Formais.

3.1 Métodos não Formais

Os métodos não formais são largamente utilizados por programadores e designers de interfaces ao longo dos anos. Tais métodos estão baseados na visão que os projetistas têm em relação ao tipo de interação que o usuário final irá ter com a IU, tomando como base a experiência de cada usuário, o histórico de interações, o estado de satisfação, entre outros. Também são observadas, as atitudes que os usuários têm durante o processo de avaliação, as quais servirão de base para os projetistas reconstruírem as referidas IU.

Dentre os métodos pertencentes a esse grupo, existem alguns que durante o processo de avaliação de usabilidade, por exemplo: as heurísticas de Nielsen, não utilizam os usuários nesse processo. Essa avaliação é realizada por um grupo de peritos em projeto de telas, que examinam uma determinada interface com o objetivo de encontrar problemas que não estejam de conformidade com os princípios gerais do projeto.

A seguir, serão descritos alguns dos métodos não formais disponíveis na literatura (PRESSMAN, 1995) (SHNEIDERMAN, 1998) (NIELSEN, 1994) sendo os mais utilizados para a construção e avaliação de IU, pela comunidade de IHC.

Design Participativo, Análise de Tarefa, Diretrizes de Projeto, Prototipagem Rápida e avaliação Heurística esse último sendo abordado no capítulo 4.

3.1.1 Design Participativo

Consiste em incorporar um ou mais usuários à equipe de projeto. Desse modo, esse método produz uma interação projetista-usuário. O emprego desse método permite que a interação ocorra durante todas as fases do projeto onde a funcionalidade é determinada, fazendo com que o projetista conheça o usuário, para o qual ele está projetando. É importante salientar que poderão ocorrer alguns problemas (embora salutares) a inclusão de usuários representativos na equipe de projeto. Porque ao tornarem-se muito próximos da equipe de projeto, os usuários poderão sentir-se inibidos, de modo que, questionamentos importantes ao bom andamento do projeto, que possam ser interpretados como crítica ou censura à equipe, não serão feitos, o que poderá acarretar danos no produto final.

3.1.2 Análise das Tarefas

Durante a análise de tarefas o projetista procura identificar a maneira na qual os usuários realizam suas tarefas, e a partir dessa análise desenvolver uma IU, que possibilite ao usuário realizar suas tarefas da maneira mais agradável possível. Para que isso ocorra, deve-se procurar avaliar o local onde o usuário utiliza a interface, e o seu comportamento durante a execução de suas tarefas no decorrer do dia, tendo assim subsídios suficientes para serem incorporados junto ao projeto. A aplicação desse método vem auxiliando consideravelmente no desenvolvimento das IU, obtendo-se aplicações interativas mais orientadas aos seus usuários.

3.1.3 Diretrizes de Projeto

As diretrizes são recomendações específicas e detalhadas para o tratamento de elementos de interface baseados no estudo cognitivo e ergonômico do usuário e da tarefa (PRESSMAN, 1995) (SHNEIDERMAN, 1998). Certos números de diretrizes estão fundamentados sobre as conclusões empíricas, demonstrando uma visão da comunidade de IHC. As várias diretrizes existentes hoje em dia são capazes de premeditar qualquer eventualidade que possa ocorrer em uma determinada interação, devido à facilidade de sua aplicação. As diretrizes têm como objetivo principal ajudar no desenvolvimento de uma IU, inserindo informações, as quais deixem as interfaces com uma melhor disposição, visando sempre os aspectos do

usuário. Entretanto, não se tem nenhuma linha de estudo com base consistente realizada sobre as diretrizes que possa servir para esclarecer ou prever o sucesso ou o fracasso de um projeto (SHNEIDERMAN, 1998).

3.1.4 Prototipagem Rápida

Tem como objetivo demonstrar a realidade, para isso é utilizado a própria IU ou desenvolvido um protótipo dessa, fazendo-se assim, uma avaliação dos problemas e indicando soluções cabíveis para a mesma (PRESSMAN, 1995; SHNEIDERMAN, 1998). Com a utilização de um protótipo, os projetistas o avaliam, economizando assim, tempo e custos, pois produzem um feedback imediato, tendo informações suficientes para a aplicação ou não do protótipo com o usuário real (NIELSEN, 1994). Tem-se ainda um ganho na avaliação de uma IU utilizando protótipos, já que, com os mesmos têm-se condições de realizar os testes antecipadamente, não precisando esperar pelo produto final.

3.2 Métodos Formais

As técnicas formais têm sido aplicadas de muitas formas no desenvolvimento de software. Elas permitem ao projetista descrever o comportamento externo de um sistema sem especificar a implementação interna (DIX, 1995). Apesar disso, não é comum a aplicação dessas técnicas para o desenvolvimento de uma IU. O pouco uso dessas técnicas vem do não conhecimento da utilização das especificações formais, e ainda da pouca disponibilidade de ferramentas necessárias para essa finalidade. Além disso, um dos problemas existentes no uso do formalismo decorre do fato de que, alguns cientistas da computação, são conhecedores dos métodos formais, mas estão desatualizados sobre a realidade do mundo real, não utilizando assim as técnicas.

Quando se utiliza métodos formais são notadas vantagens em relação a outras técnicas. Pois eles permitem uma análise completa no início do processo de desenvolvimento, uma diminuição elevada nos custos de desenvolvimento em todas as fases desse processo, a possibilidade de analisar através do emprego de cálculos matemáticos a consistência e provar que a implementação corresponde à especificação (SOMMERVILLE, 1996).

Uma especificação formal é baseada na análise de requisitos do software e através dessa análise é possível observar exatamente o que o usuário espera

dessa. Além disso, pode-se verificar as limitações, a durabilidade, a interatividade, entre outros elementos que descrevem as funções do sistema. Quando essa análise de requisitos está sendo feita, quase sempre é escrita em linguagem comum. Porém, deve-se ter o apoio de linguagens de especificações, pois o uso das especificação trás grandes vantagens no processo de desenvolvimento de uma IU, uma vez que, ele permite que a funcionalidade e o modelo de interação possam ser avaliados por desenvolvedores e usuários, antes mesmos de serem implantados (DINIZ, 2000).

Atualmente, notações formais e os formalismos são usados em diversas áreas de IHC, por exemplo, modelagem cognitiva e análise de tarefas. Essas notações sempre tentam demonstrar a melhor maneira que o sistema é programado, e ainda, continuarem precisas em relação ao comportamento do mesmo. Uma especificação informal poderá fazer o mesmo, entretanto ao se empregar uma especificação formal, têm-se a vantagem de poder realizar a verificação da descrição do sistema (DIX, 1995).

Embora seja notória a importância do uso de formalismo, em Sommerville (1996) pode-se encontrar alguns contra-exemplos, tais como:

- Conservadorismo. Dificuldade de mostrar as vantagens da especificação formal de uma maneira objetiva.
- Falta de treinamento em matemática discreta por parte dos engenheiros de software.
- Algumas Classes de software são difíceis de especificar usando as técnicas formais.
- Existe um desconhecimento generalizado a respeito da aplicabilidade das especificações formais.
- Poucas ferramentas disponíveis para notações formais.

Todavia, vem se observando que vários fatores positivos vêm sendo ressaltados, os quais acarretam cada vez mais a utilização dos métodos formais pelos projetistas, trazendo benefícios que levam a sua utilização em projetos de software. Os formalismos são um meio de completar uma especificação informal, deixando-a mais coerente e livre de duplicidades, com a manutenção facilitada e

como conseqüência, a diminuição de custos, porque erros de especificação são encontrados no início do projeto.

3.2.1 Métodos Formais e IHC

Cada vez mais aumenta o interesse na aplicação de formalismo em IHC. Principalmente, no projeto de IU, mas sem ter uma aplicação de um método definido. Entretanto, no assunto referente ao projeto de IU, os esforços na construção de sistemas formais, cujos agentes são de outra classe, fazem com que a escolha de um sistema formal, entre alternativas de projeto, passe a ser uma necessidade a ser estudada, devendo os projetistas ter o máximo cuidado quanto à escolha da modelagem a ser utilizada, pois essa é uma atividade crítica da maioria dos métodos existentes.

Quando se seleciona uma técnica para especificar uma IU deve-se ter o cuidado de seguir propriedades, que muito, provavelmente, diminuirão o tempo dedicado pelos projetistas na criação da interface, desde o momento da especificação até sua implementação. Segundo Jacob (1983), as propriedades que devem ser seguidas são:

- A especificação da IU deve ser de fácil entendimento. Deve ser mais fácil do que o próprio software que implementa a interface.
- Precisão na especificação. Não se pode ter nenhuma dúvida em relação ao comportamento do sistema para cada entrada possível.
- Deve ser fácil testar a integração das partes.
- A técnica de especificação deve ser rica em detalhes o bastante para descrever comportamentos incomuns com o mínimo de complexidade.
- A técnica deve possibilitar a descrição do comportamento de uma interface sem se preocupar como ela vai ser implementada.
- Deve ser possível a construção de um protótipo, diretamente da especificação da IU.
- A estrutura da especificação deve estar bem relacionada com o modelo ao qual o usuário idealizou, ou seja, como ele pensa. A base de construção deve representar idéias que expressem algo para o usuário.

Segundo Carvalho e Oliveira (1998), “os métodos formais procuram retratar ou reproduzir formal ou mecanicamente certos estados de acontecimentos ou situações”. Há gramáticas, que podem ser utilizados para selecionar aspectos estruturais do conhecimento e com isso se tornam ferramentas de uso geral, já que, são capazes de obter uma parte dos significados semânticos através de recursos de representação. Alguns desses métodos formais têm relação com as duas abordagens que serão apresentadas em 3.2.2 e 3.2.3.

3.2.2 Abordagem Orientada a Propriedade

Nessa abordagem as especificações são dadas em axiomas, definindo assim o envolvimento das operações de todos os elementos que constituem o sistema. Levando em consideração unicamente a IU, a especificação algébrica está voltada para operações em vez de utilizar um modelo. Na construção de uma IU, pode-se especificar cada etapa em diferentes níveis de abstração, sem que isso venha acarretar perdas na sua generalidade. São utilizados módulos de especificação para descrever o sistema, onde cada um descreve um conjunto com estrutura algébrica precisa. Tendo assim, um conjunto de axiomas sobre a função do módulo, que foram expressas através da apresentação de termos de uma IU, com uma coleção de funções e um conjunto de condições (CABRERA, 1995).

Segundo COHEN (1986 apud MUNHOZ, 2004) existem uma série de vantagens e desvantagens sobre o emprego dessa abordagem. As quais serão apresentadas a seguir:

Vantagens:

- Nenhum valor é definido para qualquer tipo de dado e nenhum modelo preciso é formado, isto é, a especificação do sistema ou dado pode ser apresentada sem referenciar qualquer tipo de dado básico ou definido anteriormente.
- Propriedades interessantes de especificações podem ser elaboradas pela manipulação lógica dos axiomas.
- A construção das especificações é feita a partir de uma parte bem definida da matemática: álgebra, lógica, categoria, etc.
- Um tipo de dado pode ser compreendido isoladamente a partir de seus operadores e seus axiomas.

Desvantagens:

- O uso de operadores parciais e totais produz dois tipos de álgebras, onde duas semânticas terão que ser consideradas. Isso trará dificuldades no tratamento de erros e de situações de exceção.
- Alguns tipos não podem ser caracterizados por um número finito de axiomas, criando com isso, a necessidade do uso de funções ocultas, fato que enfraquece a compreensão do tipo somente pelos operadores e axiomas.
- As especificações são mais abstratas e matemáticas, bem definidas e com bom fundamento, porém, mais difícil de construir, especialmente para softwares grandes e por projetistas com pouca base matemática.
- A dificuldade da escolha do tipo de álgebra que melhor semântica dará à especificação. Isso decorre do fato, da interpretação dos axiomas ser uma álgebra e de existirem três tipos de interpretação: inicial, livre e final.

Embora a abordagem orientada à propriedade, seja uma base rigorosa para o desenvolvimento de sistemas interativos, infelizmente, na maioria dos casos, o seu uso não gera na totalidade resultado satisfatório, uma vez que, as notações usadas por eles, são muito difíceis de serem entendidas por pessoas não familiarizadas com notações matemáticas, tais como os típicos usuários de software (KNEUPER, 1997). Dessa forma, para diminuir esse problema, é fundamental que a especificação matemática seja acompanhada por uma escrita na linguagem natural que explique o que a especificação significa em termos do mundo real e demonstre as conseqüências da especificação (HALL, 1990).

3.2.3 Abordagem Orientada a Modelo

Essa abordagem se baseia na especificação da IU como um modelo construído utilizando-se estruturas matemáticas bem conhecidas, tais como: conjuntos, grafos e seqüenciais etc. A utilização dessas estruturas garante a concisão de uma interface criada através desses métodos. Os métodos orientados a modelos são usados para construir um modelo comportamental do sistema. Os modelos relevantes em IHC são: Modelo de Tarefa, Modelos Cognitivos, Modelos do Usuário, Modelos de Domínio, Modelo do Contexto, Modelo de Apresentação e o

Modelo do Diálogo. Esse tipo de abordagem possui algumas vantagens (COHEN, 1986):

- Cada operação possui uma pré-condição e uma pós-condição permitindo definir facilmente funções parciais.
- Manipulação de operações separadamente.
- Facilidade para especificar softwares grandes.
- É conveniente que alguns resultados sejam especificados pela união de propriedades.
- A existência de um modelo é bem aceita para especificação. É consistente e pode dar importantes sugestões ao projetista, aumentando a possibilidade de implementação.

A utilização da orientação a modelo é bem aceita para especificação de sistemas reativos, porque encapsulam estados (funciona como uma caixa preta). Além disso, segundo Palanque (1993) “a maioria dos sistemas de gerenciamento de interfaces do usuário corrente, é altamente baseada em objeto, a comunidade de IHC se interessa em seguir essa abordagem durante o projeto de interfaces dirigidas ao usuário”.

3.3 A Escolha de uma Abordagem

Neste capítulo são apresentados vários métodos para avaliação e construção de uma Interface, contudo entre elas duas se destacam: a avaliação Heurística de Nielsen (permite fazer um diagnóstico dos problemas da IU diretamente com os usuários, através de um questionário) e abordagem orientada a modelo (permite a construção de um modelo comportamental do sistema), ambas têm as aplicações necessárias para o desenvolvimento de um protótipo, de uma interface em uso.

Nesse trabalho a aplicação dos métodos formais para a construção de uma IU, se dará após o levantamento dos problemas existentes em uma interface que já está em uso. Para se realizar essa avaliação foi necessária a aplicação de um mecanismo que permitisse junto aos usuários detectar os problemas na IU em questão.

4. Avaliação Heurística

O método de avaliação heurística é um método que tem como objetivo identificar problemas de usabilidade conforme um conjunto de heurísticas (NIELSEN,1994). Ele se baseia na melhoria das práticas definidas por profissionais experientes e especialistas em IHC, ao longo de diversos anos de trabalho nesta área.

A Avaliação é realizada por pequenos grupos de peritos de telas (no mínimo 3 e máximo 5) onde examinam uma determinada interface e procuram por problemas que violem alguns dos princípios gerais do projeto (BRANDÃO, 2005).

Este método é bastante rápido, e de menor custo que a maior parte dos métodos de avaliação que são mais conhecidos. Como em todo método de avaliação, a avaliação heurística envolve uma fase de preparação, na qual se definem:

- Proposta de design (papel ou protótipo)
- Hipóteses sobre os usuários (opcional)
- Cenário de tarefas (opcional)

Embora a Avaliação Heurística seja realizada na maioria das vezes por peritos, nesse trabalho ela foi aplicada diretamente com o usuário, tendo o acompanhamento de um perito para auxiliar na interpretação da avaliação.

4.1 Procedimentos da Avaliação Heurística

4.1.1 Avaliação Individual

A avaliação desenvolvida deve ser realizada de acordo com o seguinte procedimento (NIELSEN; MOLICH, 1994):

1. Sessões curtas (1 a 2 horas) de avaliação individual, onde cada usuário deve executar as seguintes tarefas:

- Julgar a conformidade da interface com um determinado conjunto de princípios (“heurísticas”) de usabilidade
- Anotar os problemas encontrados e sua localização
- Julgar a gravidade destes problemas
- Gerar um relatório individual com o resultado de sua avaliação e comentários adicionais.

É importante que estas sessões sejam individuais, para que um avaliador não seja influenciado pela opinião de outros. Durante cada sessão de avaliação, o avaliador percorre a interface diversas vezes, inspecionando os diversos elementos de interface e comparando-os com a lista de heurísticas de usabilidade.

4.1.2 Consolidação da avaliação dos usuários

- Novo julgamento sobre o conjunto total dos problemas encontrados
- Relatório unificado dos problemas de usabilidade

Nesta etapa, cada avaliador tem acesso aos relatórios individuais de todos os avaliadores, e pode expressar suas idéias sobre os problemas apontados pelos outros avaliadores. Ao final desta etapa, deve-se gerar um relatório único e consolidado sobre todos os problemas encontrados.

4.1.3 Seleção dos problemas que devem ser corrigidos

Esta etapa deve ser realizada junto ao cliente ou ao gerente de projeto. Trata-se de uma análise de custo/benefício das correções aos problemas encontrados na etapa anterior. Esta análise deve levar em conta não apenas a gravidade dos problemas, mas também os prazos e o orçamento do projeto, bem como a adequação da equipe de desenvolvimento.

4.1.4 Heurísticas

Nielsen (1993) propôs um conjunto básico de heurísticas. Cada elemento de interface (ou conjunto de elementos) deve ser analisado para verificar sua conformidade com cada uma das seguintes heurísticas:

- Visibilidade do estado do sistema - manter os usuários informados sobre o que está acontecendo, através de *feedback* adequado e no tempo certo.
- Correspondência entre o sistema e o mundo real - utilizar conceitos, vocabulário e processos familiares aos usuários.
- Controle e liberdade do usuário - fornecer alternativas e “saídas de emergência”; possibilidades de *retorno* e *avanço*.
- Consistência e padronização - palavras, situações e ações semelhantes devem significar conceitos ou operações semelhantes; caso haja convenções para o ambiente ou plataforma escolhidos, estas devem ser obedecidas.
- Prevenção de erro- tentar evitar que o erro aconteça, informando o usuário sobre as conseqüências de suas ações ou, se possível, impedindo ações que levariam a uma situação de erro.
- Ajudar aos usuários para reconhecer, diagnosticar e se recuperar de erros - mensagens de erro em linguagem simples, sem códigos, indicando precisamente o problema e sugerindo de forma construtiva um caminho remediador.
- Reconhecimento em vez de memorização - tornar objetos, ações e opções visíveis e compreensíveis.
- Flexibilidade e eficiência de uso - oferecer aceleradores e caminhos alternativos para uma mesma tarefa; permitir que os usuários customizem ações freqüentes.
- Design estético e minimalista - evitar porções de informação irrelevantes. Cada unidade extra de informação em um diálogo compete com as unidades de informação relevantes e reduz sua visibilidade relativa.
- Ajuda e documentação - devem ser fáceis de buscar, focadas no domínio e na tarefa do usuário, e devem listar passos concretos a serem efetuados para atingir seus objetivos.

Para cada problema encontrado, ou seja, para cada heurística violada, deve-se definir ainda a localização do problema, ou seja, onde ele ocorre na interface, e sua gravidade.

Com relação à localização, o problema pode estar:

- Em um único local na interface;
- Em dois ou mais locais na interface, casualmente;
- Na estrutura geral da interface, de forma ordenada; ou
- Pode ser algo que “não está lá”, ou seja, precisa ser incluído na interface

Já a gravidade (severidade) do problema é calculada, por cada especialista, como uma combinação dos fatores:

- Frequência com que o problema ocorre- É um problema comum ou raro?
- Impacto do problema - Será fácil ou difícil para os usuários superarem o problema?
- Persistência do problema - É um problema que ocorre apenas uma vez e que os usuários conseguem superar facilmente, ou os usuários serão incomodados pelo problema repetidas vezes?

A gravidade do problema é definida por um valor da seguinte escala:

- 0 – Sem importância: este valor pode resultar da avaliação de um usuário sobre um problema apontado por outro usuário, caso o mesmo analise a avaliação dos demais avaliadores
- 1 – Problema cosmético: não precisa ser consertado a menos que haja tempo extra no projeto
- 2 – Problema simples: o conserto deste problema é desejável, mas deve receber baixa prioridade
- 3 – Problema grave: importante de ser consertado; deve receber alta prioridade
- 4 – Catastrófico: é imperativo consertar este problema antes do lançamento do produto, ou imediatamente em uma interface que já está em uso.

Como produto da avaliação heurística, o responsável pela avaliação escreve um relatório consolidado. Este relatório pode conter, por exemplo, os seguintes itens:

- Problemas esperados (e possíveis consertos)
- O quão bem o sistema apóia as tarefas dos usuários
- Caminhos de interação primários (importantes e/ou freqüentes)
- Caminhos de interação alternativos ou pouco utilizados
- Consistência
- Elementos de estilo
- Recomendações de projeto

Também é possível realizar uma avaliação heurística nas etapas iniciais do ciclo de projeto e desenvolvimento. Esta avaliação pode ser feita sobre interfaces que ainda não tenham sido implementadas, representadas somente no papel.

Somando-se os capítulos anteriores a esse, se tem subsídios suficientes para fazer a avaliação da interface (alvo de estudo). Esses dados obtidos serão analisados e colocados em ordem de acordo com a gravidade do problema mencionado pelos usuários. Os dados coletados servirão de base para ser realizada a modelagem da nova interface, utilizando para isso um método formal que satisfaça as condições pré-estabelecidas, para se atingir uma maior interatividade.

5. Formalismo StateChart

O StateChart é um formalismo visual utilizado, geralmente, para especificar sistemas complexos, com concisão e em diferentes níveis de abstração. Sua capacidade de focar em determinados cenários e seu mecanismo histórico proporcionam ao projetista a possibilidade de especificar um sistema completo mais rápido, tornando o projeto mais legível do que se fosse especificado somente em palavras. Uma das principais características do *StateChart*- SC é sua representação hierárquica. Essa representação permite a introdução de contêineres, com subestados e as transições. Do exposto, pode-se observar que este tipo de formalismo é apropriado para a modelagem de IU, permitindo que o projetista divida a interface em módulos, podendo especificar cada módulo separadamente (SUN, 2003).

Nas seções seguintes serão apresentados os elementos básicos do SC e também exemplos de sua utilização na especificação da IU.

5.1 Elementos Básicos de StateChart

Os elementos que compõem a notação do SC são basicamente os estados representados por bolhas e setas de transições que fazem a ligação entre esses estados (ANDERSON, 2003). Por exemplo, uma tela que o usuário vê em um determinado momento é toda desenhada dentro de uma bolha, representando um estado. Para que haja uma mudança de estado (geração de uma nova bolha) se faz necessário algum tipo de interação do usuário.

Na Fig. 2, demonstram-se os principais elementos desse formalismo, as setas com pontos pretos são estados iniciais e o ponto circulado é o estado final. Uma transição é representada por uma seta, evento (parâmetros) [condição] /ação, todos ou qualquer um desses parâmetros podem estar presentes numa seta de transição.

Cada seta de transição pode conter a escrita acoplada com o nome do evento, lista de parâmetros, condição e o evento que deve ser realizado para completar a transição. Um estado (*state* 5) pode começar de um subestado inicial (*state* 3) ou de seu mais recente subestado (*state* 7) como mostra o diagrama com a letra “H” na Fig. 2. Esses recursos de estados históricos permitem ao projetista a modelagem de comportamentos de IU, tal como, o botão de “Cancelar”, que retorna ao estado anterior da interface de uma aplicação (ANDERSON, 2003).

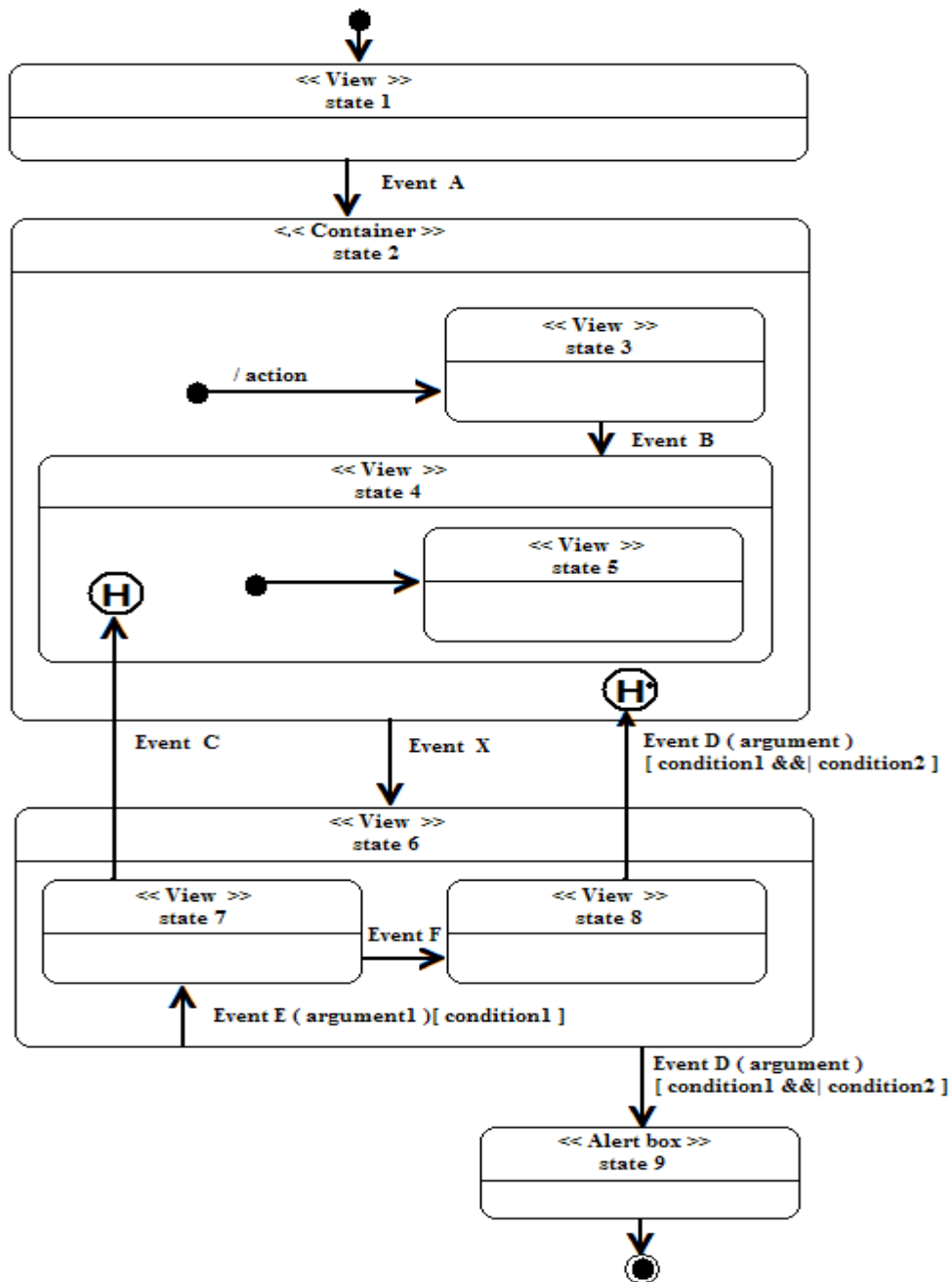


Figura 2 – Elementos Básicos de um StateChart

FONTE: (ANDERSON, 2003)

De posse desse conhecimento é possível demonstrar melhor o poder do SC na especificação de sistemas reais. Usa-se como exemplo, o caso de uma caixa de diálogo (Fig. 3), contendo *login* e senha para se ter acesso ao sistema.

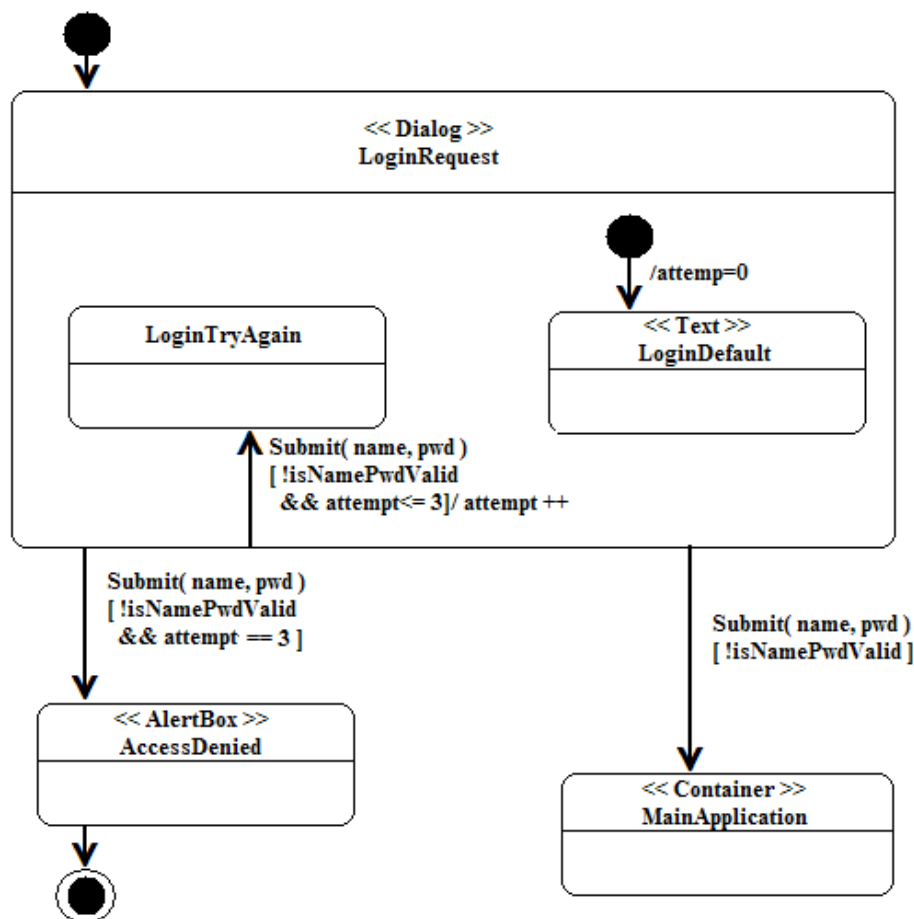


Figura 3 – Exemplo de Diálogo de *Login*

FONTE: (ANDERSON, 2003)

O diálogo de solicitação de *login* (*loginRequest*) contém dois possíveis subestados: o estado padrão, que irá colocar o *prompt* no *login* do usuário, e o subestado alternativo que mostra ao usuário que a tentativa de *login* foi incorreta e pede que ele tente novamente. Quando é a primeira entrada do usuário no diálogo, o número de tentativas é zerado, isso é demonstrado como uma ação (*/attempt = 0*) na flecha que inicia o processo de *login*. A partir do diálogo de solicitação de *login*, existe somente uma ação possível, o envio do nome e da senha. Esse processo possui três possíveis transições (ANDERSON, 2003):

- Se o nome e a senha estão corretos, a aplicação vai para a janela principal (*MainAppllication*).

- Se houve um erro no nome ou na senha, mas o usuário tentou menos de três vezes, a aplicação levará a um subestado para se tentar novamente (*LoginTryAgain*).
- Se houve um erro no nome ou na senha e o usuário já tentou mais três vezes o *login*, a aplicação mostrará uma mensagem dizendo que seu acesso foi negado.

Esse exemplo mostra como foi feita a especificação parcial de um sistema maior, demonstrando um simples diálogo de *login*.

Como o Formalismo StateChart apresenta uma forma de decompor o projeto em subestados e criando também uma estrutura hierárquica, que faz o número de setas de transição ficar mais reduzido, dando mais clareza a toda a modelagem. Obteve-se conhecimento suficiente para se realizar a avaliação mencionada no capítulo 4 e também a modelagem de uma interface que é o alvo principal desse trabalho. A fim de, conseguir melhorias em uma IU, aumentando assim a produtividade e diminuindo o tempo e custo na utilização da mesma.

6. Estudo de Caso

Com o aprendizado e estudos feitos sobre os assuntos mencionados nos capítulos anteriores, adquiriu-se vários conhecimentos com os quais possibilitam-se realizar as avaliações necessárias sobre a IU, a fim de, diagnosticar possíveis problemas. Com a ocorrência de problemas na IU, serão criadas novas modelagens, utilizando para isto, o conhecimento obtido anteriormente, e assim poderá-se desenvolver uma nova interface.

6.1 Resultados Obtidos com a avaliação Heurística sobre o sistema ABBA.

De acordo com o capítulo 4 realizou-se o Processo de Avaliação Heurística, com 3 usuários que trabalham diariamente com a interface e 1 usuário que teve seu primeiro contato com a interface do ramo óptico, também conhecido como ABBA .

Para se ter uma idéia mais concisa do(s) problema(s) existente(s), será identificada com a avaliação de Nielsen, a sessão e o grau de severidade. Também serão demonstrados no final as imagens e seus devidos problemas de interface numerados de acordo com os problemas mencionados abaixo.

1. Visibilidade do Status do Sistema

- Problema

Gravidade – Grave para 2 usuários.

Não é informado ao usuário o que está ocorrendo no exato instante quando pressionado uma tecla, assim como há algumas áreas onde não se tem informações do que deve ser colocado nos campos. Como mostram as Fig. 5, 7 e 10.

2. Compatibilidade entre o sistema e o mundo real.

- Problema

Gravidade – Grave para 2 usuários; e cosmético para 1 e simples para o outro.

O usuário deve possuir um grau de conhecimento tanto na área óptica, como também sobre a interface para efetuar tarefas simples, uma vez que, são utilizados termos de difícil entendimento. Como mostram as Fig. 7 e 9.

3. Liberdade e controle do usuário

- Problema

Gravidade – Grave para 1 usuário e sem importância para outro.

Quando é realizada uma venda, a primeira opção é a escolha do vendedor, no caso tenha ocorrido um descuido e digitou-se o código do vendedor de forma incorreta, o mesmo só poderá ser substituído caso se faça a venda novamente e se exclua a anterior.

4. Consistência e padrões

- Problema

Gravidade – Grave para 1 usuário e indicado simples por outro.

Encontra-se entre as interfaces alguns nomes iguais para funções diferentes, deixando o usuário em dúvida em qual opção selecionar. Como mostra a Fig. 5.

5. Prevenção contra erros

Foram identificados mais de um problema sendo divididos por categoria.

- Problema 1

Gravidade – Catastrófico para 1 usuário.

Caso o vendedor não tenha fechado a venda, mas imprimiu-se o carnê, fechando desse modo a tela; à venda não ficará cadastrada no caixa do dia.

- Problema 2

Gravidade – Grave para 2 usuários.

A falta de informação de alguns botões ou campos de textos, ou seja, faltam informações mais específicas, para não cometer algum erro. Como mostram as Fig. 4 e 5.

6. Reconhecimento em lugar de lembrança

- Problema

Gravidade – Grave para 1 usuário, simples para 2, e cosmético para o outro.

É necessário na janela vendas ao cliente lembrar o código do vendedor, não tendo esse nenhuma opção de procura por nome caso o usuário não saiba o código do vendedor. A procura por um cliente tem a opção de localizar em um campo o nome do cliente, mas deve ser pressionado F4, ou clicar em um botão o qual está identificado com pontos "...", deixando assim uma informação dúbia do que está abordando. Como mostra a Fig. 10.

7. Flexibilidade e eficiência de uso

Foram identificados mais de um problema sendo divididos por categoria.

- Problema 1

Gravidade – simples para 2 usuários.

Há existência de atalhos na interface, mas que são pouco utilizados, uma vez que, são de difícil aprendizagem e não são eficientes na grande maioria.

- Problema 2.

Gravidade – simples para 1 usuário.

Tem-se poucos campos para preenchimento de dados para o cônjuge. Como mostra a Fig. 8.

- Problema 3

Gravidade – Grave para 1 usuário.

Não há como informar em suprimentos de onde provem o dinheiro que entrou no caixa do dia. Como mostra a Fig. 11.

8. Projeto minimalista e estético

Foram identificados mais de um problema sendo divididos por categoria.

- Problema 1

Gravidade– grave para 2 usuários.

Existem alguns botões e campos de textos que não são utilizados pela empresa, mas alguns desses devem ser preenchidos da mesma forma, já que, o sistema não salva os dados caso isso não ocorra. Como mostram as Fig. 5, 6, 7, 9 e 11.

- Problema 2

Gravidade – grave para 1 usuário e simples para outro.

Contém alguns diálogos irrelevantes para os usuários finais, que são compreendidos somente pelo programador. Como mostram as Fig. 10, 11 e 13.

9. Auxiliar os usuários a reconhecer, diagnosticar e recuperar-se de erros.

Foram identificados mais de um problema sendo divididos por categoria.

- Problema 1

Gravidade – grave para 2 usuários.

Não há identificação na maioria dos erros e quando há deixam o usuário com dúvidas.

- Problema 2

Gravidade – grave para 1 usuário.

No recebimento de carnê quando o usuário clicar para sair o mesmo não executa a ação. Como mostra a Fig. 12.

10. Ajuda e documentação.

Gravidade – Catastrófico, selecionado entre 3 dos 4 usuários que realizaram o teste.

- Problema

Encontra-se somente um tópico de ajuda, entre todas as janelas do ABBA como pode ser observado na Fig. 5, e a mesma quando selecionada não aparece nada. Esta sessão foi descrita, como o mais grave dos problemas, com a qual os usuários depararam-se.

6.1.1 Figuras demonstrando problemas da interface atual

As figuras abaixo estão marcadas com um círculo, indicando os problemas correspondentes, e numeradas de acordo com a sua sessão descritas no princípio desse capítulo.



Figura 4 – Caixa Principal – Menu de opções

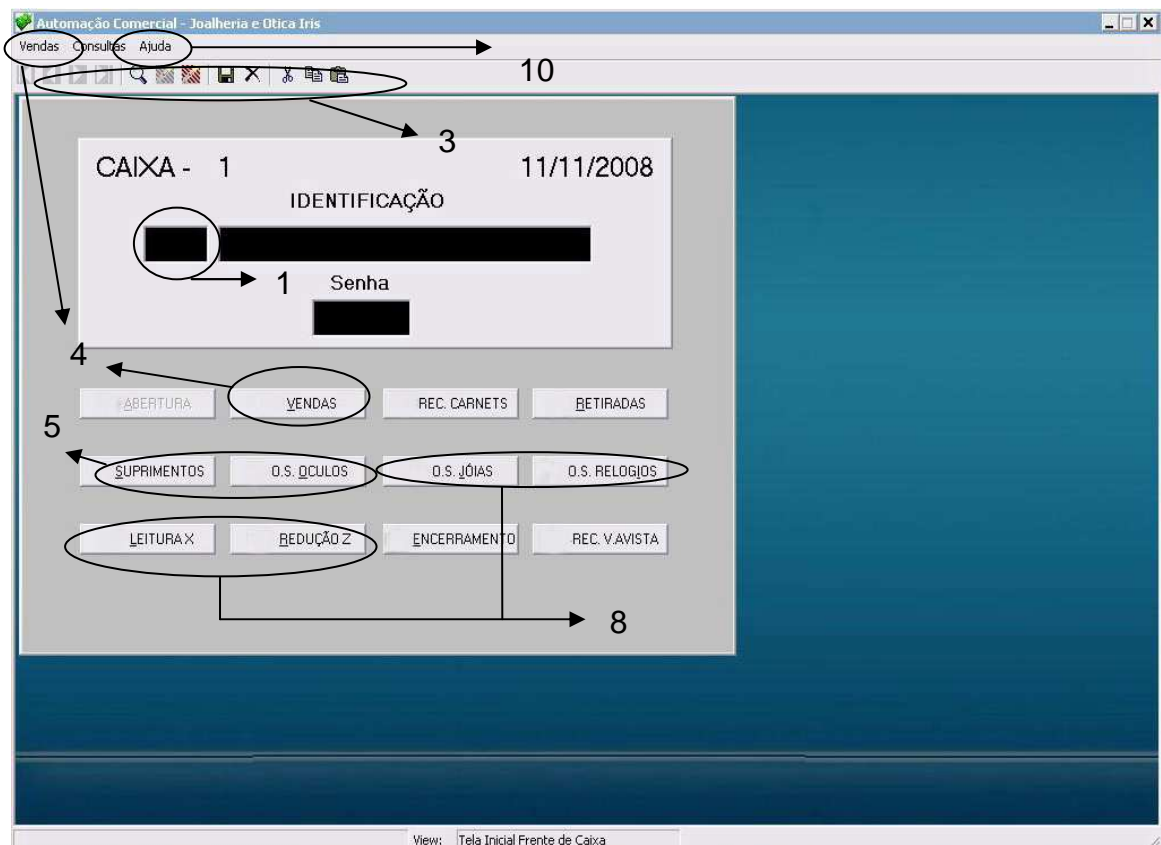


Figura 5 - Caixa do Dia – Tela de operações principais do Caixa

Figura 6 - Cadastro Clientes - Dados Pessoais

Figura 7 - Cadastro Clientes - Dados Profissionais

Figura 8 - Cadastro Clientes - Cônjuge e referências

Figura 9 - Cadastro da Dioptria (grau) do Cliente

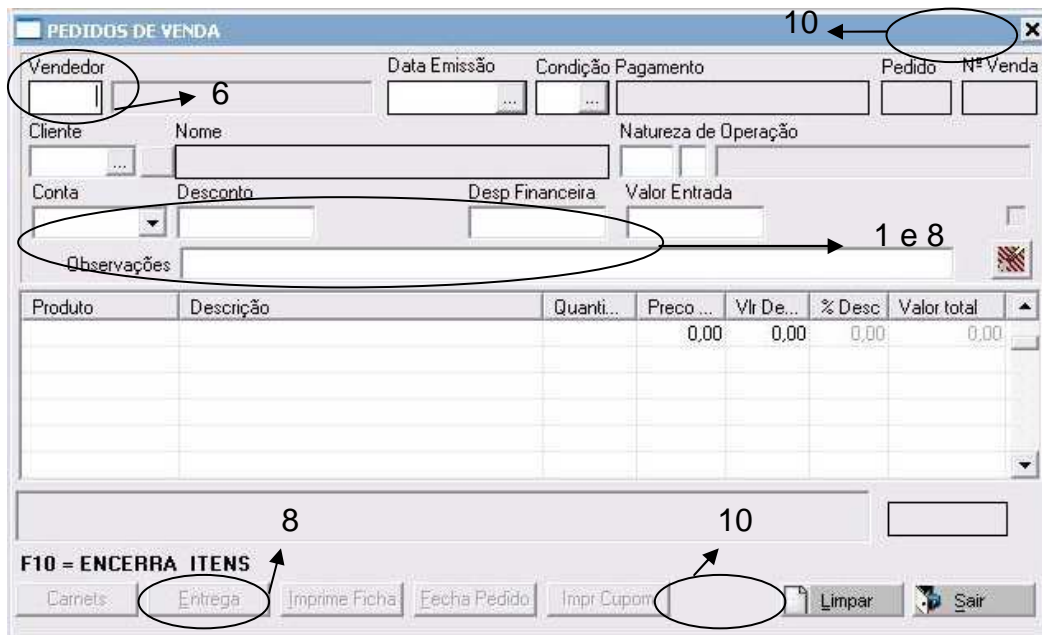


Figura 10 - Venda ao Cliente – Formas de Pagamento

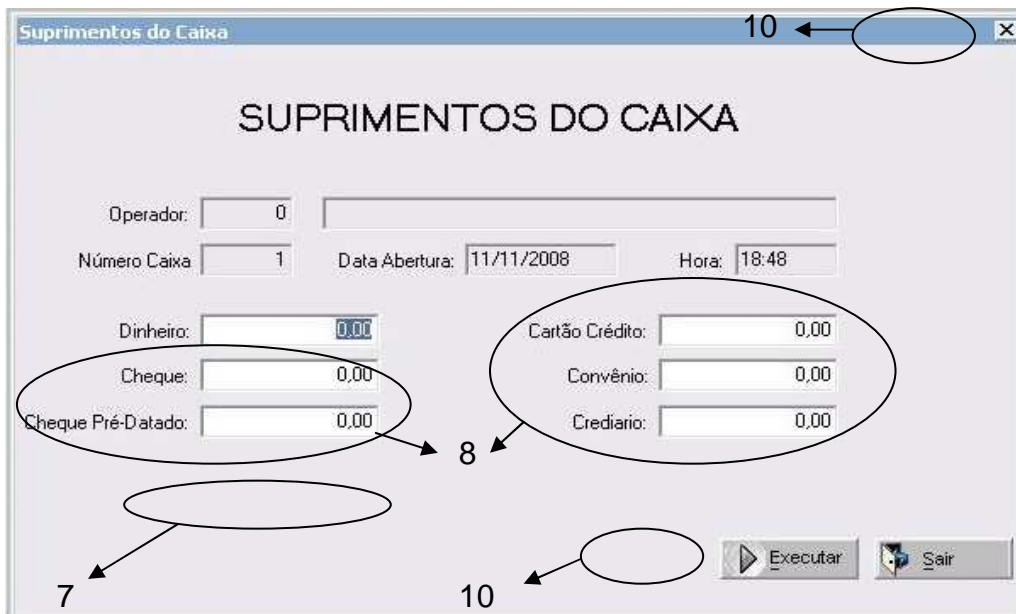


Figura 11 - Entrada de Dinheiro

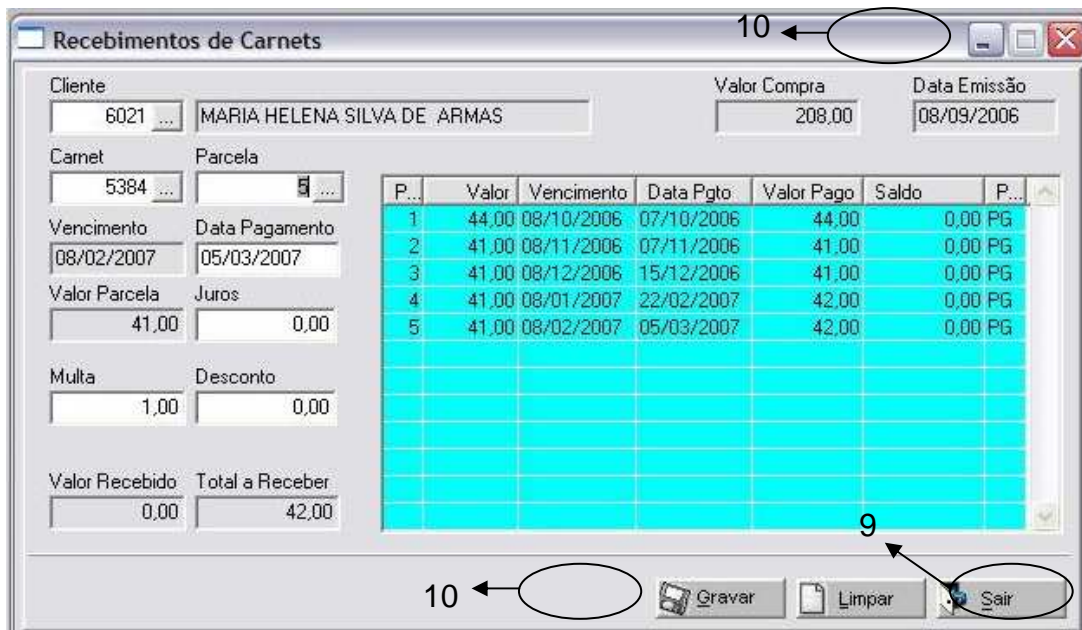


Figura 12 - Recebimento Carnê

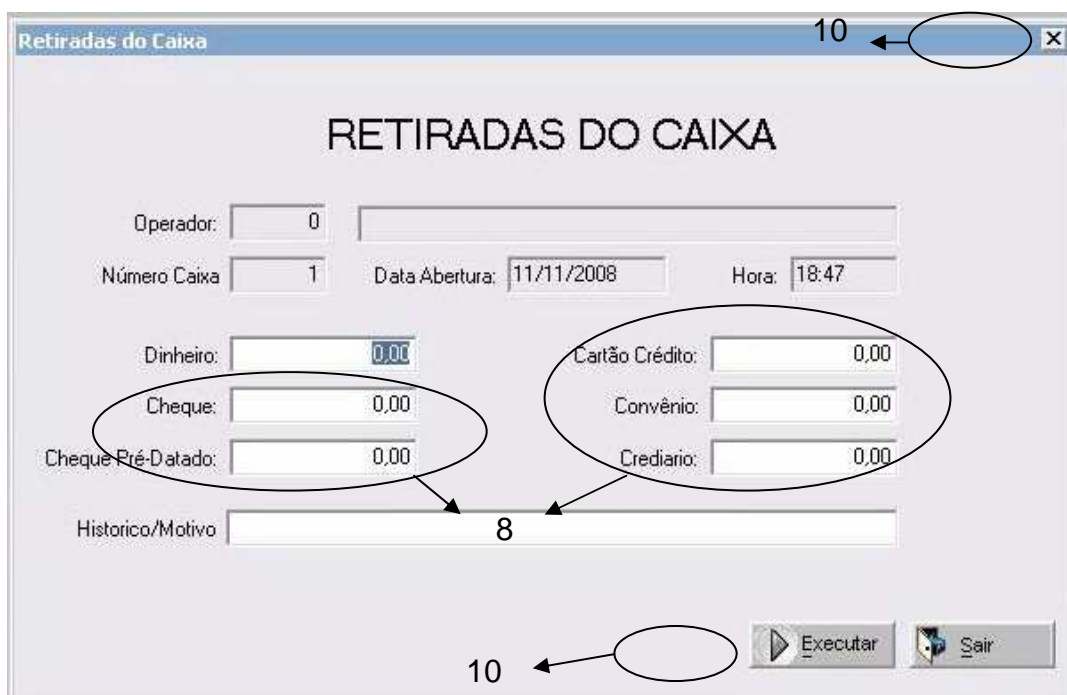


Figura 13 - Retirada de Dinheiro

6.2 Modelando a nova interface com o StateChart

Utilizando os conhecimentos obtidos no estudo do capítulo 5, referentes ao formalismo StateChart, foi desenvolvida uma nova modelagem para cada janela da interface que apresentou problemas, diagnosticados a partir da avaliação Heurística mencionada.

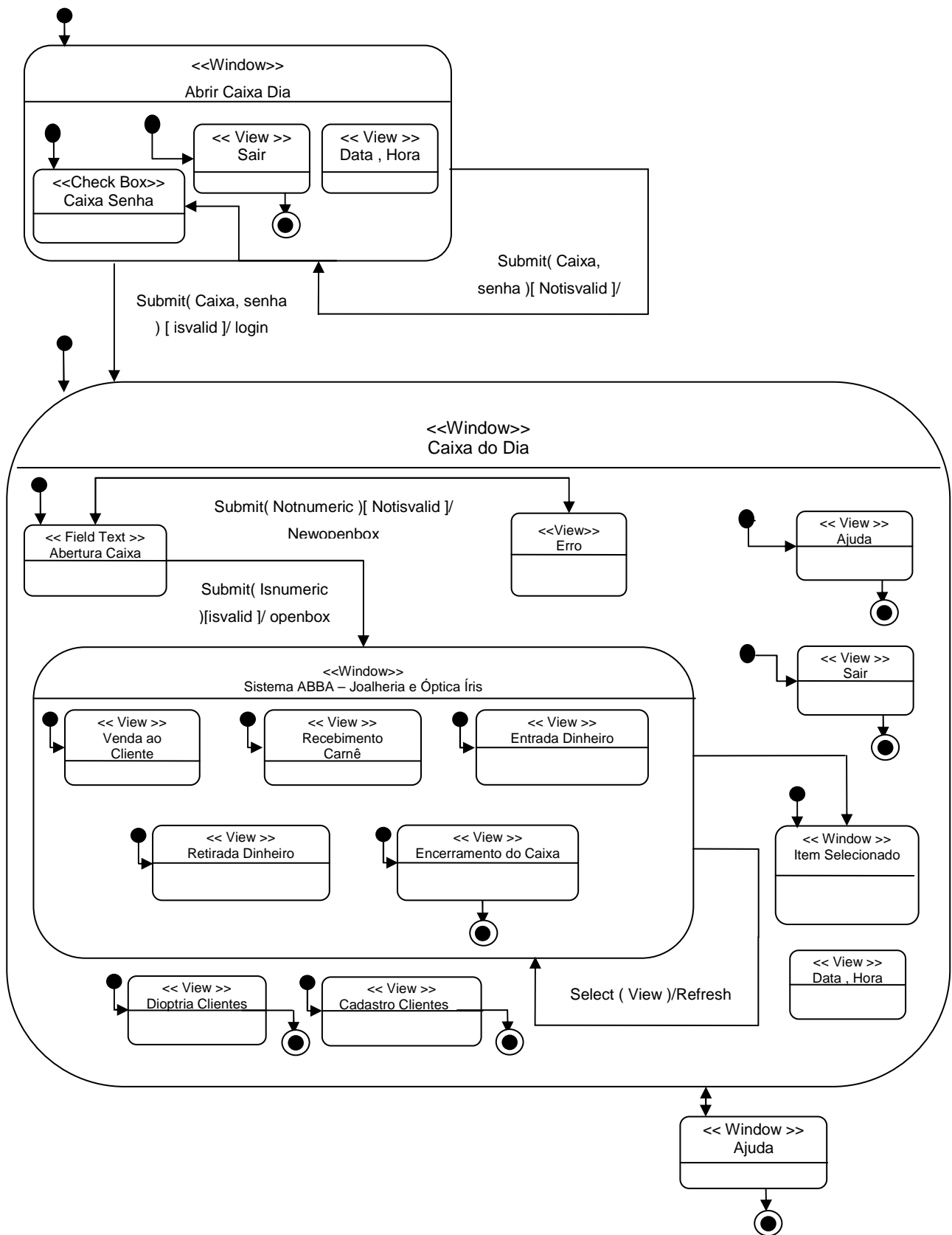


Figura 15 - Modelagem StateChart sobre as avaliações da Fig. 5.

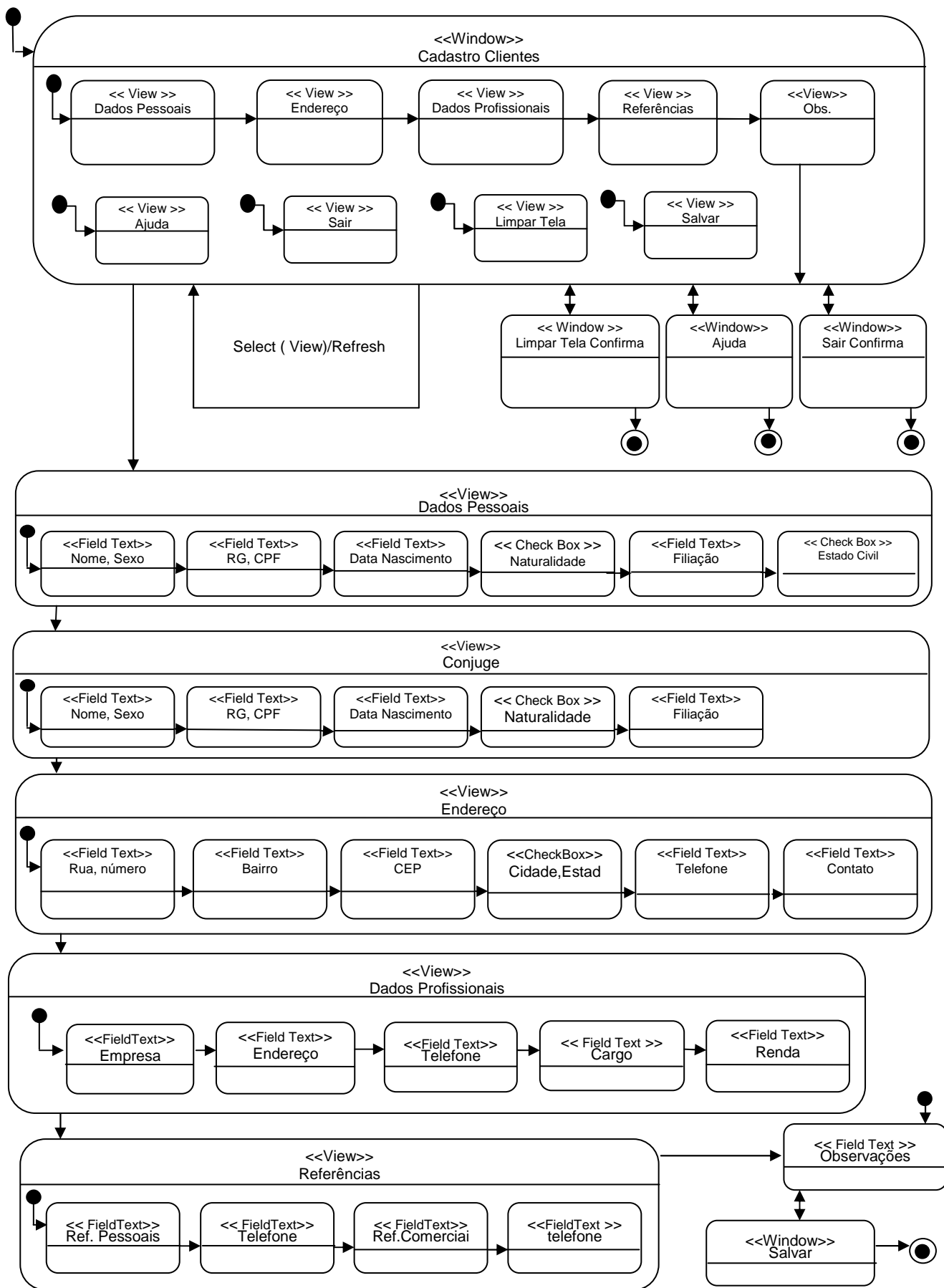


Figura 16 - Modelagem StateChart sobre as avaliações das Fig. 7,8 e 9.

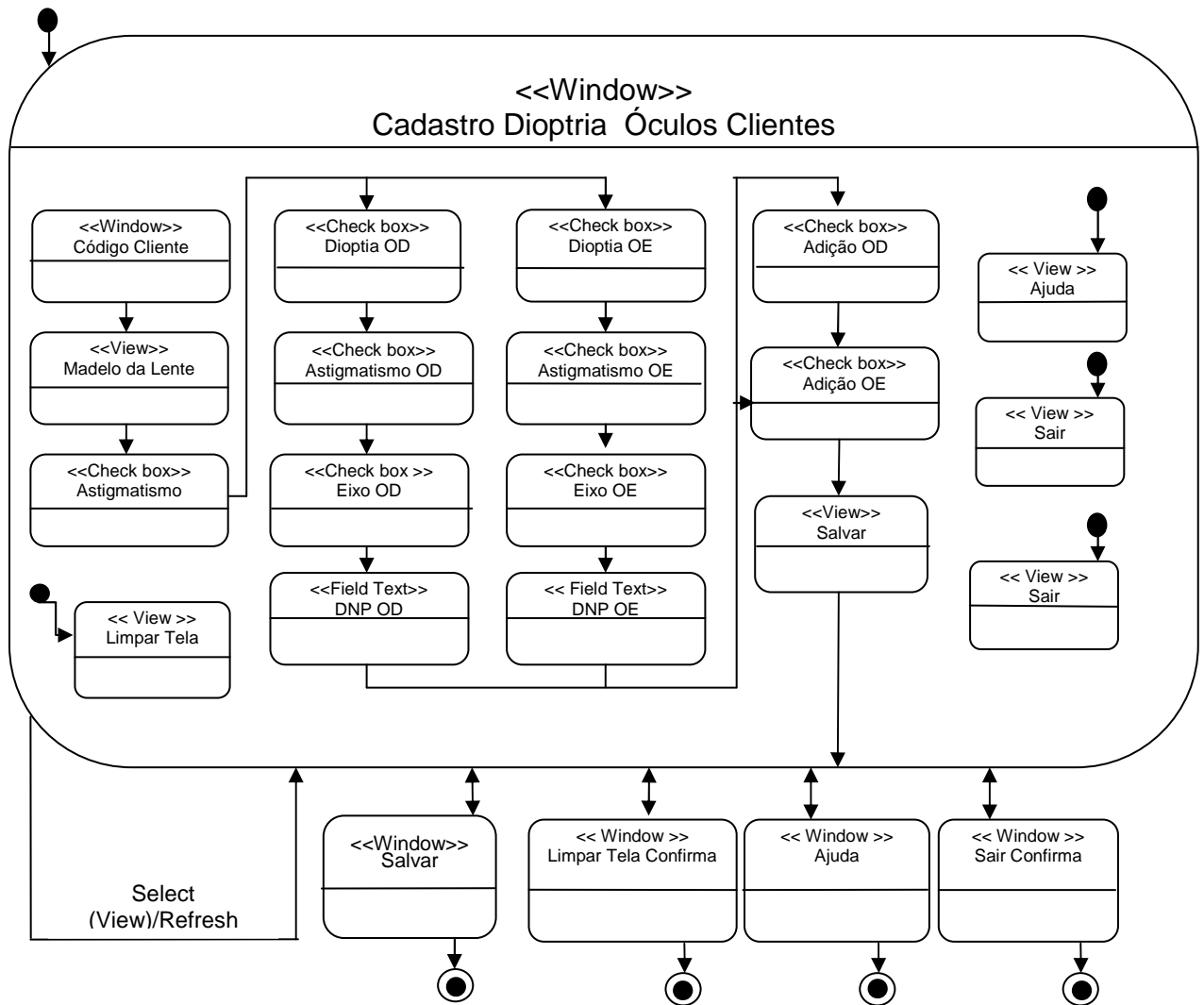


Figura 17 - Modelagem StateChart sobre as avaliações da Fig. 9.

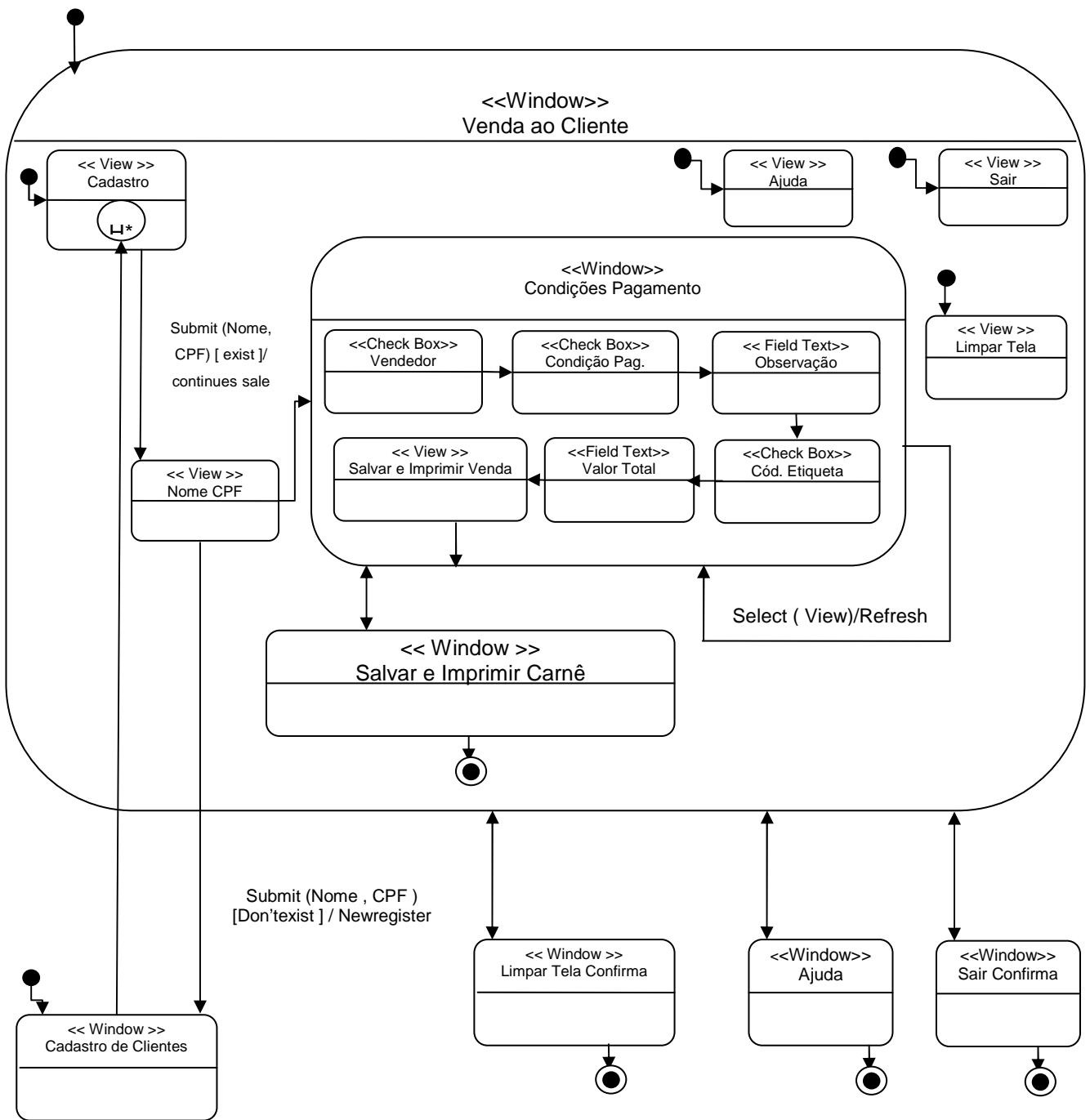


Figura 18 - Modelagem StateChart sobre as avaliações da Fig. 10.

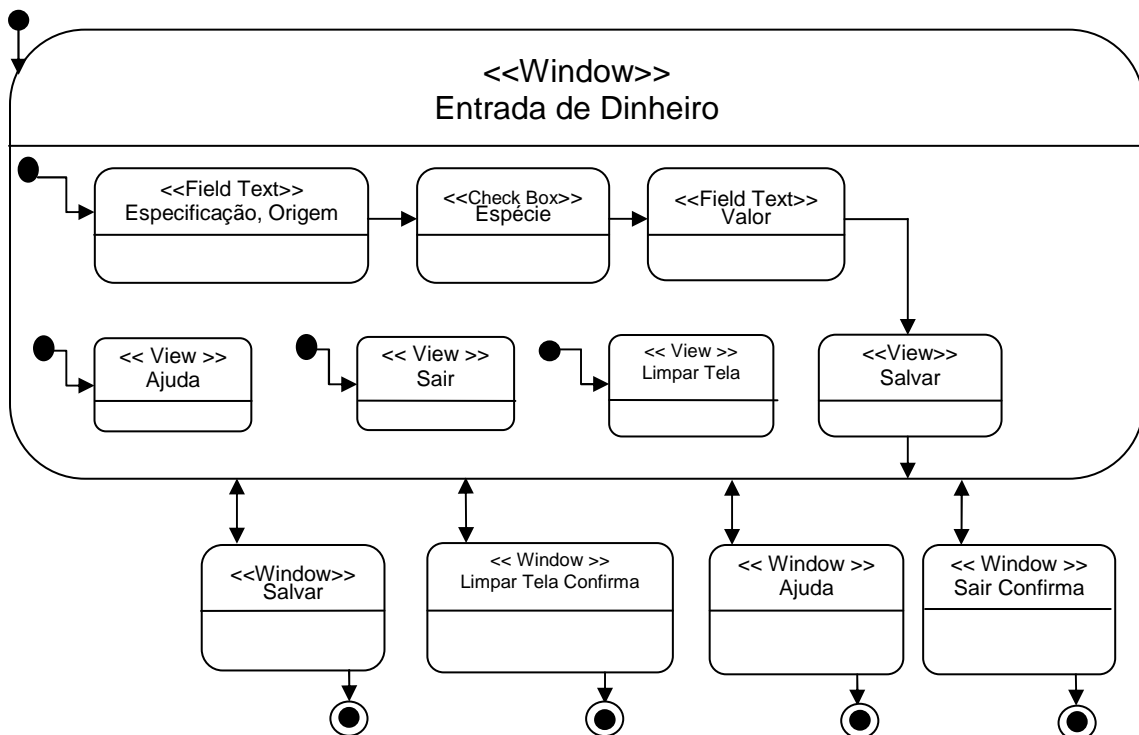


Figura 19 - Modelagem StateChart sobre as avaliações da Fig. 11

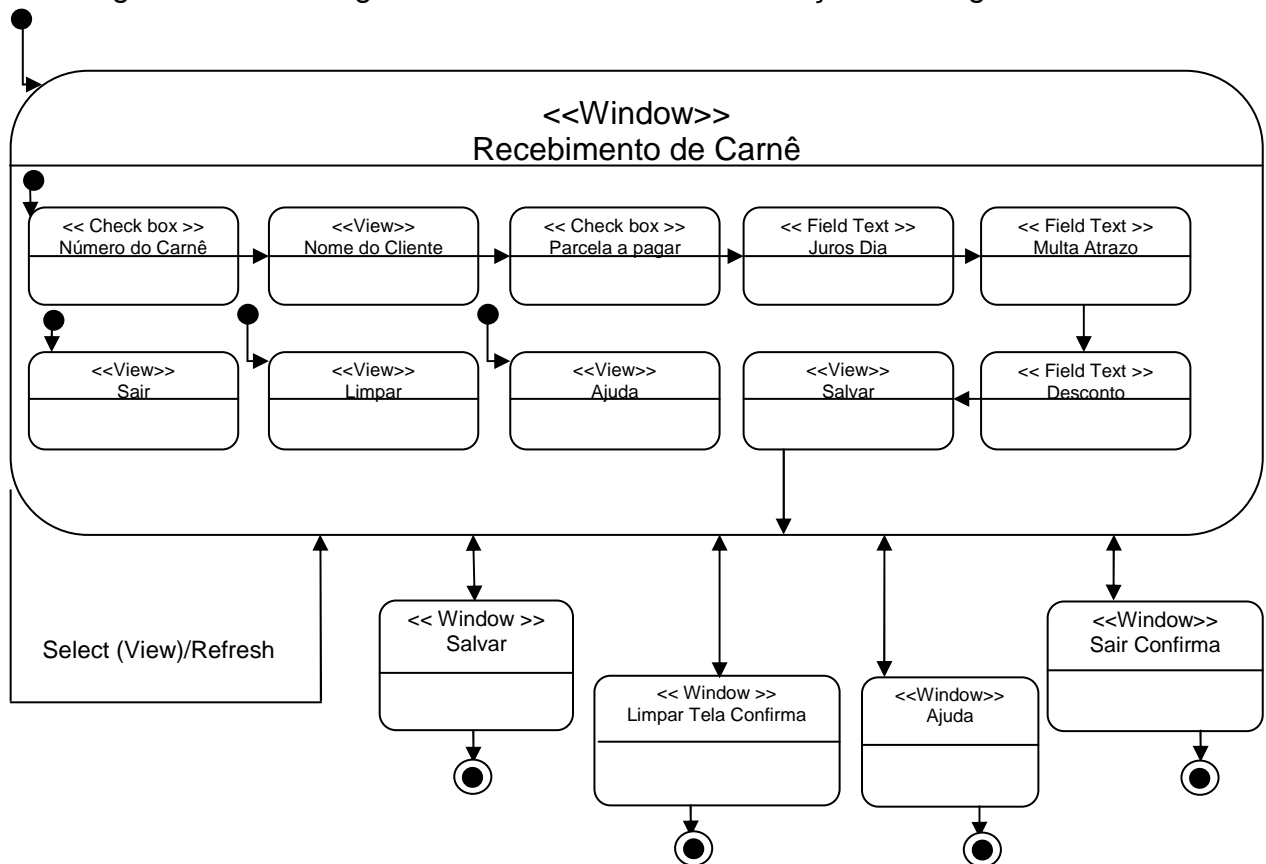


Figura 20 - Modelagem StateChart sobre as avaliações da Fig. 12

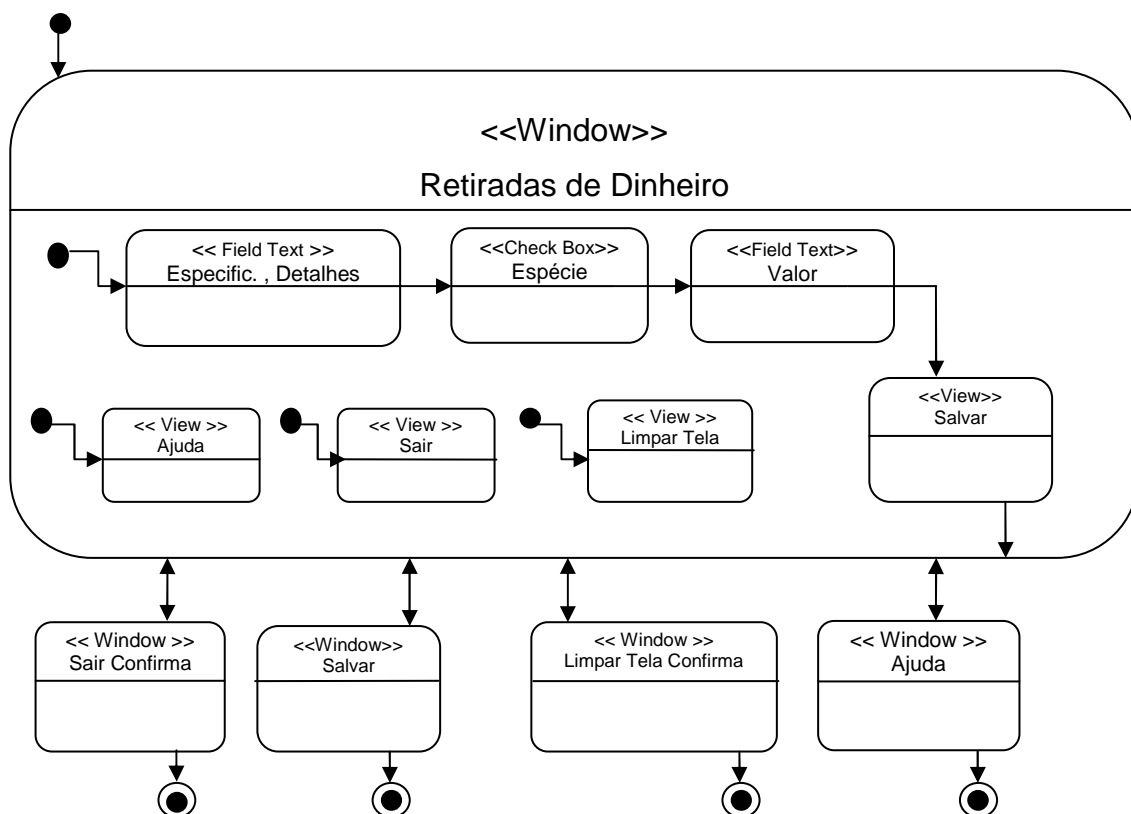


Figura 21 - Modelagem StateChart sobre as avaliações da Fig. 13.

6.3 Novas Interfaces projetadas decorrentes da aplicação do StateChart

De acordo com as modelagens desenvolvidas na sessão anterior, obtiveram-se informações suficientes para desenvolver a nova IU. Esta foi desenvolvida em Java, utilizando a plataforma conhecida como NetBeans, versão 6.1.

A interface foi modificada em alguns aspectos, os quais não foram descritos na avaliação heurística, através da aplicação do capítulo 2, verificou-se que tais mudanças poderiam proporcionar uma maior usabilidade.

Na janela do Caixa principal como mostra a Fig. 22, foram introduzidos atalhos, utilizando para isto a tecla ALT + Letra Principal do Ícone, para os usuários mais experientes. Além disso, foi introduzido um ícone de ajuda para usuários iniciantes, na qual o usuário, caso tenha alguma dificuldade, poderá encontrar informações sobre que procedimento adotar.

Na versão anterior detectou-se ambigüidade em certos ícones que também foram alterados, a fim de se ter, uma melhor compreensão das áreas que os mesmos abrangem.



Figura 22 - Nova IU do Caixa Principal - Menu de Opções

No Caixa do dia, como mostram as Fig. 23 e 24, foram atribuídos para os ícones tanto teclas de atalho como também a utilização do ALT+ Letra ou Número correspondentes ao ícone ou barra de ferramentas. Com a passagem do mouse sobre os ícones também são informados os procedimentos que devem ser tomados pelos usuários, como mostra na Fig. 24.

Caso o usuário aperte a opção: Sair, aparecerá na tela uma janela confirmando, ou não a sua decisão.

Foram ainda alterados alguns ícones que também estavam com nomes idênticos, porém com semântica diferente, fazendo com que o usuário cometesse equívocos. Alterou-se ainda alguns nomes, os quais eram de difícil entendimento para usuários iniciantes no ramo.

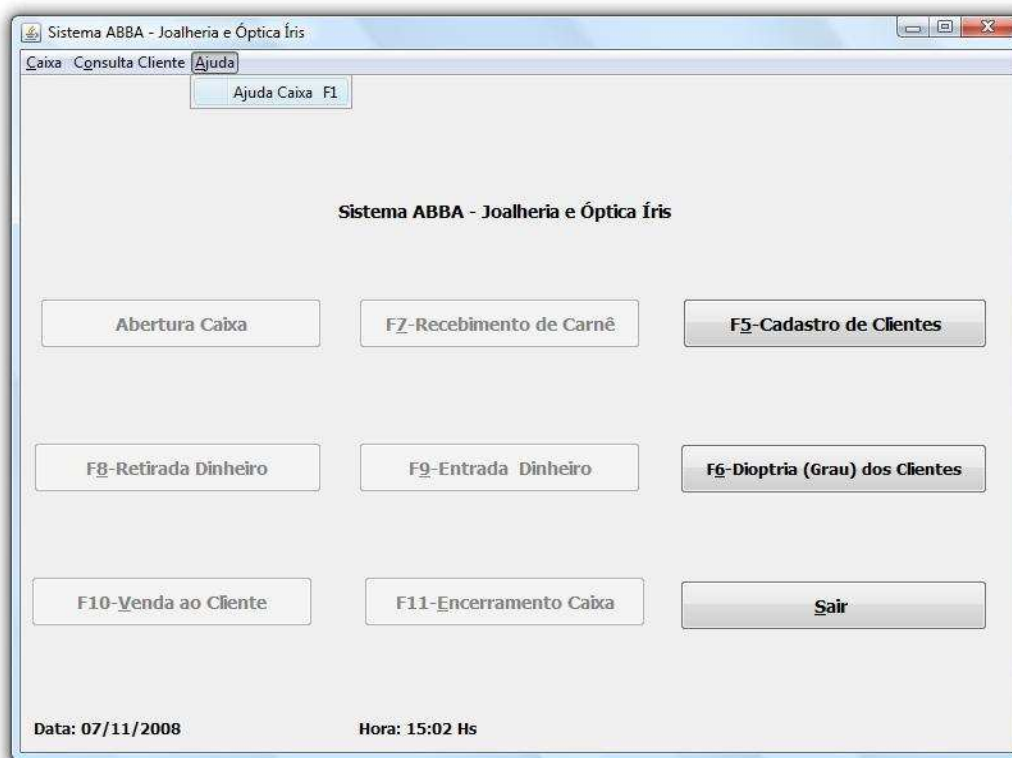


Figura 23 - Caixa do dia – Tela de operações principais do Caixa

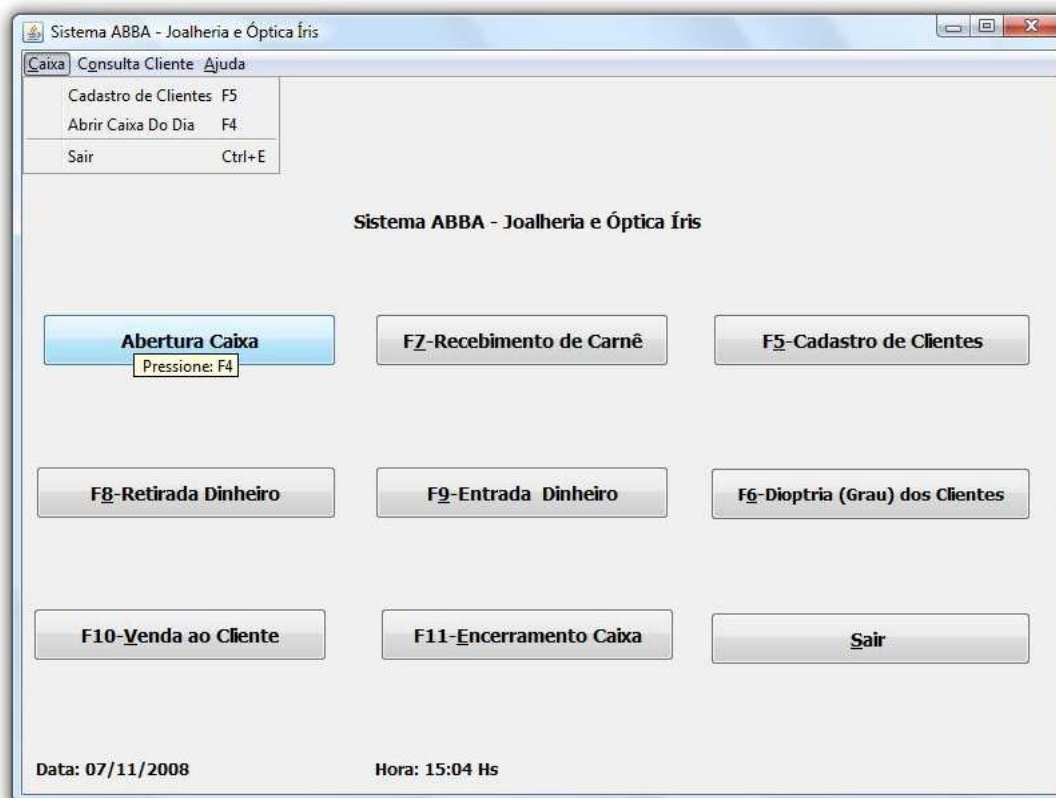


Figura 24 - Caixa do dia – Tela de operações principais do Caixa

Nas Fig. 25, 26, 27, 28 e 29 que referem-se ao Cadastro de Clientes, foram feitas mudanças no layout, visando deixá-los de uma forma similar aos documentos dos clientes, e seguindo uma ordem normal dos campos de preenchimento. Foram adicionados ainda campos de seleção no lugar de preenchimento, devido a se ter um ganho na realização da tarefa e descartando possíveis erros de digitação.

Outros pontos onde ocorreram mudanças foram no cadastro do cônjuge, onde se tem os mesmos dados para preenchimento como os do cliente principal. E também a tela de ajuda, a qual tem um comentário para todas as janelas do Cadastro de Clientes.

Foram retirados campos que não tinham utilidade para a empresa e adicionados botões para retorno e avanço das janelas internas, a fim de auxiliar usuários com pouca experiência.

Sistema ABBA - Joalheria e Óptica Íris

Cadastro de Clientes

Dados Pessoais | Endereço | Dados Profissionais | Referências | Obs.

Nome

RG CPF Data de Nascimento Naturalidade Sexo

Pelotas Feminino Masculino

Filiação

Pai

Mãe

Estado Civil

Retornar Avançar

<< >>

Salvar Limpar Tela Ajuda Sair

Figura 25 - Cadastro de Clientes – Dados Pessoais

Sistema ABBA - Joalheria e Óptica Íris

Cadastro de Clientes

Dados Pessoais | **Conjuge** | Endereço | Dados Profissionais | Referências | Obs.

Nome

RG CPF Data de Nascimento Naturalidade Sexo

 Pelotas Feminino
 Masculino

Filiação

Pai

Mãe

Retornar Avançar

Figura 26 - Cadastro de Clientes – Cônjuge

Sistema ABBA - Joalheria e Óptica Íris

Cadastro de Clientes

Dados Pessoais | Endereço | Dados Profissionais | Referências | Obs.

Endereço Nº

Complemento (Apartamento , Bloco , Informações Adicionais)

Bairro

CEP Estado Cidade

 Rio Grande do sul Pelotas

Telefone Residencial Telefone Celular Contato

Retornar Avançar

Figura 27 - Cadastro de Clientes – Endereço

Sistema ABBA - Joalheria e Óptica Íris

Cadastro de Clientes

Dados Pessoais | **Endereço** | Dados Profissionais | Referências | Obs.

Empresa

Endereço
 Nº

Telefone Comercial

Cargo

Renda

Retornar << Avançar >>

Salvar Limpar Tela Ajuda Sair

Figura 28 - Cadastro de Clientes – Dados Profissionais

Sistema ABBA - Joalheria e Óptica Íris

Cadastro de Clientes

Dados Pessoais | Endereço | Dados Profissionais | **Referências** | Obs.

Referência Pessoal <input type="text"/>	Telefone <input type="text"/>
Referência Pessoal <input type="text"/>	Telefone <input type="text"/>
Referência Comercial <input type="text"/>	Telefone <input type="text"/>
Referência Comercial <input type="text"/>	Telefone <input type="text"/>

Retornar << Avançar >>

Salvar Limpar Tela Ajuda Sair

Figura 29 - Cadastro de Clientes – Referências

Na janela de cadastro de Dioptria (Grau) dos Clientes como mostra a Fig. 30, foram extraídos vários campos que já haviam sido mencionados na Fig. 9 sem utilidade para uma possível pesquisa.

Observou-se que com a utilização de campos de preenchimento, muitas vezes informações eram escritas em vários formatos distintos, isto dificultava a utilização das informações, já que, nem sempre são os mesmos usuários que utilizam a Interface. Assim foi aplicada a utilização de campos de seleção, tendo-se informações mais consistentes e de fácil entendimento pelos usuários.

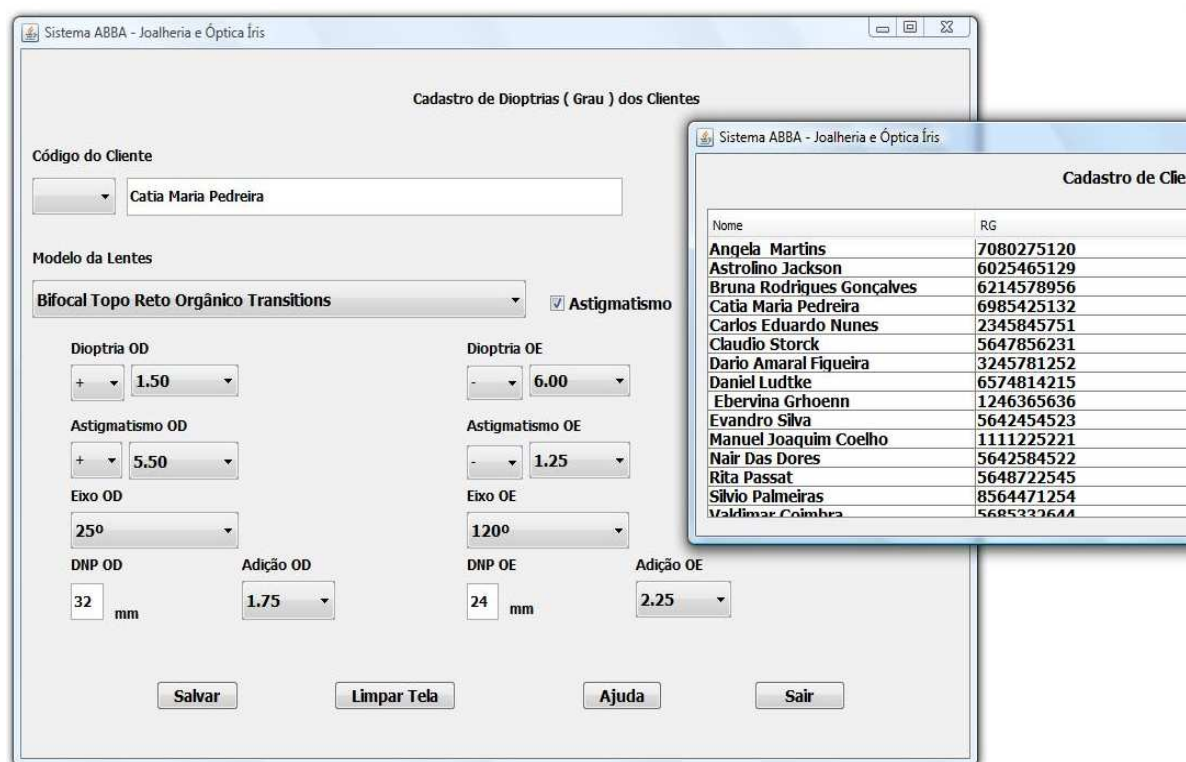


Figura 30 - Cadastro da Dioptria (grau) de Clientes

A principal mudança que ocorreu na janela de Venda ao Cliente, como é demonstrada na Fig. 31, foi à procura pelo Nome do Vendedor, que antes era realizada pelo código do mesmo, pois, os usuários não são obrigados a memorizar os códigos dos vendedores.

Foram alteradas também as formas de procura da Condição de Pagamento e do código da etiqueta do produto. Ainda foi interligada a função de Impressão do Carnê com a confirmação da gravação do Cadastro da Venda, evitando a ocorrência de vendas não salvas no sistema. Além da retirada de vários campos de seleção e preenchimento que não eram utilizados pela empresa.

Nome	RG
Caetano Swick	3047030251
Dario Amaral Figueira	3245781252
Daniel Ludtke	6574814215
Ebervina Grhoenn	1246365636
Evandro Silva	5642454523
Manuel Joaquim Coelho	1111225221
Nair Das Dores	5642584522
Rita Passat	5648722545

Figura 31 - Venda ao Cliente - Forma de pagamento

Na janela de Entrada de Dinheiro (antigo Suprimento Caixa), como mostra a Fig. 32, a principal mudança foi à introdução de um campo de especificação, ou seja, de onde provém o dinheiro que está entrando no caixa do dia. Além disso, foi introduzido um campo para seleção da espécie, que identifica qual o formato da moeda (dinheiro, cheque, cartão, etc.).

The image shows a software window titled "Sistema ABBA - Joalheria e Óptica Íris" with a sub-header "Entrada de Dinheiro". The form contains the following elements:

- Especificação - Origem:** A large empty text input field.
- Espécie:** A dropdown menu currently displaying "Dinheiro".
- Valor:** A smaller empty text input field.
- Buttons:** Four buttons are located at the bottom: "Salvar", "Limpar Tela", "Ajuda", and "Sair".

Figura 32 - Entrada de Dinheiro

O Recebimento de Carnê que é demonstrado na Fig. 33, recebeu alterações na retirada de campos que não eram utilizados, também na forma de recebimento (podendo ser feito tanto pelo número do carnê como também pelo nome do cliente). Houve ainda introdução da opção Ajuda e alteração da tecla sair, a qual não estava executando a sua ação de acordo com a função.

Recebimento Carnê

Número do Carnê **Nome do Cliente**

2002 Astrofino Jackson

Parcela a Pagar

4

Juros Dia

Multa Atrazo

Desconto

Parcela	Valor	Vencimento	Data do Paga...	Valor Pago	Saldo
1/5	50,00	10/10/2008	08/10/2008	49,50	
2/5	50,00	10/11/2008	12/11/2008	50,00	
3/5	50,00	10/12/2008	02/12/2008	49,50	
4/5	50,00	10/01/2009			
5/5	50,00	10/02/2009			

Figura 33 - Recebimento Carnê

Na janela de Retirada de Dinheiro, como mostra a Fig. 34, foi introduzido um campo para seleção da espécie, que identifica qual o formato da moeda (dinheiro, cheque, cartão, etc.) que está sendo retirada do caixa.

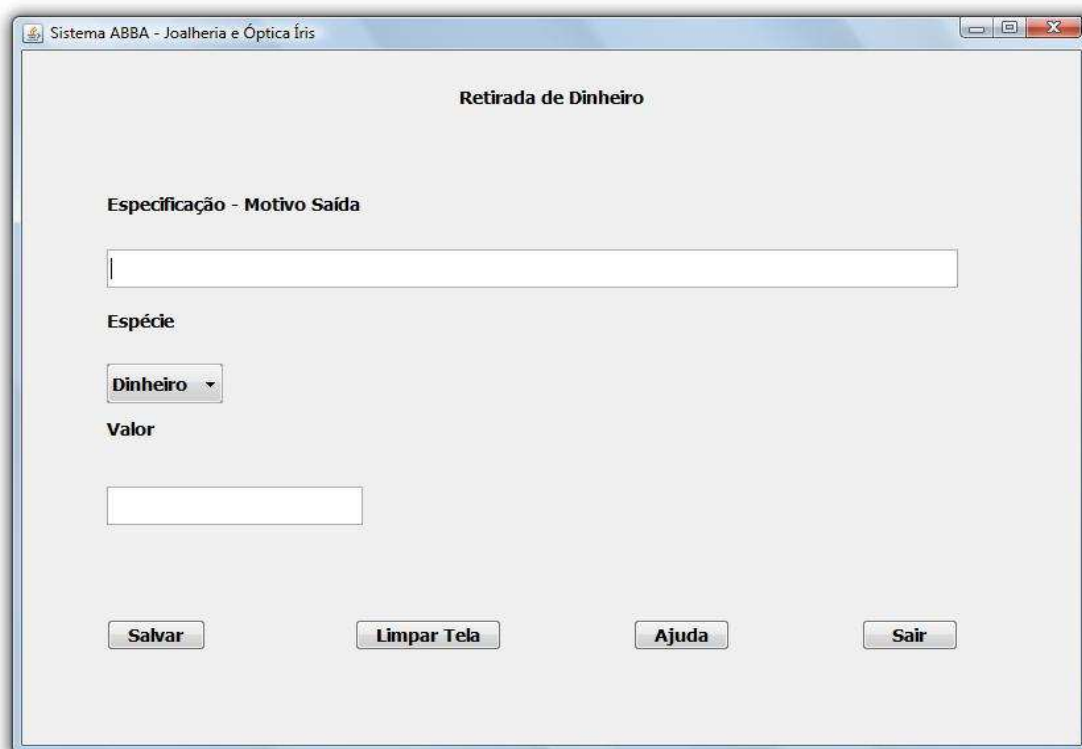


Figura 34 - Retirada de Dinheiro

As Fig. 35, 36 e 37 são em parte mudanças realizadas para ter-se maiores informações do que realmente está sendo realizado quando é clicado em algum ícone. Além disso, tem a demonstração de uma janela de Ajuda a qual foi realizada para todas as janelas.

Também foram criadas janelas informativas, caso tenha ocorrido algum problema ou durante a conclusão de alguma tarefa. Esta área foi desenvolvida com o objetivo de deixar o usuário sempre informado sobre o que o sistema está processando, para se ter uma melhora durante o processo interativo.

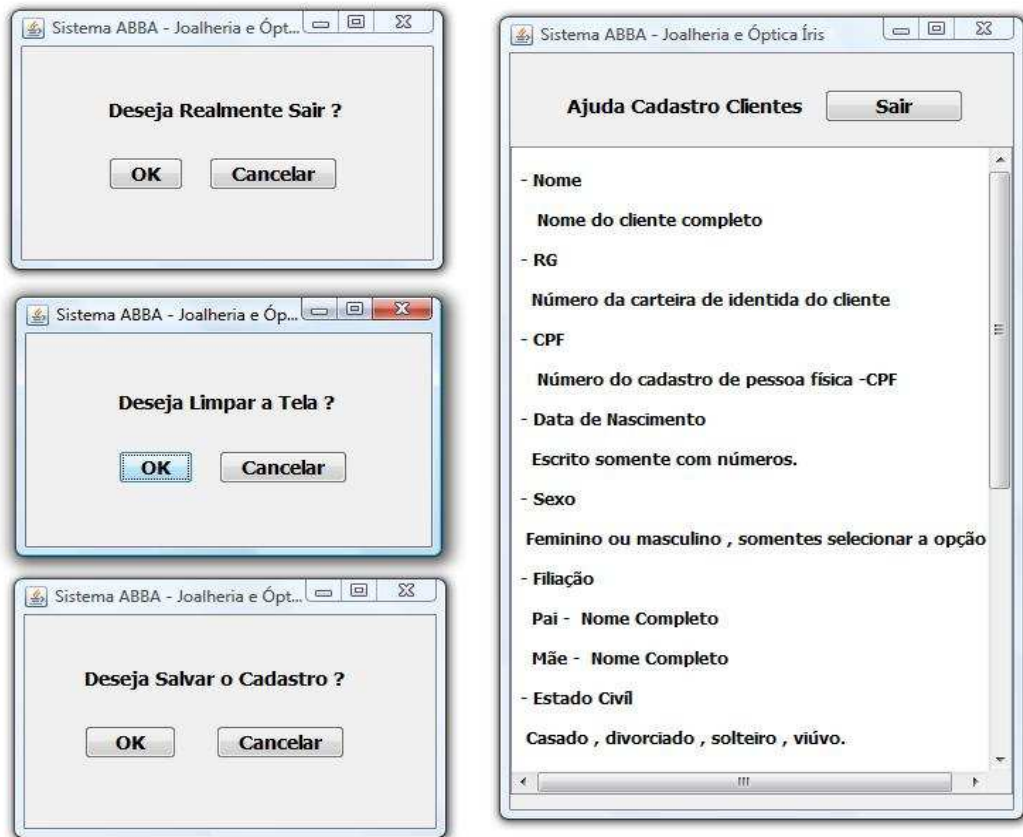


Figura 35 - Janelas de confirmação e Ajuda

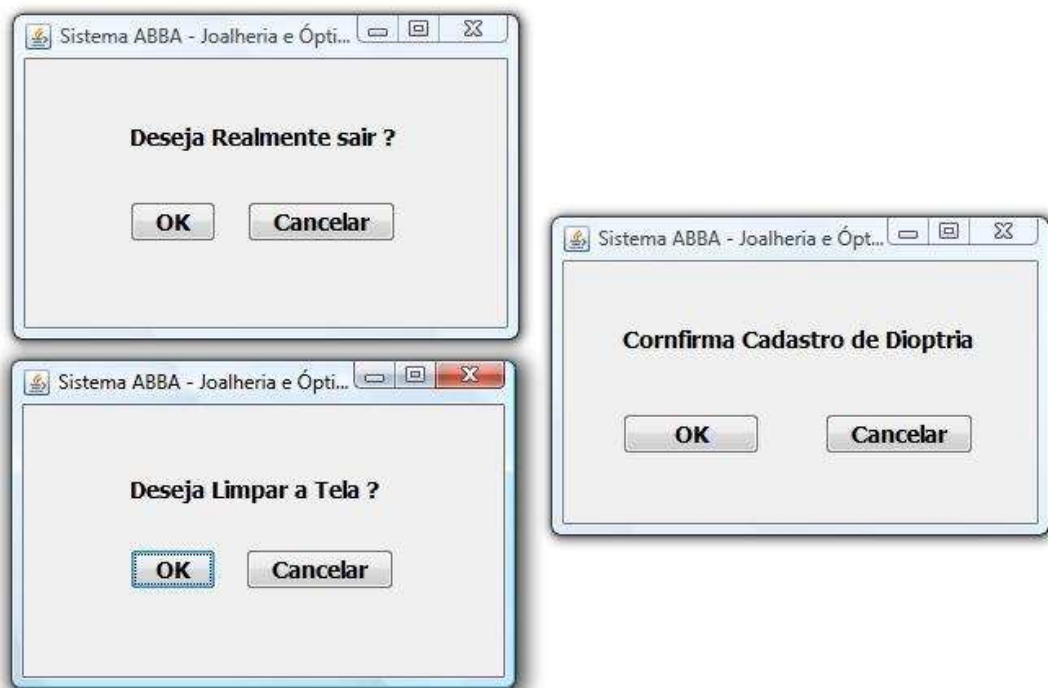


Figura 36 - Janelas de confirmação

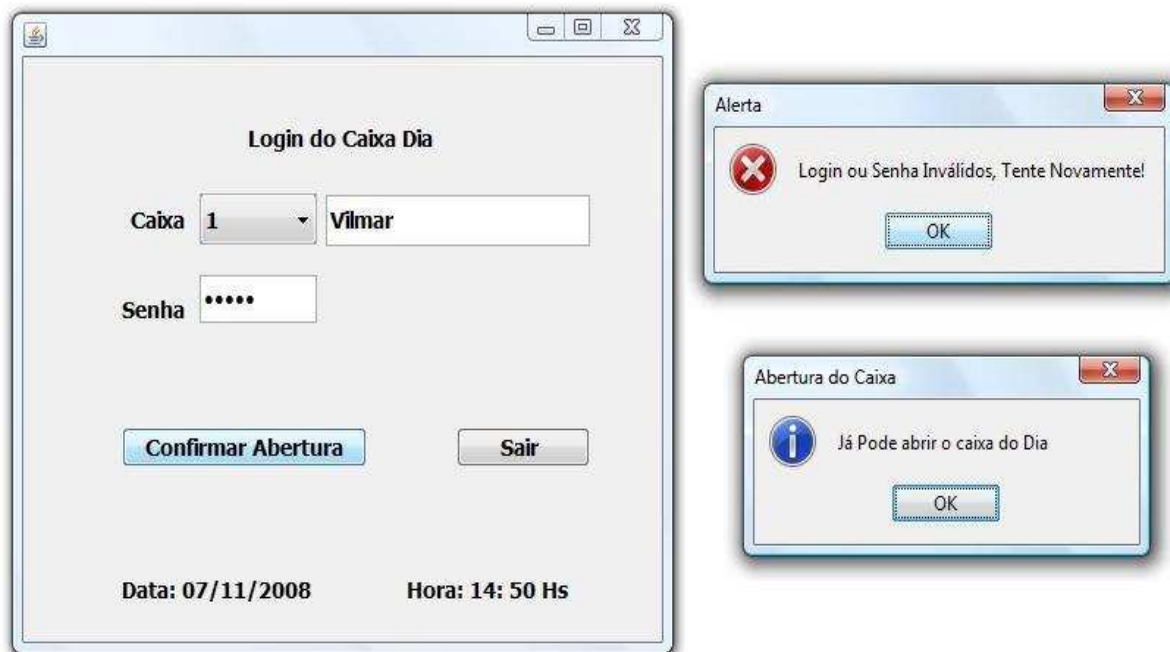


Figura 37 - Janela de Login do Caixa Dia e Janelas Informativas

6.4 Resultados Obtidos com a nova avaliação sobre a interface modificada

Utilizando os mesmos padrões de avaliação dos capítulos 4.1 e 6.1 sobre a nova interface, foram diagnosticados problemas. Estes foram resolvidos após a avaliação, feita entre um usuário e outro, tendo com isso um ganho na qualidade da interface e diminuindo o tempo gasto pelo usuário para diagnosticar os problemas. Além disso, também já foram realizadas as modificações como pode ser visto nas Fig.: 24, 27 e 30, onde encontravam-se os problemas, mencionados abaixo.

Será identificado o problema e logo após, será descrito a solução, para uma melhor identificação dos temas em questão.

1. Visibilidade do Status do Sistema

- Problema

Gravidade – Simples por um usuário.

Ao abrir a janela Caixa do Dia, o usuário encontra certa dificuldade em descobrir que deve fazer primeiro Login no Caixa na barra de ferramentas, ou clicar no ícone Abertura Caixa.

Solução

Aplicou-se no ícone Abertura Caixa um campo de texto informativo, que torna-se visível na passagem ou no clique do mouse sobre o ícone correspondente, como mostra a Fig. 24. Ainda foram adicionadas mais explicações sobre a Abertura do Caixa do Dia na janela de Ajuda.

3. Compatibilidade entre o sistema e o mundo real

- Problema 1

Gravidade – Sem importância para um usuário.

O botão Dioptria dos Clientes, para quem não tem conhecimento do ramo óptico é de difícil interpretação.

Solução

Modificou-se o nome do ícone para Dioptria (Grau) dos Clientes, já que, a palavra Grau é utilizada no lugar de Dioptria na linguagem popular. Como mostra a Fig. 24.

- Problema 2

Gravidade – Cosmética para um usuário.

Na janela de Cadastro de Clientes na parte do Endereço a linha do complemento deixa o usuário em dúvida sobre o que escrever na mesma.

Solução

Foram colocados ao lado, em observação e entre parênteses, quais os dados que devem ser descritos no campo, deixando mais concisa a informação passada ao usuário. Como mostra a Fig. 27.

8. Projeto minimalista e estético

- Problema 1

Gravidade – simples para um usuário.

Não há necessidade de ter o ícone sobre Ametropias na Janela de Dioptria (Grau) dos Clientes, já que, o usuário que trabalha no ramo deve ter conhecimento do mesmo.

Solução

Foi retirada a opção de seleção de Ametropias, ficando assim mais clara a informação, sendo passada somente a Dioptria do Cliente. Como mostra a Fig. 30.

Através dos resultados obtidos com as avaliações, somado-se estas as mudanças realizadas na interface (presentes no protótipo), tem-se dados suficientes para realizar uma comparação entre as duas interfaces. Visando ter-se uma idéia mais sólida, sobre as diferenças proporcionadas decorrentes das modificações realizadas na interface.

6.5 Análise de tempo em virtude da utilização da antiga (ABBA) e nova IU.

Nesta seção será realizada uma comparação de tempo utilizado tanto na antiga quanto na nova IU, denominada como ABBA, a fim de, analisar se houve alguma melhora na produtividade durante a interação do usuário com a interface em questão.

6.6 Tabelas de comparação de tempos entre a antiga (ABBA) e nova IU.

Foram feitas avaliações de tempo (utilizando cronômetro) com 4 usuários, dos quais 3 tinham contato diário com a interface enquanto que o outro passou a ter contato com a referida interface somente a primeira seção de avaliação. Foram analisados os tempos para se fazer o Cadastro de Clientes, Venda ao Cliente e Dioptria (grau) de Clientes, já que, as demais janelas não demandam tempo mensurável para serem preenchidas.

Deve-se levar em conta uma margem de tolerância de 5% tanto para mais como para menos, devido ao nível de treinamento e conhecimentos gerais distintos. Obteve-se esse percentual devido a realização de 3 testes com cada usuário com cadastros distintos.

Tabela 1 - Tempo gasto para utilização das Interfaces.

Usuário 1, experiência com a Interface antiga: 8 meses

Janelas	Interface Antiga		Protótipo da Interface Nova	
	Tempo (segundos)	Observação	Tempo (segundos)	Observação
Cadastro de Clientes	245		250	
Venda ao Cliente	110		90	
Dioptria (grau) dos Clientes	75		70	Ficou em dúvida quanto ao nome da Janela

Tabela 2- Tempo gasto para utilização das Interfaces.

Usuário 2, experiência com a Interface antiga: Nenhuma

Janelas	Interface Antiga		Protótipo da Interface Nova	
	Tempo (segundos)	Observação	Tempo (segundos)	Observação
Cadastro de Clientes	500	Somente com ajuda.	240	
Venda ao Cliente	450	Somente com ajuda.	120	
Dioptria (grau) dos Clientes	180		185	

Tabela 3- Tempo gasto para utilização das Interfaces.

Usuário 3, experiência com a Interface antiga: 6 anos

Janelas	Interface Antiga		Protótipo da Interface Nova	
	Tempo (segundos)	Observação	Tempo (segundos)	Observação
Cadastro de Clientes	240		220	
Venda ao Cliente	100		75	
Dioptria (grau) dos Clientes	85		40	

Tabela 4- Tempo gasto para utilização das Interfaces.

Usuário 4, experiência com a Interface antiga: 1 ano

Janelas	Interface Antiga		Protótipo da Interface Nova	
	Tempo (segundos)	Observação	Tempo (segundos)	Observação
Cadastro de Clientes	260		240	
Venda ao Cliente	140		95	
Dioptria (grau) dos Clientes	110	Encontrou dificuldades	85	

6.7 Considerações finais sobre os tempos

Analisando-se os tempos encontrados, pode-se observar que houve algumas diferenças, da antiga para nova IU. Mas em alguns casos não se teve alterações significativas, devido à inserção de um número maior de campos, os quais não constavam na antiga Interface e que são de fundamental importância para uma melhora da performance do usuário durante a interação.

Outro ponto fundamental a ser observado é o caso do quarto usuário, o qual nunca havia utilizado a Interface. O mesmo não conseguiu realizar algumas pesquisas ficando estagnado em um ponto e precisando de auxílio de outro usuário para completar as tarefas.

Constatou-se que com a utilização dos métodos no desenvolvimento da interface além das vantagens proporcionadas aos projetistas, no caso de uma possível manutenção da interface, também há influência direta da mesma sobre os usuários, os quais apresentaram uma melhora no manuseio da IU. Ocasionalmente um ganho na produtividade e diminuição de tempo para realização das atividades.

7. Conclusão

Neste trabalho realizou-se um estudo sobre os diversos tipos de métodos para a avaliação e construção de uma IU. Para tanto, apresentou-se uma visão geral de como é realizado a avaliação sobre uma interface pronta ou que esteja em construção, como também foi demonstrada a modelagem de uma nova IU, através de um formalismo.

Os métodos abordados foram os mesmos usados pela maioria dos projetistas, entre eles estão: os métodos não Formais e os formais. A partir do estudo feito sobre os métodos não Formais foi escolhido o método de avaliação Heurístico, já que esse apresenta os problemas de usabilidade da IU, com uma porcentagem bem significativa em relação os demais métodos mencionados. No caso dos métodos formais, foi proposta a utilização do método orientado a modelo, uma vez que esse mostra-se mais apropriado para a modelagem de IU.

Dentre os métodos selecionados, foram abordados, especificamente, Avaliação Heurística de Nielsen e o formalismo StateChart, os quais foram fundamentais para a conclusão desse trabalho.

Durante o trabalho observou-se que a avaliação Heurística de Nielsen era a melhor opção para a avaliação da Interface atual, devido à facilidade da sua aplicação e do alto índice de descobertas de problemas de usabilidade nas IU. Já o formalismo StateChart, apresenta uma forma de decompor o projeto em subestados, criando também uma estrutura hierárquica, que faz o número de setas de transição ficar mais reduzido, dando mais clareza a toda a modelagem. Com a aplicação desses métodos na interface obteve-se uma melhor apresentação de como proceder na execução de tarefas numa futura avaliação da IU. Com a modelagem

utilizada pertence- se ainda que os projetistas possam encontrar facilmente os problemas da IU e fazer as respectivas modificações.

Constatou- se ainda nesse trabalho, que os referidos métodos, se forem perfeitamente selecionados de acordo com suas peculiaridades e aplicabilidades, só tendem a trazer vantagens na utilização dos mesmos. Essas vantagens poderão ser percebidas, durante a especificação e, principalmente, na fase de manutenção, onde o tempo e o custo são as maiores dificuldades enfrentadas pelos projetistas. Com a utilização dos métodos nesse trabalho, pode-se observar que houve uma diminuição desses problemas mencionados anteriormente.

7.1 Trabalhos Futuros

Espera-se que, como sugestões de trabalhos futuros, sejam feitos mais estudos tanto sobre os métodos não Formais como os formais, para que de posse desses conhecimentos sejam divulgados os benefícios que acarretam na sua utilização, tentando assim mudar a visão e a cultura dos projetistas, a fim de implantar a utilização na especificação de interfaces com os Usuários em todos os segmentos.

8. Bibliografia

ANDERSON, David J.O'BYRNE, Brian. Lean Interaction Design and Implementation: Using StateCharts with Feature Driven Development. Proceedings of Use 2003. 2003.

BRANDÃO, Eduardo. Métodos Para Avaliação De Usabilidade De Interfaces Gráficas Digitais. 2005. P.166 - 192.

CABRERA, M., GEA, M, GUTIERREZ,F.,TORRES,J.C. Algebraic Specification Of User Interfaces. In: STEPHANIDIS, Constantine, AT and HCI LAB, ICS-FORTH (ed). 1. ERCIM WORKSHOP ON USER INTERFAC FOR ALL, 1995, Greece.

CARVALHO, R. L., OLIVEIRA, C. M. Modelos de Computação e Sistemas Formais. Rio de Janeiro: DCC/IM,COPPE/ sistemas, NCE/UFRJ, 1998.p 276.

COHEN, Bernard, HARWOOD, W. T., JACKSON, M. I. The Specification of complex Sytems, Wokingham-England: ADDISON-WESLEY Publishing Company, 1986. p.143.

DINIZ, Eliane S.A.; PIMENTA, M.S. Formalismos Para Projeto de Interfaces Baseados em Tarefas. In: WMF-WORKSHOP DE MÉTODOS FORMAIS, 2000, João Pessoa/PB. SBES' 2000, - WORKSHOPS ANAIS. João Pessoa/PB: SBC, 2000, p. 80-91.

DINIZ, Eliane da S. A. Ferramenta para Avaliação de Interface em Ambiente Windows, a partir da Monitoração de Dados – FAIWIN. Campina Grande – PB: Universidade Federal da Paraíba,1996. p.1-29. (Dissertação, Mestrado em Informática).

DIX, Alan J.: Formal Methods an introduction to and overview of the use of formal methods within HCI. 'Chapter 2 Formal Methods' in Perspectives on HCI:

Diverse Approaches, Eds. A. Morak and N. Gilbert. London, Academic Press. 1995. p.9-43.

FISCHER, G. From consumers to Designers. IEEE Computer Society. 1998. p.2-9.

HALL A. Seven myths of formal methods. In: IEEE Software, September 1990, p. 11-19.

HARTSON, H.R. Human-Computer Interaction: Interdisciplinary roots and trends., 1998. p.103-118.

JACOB, Robert J. K. Using Formal Specification in the Design of a Human-Computer Interface. Naval Research Laboratory, 1983.

KARAT, J. The cost-benefit and business case analysis of usability engineering. 1993.

KNEUPER, Ralf, Limits of Formal Methods. In: FORMAL ASPECTS OF COMPUTING – The International Journal of Formal Methods, UK, v. 9, n.4, p.379-394, July-Aug. 1997.

MOLICH, R.; Heuristic evaluation of user interfaces, Seattle, p249-256, 1990.

MORAN, T. A representation for the user interface of interactive computer systems. Academic Press. 1981. p.3-50.

MUNHOZ, Felipe S. O Uso de Formalismo na Especificação de um Documento para Web. 2004. p.22-30.

NIELSEN, Jakob. Usability Engineering, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1993. p.10-23.

PALANQUE, Philippe, BASTIDE, R., DOURTE, L. ET AL.. Design of User-Driven Interfaces Using and Objects. In: CONFERENCE ON ADVANCE INFORMATION SYSTEM ENGINEERING, 1993. June.

PRATES, Raquel O.; BARBOSA, Simone D.J. et al. Avaliação de Interfaces de Usuário – Conceitos e Métodos , 2003. p.49.

PREECE, J.; ROGERS,Y.; SHARP, H. (2002) Interaction Design: Beyond Human-computer Interaction. New York, NY: John Wiley & Sons. 2002.

PREECE, I.; ROGERS, Y.; SHARP, E.; BEYON, D.; HOLLAND, S.; CAREY, T. Human-Computer Interaction. England: Addison-Wesley, 1994.

PRESSMAN, Roger S. "Engenharia de Software", Makron Books, 1995.

RAMOS, Ricardo A. Treinamento Prático em UML. 1º edição. São Paulo – SP. Universo dos Livros Editora Ltda., 2006. p.45-127.

SHNEIDERMAN, Ben. Designing the User Interface Strategies for Effective Human-Computer Interaction. New York. Addison-Wesley, 1987. p.448.

SHNEIDERMAN, B.; Designing the User Interface: Strategies for Effective Human-Computer Interaction. 3 ed. USA: Addison-Wesley Longman Inc., 1998. p.640.

SCOTT W. Ambler. Building Object Applications That Work. User Interface Design: Tips and Techniques, SIGS Books/Cambridge University Press. AmbySoft Inc. 1998. p.23-25.

SOMMERVILLE, I.. Software Engineering, 5th edn, Addison-Wesley Publishing Company, Wokingham England. 1996.

SPOLSKY, Joel. User Interface Design for Programmers, 2001.

SUN, Chenliang. Statecharts Based GUI Design. School of Computer Science, McGill University. 2003.

THIIMBLEY, Harold. User Interface Design. New York: ACM Press, 1991. p.470.

VERA, A. S. Castro. Especificação Formal: Uma Abordagem Algébrica. Porto Alegre – RS: UFRGS II- Instituto de Informática, dezembro 1988. p.82. (Trabalho Individual I).

VERA, A. S. Castro. Métodos e Linguagens de especificação Formal. Porto Alegre – RS: UFRGS II- Instituto de Informática, JUNHO 1992. (Exame de Qualificação III).

WINOGRAD T. Environments for software design. In: Communications of the ACM, v.38, n.6, 1995, p.65-74.

Apêndice

Avaliação Heurística de uma Interface específica de uma empresa do ramo Óptico, visando usabilidade e acessibilidade

Dez Heurísticas de Nielsen [NIE 94]

Descrição da avaliação

Avaliação heurística, definida por Nielsen e Molich [NIE 94] [NIE 94], é um método de avaliação de usabilidade onde um avaliador procura problemas de usabilidade numa interface com o usuário através da análise e interpretação de um conjunto de princípios ou heurísticas. Este método de avaliação é baseado no julgamento do avaliador e normalmente, descobre 75% dos problemas de usabilidade.

1. Primeiramente, leia e analise as dez heurísticas (ver Tabela 1).

Tabela 5 - Conjunto de heurísticas de Nielsen (1994)

Heurísticas	Descrição
Visibilidade do status do sistema:	O sistema deve sempre manter os usuários informados sobre o que está acontecendo através de feedback apropriado, em um tempo razoável.
Compatibilidade entre sistema e mundo real:	O sistema deve utilizar a linguagem do usuário, com palavras, frases e conceitos familiares para ele, ao invés de termos específicos de sistemas. Seguir convenções do

	<p>mundo real, fazendo com que a informação apareça em uma ordem lógica e natural.</p>
<p>Controle e liberdade para o usuário:</p>	<p>Estão relacionados à situação em que os usuários freqüentemente escolhem as funções do sistema por engano e então necessitam de "uma saída de emergência" claramente definida para sair do estado não desejado sem ter que percorrer um longo diálogo, ou seja, é necessário suporte a <i>undo e redo</i>.</p>
<p>Consistência e padrões:</p>	<p>Referem-se ao fato de que os usuários não deveriam ter acesso a diferentes situações, palavras ou ações representando a mesma coisa. A interface deve ter convenções não-ambíguas.</p>
<p>Prevenção de erros:</p>	<p>Os erros são as principais fontes de frustração, ineficiência e ineficácia durante a utilização do sistema.</p>
<p>Reconhecimento em lugar de lembrança:</p>	<p>Tornar objetos, ações, opções visíveis e coerentes. O usuário não deve ter que lembrar informações de uma parte do diálogo para outra. Instruções para o uso do sistema devem estar visíveis ou facilmente acessíveis.</p>
<p>Flexibilidade e eficiência de uso:</p>	<p>A ineficiência nas tarefas pode reduzir a eficácia do usuário e causar-lhes frustração. O sistema deve ser adequado tanto para usuários inexperientes quanto para usuários experientes.</p>
<p>Projeto minimalista e estético:</p>	<p>Os diálogos não devem conter informações irrelevantes ou raramente necessárias. Cada unidade extra de informação em um diálogo compete com unidades relevantes e diminui sua visibilidade relativa.</p>
<p>Auxiliar os usuários a reconhecer, diagnosticar e</p>	<p>Mensagens de erro devem ser expressas em linguagem natural (sem códigos), indicando precisamente o erro e</p>

recuperar erros:	sugerindo uma solução.
Ajuda e documentação:	Mesmo que seja melhor que o sistema possa ser usado sem documentação, pode ser necessário fornecer ajuda e documentação. Tais informações devem ser fáceis de encontrar, ser centradas na tarefa do usuário, listar passos concretos a serem seguidos e não ser muito grandes. A ajuda deve estar facilmente acessível e on-line.

2. A seguir, utilize a Interface da empresa, também denominada de ABBA livremente, procurando possíveis problemas de usabilidade.

3. Quando um problema qualquer for detectado, classifique-o em uma das dez heurísticas de Nielsen, anotando o problema na tabela correspondente e atribuindo o **grau de severidade** (0 até 4) para este problema (dado pela tabela 2) e recomece novamente até não encontrar mais problemas de usabilidade.

Tabela 6 - Grau de severidade dos problemas de usabilidade

Grau de severidade	Tipo	Descrição
0	Sem importância	Não afeta a operação da interface
1	Cosmético	Não há necessidade imediata de solução
2	Simple	Problema de baixa prioridade (<u>pode</u> ser reparado)
3	Grave	Problema de alta prioridade (<u>deve</u> ser reparado)
4	Catastrófico	Muito grave, deve ser reparado de qualquer forma.

Heurística de Nielsen

Avaliador:

Tabela 7 - Heurísticas de Nielsen - Sessão de avaliação

<i>1. Visibilidade do status do sistema</i>	
Verificação: Os usuários são mantidos informados sobre o progresso do sistema com apropriado <i>feedback</i> (informações sobre o processo) em um tempo razoável?	Grau de severidade
Problema 1:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3

	() Catastrófico – 4
Problema 2:	() Sem importância - 0 () Cosmético - 1 () Simples - 2 () Grave - 3 () Catastrófico – 4
Problema 3:	() Sem importância - 0 () Cosmético - 1 () Simples - 2 () Grave - 3 () Catastrófico – 4
Problema 4:	() Sem importância - 0 () Cosmético - 1 () Simples - 2 () Grave - 3 () Catastrófico – 4
<i>2. Compatibilidade entre o sistema e o mundo real</i>	
Verificação: O sistema utiliza conceitos e linguagem familiar com o usuário em vez de termos orientados ao sistema? O sistema utiliza convenções do mundo real, exibindo informações com uma ordem lógica e natural?	Grau de severidade

Problema:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 2:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 2:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 3:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 4:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4

<i>3. Liberdade e controle do usuário</i>	
Verificação: Os usuários podem fazer o que querem quando querem?	Grau de severidade
Problema 1:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 2:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 3:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 4:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4

4. Consistência e padrões	
Verificação: O projeto de elementos como objetos e ações tem o mesmo significado ou efeito em diferentes situações?	Grau de severidade
Problema 1:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico - 4
Problema 2:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico - 4
Problema 3:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico - 4
Problema 4:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico - 4

<i>5. Prevenção contra erros</i>	
Verificação: Os usuários podem cometer erros dos quais bons projetos poderiam prevenir?	Grau de severidade
Problema 1:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico - 4
Problema 2:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico - 4
Problema 3:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico - 4
Problema 4:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico - 4

6. Reconhecimento em lugar de lembrança	
Verificação: Os elementos de projeto como objetos, ações e opções são possíveis? O usuário é forçado a relembrar informações de uma parte do sistema para outra?	Grau de severidade
Problema 1:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 2:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 3:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 4:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3

	() Catastrófico – 4
<i>7. Flexibilidade e eficiência de uso</i>	
<p>Verificação:</p> <p>As tarefas de usuário são eficientes e podem se adaptar ao gosto do usuário em suas ações mais freqüentes ou ele utiliza atalhos?</p>	Grau de severidade
Problema 1:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 2:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 3:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 4:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2

	<input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
8. Projeto minimalista e estético	
Verificação: Os diálogos contém informações irrelevantes ou raramente necessárias?	Grau de severidade
Problema 1:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 2:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 3:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 4:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2

	<input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
9. Auxiliar os usuários a reconhecer, diagnosticar e recuperar-se de erros	
Verificação: As mensagens de erro são expressas em linguagem simples (sem códigos) descrevendo exatamente o problema e sugerindo uma solução?	Grau de severidade
Problema 1:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 2:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 3:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
Problema 4:	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1

	<input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
<i>10. Ajuda e documentação</i>	
<p style="text-align: center;">Verificação:</p> <p>São fornecidas apropriadas informações de ajuda, e estas informações são fáceis de procurar e de focalizar nas tarefas do usuário?</p>	Grau de severidade
<p style="text-align: center;">Problema 1:</p>	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
<p style="text-align: center;">Problema 2:</p>	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
<p style="text-align: center;">Problema 3:</p>	<input type="checkbox"/> Sem importância - 0 <input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
<p style="text-align: center;">Problema 4:</p>	<input type="checkbox"/> Sem importância - 0

	<input type="checkbox"/> Cosmético - 1 <input type="checkbox"/> Simples - 2 <input type="checkbox"/> Grave - 3 <input type="checkbox"/> Catastrófico – 4
--	---