

Avaliação de Interfaces para Aplicativos em Sistemas Android

Julie Munhoz Barbosa¹, Iara Carnevale de Almeida¹

¹ Centro Universitário Franciscano

Caixa Postal – 91.501-970 – Santa Maria – RS – Brasil

lilasm8@gmail.com, iara.almeida@unifra.br

***Abstract.** In this sense, Google has launched the Google Android operating system, one of the most consolidated and adopted for mobile systems, having advantages as multitasking, open source, and is compatible with various devices. However, despite the large amount of applications available for Android, some significant gaps in its interface, such usability questions. This study aims to identify how an application for Android system should be developed following usability criteria.*

***Resumo.** A empresa Google lançou o sistema Android, sendo um dos sistemas mais consolidados e adotados para dispositivos móveis, possuindo vantagens como ser multitarefa, ter código aberto e além de ser compatível com diversos dispositivos. Porém, apesar da grande quantidade de aplicações disponíveis para Android, algumas apresentam deficiências na sua interface, em quesitos de usabilidade. Assim, este trabalho tem o objetivo de identificar como uma aplicação para sistema Android deve ser desenvolvida seguindo critérios de usabilidade.*

1. Introdução

A usabilidade é um atributo principal da qualidade de qualquer produto interativo. Usabilidade baseada em dispositivos móveis é algo essencial e deve ser considerado quando do lançamento de um novo produto, que poderia ser um diferencial em um mercado ágil, como é o dos dispositivos móveis hoje em dia [Nielsen 2003].

Segundo Nielsen (1993), usabilidade é a medida pela qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficiência, eficácia e satisfação em um contexto de uso específicos. Portanto, duas expressões normalmente usadas, no teste de usabilidade: usabilidade e experiência do usuário. A usabilidade está associada a cinco atributos, os quais são: **Facilidade de aprendizado**: a utilização do sistema requer pouco treinamento, o sistema deve ser fácil de aprender de forma que o usuário possa começar a utilizá-lo rapidamente; **Fácil de Memorizar**: o usuário deve lembrar como utilizar a interface depois de algum tempo; **Maximizar a produtividade**: a interface deve permitir que o usuário realize a tarefa de forma rápida e eficiente; **Minimizar a taxa de erros**: caso aconteçam erros, a interface deve avisar o usuário e permitir a correção de modo fácil; **Maximizar a satisfação do usuário**: a interface deve dar-lhe confiança e segurança. Na sequência, Nielsen (1993) apresentou

os “dez princípios gerais do design de interface do usuário”¹: Visibilidade do status do sistema: o usuário deve ser informado sobre o que está acontecendo, através de *feedback* apropriado e em tempo razoável. **Relacionamento entre a interface do sistema e o mundo real**: devem ser utilizadas palavras, frases e conceitos familiares ao usuário, ao invés de termos orientados ao sistema. Devem ser seguidas as convenções do mundo real, tornando as informações que aparecem em uma ordem natural e lógica. **Liberdade e controle do usuário**: o usuário frequentemente escolhe algumas funções do sistema por engano e precisará de uma “saída de emergência” para sair daquele estado indesejado, sem ter que passar por um extenso diálogo. Deve haver apoio no desfazer e refazer. **Consistência e padrões**: o usuário não precisa adivinhar que diferentes palavras, situações ou ações tem o mesmo significado. Devem ser seguidas as convenções da plataforma. **Prevenção de erros**: melhor do que uma mensagem de erro é um projeto cuidadoso que impedirá que o erro ocorra. Devem ser eliminadas as condições passíveis de erros, apresentado sempre ao usuário uma opção de confirmação antes de se comprometerem com uma determinada ação. **Reconhecimento ao invés de lembrança**: minimizar a carga de memória do usuário tornando objetos, ações e opções visíveis. O usuário não deve ter que se lembrar da informação de uma parte do diálogo para outra. Instruções de uso do sistema devem estar visíveis e serem facilmente recuperáveis quando necessário. **Flexibilidade e eficiência de utilização**: disponibilizar aceleradores para o usuário experiente, mas que fiquem invisíveis para o usuário novato, assim o sistema pode atender ambos os perfis de usuário. Permitir ao usuário personalizar suas ações frequentes. **Estética e design minimalista**: os diálogos não devem conter informações irrelevantes ou raramente necessárias. Cada unidade extra de informação num diálogo compete com as unidades relevantes de informação, diminuindo sua visibilidade relativa. **Ajude aos usuários a reconhecer, diagnosticar e resolver erros**: mensagens de erros devem ser expressas em linguagem clara (sem códigos), indicando com precisão o problema e sugerindo uma solução construtiva. **Ajuda e documentação**: mesmo que seja ideal um sistema ser usado sem documentação, deve ser fornecida uma documentação de apoio. Qualquer informação deve ser sucinta e fácil de ser pesquisada, com foco na atividade do usuário, com lista de passos concretos a serem realizados.

Segundo Pádua (2012), avaliações de usabilidade visam à verificação da qualidade da interface. Comparando-se os resultados obtidos com metas definidas conforme as heurísticas de Nielsen (1993), se consegue determinar se o produto possui um nível aceitável de qualidade, ou se será necessário uma nova interação de avaliação com base nas heurísticas.

O restante do texto está organizado da seguinte forma. A seção 2 refere-se a conceitos de usabilidade, eficiência e eficácia, e demais aspectos relevantes quanto à usabilidade; a seção 3 refere-se aos métodos de avaliação das interfaces, a seção 4 informa as etapas de avaliação de usabilidade, das heurísticas envolvidas para cada aplicativo e na seção 5 serão abordadas as propostas de Layouts para o aplicativo do WorkAssure, a seção 6 aborda o conjunto de checklist para o aplicativo MotoBoyOnline e na seção 7 as propostas de Layouts para o aplicativo

¹ Note que estes são chamados de heurísticas, pois estão mais na natureza de regras do que como diretrizes de usabilidade específicas.

MotoBoyOnLin, na seção 8 será feito a análise dos resultados dos aplicativos e finalizando a conclusão deste trabalho na seção 9.

2. Avaliação de Usabilidade

Nesta secção serão apresentadas alguns fundamentos e métodos que vêm sendo utilizados para avaliação de sistemas móveis [Oliveira 2010], para que seja avaliada a funcionalidade do sistema e o efeito da interface junto ao usuário, e identificando os problemas do sistema.

A usabilidade das aplicações é considerada um dos fatores mais importantes para a qualidade e a aquisição de produtos e, por consequência, no aumento da competitividade deste produto [Hitz 2006]. Várias abordagens têm sido empregadas para avaliar a usabilidade de aplicações no contexto móvel. Conforme Nielsen (1993), algumas têm foco nos testes (isto é, na atenção do usuário e no contexto de uso), enquanto outras tem foco na inspeção (ou seja, propõem técnicas executadas por especialistas).

Pode-se definir um método de avaliação de usabilidade como um procedimento do sistema, para registrar os dados relacionados à interação do usuário com um sistema ou produto, que serão então utilizados para determinar sua usabilidade [Ferreira 2013]. Unindo os métodos de usabilidade, podem-se elaborar testes e, por consequência, avaliar um dado sistema. Esses testes, segundo Nielsen (1993), consideram se o sistema é fácil de usar, de aprender, de memorizar, se é isento de erros e se causam satisfação ao usuário.

Para garantir o sucesso da avaliação de usabilidade é fundamental seguir alguns passos para o correto planejamento desta avaliação: estabelecer um objetivo, definir métricas, providenciar recursos, escolher as técnicas de avaliação e recrutar usuários.

O objetivo, para a avaliação de usabilidade, deve ser bem estabelecido para que as técnicas e métricas possam ser melhor definidas. Um teste de usabilidade pode ter vários objetivos, tais como: estabelecer uma cultura em usabilidade, a ser aplicada em futuros projetos da empresa; minimizar riscos de falhas de usabilidade, principalmente em aplicativos usados em “situações limite” (por exemplo, na aviação ou medicina); criar um diferencial do aplicativo para os produtos concorrentes; minimizar os custos de manutenção e suporte; aumentar as vendas e a fidelização de clientes.

Nos testes com usuários, diferentes métricas podem ser adotadas e utilizadas nas avaliações. Dentre as métricas, podemos destacar: tempo de execução de uma tarefa; número de erros cometidos durante cada tarefa; tempo em que o usuário fica sem executar nenhuma função no aplicativo; opiniões do usuário sobre o aplicativo.

Nielsen (1993) recomenda que os testes sejam feitos com cinco usuários, por que esse número apresenta a melhor relação entre o custo e o benefício.

3. Métodos de Avaliação Heurística

As avaliações de usabilidade visam à verificação da qualidade da interface, comparando-se os resultados obtidos com as metas definidas anteriormente, com o

objetivo de se definir se o produto está concluído, com um nível aceitável de qualidade, ou se será necessário uma nova iteração pelo fluxo de usabilidade [Padua 2012].

A avaliação heurística tem sido utilizada com frequência na avaliação de usabilidade de aplicações. Esta é uma técnica de inspeção e é realizada através de um conjunto de princípios e diretrizes (isto é, de heurísticas) onde os envolvidos são inspetores especialistas que direcionam seu foco em encontrar problemas a serem corrigidos e aplicar melhorias de usabilidade. Avaliações de Heurísticas foram desenvolvidas como uma alternativa para minimizar custos ao realizar inspeções de software [Nielsen 1993].

O procedimento de aplicação de uma inspeção é feito pelos inspetores, através da análise das interfaces, que verificam se estas estão de acordo com os “dez princípios gerais do design de interface do usuário, acima apresentados. Cada inspetor deve relatar problemas encontrados, categorizá-los conforme as heurísticas e, por fim, determinar o grau de severidade do defeito. Depois de toda execução da inspeção é realizada uma reunião de discriminação onde é feito a separação dos problemas em falso-positivo. Este procedimento é útil para avaliação em interfaces em desenvolvimento ou sistemas já completos. E, por utilizar heurísticas gerais, para avaliar interfaces, diminui o tempo de avaliação e a facilidade de aplicação da mesma. No entanto, a habilidade do inspetor para identificar os problemas de usabilidade depende de conhecimentos e experiência na área de avaliação.

A avaliação heurística é considerada um conjunto de regras, ou diretrizes, que são observadas para identificar possíveis problemas na interação humano-computador que provavelmente os usuários encontrarão. Esse tipo de avaliação é baseado tanto no conhecimento quanto na experiência de avaliadores especialistas, que analisando as interfaces de um determinado sistema, fazem o levantamento dos possíveis problemas e sugerem soluções [Pádua, 2012].

A avaliação heurística é um método de avaliação de interfaces baseado em princípios de usabilidade. A eficiência deste método consiste na capacidade do(s) avaliador (es) reconhecer(em) problemas de usabilidade. Qualquer pessoa pode ser treinada para aplicação deste método, embora melhores resultados tenham sido obtidos com avaliadores experientes [Nielsen, 1993]. Posteriormente, os resultados das avaliações dos avaliadores são organizados em um único relatório, tal que resultados iguais/similares são agrupados e depois categorizados em função da gravidade do problema. Segundo Nielsen (1993), três a cinco avaliadores são suficientes para identificar a maior parte dos problemas [Pádua, 2012].

A avaliação de usabilidade por inspeção com listas de conferência (*checklist*) é realizada por meio de vistorias através das quais profissionais diagnosticam rapidamente problemas gerais e repetitivos das interfaces [Bastien & Scapin, 2009] e [Cybis, Betiol, Faust, 2007]. Esses profissionais podem ser programadores, analistas, ou especialistas em usabilidade. Nesse tipo de técnica, a qualidade das listas determina as possibilidades de avaliação [Pádua 2012].

A avaliação com listas de conferência pode ser combinada com a avaliação heurística para se alcançar as vantagens das duas abordagens [Pádua, 2012]. Conforme Cybis (2000), nesse tipo de técnica, ao contrário da avaliação heurística, são as qualidades do *checklist*, e não dos avaliadores, que determinam a qualidade da avaliação.

4. Avaliação do *checklist* de usabilidade para o aplicativo WorkAssure

A empresa de telecomunicações Net com objetivo de maximizar a eficiência operacional e aumentar a satisfação do cliente implementou o sistema de baixa de ordem de serviço por dispositivo móvel que utilizam sistema operacional android denominado WorkAssure, desenvolvido pela empresa Accenture.

Nesta seção apresentamos a avaliação do aplicativo *WorkAssure*. Salientamos que as heurísticas e as descrições do aplicativo com base em um modelo de aplicação de *Checklist* (lista de verificação) foram feitas no TFGI. Este trabalho pretende trabalhar sobre a avaliação das heurísticas que não foram satisfeitas, de forma a apresentar uma proposta de melhoramento.

Nesta seção serão abordadas as etapas de avaliação do conjunto de heurísticas/*checklist* informando às heurísticas que não estavam em conformidade e apresentando soluções:

4.1. Heurística “Relacionamento entre a Interface do Sistema e o Mundo Real”

Como mostra a Figura 1, o usuário deverá inserir o login de acesso e sua respectiva senha. A partir do momento que o mesmo se conecta ao sistema, o aplicativo informa que a conexão com o servidor foi feita. Caso a senha seja esquecida, é possível, através do Menu, mudar a senha de acesso.



Figura 1: Acesso e Mudança de Senha

Problema 1: Não é todas as telas que mantêm acessíveis menus e funções comuns do aplicativo. A solução possível é ter os menus mais acessíveis, utilizar ícones e símbolos fáceis de reconhecer e relacioná-los com as tarefas as quais estão associadas. Para facilitar a busca no “menu”, colocar os itens em listas de seleção (isto é, os nomes, produtos, etc...), ordenados por um critério adequado, por exemplo, em ordem alfabética.

Problema 2: O aplicativo não oferece informações sobre sua versão. A solução possível é apresentar no aplicativo a versão que está utilizando.

4.2. Heurística “Liberdade e controle do usuário”

A Figura 2 mostra os trabalhos pendentes durante o dia de serviço. Para obter mais informações sobre o serviço o usuário deverá dar um duplo clique em cima da ordem de serviço.



Figura 2: Acesso a Ordem de Serviço

Problema 1: Não é possível identificar o número de passos necessários para a realização de uma tarefa. Possível solução é, por exemplo, partir de uma indicação numérica (1-5) da quantidade de páginas ou passos, ou apresentar um tutorial, ou ainda dividir a tarefa em abas.

Problema 2: O aplicativo não oferece mensagens de fechamento para tarefas que envolvam um conjunto de etapas. Possível solução é inserir um botão “voltar” do aplicativo.

Problema 3: Não é possível retornar a tela anterior a qualquer momento. O usuário não pode desfazer uma ação. Possível solução é ter um botão para avançar ou nota de explicação, para que o próximo passo seja realizado.

Problema 4: Não é possível sair da tela atual a qualquer momento. Solução possível é deveria ter opções de cancelamento de ação.

Problema 5: O usuário não pode refazer uma ação. A solução é implementar uma maneira que o aplicativo possa armazenar o que já foi feito, e permitir retorno a esta ação.

4.3. Heurística “Consistência e padrões”

A Figura 3 mostra os procedimentos para acessar a Ordem de Serviço a ser executada.



Figura 3: Iniciando a Ordem de Serviço

Problema 1: As informações textuais do aplicativo não utilizam o mesmo idioma. Possível solução é ter todas as telas de busca com o mesmo título. Por exemplo, na situação da Figura 3, um texto apropriado conteria todas as telas com a alternativa “Ativas”.

4.4. Heurística “Prevenção de erros”

A Figura 4 mostra como executar o serviço, neste caso uma troca de equipamento:

1. O usuário vai em “Sumário” e clica em “Equipamento”.
2. Confere na outra tela que surgirá o número do equipamento a ser trocado.
3. Dando um duplo clique surgirá outra tela onde será inserido o novo número do Equipamento.
4. Após isso clicar em “Salvar”.

Note que o usuário deve ter muita atenção neste momento, pois no instante em que é inserido o código do equipamento e, sendo salva esta operação, não é permitido voltar, se o código inserido estiver errado.



Figura 4: Baixa da Ordem de Serviço

Problema 1: Nas primeiras interações do usuário com o aplicativo não são apresentadas instruções básicas. A solução possível é programar um botão que possa ter informações do sistema, como por exemplo um “Tutorial sobre o Sistema”.

Problema 2: Não há botões e controles que desencadeiam ações irreversíveis. Possível solução é uma implementação de uma ação para pedir confirmação do número inserido, como por exemplo na “Troca de equipamento”. Outra solução mais complexa seria cada equipamento ter no código um dígito verificador, válida para equipamentos futuros.

4.5. Heurística “Ajude aos usuários a reconhecer, diagnosticar e sanar os erros”

Problema 1: O aplicativo não permite fácil recuperação de erros, por exemplo, a situação acima apresentada não permite ao usuário reconhecer que o código foi mal inserido. Possível solução é permitir que houvesse opções de navegação como, por exemplo, um link de “Voltar” para retornar à página anterior inserindo novamente os dados incorretos, ou um link “Refazer” para fazer novamente a transação com os dados corretos.

4.6. Heurística “Reconhecimento ao invés de lembrança”

Depois da baixa da ordem de serviço (conforme Figura 4), o usuário clica em “Tipo de OS” e escolhe a opção “Executada”, na tela que surge a seguir, o mesmo irá no “Menu do aparelho” e clica em “Finalizar”. A próxima tela que segue é a tela “Trabalhos Pendentes”, indicando se há mais algum trabalho a ser executado. A Figura 5 mostra então como finalizar a Ordem de Serviço no dispositivo móvel.



Figura 5: Finalizando a Ordem de Serviço

Problema 1: Nesta situação não é utilizado o nome da tela anterior ao invés da opção “Voltar” para nomes de botões e links. Possível solução é de não assumir que o usuário compreenda que deve utilizar o “Menu” do aparelho, mas sim o aplicativo ter opções para verificar os trabalhos pendentes.

4.7. Heurística “Flexibilidade e eficiência de uso”

A Figura 6 mostra como executar o serviço com troca de equipamento. Porém utilizando termos técnicos com muitas informações.



Figura 6: Baixa da Ordem de Serviço

Problema 1: As tarefas não são relativamente simples de serem executadas. Possível solução é colocar o aplicativo a funcionar ao primeiro clique sem travar.

Problema 2: O aplicativo não oferece mecanismos para uma navegação dinâmica na qual os usuários podem ir de uma tela para outra com apenas um *click* sem ter que ir para uma outra página antes. Possível solução é que as tarefas sejam executadas em poucos passos.

Problema 3: Não é possível personalizar a interface de acordo com a preferência de cada usuário. Possível solução é permitir que o usuário altere, por exemplo, o tamanho das fontes ou cores, e escolha trabalhar com ícones ou botões.

4.8. Heurística “Estética e design minimalista”

A Figura 5 mostra como é finalizada a ordem de serviço. Note que o “Menu” não é esteticamente simples e claro, com linguagem muito técnica e as instruções textuais muito longas.

A possível solução é usar uma linguagem mais usual, mais simples, sem detalhes técnicos, e instruções textuais claras e com frases curtas.

4.9. Heurística “Ajuda e documentação”

Problema 1: O aplicativo não possui uma opção de ajuda. Possível solução é ter uma opção “Ajuda”, um botão “Configuração” onde será fácil buscar informações sobre sistema, como foi explicado anteriormente em um “Tutorial do Sistema” sobre aplicações de como mudar fontes e letras, verificação da rede e bateria.

5. Proposta de *layouts* para o aplicativo WorkAssure

Nesta seção serão apresentadas as soluções para os problemas citados na seção anterior.

A ferramenta utilizada para geração destes *layouts* é o Eclipse com emulador para Android. Os emuladores para Android representam bem os dispositivos reais, permitindo ao desenvolvedor reproduzir o ambiente real com qualidade. Salienta-se que nesta proposta a preocupação é de resolver os problemas de usabilidade e não de *design*. Portanto, sabe-se que se faz necessário o melhoramento destes *layouts*.

5.1. Login

A Figura 7 ilustra o “Login” na aplicação móvel que permite o acesso mediante a autenticação de usuário e senha. Na primeira tela são inseridos os dados do usuário. Na tela seguinte, em caso de falha, será exibida uma mensagem de erro. Para redefinir a senha é necessário que o usuário ligue para a central e peça um novo “login”. Em caso de sucesso na autenticação é apresentada a tela principal da aplicação.



Figura 7: Login

5.2. Executar Ordem de Serviço

A Figura 8 mostra como executar a ordem de serviço. Neste caso, com troca de equipamento, o usuário clica em “Iniciar Atendimento”. Na próxima tela é inserido o número do equipamento. Caso o código seja inserido errado o sistema dispara uma mensagem de erro informando que o número está incorreto. Após digitar o número correto é só clicar em “Salvar”.



Figura 8: Executar Ordem de Serviço

5.3. Atendimento

A Figura 9 demonstra como começar um atendimento. Na tela do aplicativo são informados dados do cliente e suas observações em caso de pane hipoteticamente em seu televisor, como exemplo. Finalizada a operação de manutenção, não havendo a necessidade de voltar até a residência clica-se em “NÃO” e “Finaliza o Atendimento”, onde será mostrada uma mensagem de “Sucesso” no aplicativo. Ao clicar no botão “SIM” é informado que é necessário retornar até a casa do assinante. Então, se faz necessário relatar o motivo do futuro retorno, logo após clica-se no botão “Emitir Alerta” para que não ultrapasse o limite de tempo de atendimento.



Figura 9: Tela e Atendimento

5.4. Configurações

A Figura 10 mostra todas as configurações disponíveis do sistema, tais como verificar a conexão/rede do aparelho, o som, as aplicações que consiste em poder mudar a fonte, cores/tamanho das letras; informações sobre o aplicativo, como atualizações do sistema e sinal; poderá ser visualizado o tutorial ou ajuda sobre o aplicativo, o quanto consome da bateria e a versão que está sendo usada atualmente. A qualquer momento é possível voltar à tela anterior ou sair do sistema.



Figura 10: Configurações do sistema

6. Avaliação do *checklist* de usabilidade para o aplicativo MotoBoyOnline

Este aplicativo foi desenvolvido pela empresa VaiMoto, fundado por Daniel Muniz Silva, com recursos próprios, começou a operar em 2013, para facilitar a chamada de motoboys por aplicativos para smartphones com sistema operacional android

Nesta subseção serão abordadas as etapas de avaliação do conjunto de heurísticas/*checklist* informando às heurísticas que não estavam em conformidade e apresentando soluções:

6.1. Heurística “Visibilidade do Status do Sistema”

A Figura 11 mostra que o usuário deve se conectar ao sistema para receber os serviços, tendo o mesmo que informar seu “login” e “senha” e, em seguida, na próxima tela o Status do sistema aparecerá OnLine, ou seja, o usuário está apto a receber serviço.

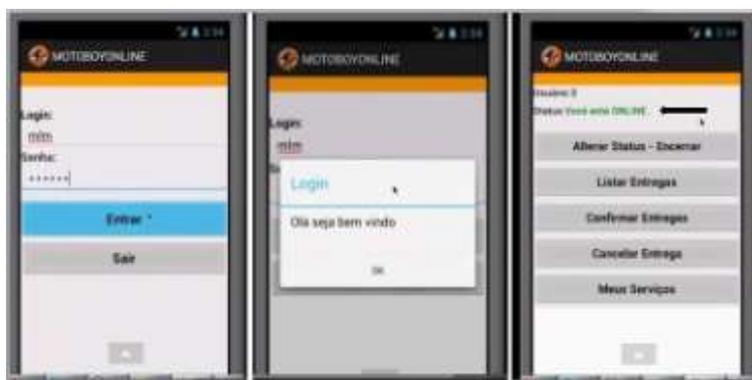


Figura 11: Login do Sistema

Problema 1: O aplicativo não prioriza mensagens relativas à informações críticas e contextuais como o status da bateria e status da rede. A solução é permitir a visualização da bateria e se o dispositivo móvel está conectado a rede para poder receber os serviços.

6.2. Heurística “Relacionamento entre a Interface do Sistema e o Mundo Real”

Problema 1: Conforme pode ser visto na figura 11, acima apresentada, além da visualização da bateria e da rede, não há informações sobre a versão que está sendo utilizada no aplicativo. A solução é de que a informação da versão do sistema do aplicativo esteja visível.

6.3. Heurística “Liberdade e controle do usuário”

Ao clicar no botão “Listar Entregas” irá aparecer a solicitação do serviço. No momento que a mensagem chega ao dispositivo é necessário abri-la e na próxima tela é mostrado o tipo de serviço que terá que ser executado como mostra Figura 12.



Figura 12: Tela de Serviços

Problema 1: Não é o usuário quem inicia e encerra tarefas, e sim, o aplicativo. A solução é que o usuário deve ter uma maior interação com o sistema e deve ter autonomia de fazer alterações ou anotações.

Problema 2: O aplicativo não oferece mensagens de fechamento para tarefas que envolvam um conjunto de etapas. A solução é que em cada etapa da tarefa deva ter uma mensagem de encerramento ou um botão de voltar.

Problema 3: Não é possível sair da tela atual a qualquer momento. A solução é que se programe uma ação onde o usuário possa interromper a execução da tarefa a qualquer instante coma alternativa da opção “sair”.

6.4. Heurística “Consistência e padrões

A partir do momento que a tarefa foi realizada, o usuário deve confirmar a tarefa (isto é, a entrega do serviço) clicando no botão “Confirmar Entregas”. Após, aparece uma tela de confirmação da realização de serviço. Deve-se clicar no botão “Sim” para que este serviço saia da lista de serviços do usuário como mostra Figura 13.

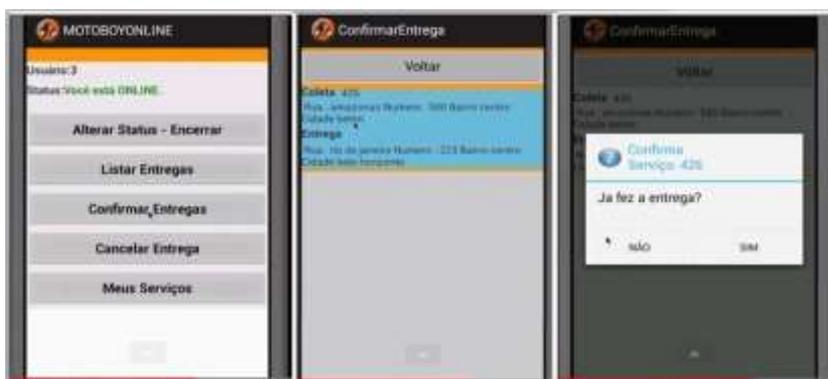


Figura 13: Confirmação da Entrega

Problema 1: O aplicativo não deixa claro qual o próximo passo para realizar a tarefa. A solução é, quando clicado no botão “SIM”, ou seja, o usuário confirma a entrega, possa haver uma maneira de deixar um comentário com informações a respeito da entrega.

Problema 2: O nome do botão/ícone não é consistente com o nome da tela que abre. A solução é que todas as telas devem seguir um padrão, o nome da tela deve ter uma relação com o botão que finaliza a tarefa.

6.5. Heurística “Reconhecimento ao invés de lembrança”

Ao clicar duas vezes sobre a solicitação do serviço, irá aparecer a tela confirmando que o usuário recebeu este serviço. Após, irá aparecer o tipo de serviço, os dados de coleta, a entrega, o valor do frete e o tipo de urgência. Logo após, é confirmado ao usuário que este serviço deve ser efetuado conforme os dados apresentados na tela, como mostra a Figura 14.

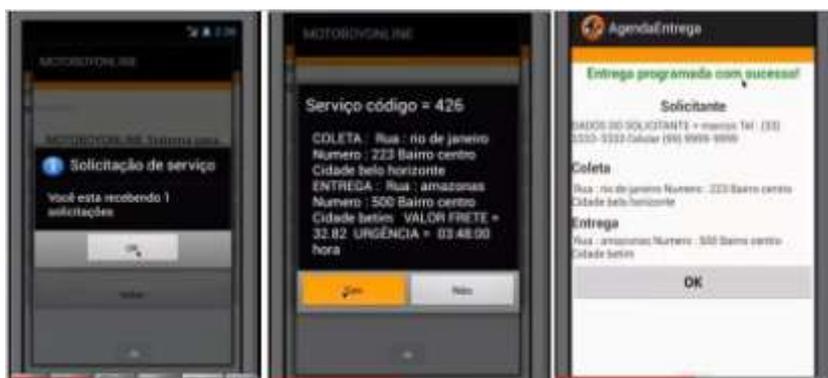


Figura 14: Dados da Ordem de Serviço

Problema 1: Os títulos das telas não são curtos. A solução é implementar botões e textos com títulos curtos e fáceis de reconhecer.

Problema 2: Os títulos das telas não descrevem adequadamente seu conteúdo. A solução é rever as informações que estão dispostas na tela, pois há muitos dados que podem confundir o usuário.

6.6. Flexibilidade e eficiência de uso

Para que o usuário pare de receber serviços é necessário encerrar as suas atividades, deixando o usuário *offline*. Para que isso ocorra, o usuário deve clicar no botão “Alterar Status - Encerrar”. O aplicativo então informará que está desligado, retornando a tela para fazer login. Esta sequência é mostrada na Figura 15.

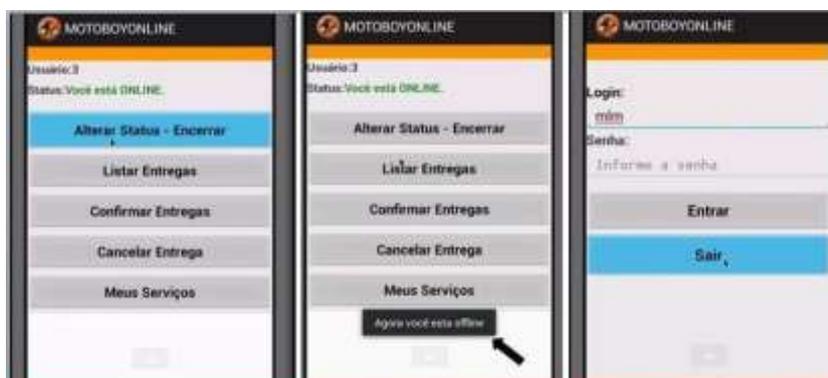


Figura 15: Encerrando Aplicativo

Problema 1: O propósito/função do aplicativo não é claro. A solução possível é programar um recurso que informe o objetivo do aplicativo com informações de uso.

Problema 2: As tarefas não são relativamente simples de serem executadas. A possível solução é implementar um botão que especifique como as tarefas devem ser executadas.

6.7. Heurística “Estética e design minimalista”

A Figura 15, mostrada a cima, referencia as heurísticas que não estão em conformidade no aplicativo.

Problema 1: Não são exibidas apenas informações relacionadas à tarefa que esta sendo realizada. A solução é aparecer na tela apenas o serviço de coleta e entrega, sem mais informações, como a do frete. As informações referentes a valores deverá ser anotada nas observações que o usuário poderá descrever.

Problema 2: O menu não é esteticamente simples e claro. A solução é implementar um botão que especifique a urgência do serviço. Com opções fáceis de encontrar, dispostas em uma ordem lógica e com títulos curtos.

Problema 3: O aplicativo exibe quantidades pequenas de informações em cada tela. A solução é diminuir a quantidade de informações existentes, como por exemplo, não oferecer instruções textuais muito longas.

6.8. Heurística “Ajude os usuários a reconhecer, diagnosticar e sanar erros”

Problema 1: As mensagens de erro não são claras e precisas. A solução é programar botões de ajuda.

7. Proposta de Layouts para o aplicativo MotoBoyOnLine

Nesta seção vamos apresentar soluções para os problemas citados na seção anterior.

7.1. Login

Como mostra a Figura 16, para se conectar ao sistema, o usuário necessita informar o seu nome e sua senha ao clicar no botão “Entrar”. Neste instante, abre outra janela informando que foi conectado corretamente. Caso os dados estejam incorretos, aparece mensagem de erro, sendo necessário inserir os dados novamente.

Se no caso o usuário não souber o login de acesso ao aplicativo, o mesmo deverá ligar para central de atendimento e pedir novo login ou senha.

Quando o aplicativo é inicializado é possível clicar em “configurações”, onde é possível verificar e editar os níveis de conexão, verificar a versão do aplicativo e o quanto de bateria consomem.



Figura 16: Login

7.2. Executar Ordem de Serviço

De acordo com a Figura 17 ao clicar em “Iniciar Ordem de Serviço”, surge à tela, com as informações do cliente, o endereço onde o Motoboy irá buscar o produto e onde deve ser entregue. Depois de visualizar as informações, o aplicativo volta para tela de “Executar Ordem de Serviço”.

Se o usuário tiver dúvidas poderá a qualquer momento voltar ou sair do aplicativo. É importante salientar que ao “Iniciar Ordem de Serviço”, é emitido um alerta informando que este cliente necessita de entrega imediata.



Figura 17: Executando o Aplicativo

7.3. Confirmar/Cancelar Serviço

Ao clicar em “Confirmar Entrega” aparece outra tela na qual o usuário digita informações a respeito do cliente (responsável pela recepção do serviço).

Para cancelar por algum motivo, clica-se em “Cancelar Entregas”. Então, o usuário informa o motivo do cancelamento e, ao final, salva a informação, como é demonstrado na Figura 18.



Figura 18: Confirmar/Cancelar

8. Análise dos Resultados do WorkAsure e MotoBoyOnline

Para avaliação das aplicações propostas nesse trabalho, foi definida uma metodologia de avaliação da qualidade dos aplicativos móveis baseado nos seguintes critérios:

Usabilidade: Conjunto de características que definem a capacidade que a aplicação possui de ser legível, atraente e navegável, envolvendo aspectos de interface, estilo e simplicidade; **Funcionalidade:** Conjunto de características que definem a capacidade da aplicação efetuar recuperação, inclusão e alteração de informação; **Eficiência:** Conjunto de características que definem a capacidade da aplicação com relação à performance; **Confiabilidade:** Conjunto de características que definem a capacidade da

aplicação com relação à recuperação de erros, processamento correto dos links e validação de dados.

Para cada item, foi verificada a heurística e o que deveria ser feito. Após, foi feita uma proposta de solução dos problemas e foi apresentado neste trabalho.

9. Conclusão

Dentro da grande maioria das empresas de desenvolvimento de software, a realização de testes de avaliação de usabilidade ainda são considerados algo desnecessário ou de pouca importância. Parte disso ocorre em função de que muitas empresas ainda lutam para assegurar que o aplicativo funcione.

Ainda inexistem tempo para avaliar outros atributos da qualidade, em especial, a usabilidade do aplicativo. No entanto, estudos acadêmicos e relatos de avaliação de usabilidade de sites e aplicativos de comércio eletrônico demonstram que melhorias na usabilidade fazem a diferença para o sucesso de um aplicativo. Há a necessidade das empresas perceberem que aplicativos com altos níveis de usabilidade facilitam a vida do usuário final durante o uso, reduz investimentos com treinamentos e suporte técnico, entre outros benefícios.

As duas empresas foram contatadas e se mostraram interessadas em receber a avaliação feita neste trabalho. Este documento foi então enviado na data de 08/Novembro/2014 para os responsáveis técnicos e, até o presente momento, não recebemos retorno.

10. Bibliografia

Cybis, Walter, Betiol, Adriana Holtz; Faust, Richard. Ergonomia e usabilidade: conhecimentos, métodos de aplicações. Rio de Janeiro: Novatec, 2007.

Bastien, C. e Scapn, D. Recomendações de Usabilidade. (1ª Edição). São Paulo. Editora Campus.2009..

Ferreira, Kátia Gomes. **Teste de Usabilidade**. Belo Horizonte: Universidade Federal de Minas Gerais. Especialização em Informática com Ênfase em Engenharia de Software, 2013.

Hitz, M., Leitner, 2006. "Usabilidade para Web aplicações". Engenharia Web : a disciplina de Desenvolvimento sistemático de aplicações Web. IX Simpósio Brasileiro de Qualidade de Software. Disponível em: http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2010/RL12_tayana.pdf. Acesso: 29 set. 2014.

Nielsen, J, 1993. Evolução das heurísticas: Métodos de inspeção. Disponível em: <http://www1.folha.uol.com.br/folha/informatica/ult124u21504.shtml>. Acesso em: 14 set. 2014

Pádua, C. I. P. da S. e. Engenharia de Usabilidade – Material de Referência. UFMG, Belo Horizonte, MG. Disponível em: <http://homepages.dcc.ufmg.br/~clarindo/arquivos/disciplinas/eu/material/referencias/apostila-usabilidade.pdf>. Acesso em: 10 set. 2014

Validação de Heurísticas de Usabilidade para Celulares Touchscreen WP_GQS_01-2013_v10 usabilidade.pdf. Acesso em: 23 out. 2012

Oliveira, Sandro R. B Oliveira. Projeto e Avaliação de Interfaces: Ambiente Web. Instituto de Ciências e Exatas e Naturais Universidade Federal de Pará. Disponível em: <http://www.ufpa.br/srbo/Disciplinas/Especializacao/Aulas/Aula07.pdf>. Acesso em: 16 set/2014