

2 Usabilidade em Interação Humano-Computador

Do ponto de vista do usuário, a interface é uma das partes mais importantes dos sistemas computacionais, já que por meio dela ele vê, ouve e sente. Essa relevância motiva pesquisadores da área de Interação Humano-Computador a estudarem maneiras de se criarem interfaces com design focado em usabilidade. A avaliação da usabilidade de interfaces visa verificar se elas atendem aos requisitos do usuário de forma que as funcionalidades do sistema sejam realizadas de modo efetivo, eficiente e que satisfaça as expectativas do usuário. O trabalho de campo realizado nessa pesquisa foi fundamentado a partir do estudo desse agrupamento teórico de princípios, heurísticas, diretrizes e normas de usabilidade e acessibilidade apresentados nesse capítulo.

Segundo a ISO 9241-210, *user experience* pode ser definido como sendo "as percepções e respostas dos usuários resultantes do uso e/ou da antecipação do uso de um produto, sistema ou serviço" (ABNT NBR ISO 9241-210:2011).

A ISO 9241-11 estabelece a seguinte definição para usabilidade: "Usabilidade é a medida em que um sistema, produto ou serviço pode ser utilizado por usuários específicos para atingir objetivos específicos com efetividade, eficiência e satisfação em um contexto específico de uso". Esta definição para usabilidade tem se tornado comumente aceita em todo o mundo (ABNT NBR ISO 9241-11:2011).

A ISO 9241-171 descreve requisitos para "Orientação sobre acessibilidade de software". Este documento define acessibilidade como a "usabilidade de um produto, serviço, ambiente ou facilidade para pessoas com a mais ampla variedade de capacidades". Entende-se produto, serviço, ambiente ou facilidade no contexto específico de software, incluindo aqueles passíveis de estar disponíveis na *web* (Ergonomics of Human-System Interaction - ISO 9241-171:2008).

A Apple desenvolveu um *built-in* de funcionalidades de acessibilidade, APIs (*Application Programming Interface*, conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web) de acessibilidade e uma série de ferramentas para desenvolvimento de

utilitários na plataforma iOS. No geral, o entendimento é possibilitar uma configuração mais acessível para facilitar a utilização do sistema (iOS) (APPLE DEVELOPER SITE, 2016).

Com o crescimento exponencial da tecnologia e o surgimento de novas soluções a cada ano, os trabalhos de pesquisa nessa área são contribuições relevantes para uma melhor interação entre os usuários e as novas tecnologias.

Os conceitos apresentados nesse capítulo, são importantes para os profissionais que atuam na área de desenvolvimento *mobile* e para a contextualização desse trabalho. São pontos de atenção baseados em estudos com o objetivo de tornar a experiência de uso de um sistema ou aplicativo a melhor possível para o usuário. Também apoiou na definição dos métodos e técnicas apresentadas no capítulo 4.

2.1. Princípios de usabilidade

Segundo Preece (1994), princípios de *design* são, a rigor, conjuntos de *guidelines* genéricos para guiar o design. *Guidelines* podem ser tanto regras específicas a serem seguidas, quanto conjuntos genéricos de princípios a serem aplicados nos mais diversos domínios.

Os princípios de *design* de sistemas interativos, que incluem não só a interface com o usuário, mas também aplicativos comerciais de *software* e dispositivos computacionais, se dividem em três categorias: aprendizibilidade, flexibilidade e robustez (DIX et al. 2004).

Para cada categoria, os autores enumeram um conjunto de princípios menos genéricos que as compõem. O princípio de aprendizagem se refere à capacidade de o sistema interativo permitir que o usuário aprenda a utilizá-lo em um primeiro momento, para que ele seja capaz de explorar todas as funcionalidades do sistema com relativa facilidade. Como o tempo de aprendizagem varia de acordo com o usuário, normalmente ela é analisada com base na eficiência de execução de atividades pelo usuário. Os cinco princípios específicos que compõem a aprendizibilidade (DIX et al. 2004):

1. *Previsibilidade*. O desejo maior dos usuários de sistemas interativos é obter do sistema as respostas que eles desejam em relação à ação realizada por

eles. Geralmente, esse objetivo pode ser alcançado por meio de um planejamento de quais atividades contemplarão um determinado anseio dos usuários. Em geral, as pessoas são capazes de prever ações de acordo com determinadas vivências que elas tiveram no mundo real. Estendendo-se esse comportamento humano ao contexto de interação entre pessoas e sistemas interativos, diz-se que sistemas com boa previsibilidade são os que deixam claro ao usuário as consequências das ações realizadas por eles.

2. *Capacidade de sintetização.* Depois que o usuário executa uma determinada ação, é indispensável (e crítico) que o sistema o retorne algum *feedback*. Existem várias formas de se realizar esse *feedback*, mas é imprescindível que ele seja confiável e efetivo no sentido de que o usuário perceba claramente que mudanças ocorreram no sistema.
3. *Familiaridade.* Esse princípio diz respeito à capacidade de o sistema criar elos entre objetos ou situações exibidas ao usuário com objetos ou situações conhecidas por ele no mundo real.
4. *Consistência.* O uso de elementos de entrada/saída deve ser análogo para situações análogas. O mesmo vale para o comportamento de todo o sistema para situações parecidas.
5. *Generalização.* Esse princípio se refere à capacidade de o usuário tomar porções pequenas do sistema e generalizá-las, para deduzir funcionalidades mais amplas.

O conceito de flexibilidade está relacionado à multiplicidade de formas com que o sistema e o usuário trocam informações, ou seja, o usuário deve dispor de caminhos alternativos para chegar a um mesmo resultado. Os cinco princípios a seguir formam o princípio de flexibilidade (DIX et al. 2004):

1. *Iniciativa de diálogo.* É evidente que todo sistema deve possuir alguma forma de interagir com o usuário, a qual caracteriza a forma pela qual eles se comunicam. O cerne desse princípio está em decidir se o sistema irá iniciar esse diálogo, ou se essa tarefa será realizada pelo usuário. Entretanto, é fundamental que o usuário seja capaz de abandonar, suspender ou continuar uma atividade do sistema a partir de qualquer ponto. É comum que abordagens híbridas de iniciativa de diálogo sejam adotadas em

sistemas interativos, para satisfazer requisitos de funcionalidades e garantir flexibilidade de uso.

2. *Multithreading*. Relaciona-se à capacidade de o usuário realizar mais de uma tarefa concomitantemente. Geralmente o termo mais usado em Interação Humano-Computador para designar esse comportamento é “multimodalidade”. Usuários podem realizar diferentes tarefas em locais diferentes, abrindo abas distintas para operações distintas, por exemplo, ou realizá-las em uma mesma tela de um mesmo dispositivo.
3. *Migração de atividades*. Permitir que usuários realizem tarefas rotineiras pode fazer com que eles se desestimulem por perderem a concentração, o que aumenta a probabilidade de erros ao longo da interação. A migração de atividades se relaciona à capacidade de o sistema realizar atividades que corriqueiramente são de responsabilidade do usuário. Esse comportamento economiza tempo e diminui a probabilidade de erros. Novamente, é necessário que se estime quão migrável pode ser uma determinada funcionalidade, já que muitas vezes não é possível dispensar totalmente as ações realizadas pelo usuário.
4. *Substitutividade*. Esse conceito se relaciona à possibilidade de usar termos equivalentes para valores de entrada/saída como, por exemplo, permitir visualização de temperatura tanto em Celsius quanto em Fahrenheit.
5. *Personalização*. Muitos usuários desejam que as interfaces sejam capazes de se adaptarem às suas necessidades. Essa adaptação pode ocorrer tanto por parte do usuário (adaptabilidade) quanto por parte do sistema (adaptatividade). As customizações de sistema mais comuns atualmente são a possibilidade de escolha de características pessoais ou de criação de atalhos para funcionalidades.

Os princípios de robustez enumerado abaixo se refere ao nível de suporte oferecido ao usuário para que ele obtenha sucesso na interação. Esse princípio foi dividido em quatro subprincípios (Dix et al. 2004):

1. *Observabilidade*. O usuário deve ser capaz de identificar o estado atual da aplicação rapidamente por meio da representação perceptível da interface. Para tanto, as informações contidas na interface devem ser intuitivas e relevantes.

2. *Recuperabilidade*. Capacidade de o usuário se recuperar de situações de erros ou desfazê-los. O sistema deve permitir que o usuário efetue uma operação de correção sempre que um erro ocorrer. Além disso, o usuário deve ser capaz de voltar a uma situação anterior ao erro. As mensagens de erro devem ser concisas, informativas, específicas e construtivas.
3. *Capacidade de resposta*. Para efeito de estabilidade, o sistema deve garantir o oferecimento de respostas rápidas ao usuário, a fim de que ele não se sinta confuso enquanto realiza uma determinada atividade. Além disso, espera-se que o sistema notifique o usuário de alguma forma, após ele ter solicitado uma determinada ação. Esse comportamento é importante para que o usuário tenha a garantia de que o sistema está cuidando da solicitação por ele enviada.
4. *Conformidade de realização de atividades*. Refere-se ao grau de suporte que o sistema oferece ao usuário em relação às tarefas que ele pode realizar e à maneira pela qual ele pretende realizá-las.

Em um contexto mais genérico, não se limitando apenas ao *design* de sistemas interativos, Williams (2005) elaborou um conjunto de princípios que ele considera básicos em qualquer obra de design, aos quais ele abreviou como **CARP**: **C**ontraste, **A**linhamento, **R**epetição e **P**roximidade.

O princípio de contraste prega que elementos importantes devem ter contraste em relação aos demais, principalmente em relação ao fundo. O olho humano tem tendência a repousar sobre elementos que estão deslocados do padrão e esse pode ser um fator importante de se explorar quando se elabora uma interface. O contraste de um elemento pode ocorrer por meio da alteração do seu tamanho, da adição de textura, da mudança da sua orientação ou da sua forma, da colorização ou da alteração do posicionamento padrão entre outros.

O princípio de alinhamento atenta para que elementos visuais não sejam dispostos de qualquer forma, mas sim de maneira alinhada. Elementos análogos colocados de forma alinhada facilitam a compreensão porque permitem que o usuário visualize os elementos de forma linear, de forma mais confortável aos olhos.

A repetição é utilizada no *design* para criar uma "âncora visual". Ela é responsável pela facilidade de se mudar de um elemento para outro, em uma determinada interface. Além disso, elementos repetidos sugerem que eles fazem parte de um mesmo grupo, o que facilita a interpretação. Nota-se que o conceito de repetição tem ligação com o princípio de consistência de interfaces (Dix et al. 2004).

Por fim, o princípio de proximidade visa garantir que elementos relacionados estejam próximos, ao passo que a distância entre elementos sem relação deve ser maior. Esse princípio ajuda a manter a unidade e ordem do *layout*.

Novamente, os princípios devem ser aplicados em conformidade com o contexto de aplicação. O contraste não pode, por exemplo, interromper o processo natural de leitura de um texto por meio da alteração do tamanho das letras utilizadas em determinadas palavras. O *design* sempre deve levar em consideração as restrições dos usuários e do dispositivo a ser utilizado.

2.2. Heurísticas de usabilidade

Avaliações de usabilidade de interfaces são processos que visam garantir, por meio da análise dessas interfaces, que o sistema funcione adequadamente e satisfaça às expectativas dos usuários e os requisitos de *software* elicitados nas fases iniciais do processo de desenvolvimento do software. Os objetivos dessas avaliações são examinar a experiência do usuário ao utilizar as interfaces, analisar a acessibilidade das funcionalidades disponibilizadas e identificar problemas de *design* (Sharp et al., 2007).

A avaliação heurística é realizada por meio de um conjunto pequeno de especialistas de *design* que, separadamente, avaliam a interface confrontando-as com regras, conhecidas como "heurísticas", para identificar eventuais erros de *design* que comprometam a usabilidade. Acredita-se que de três a cinco avaliadores especialistas sejam capazes de identificar de 75% a 80% dos problemas de usabilidade de interfaces computacionais (Nielsen, 1994).

Provavelmente o conjunto de heurísticas mais conhecido seja o elaborado por Nielsen (1994), a seguir:

1. *Visibilidade do status do sistema.* O usuário deve estar completamente informado do que está acontecendo, por meio de *feedback* imediato da interface.
2. *Compatibilidade do sistema com o mundo real.* A terminologia deve ser adequada à linguagem do usuário e não orientada ao sistema. As informações devem ser organizadas de acordo com o modelo mental do usuário. Segundo Helander [1997], o modelo mental se refere à expectativa que um usuário possui em relação ao comportamento do computador.
3. *Controle e liberdade do usuário.* Disponibilize “saídas de emergência” ao usuário, para que ele possa desfazer ou refazer ações, a fim de que ele se situe em um ponto recente da interação.
4. *Consistência e padrões.* Nunca identifique uma mesma ação por ícones ou metáforas diferentes. Elementos similares devem ser usados para propósitos semelhantes, assim como funcionalidades semelhantes devem possuir uma sequência de ações semelhantes.
5. *Prevenção de erros.* Idealmente, interfaces não precisam de mensagens de erro por serem capazes de prevenir que erros ocorram. Ações definitivas podem ter um tratamento anterior para que o usuário as confirme por meio de *checkboxes*, por exemplo.
6. *Reconhecimento ao invés de lembrança.* O usuário não deve precisar memorizar o que está realizando. Permita que a interface atue como um meio de dialogar com o usuário, em tempo de execução.
7. *Flexibilidade e eficiência de uso.* O sistema deve ser fácil de ser operado por usuários novatos, mas também robusto o suficiente para permitir eficiência de uso a usuário avançados. Teclas de atalho e de comandos por voz podem ser alternativas para tornarem a interface flexível. Em outras palavras, flexibilidade implica em permitir que uma mesma funcionalidade seja realizada por comandos distintos.
8. *Estética e design minimalista.* As informações devem ser sucintas e não devem informar mais do que os usuários necessitam para realizar a funcionalidade corrente. Os diálogos do sistema precisam ser diretos e naturais e devem aparecer nos momentos adequados.

9. *Ajuda para usuário identificar, diagnosticar e corrigir erros.* As mensagens de erros devem ser claras e simples e não podem intimidar o usuário. Ao contrário, devem estimulá-lo ao oferecer formas de corrigir o erro.
10. *Ajudas (Help) e documentação.* Um bom *design* evita que o usuário tenha que usar opções de ajuda com frequência. Entretanto, é fundamental que o sistema possua telas específicas de ajuda, para orientar o usuário em casos de dúvidas.

Nitidamente, existe uma dependência estreita entre as heurísticas criadas por Nielsen (1994) e as diretrizes divulgadas por empresas detentoras de sistemas operacionais para dispositivos móveis.

Segundo Bertini et al. (2006), o conjunto de heurísticas elaborado por Jacob Nielsen é utilizado como forma de precaver erros de usabilidade e satisfazer requisitos de qualidade de interfaces em vários domínios de aplicação. Por conseguinte, é esperado que tais heurísticas também sejam usadas como base científica para avaliação de interfaces de dispositivos móveis. Entretanto, os autores ressaltam que, por conta de a interação nesses dispositivos ser muito peculiar, o ideal é que se usem heurísticas apropriadas para eles. Por conta dessa necessidade, Bertini et al. (2006) elaboraram regras próprias para esse fim. O conjunto de heurísticas elaborado por Bertini et al. (2006) como refinamento das regras de Nielsen (1994) e adaptação delas para o contexto de interfaces de computadores móveis contém oito regras, a seguir:

1. *Visibilidade do status do sistema e facilidade de encontrar o dispositivo móvel.* O sistema deve sempre manter o usuário informado sobre o que está ocorrendo. Além disso, o sistema deve dar prioridade a mensagens relativas a aspectos críticos do sistema, como capacidade da bateria, condições do ambiente de utilização e informações de conectividade. Tendo em vista que dispositivos móveis são perdidos com certa facilidade, medidas de encriptação de dados devem ser consideradas e o sistema deve prover formas de o usuário encontrar o dispositivo, caso ele tenha sido esquecido fora de lugar.
2. *Compatibilidade entre o sistema e o mundo real.* Permita que o usuário entenda a informação sendo exibida de forma correta, por meio de uma disposição de elementos em ordem natural e lógica. Sempre que possível, o

sistema deve permitir identificar condições ambientes locais e informações de uso automaticamente e exibi-las de forma adequada ao usuário.

3. *Consistência e mapeamento.* O modelo conceitual que o usuário possui acerca da relação entre função e interação deve ser consistente com o contexto de utilização. É crucial que haja um mapeamento adequado entre ação a ser realizada e modo de realizar esta mesma ação no mundo real.
4. *Boa ergonomia e design minimalista.* Dispositivos móveis devem ser fáceis de manusear com apenas uma das mãos e ser resistentes a degradação por ações do ambiente, como umidade. Além disso, nenhuma informação desnecessária deve ser exibida ao usuário.
5. *Facilidade de entrada de dados, legibilidade e capacidade de assimilação.* Os dispositivos móveis devem prover modos simples para que o usuário informe dados de entrada, preferencialmente sem que o usuário precise usar as duas mãos para executar tal tarefa. A tela deve possuir todas as informações visíveis ao usuário, independentemente das condições de luminosidade do ambiente. Idealmente, o usuário deve ser capaz de assimilar a informação sendo exibida imediatamente.
6. *Flexibilidade, eficiência de uso e personalização.* Permita que os usuários personalizem as ações de acordo com as necessidades deles. Sempre que possível, o sistema deve ser capaz de sugerir ao usuário formas de personalizar ações que porventura sejam benéficas em algum contexto de utilização.
7. *Convenções estéticas, sociais e de privacidade.* Leve em consideração aspectos emocionais e estéticos dos usuários que utilizarão o dispositivo. As informações do usuário deverão ser mantidas com segurança e privacidade. As interações devem respeitar convenções sociais dos usuários.
8. *Gerenciamento de erros realístico.* Proteja o usuário dos erros de interação. Se não for possível fazê-lo, permita que o usuário identifique o erro, o diagnostique e, se possível, o corrija. Mensagens de erros devem ser claras e sucintas. Se o erro for irreversível, certifique-se que o usuário entenderá a condição em que ele ocorreu.

Para criar as dez heurísticas para avaliação de interfaces de usuário, Moraveji e Soesanto (2012), assim como Bertini et al. (2006), acabam sobrepondo

algumas heurísticas de Nielsen (1994). Entretanto, é nítida a preocupação dos autores em elaborarem regras que mitiguem os aspectos estressores das interfaces. A primeira característica (C1) levantada por Lupien et al. (2007) é a de que elementos estressores parecem imprevisíveis, incertos ou não familiares em algum contexto. A segunda (C2) deriva de estudos realizados por Henry e Grim (1990) e diz respeito à sensação de que a pessoa está perdendo o controle da situação. A característica seguinte (C3), identificada por Dienstbier (1989), é a propensão de algo causar perda ou dano ao indivíduo, ou seja, o indivíduo perde o sentimento de posse dele por algo. Por fim (C4), Lupien et al. (2007) citam a capacidade de o elemento representar uma ameaça à identidade ou à autoestima do indivíduo, quando julgada do ponto de vista social.

As dez regras levantadas por Moraveji e Soesanto (2012) são acompanhadas da listagem das características estressores que elas combatem, a seguir:

1. *Capacidade de controlar interrupções (combate C1 e C2)*. Interrupções imprevisíveis comprometem o controle do usuário sobre a interface. Esse é um fator tão importante de ser considerado que existe um ramo da Interação Homem-Computador que estuda especificamente os problemas causados por interrupções (Iqbal e Horvitz, 2007). Essa heurística visa verificar se a interface dispõe de formas de bloquear, controlar, ou desabilitar temporariamente interrupções durante apresentações, ligações telefônicas, ou durante outras funcionalidades que demandam atenção do usuário. Opções como “Não mostrar essa mensagem novamente” ou “Lembrar-me dessa preferência no futuro” são boas formas de o usuário controlar interrupções.
2. *Não sobrecarregar o usuário (combate C2 e C4)*. Grandes quantidades de dados são comuns em aplicativos usados por muitos usuários ou que possuam dados armazenados em servidores *Web*. Nesses casos, o estresse do usuário está em sentir que ele não é capaz de lidar com a quantidade de informação, de modo que a interação parece que não terá fim. O usuário pode imaginar que não está usando o aplicativo da forma correta, que não está conseguindo se comunicar com outros usuários (em casos de programas colaborativos) ou que não ofereceu a quantidade de dados suficiente para o aplicativo funcionar. Ademais, o usuário pode se sentir sobrecarregado por telas com muitas opções para serem visualizadas e/ou preenchidas. Em

aplicativos de email, por exemplo, deve-se evitar mostrar várias mensagens não lidas de uma só vez.

3. *Considerar a percepção de tempo dos seres humanos (combate C1 e C2).* A barra de progresso indica ao usuário o tempo a transcorrer de uma determinada atividade, do ponto de vista do sistema. Entretanto, a percepção de tempo humana não é linear, porque os indivíduos tendem a imaginar que o tempo passa mais devagar quanto mais eles esperam por algo (Harrison et al., 2007). Esse comportamento da barra de progressos pode gerar imprevisibilidade e falta de controle do momento em que o usuário poderá usar o sistema. Uma forma de amenizar esse problema é permitir que processos mais longos iniciem o processamento mais cedo, em paralelo com outras interações mais leves. Outra alternativa é distrair o usuário com elementos lúdicos ou informações adicionais que sejam interessantes.
4. *Usar entonação apropriada (combate C1 e C4).* Usuários simpatizam com aplicativos com comportamento baseados em polidez e reciprocidade (Nass et al., 1994). Quando o sistema não é capaz de gerar esse comportamento, os usuários tendem a se sentirem constrangidos pelo tom das mensagens. Por conta disso, é importante que os *designers* se valham de comunicação que se aproxime da humana, em termos de entonação e de emoção. Alguns exemplos incluem uso de mensagens bem-humoradas e que fazem apologia ao aplicativo. Pedidos com polidez devem ser preferidos às mensagens imperativas que indicam obrigatoriedade de ações.
5. *Prover feedback positivo à entrada do usuário e eventos de interação (combate C1 e C4).* *Feedback* negativo pode causar estresse no sentido de que os usuários podem sentir que não são capazes de fornecer ao sistema as informações das quais ele necessita. Esse tipo de *feedback* tende a negligenciar a forma natural de conversação entre indivíduos, o que pode causar estranheza. O sistema pode informar o usuário de que o problema ocorrido é comum, com mensagens como “Outras pessoas cometeram este mesmo equívoco. Sem problemas” e mostrar mensagens positivas quando o usuário completa uma determinada tarefa, como “Obrigado por preencher nosso formulário”.
6. *Estimular interações sociais (combate C4).* Dispor de elementos simples como botões de “curtir” ou “retwitter” ajudam a usuários quebrar barreiras

de interação social, por serem diretas e estarem embutidas na interface.

7. *Amenizar a pressão ao usuário em relação ao tempo (combate C1, C2 e C4).* A maneira pela qual o usuário se sente estressado por conta da pressão imposta pelo tempo é bem documentada por Maule e Hockey (1993), mas ainda não foi aplicada a dispositivos computacionais de modo geral, tampouco para dispositivos móveis. Ao contrário da heurística 3, que trata do tempo enquanto o usuário aguarda, essa heurística considera o tempo enquanto o usuário interage por meio da interface. Usuários tendem a se incomodar e perder o controle da situação quando são pressionados por tempo para realizarem as atividades. Em contextos competitivos, eles podem se frustrar ao pensarem que estão realizando as tarefas mais demoradamente do que outros usuários.
8. *Usar elementos naturalmente calmantes (combate C1).* Chamar a atenção ou gerar fascínio ao usuário de forma involuntária por meio de elementos naturais tende a fazer com que ele se sintam bem, porque os níveis de atividade fisiológicas do indivíduo são alterados de forma positiva (Ulrich et al., 1991). Essa heurística valida se a interface embute elementos naturais (sons, imagens ou animações) de forma que o usuário as assimile e as utilize com facilidade.
9. *Esclarecer as ações ao usuário (combate C1 e C2).* Em cada interface navegada, o usuário espera poder resolver uma determinada atividade que, caso esteja desabilitada, pode gerar irritação e estresse. Nesses casos em que o sistema não pode disponibilizar uma determinada funcionalidade, a interface deve esclarecer ao usuário o motivo pelo qual a funcionalidade não pode ser realizada, ou sugerir formas alternativas de o usuário realizar a atividade.
10. *Desmistificar a interface.* Usuários podem se estressar quando interfaces o apresentam uma miríade de escolhas a fazer de uma única vez. Além disso, sugerir ao usuário que ele procure ajuda no sistema pode prejudicar a autoestima dele. De modo a complementar a heurística de “Ajuda e Documentação” proposta por Nielsen (1994), o sistema deve desmistificar os elementos da interface ao usuário, antes que ele tenha que recorrer à opção de Ajuda.

2.3. Diretrizes de usabilidade

Existem menos pesquisas de usabilidade de interfaces para dispositivos móveis se comparadas a quantidade de pesquisas relacionadas ao ambiente de computadores pessoais, por fatores diversos: a interação em dispositivos móveis é muito específica e diferente da interação com computadores pessoais, o mercado de dispositivos móveis aqueceu após o mercado de computadores pessoais já estar consolidado, dispositivos móveis possuem características e limitações físicas distintas.

As *guidelines* encontradas com o propósito de auxiliar o *design* de interfaces de dispositivos móveis são listadas a seguir (NETO, 2013):

1. *Destaque a principal atividade da aplicação e garanta que o usuário complete qualquer tarefa.* Quando o foco da aplicação é estabelecido e mantido na tarefa principal da aplicação, o nível de satisfação do usuário tende a ser maior. Em cada tela exibida, verifique a se toda informação sendo exibida é necessária no momento para que o usuário finalize a atividade sendo realizada. Em caso negativo, verifique se haverá algum momento em que a informação se tornará crítica. Em caso afirmativo, exiba a informação (APPLE DEVELOPER SITE, 2016). Além disso, a estrutura do conteúdo deve ser agrupada em porções pequenas e homogêneas de informação (GRASSO e ROSELLI, 2005).
2. *Invista os maiores esforços nos fatores mais importantes do ponto de vista do usuário.* Em um jogo de videogame, os usuários não estão interessados em gerenciar contas pessoais ou em gerar e compartilhar conteúdos, mas sim em interagir com o jogo com baixos tempos de resposta por meio de um roteiro que os fascinem e por meio de cenários atrativos. Conteúdos que prendem a atenção de usuário podem ser destacados por meio de efeitos de esmaecimento de controles que não estão sendo usados no momento (APPLE DEVELOPER SITE, 2016).
3. *O design de interface mobile é uma atividade a ser preenchida de cima para baixo.* Em *smartphones*, o topo da tela é o local mais visualizado pelo usuário, porque a interação com o aparelho ocorre enquanto o usuário o segura: com a mão não dominante, enquanto interage com um dos dedos da outra mão; com mão dominante enquanto interage com o polegar dessa

mão; entre as duas mãos, enquanto interage com ambos os polegares (HAYHOE, 2001). Por conta disso, as informações genéricas mais importantes devem ser dispostas no topo de modo que, à medida que o usuário desliza a tela para baixo, ele encontra informações mais específicas (APPLE DEVELOPER SITE, 2016). Idealmente, toda a informação deve ficar visível na tela de uma só vez (GRASSO e ROSELLI, 2005).

4. *Proporcione um caminho lógico para o usuário.* Usuários gostam de saber em que ponto estão da aplicação, por meio de caminhos apresentados na tela. Opções de “voltar” são sempre importantes e, sempre que possível, permita que apenas uma sequência lógica seja usada para que uma tela seja acessada (APPLE DEVELOPER SITE, 2016).
5. *Torne a interação fácil e óbvia.* Facilitar o entendimento de texto em dispositivos móveis implica em exibir frases curtas. Use interações padrões do dispositivo para executar funcionalidades que sejam consistentes com tais interações, como a operação de *swipe* lateral para mudar fotos de um álbum (APPLE DEVELOPER SITE, 2016). Se possível, explore o uso de áudio para substituir conteúdos textuais extensos (DIMAKOPOULOS e MAGOULAS, 2009).
6. *Facilite a entrada de dados.* A entrada de dados costuma demandar muito tempo de interação, de forma que, caso o usuário tenha que fornecer uma grande quantidade de dados antes que qualquer ação relevante ocorra, é provável que ele se desestime. Balanceie a quantidade de dados requeridos com o que a aplicação pode fornecer por meio da interface. Utilize o componente correto para cada tipo de informação a ser coletada, para minimizar o tempo de interação e evitar erros do usuário (APPLE DEVELOPER SITE, 2016).
7. *Estimule a conectividade e o comportamento colaborativo.* Sempre que oportuno, permita que o usuário compartilhe informações com outras pessoas, tais como a localização atual, opiniões, resultados, dentre outras. A interface pode permitir que o dispositivo se comunique com outros dispositivos móveis (APPLE DEVELOPER SITE, 2016).
8. *Torne a interface mais realista possível.* Usuários se sentem mais motivados a interagirem com interfaces que se assemelham com artefatos do mundo real. Por exemplo, a aplicação “agenda” é de fácil entendimento por conter

elementos que remetem a uma agenda de papel. Pense nos objetos do mundo real como oportunidades de comunicação simples e efetiva com o usuário (APPLE DEVELOPER SITE, 2016).

9. *Dê suporte à mudança de orientação.* As pessoas possuem diferentes preferências de orientação, que podem se alterar de acordo com a tela com a qual elas interagem. Por conta disso, crie interfaces que mantenham foco nas atividades e nas informações mais importantes, independentemente da orientação do dispositivo móvel (vertical ou horizontal). Em casos que o aplicativo deve funcionar apenas em uma orientação, exibe a interface nessa orientação, mesmo que o dispositivo esteja noutra orientação, para que o usuário entenda que ele deve girar o aparelho para interagir com a interface. A mudança de orientação nem sempre deve significar um simples redimensionamento de conteúdo. Às vezes, ela deve acarretar em uma total reorganização do conteúdo exibido, para atender as expectativas do usuário (APPLE DEVELOPER SITE, 2016).
10. *Mantenha o usuário ciente de qualquer ação.* Jamais termine um aplicativo sem avisar o usuário ou sem que ele tenha escolhido explicitamente encerrar a interação, porque ele tenderá a imaginar que a aplicação parou de funcionar. Diálogos e informações de alerta devem ser exibidos apenas quando o usuário seleciona uma opção que inicia uma funcionalidade com problemas (APPLE DEVELOPER SITE, 2016).
11. *Dê controle ao usuário.* A interface deve exibir automaticamente o máximo de informação possível, para evitar que o usuário forneça seus dados dispendiosamente. Entretanto, deixe claro que o usuário está no controle no aplicativo, permitindo que ele explicita suas preferências de utilização e que ele desfça operações realizadas recentemente (APPLE DEVELOPER SITE, 2016).
12. *Aposte em um design minimalista.* Elabore uma interface simples, mas de fácil entendimento, por meio de um uso balanceado de elementos e de cores. Use gráficos e animações para melhorar o entendimento, mas os evite no momento da inicialização do aplicativo (APPLE DEVELOPER SITE, 2016).

13. *Use imagens e gráficos em alta definição e editados profissionalmente.* Dedique um bom tempo de trabalho para a criação do ícone do aplicativo, que aparecerá ao longo de toda a interação, porque ele é a imagem que ficará na mente do usuário associada à aplicação. Edite imagens e gráficos para que gerem impacto visual e surpreenda o usuário com a beleza das animações e imagens utilizadas (APPLE DEVELOPER SITE, 2016).

2.4. Normas de usabilidade e acessibilidade da Apple na plataforma iOS

As normas de usabilidade e acessibilidade da Apple, aqui apresentadas, são definidas para a plataforma iOS que engloba os dispositivos móveis da marca (APPLE DEVELOPER SITE, 2016).

1. *Substitutividade.* Esse conceito se relaciona à possibilidade de usar termos equivalentes para valores de entrada/saída como, por exemplo, permitir visualização de temperatura tanto em Celsius quanto em Fahrenheit.
2. *Legendagem e áudio Descrições (Captioning and Audio Descriptions).* Essas funcionalidades permitem ao usuário configurar legendas e descrições de áudio em aplicativos para exibir durante a reprodução de um vídeo. Aplicativos que usam AVFoundation (Kit/biblioteca de áudio e vídeo para desenvolvimento iOS) tem suporte para legendas e áudio descrição durante a reprodução através dessas APIs (APPLE DEVELOPER SITE, 2016).
3. *Personalização da tela (Display Customization).* O sistema iOS suporta uma gama de recursos para personalizar a exibição das informações na tela do dispositivo. Incluindo texto Dinâmico, formas do botão, reduzir a transparência, escurecer as cores, e reduzir o movimento (APPLE DEVELOPER SITE, 2016).
4. *Discurso (Speech).* Dispositivos iOS podem ler o texto selecionado do seu aplicativo em voz alta em mais de 30 línguas. A classe *AVSpeechSynthesizer* produz voz sintetizada a partir de texto em um dispositivo iOS, e fornece métodos para controlar ou monitorar o progresso do discurso contínuo (APPLE DEVELOPER SITE, 2016).
5. *VoiceOver.* É um leitor de tela que interage com os objetos em seus aplicativos para que o usuário possa conduzir a interface, mesmo sem vê-la.

Certifique-se de que os elementos da interface de usuário em seus aplicativos são acessíveis e úteis.

6. *Acesso Guiado (Guided Access)*. Ajuda as pessoas com autismo ou outra deficiência e desafios sensoriais, a ficarem focados na tarefa. Um pai, professor ou terapeuta pode limitar um dispositivo iOS para ficar em um app, desativando o botão Home, e até mesmo restringir a entrada de toque em certas áreas da tela. A implementação do Protocolo de acesso guiado em aplicativos permite que o usuário especifique quais partes de seus aplicativos são funcionais, dependendo da necessidade do usuário.
7. *Deferência*. A atenção no desenvolvimento da interface do usuário ajuda as pessoas a entender e interagir com o conteúdo, mas nunca competir com ele.
8. *Clareza*. Texto legível, ícones precisos e intuitivos. Adornos sutis e apropriados e muito foco na funcionalidade para ajudar o design.
9. *Profundidade*. Camadas visuais claras e movimento fluido transmite vitalidade, torna a experiência mais prazerosa e aumenta a compreensão do usuário.
10. *Adaptatividade*. Proporcionar aos usuários o uso do aplicativo em diferentes dispositivos e orientação da tela (horizontal e/ou vertical).
11. *Proporcione uma grande experiência em cada ambiente*. Tomando vantagem da adaptatividade, você pode garantir que a interface do usuário responda apropriadamente a mudanças de dispositivos e telas.
12. *Mantenha o foco no conteúdo principal de cada ambiente*. Essa é a maior prioridade. As pessoas usam um determinado aplicativo para ver e interagir com o conteúdo que interessa a elas.
13. *Evite mudanças desnecessárias no layout do aplicativo*. Uma experiência consistente em diferentes ambientes faz com que o usuário siga com o mesmo padrão de uso em diferentes orientações da tela ou em dispositivos diferentes.

Desde o início da existência dos iPhones, dispositivos baseados em iOS incluem várias características que fazem o dispositivo ser fácil de utilizar para todos, incluindo correio de voz visual, grandes fontes no e-mail, e zoom em páginas web, fotos e mapas. Com a adição dos seguintes recursos de

acessibilidade, é ainda mais fácil para as pessoas com deficiência visual, auditiva e deficiência física usarem seus dispositivos:

- *Zoom*: Amplia a tela do dispositivo inteiro.
- *Branco no preto*: Inverte as cores na tela.
- *Áudio Mono*: Combina o som dos canais esquerdo e direito em um sinal mono distribuído em ambos os lados.
- *Ações por áudio*: Falar Seleção, falar conteúdo da tela, falar texto automático (verbaliza as correções de texto).
- *Controle de voz*: Permite ao usuário fazer chamadas telefônicas e utilizar outras funções do dispositivo usando comandos de voz.
- *VoiceOver*: O usuário com deficiência visual pode contar com essa funcionalidade para ajudá-lo a usar seu dispositivo.

O *VoiceOver* é uma tecnologia de leitura de tela, da Apple, que dá aos usuários o controle sobre seus dispositivos sem ver a tela. Ele faz isso agindo como um intermediário entre a interface do usuário do aplicativo e o toque do usuário, fornecendo descrições sonoras de elementos e ações no aplicativo. Quando o *VoiceOver* está ativo, os usuários não precisam se preocupar em apagar acidentalmente um contato ou ligar para um número de telefone, porque o *VoiceOver* diz onde eles estão na interface do usuário, que ações eles podem tomar e quais serão os resultados dessas ações.

Um aplicativo é acessível quando todos os elementos da interface, com o qual os usuários podem interagir, são acessíveis. Um elemento de interface do usuário está acessível quando ele se reporta corretamente como um elemento de acessibilidade (APPLE HUMAN INTERFACE GUIDELINE, 2016).

2.5. Considerações parciais

Esse capítulo explora e consolida princípios, heurísticas, diretrizes e normas de usabilidade e acessibilidade, diretamente ligados a IHC (Interação Humano-Computador). Baseadas nas ISOs 9241 e suas extensões 210 (*user experience*), 11 (usabilidade) e 171 (acessibilidade). Apresentando a importância do usuário no processo de desenvolvimento de uma solução digital e auxiliando os profissionais

da área de desenvolvimento de sistemas e aplicações a fundamentarem melhor suas propostas de solução. Dando mais atenção e consideração as necessidades do usuário, aplicando conceitos sólidos gerados por pesquisadores reconhecidos da área de IHC.

Mesmo em se tratando de diferentes definições e aplicações teóricas, ter a possibilidade de consolidar esses conceitos em um único documento é uma oportunidade de formar um agrupamento teórico pesquisado por especialistas da área e, também, foi muito importante para contextualizar esse trabalho de pesquisa.