

Unidade IV

7 PROCESSO DE PROJETO DE INTERFACE COM O USUÁRIO — AVALIAR

Não basta o projetista seguir *guidelines* para projetos de interface para garantir altos níveis de usabilidade. As avaliações não devem ser realizadas apenas na fase final do processo de projeto de interface, como uma única fase. As avaliações podem ser realizadas com especialistas (avaliação de usabilidade) e com usuários (teste de usabilidade).

7.1 Importância das avaliações

A primeira vez que um usuário utiliza um produto depois de pronto, ele forma uma opinião. De acordo com Weiss (2002), se esta experiência for insatisfatória, será improvável que o usuário compre-o ou use-o novamente. Portanto, para assegurar o sucesso do produto, as avaliações de usabilidade não são opcionais, e sim obrigatórias.

No entanto, a avaliação não deve ser vista como uma fase única dentro do processo de desenvolvimento ou ser uma atividade a ser feita somente no final do processo e "se der tempo" (ROCHA; BARANAUSKAS, 2003). Avaliações podem ser realizadas durante todo o ciclo de desenvolvimento da interface, desde as fases iniciais do projeto, nas quais são produzidos os primeiros esboços, até os estágios mais avançados, já com o *software* ou protótipo executável.

Os protótipos podem ser avaliados por especialistas e, particularmente, pelos usuários, por meio de algumas técnicas. Possivelmente ocorrerão algumas iterações de avaliação, *design* e prototipagem até que se atinjam os requisitos de usabilidade definidos. Os resultados das avaliações podem indicar a necessidade de voltar à fase Identificar e Analisar o Contexto de Uso. Dificilmente se conseguirão atingir os níveis de usabilidade definidos já no primeiro protótipo.

Segundo Rocha e Baranauskas (2003), as avaliações têm três grandes objetivos: a) avaliar a funcionalidade do sistema; b) avaliar o efeito da interface junto ao usuário; e c) identificar problemas específicos do sistema.

- Avaliar a funcionalidade: o sistema deve permitir ao usuário realizar a tarefa pretendida de modo mais fácil e eficiente, ou seja, adequada aos requisitos da tarefa do usuário. Neste nível, também faz parte da avaliação medir o desempenho do usuário junto ao sistema, ou seja, avaliar a eficiência do sistema na execução da tarefa pelo usuário.
- Avaliar o efeito da interface junto ao usuário: é preciso avaliar a usabilidade da interface, e isso inclui avaliar a facilidade de aprender a usar o sistema; a atitude do usuário com relação ao sistema; identificar áreas da interface que sobrecarregam o usuário de alguma forma etc.

Frequentemente, os métodos concentram a avaliação sobre aspectos de padrão de usabilidade, como o uso de *guidelines*.

- Identificar problemas específicos do sistema: avaliar a interface para identificar aspectos do *design* que podem levar a resultados inesperados, dúvidas ou confusão nos usuários quando utilizados em um contexto específico de uso. Contudo, isso está correlacionado tanto com a funcionalidade quanto com a usabilidade da interface.

Ainda de acordo com Rocha e Baranauskas (2003), atendendo a esses objetivos, podem-se classificar os métodos de avaliação em duas dimensões: se usuários reais estão ou não envolvidos e se o sistema (ou protótipo executável) está ou não implementado.

- Inspeção de usabilidade (avaliação preditiva): não envolve usuários e pode ser realizada em qualquer fase do desenvolvimento do sistema (com o sistema implementado ou não).
- Teste de usabilidade: método de avaliação centrado no usuário. Realizado em uma implementação real do sistema, podendo ser desde uma simulação da capacidade interativa do sistema, sem nenhuma funcionalidade, um protótipo básico implementando um cenário, ou até o sistema completamente implementado.

A inspeção de usabilidade pode ser realizada para: remover alguns dos principais problemas antes de se efetuar testes com o usuário; comparar ideias alternativas de projeto para verificar se este atende a padrões e guias de estilo; ajudar na preparação de testes com o usuário com questões não esclarecidas no processo de inspeção etc. Os testes com o usuários precisam ser feitos em casos de decisões de risco elevado que envolvam riscos a vidas humanas, ao meio ambiente etc., bem como em casos nos quais existam fatores considerados críticos para os usuários que não podem ser mensurados com inspeções (LIESENBERG, 2005).

7.2 Inspeção de usabilidade

Inspeção de usabilidade é um conjunto de métodos utilizados para avaliação de aspectos relacionados à usabilidade de uma interface de usuário. É realizada por avaliadores especialistas em usabilidade, por desenvolvedores de *software* habilitados, por especialistas em um determinado padrão de interface etc.

7.2.1 Avaliação heurística

Avaliação heurística é um método de inspeção de usabilidade utilizado para encontrar problemas de projeto de interfaces. Tem como base princípios de usabilidade amplamente reconhecidos e aceitos denominados heurísticas. Avaliação heurística é um método considerado econômico, de fácil aprendizagem e fácil de ser aplicado.

Este método é realizado por diversos avaliadores. O motivo para isso é que uma única pessoa não poderia encontrar todos os problemas, ao passo que diferentes pessoas encontrarão problemas parecidos e distintos.

Neste método, as sessões de avaliação individual duram de normalmente uma a duas horas. O avaliador navega pela interface pelo menos duas vezes (uma vez para obter uma percepção mais global de possíveis fluxos de interações e o escopo do sistema e uma segunda vez para analisar elementos específicos da interface) e inspeciona diversos elementos de diálogo e os compara com os princípios aceitos e possivelmente outros que, na sua experiência, se apliquem (LIESENBERG, 2005).

Ao encontrar algum problema, este é relatado e associado às heurísticas de usabilidade que foram violadas.

Os problemas encontrados pelos avaliadores são discutidos em grupo com o objetivo de formar uma avaliação consolidada no final.

A avaliação heurística não deve substituir os testes de usabilidade. Os problemas identificados em uma avaliação heurística são diferentes daqueles encontrados em um teste de usabilidade.

7.3 Teste de usabilidade

O teste de usabilidade é utilizado para avaliação de um produto ou serviço envolvendo os usuários. Durante um teste, os participantes tentarão completar tarefas típicas passadas a eles enquanto observadores assistem, ouvem e fazem anotações. O objetivo da aplicação dos testes de usabilidade é identificar os problemas, coletar dados qualitativos e quantitativos e medir a satisfação do participante com o produto (USA, [s.d.]).

Antes de aplicar um teste de usabilidade, é necessário desenvolver um sólido plano de teste, recrutar os participantes e, em seguida, analisar e relatar suas descobertas.

De acordo com Rocha e Baranauskas (2003), para aplicar um teste de usabilidade, deve ser desenvolvido um plano detalhado de teste no qual, dentre outras mais específicas, as seguintes questões devem ser respondidas:

- O objetivo do teste: o que se deseja obter?
- Quando e onde o teste irá acontecer?
- Qual a duração prevista de cada sessão de teste?
- Qual o suporte computacional necessário?
- Qual *software* precisa estar à disposição?
- Qual deverá ser o estado do sistema no início do teste?
- Quem serão os experimentadores?
- Quem serão os usuários e como serão conseguidos?

- Quantos usuários são necessários?
- Quais as tarefas que serão solicitadas aos usuários?
- Qual critério será utilizado para definir que os usuários terminaram cada tarefa corretamente?
- Quanto o experimentador poderá ajudar o usuário durante o teste?
- Quais dados serão coletados e como serão analisados uma vez que tenham sido coletados?
- Qual o critério para determinar que a interface é um sucesso?

O teste de usabilidade permite identificar problemas antes de a interface ser desenhada e implantada, tornando menores os custos das correções. Entretanto, os testes de usabilidade não são apenas uma atividade do processo de desenrolamento da interface. Os testes devem ser realizados, e os resultados, aplicados.

7.3.1 Planejamento do teste de usabilidade

O objetivo do plano de teste de usabilidade é documentar o que será feito, como será conduzido o teste, quais métricas serão capturadas, o número de participantes no teste e quais os cenários que serão utilizados.



Observação

O cenário (contexto para usuários específicos alcançarem seus objetivos, ou seja, realizarem suas tarefas), em testes de usabilidade, não deve incluir qualquer informação sobre como realizar uma tarefa. O teste de usabilidade irá mostrar como o participante (usuário) realiza uma determinada tarefa e mostrará se a interface facilita a realização dessa tarefa.

Elementos de um plano de teste

Um plano de teste de usabilidade deverá conter os elementos que seguem (USA, [s.d.]a):

Escopo: dê um nome à aplicação, *site* ou outro produto qualquer que será testado. Especifique quanto do produto, dos módulos ou dos recursos será coberto pelo teste.

Objetivo: identifique detalhadamente os objetivos das sessões de teste, bem como suas metas.

Cronograma e local: indique quando (e quantas sessões você vai realizar em um dia) e onde os testes serão realizados.

Sessões: descreva as sessões e o tempo de duração (geralmente de 60 a 90 minutos).

Equipamento: indique o tipo de equipamento que será utilizado no teste (*desktop*, *laptop*, *smartphone*, *tablet* etc.). Especificar informações sobre o tamanho do monitor e a resolução, o sistema operacional, o navegador etc. Indicar também se as sessões serão gravadas ou se algum equipamento/ferramenta será necessário para apoiar o teste.

Participantes: indique o número e o perfil dos usuários que participarão dos testes. Descreva como esses participantes foram ou serão recrutados.

Cenários: indique o número e os tipos de tarefas incluídas nos cenários de teste. Tipicamente, para 60 minutos de teste, cerca de dez (mais ou menos dois) cenários de testes para sistemas de *desktop* ou *laptop* e oito (mais ou menos dois) cenários de testes para sistemas de dispositivos móveis (como *tablet* e *smartphone*).

Métricas: subjetivas, como facilidade de uso e satisfação em utilizar o sistema. Definir perguntas que serão feitas aos participantes antes do início das sessões, depois da conclusão de cada cenário de testes e após a conclusão das sessões de testes.

Métricas quantitativas: indique os dados quantitativos que serão medidos nos testes (por exemplo, as taxas de tarefas concluídas corretamente, as taxas de erro, o tempo para realizar a tarefa).

Funções: inclua uma lista das pessoas que participarão dos testes de usabilidade e qual o papel que cada uma irá desempenhar. O especialista em usabilidade deve ser o facilitador das sessões.



Saiba mais

Um exemplo de plano de teste de usabilidade pode ser encontrado em:

BRIGHT, L.; GILG, R.; NODLER, H. *Usability test plan*. Austin: University of Texas, 2005. Disponível em: <http://www.rachaelgilg.com/docs/testplan_final.pdf>. Acesso em: 6 mar. 2015.

7.4 Escala de Usabilidade do Sistema — *System Usability Scale (SUS)*

A *System Usability Scale (SUS)* – Escala de Usabilidade do Sistema – fornece uma ferramenta “rápida e suja”, confiável para medir a usabilidade. Consiste em um questionário de dez perguntas com cinco opções de resposta (de “Discordo completamente” até “Concordo completamente”) para os participantes e permite avaliar uma grande variedade de produtos e serviços, incluindo *hardware*, *software*, dispositivos móveis, *websites* e aplicações. A SUS não é um diagnóstico – seu uso tem como objetivo classificar a facilidade de uso do *site*, aplicativo ou ambiente a ser testado.



Observação

A avaliação "rápida e suja" é utilizada em qualquer estágio do projeto para obtenção de *feedback* informal dos usuários e consultores com o objetivo de confirmar se as ideias dos projetistas atendem às necessidades dos usuários. É uma avaliação muito usada no *webdesign*. O termo "rápida e suja" é utilizado em razão de a avaliação ser realizada em curto espaço de tempo.

Para utilizar a SUS, os participantes precisam responder às dez perguntas com um X na resposta equivalente ("Discordo completamente", "Discordo", "Neutro", "Concordo" e "Concordo completamente").

Embora a interpretação da pontuação seja um pouco complexa, a escala pode ser bem eficiente. Para calcular a pontuação SUS, o primeiro passo é somar a pontuação de cada resposta. A pontuação das respostas vai de 0 a 4. Para as respostas 1, 3, 5, 7 e 9, a pontuação é a posição da escala menos 1. Para as respostas 2, 4, 6, 8 e 10, a pontuação é de 5 menos a posição de escala. O segundo passo é somar a pontuação e multiplicar o resultado por 2,5 para obter o valor total.

Segundo USA ([s.d.]b), uma pontuação SUS acima de 68 seria considerada acima da média, e abaixo desse número seria considerada abaixo da média.

System Usability Scale	
	Discordo completamente Concordo completamente
1. Acho que gostaria de usar este sistema frequentemente.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 1 2 3 4 5
2. Achei o sistema desnecessariamente complexo.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 1 2 3 4 5
3. Achei que foi fácil usar o sistema.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 1 2 3 4 5
4. Acho que eu iria precisar da ajuda de um técnico para conseguir usar este sistema.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 1 2 3 4 5
5. Achei que as várias funções deste sistema foram bem-integradas.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 1 2 3 4 5
6. Achei que havia muita inconsistência neste sistema.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 1 2 3 4 5
7. Imagino que a maioria das pessoas aprenderia a usar este sistema muito rapidamente.	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 1 2 3 4 5

8. Achei o sistema muito incômodo de usar.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
9. Eu me senti muito confiante usando o sistema.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5
10. Eu precisava aprender muitas coisas antes de utilizar este sistema.	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	1	2	3	4	5

Adaptado de: Brooke ([s.d.], p. 4).



Saiba mais

O documento original da escala SUS de usabilidade, proposta por John Brooke, com detalhes de aplicação e um exemplo de cálculo da pontuação, pode ser encontrado em:

BROOKE, J. *SUS: a quick and dirty usability scale*. Earley: Redhatch Consulting, [s.d.]. Disponível em: <http://cui.unige.ch/isi/icle-wiki/_media/ipm:test-suschart.pdf>. Acesso em: 6 mar. 2015.



Lembrete

Para garantir o sucesso do produto em desenvolvimento, as avaliações de usabilidade não são opcionais, mas sim obrigatórias.

7.5 Integração e implantação final

Agora que a interface atingiu os níveis definidos de usabilidade, deverá ser integrada à aplicação.

Uma vez que a implementação e integração tenha ocorrido, uma avaliação adicional pode ser realizada para assegurar que todas as questões que se relacionam ao uso do sistema integrado sejam abordadas. Finalmente, uma vez que o sistema computacional for liberado para os usuários, uma avaliação do seu uso em contextos reais poderá ser altamente benéfica. Ambas as avaliações finais podem servir de base para a compreensão dos usuários, de suas tarefas e contextos, bem como para o processo de *design*. Ainda que os resultados não sejam utilizados na atual versão do sistema computacional, poderão ser usados para melhorar as versões futuras.

Embora as dificuldades possam ser maiores para a realização de avaliações de usabilidade no ambiente real de uso, durante a fase de integração e implantação, quando possível, o objetivo é realizá-las.

No entanto, envolve também a atividade de realimentação dos usuários, ou seja, os usuários informam as dificuldades de uso encontradas após a implantação do produto. Essas informações servem como base para melhoria ou novas versões da interface com o usuário e para novos projetos.

8 DIRETRIZES PARA PROJETOS DE INTERFACE COM O USUÁRIO

Diretrizes para projetos de interfaces são conjuntos de regras com informações e recomendações que visam padronizar decisões de projeto de interface com o intuito de tornar as interfaces mais consistentes, contribuindo, portanto, para a melhoria do nível de usabilidade. Foram desenvolvidas para ajudar os projetistas no desenvolvimento de interfaces de usuários com usabilidade. Esse conjunto de regras, também conhecido como *guidelines*, não deve ser entendido como uma receita de projeto. Deve ser visto como um conjunto de recomendações e princípios norteadores do projeto e deve ser aplicado de forma contextualizada.

Diversos pesquisadores publicaram importantes diretrizes de projetos de interface que são aplicáveis à maioria dos sistemas interativos. Algumas das diretrizes mais relevantes serão apresentadas nas próximas seções.

8.1 As regras de ouro de Mandel

Em 1997, Theo Mandel apresentou em seu livro sobre projeto de interfaces três regras de ouro, formando uma base para um conjunto de princípios para o projeto de interfaces de usuário. São as expostas a seguir.

Deixar o usuário no comando

Os projetistas de interface não devem introduzir restrições e limitações de interação para simplificar a implementação da interface. Theo Mandel define uma série de princípios de projeto que permitem a um usuário manter o controle:

I – Defina modos de interação para não forçar o usuário a realizar ações desnecessárias ou indesejadas. Modo de interação é o estado atual da interface no momento de sua utilização. Por exemplo: o usuário pode escolher o comando de correção ortográfica no menu de um processador de texto para apenas fazer uma edição de uma frase. Ele deve conseguir entrar nesse modo e sair dele com pouco ou nenhum esforço.

II – Proporcione interação flexível. O *software* deveria permitir a um usuário interagir por meio de comandos de teclado, movimentação do *mouse* ou, por exemplo, por comandos de reconhecimento de voz (embora nem toda ação seja suscetível a todo mecanismo de interação).

III – Possibilite que a interação de usuário possa ser interrompida e desfeita. O usuário deve ser capaz de interromper uma sequência de ações para realizar alguma outra sem perder o trabalho que já havia feito. Também deve ser capaz de "desfazer" qualquer ação.

IV – Simplifique a interação à medida que os níveis de competência avançam e permita que a interação possa ser personalizada. Para os usuários que realizam a mesma sequência de interações repetidamente, é interessante criar um mecanismo de "macros" que permita ao usuário avançado personalizar a interface para facilitar sua interação.

V – Oculte os detalhes técnicos de funcionamento interno do usuário casual. O usuário não deve se preocupar com o sistema operacional ou com alguma outra tecnologia computacional enigmática para ele.

VI – Projete para interação direta com objetos que aparecem na tela. Uma interatividade direta com objetos na tela, como permitir ao usuário redimensionar o objeto, como aumentar e diminuir a escala de seu tamanho, passa a sensação de controle para o usuário.

Reduzir a carga de memória do usuário

Uma interface bem-projetada não exaure a memória do usuário. Para reduzir a carga de memória do usuário, as recomendações de princípios de projeto definidos por Mandel (1997) são:

I – Reduza a demanda de memória recente. A interface deve ser projetada para minimizar a exigência de recordar ações, entradas e resultados passados. Uma interface bem-projetada não deve sobrecarregar a memória do usuário. A interface deve proporcionar um cenário que ajude o usuário a lembrar.

II – Estabeleça *defaults* significativos. A configuração, ou conjunto de parâmetros iniciais (*defaults*), deve ser adequada para um usuário comum. No entanto, o usuário deve ter a opção de especificar suas preferências individuais. Uma opção de *reset* na interface também é necessária para que se permita o restabelecimento da configuração padrão original.

III – Defina atalhos intuitivos. Utilize mnemônicos para realizar alguma função de sistema (por exemplo, "alt-P" para chamar a função de impressão). O mnemônico deve estar ligado a uma ação que seja facilmente memorizada pelo usuário (por exemplo, a primeira letra da tarefa a ser solicitada).

IV – O *layout* visual da interface deve se basear na metáfora do mundo real. Isso permitirá ao usuário que se apoie em indicações visuais bem-compreensíveis do seu mundo físico, em vez de ter de memorizar uma sequência de interações.

V – Revele as informações de maneira progressiva. A interface deve ser organizada hierarquicamente, abstraindo informações, e apresentar os seus detalhes conforme o necessário e as solicitações de interesse do usuário.

Tornar a interface consistente

A interface de usuário deve apresentar e obter informações de forma consistente. Para isto, Mandel (1997) recomenda:

I – Permita ao usuário inserir a tarefa atual em um contexto significativo. É importante que a interface permita que o usuário se situe no contexto da ação. Isto é possível por meio de indicadores como títulos para as janelas, ícones gráficos, imagens, sistema de cores consistente etc. A interface precisa também possibilitar que o usuário saiba de onde ele veio e quais alternativas existem para transição para uma nova tarefa.

II – Mantenha a consistência ao longo de uma família de aplicações. Cada produto de *software* do conjunto de aplicações deve implementar as mesmas regras de projeto de modo que a consistência da interface e as expectativas criadas anteriormente sejam mantidas para toda a interação.

III – Se modelos interativos anteriores tiverem criado expectativa nos usuários, não faça alterações a menos que haja uma forte razão para isso. Não altere as sequências de interação utilizadas pelos usuários quando já tiverem se tornado padrão, como exemplo, o "alt-S" para salvar um arquivo, pois o usuário pressupõe que essa sequência vá acontecer na nova aplicação.

8.2 As oito regras de ouro de Shneiderman

As oito "regras de ouro" publicadas por Shneiderman são aplicáveis à maioria dos projetos de sistemas interativos. Essas oito regras de ouro devem ser interpretadas, refinadas e estendidas para cada domínio específico. Eles têm as suas limitações, mas fornecem um bom ponto de partida para projetos de interface de usuário de computadores, celulares e *websites*.

Estes princípios são derivados da experiência e refinados ao longo de três décadas (SHNEIDERMAN; PLAISANT, 2010).

1 – Esforçar-se para manter a consistência

Sequências consistentes de ações devem ser exigidas em situações semelhantes. Terminologia idêntica deve ser utilizada em instruções, menus e telas de ajuda. Por fim, cores consistentes, *layout* e fontes devem ser utilizados em todo o sistema.

2 – Atender à usabilidade universal

Reconhecer as necessidades da diversidade de usuários e projetar uma interface que atenda a essa diversidade. Diferenças entre usuários novatos e experientes, com diversas faixas etárias, portadores de necessidades especiais e uma enorme diversidade tecnológica, aumentam os requisitos para projetos de interface. Adicionar recursos de explicações de uso para os usuários sem experiência e recursos de atalhos para os mais experientes pode enriquecer o desenho da interface e melhorar a percepção da qualidade do sistema.

3 – Oferecer *feedback* informativo

Para cada ação do usuário, deve haver algum *feedback* (resposta) do sistema. Esse *feedback* deve ser proporcional à importância da ação. Para ações frequentes e simples, a resposta pode ser modesta. Para as ações esporádicas e mais relevantes, a resposta deve ser mais substancial.

4 – Projetar diálogos para encerrar as ações

Sequências de ações devem ser organizadas em grupos com começo, meio e fim. O *feedback* informativo na conclusão de um conjunto de ações dá ao usuário a convicção do que foi realmente realizado. Desse modo, o usuário saberá se pode avançar para sua próxima ação ou se deve rever e voltar à sua ação anterior.

5 – Evitar erros

Na medida do possível, projete o sistema de tal forma que os usuários não possam cometer erros graves. Se um usuário cometer um erro, a interface deverá detectar o erro e oferecer instruções simples, construtivas e específicas para corrigir o erro.

6 – Permitir facilmente a reversão de ações

Os projetistas devem procurar oferecer aos usuários meios satisfatórios de reverter suas ações. Este recurso reduz a ansiedade dos usuários que têm receio de utilizar alguns recursos do sistema achando que vai causar algum problema irreversível e incentiva a exploração de opções desconhecidas, já que poderão desfazer seus erros. Este recurso deve ser possível em vários pontos ao longo da realização de uma tarefa, como após uma única ação, após a entrada de dados ou após uma sequência de ações.

7 – Fornecer a sensação de controle ao usuário

Os usuários mais experientes desejam muito sentir que estão no comando do sistema e que o sistema responde às suas ações. Os usuários não gostam de mudanças no comportamento da interação com a qual já estão familiarizados, de sequências tediosas de entrada de dados, de dificuldade em obter informações necessárias e de incapacidade de conseguir o resultado desejado.

8 – Reduzir a carga de memória de curta duração

Em função da limitada capacidade dos seres humanos para o processamento de informações na memória de curta duração, os projetistas devem evitar interfaces que exijam dos usuários informações apresentadas em outras telas. As interfaces precisam ser, na medida do possível, simples, e o treinamento, com tempo mínimo suficiente, deve ser realizado para as ações de sequências complexas.



Saiba mais

Estas regras de ouro de projeto de Interface de usuário estão no livro *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Elas foram originalmente criadas em 1987 a partir de pesquisas feitas por Shneiderman sobre Interação Humano-Computador e vêm sendo aprimoradas pelo autor. São aplicáveis para a maioria dos sistemas interativos e podem ajudar os projetistas de interface a criarem interfaces de usuário com mais usabilidade.

SHNEIDERMAN, B.; PLAISANT, C. *Designing the user interface: strategies for effective human-computer interaction*. 5. ed. Reading: Addison-Wesley, 2010.

8.3 Dez heurísticas de usabilidade para projeto de interface de usuário

Jakob Nielsen, depois de analisar centenas de problemas de usabilidade, estabeleceu dez princípios gerais para projeto de interação (NIELSEN, 1995).

Visibilidade do estado do sistema

O sistema deve sempre manter os usuários informados do que está acontecendo, por meio de *feedback* apropriado em tempo hábil.

Correlação entre o sistema e o mundo real

O sistema deve apresentar uma analogia ao mundo real, usando uma linguagem com palavras, frases e conceitos familiares para o usuário, em vez de termos orientados ao sistema. Siga as convenções do mundo real, fazendo as informações aparecerem numa ordem natural e lógica.

Liberdade e controle do usuário

Usuários costumam escolher algumas funções do sistema por engano e precisarão identificar opções de saída e cancelar a ação realizada. O sistema deve ter mecanismos que possam desfazer e refazer as ações.

Consistência e padrões

Os usuários não devem ter de se perguntar se diferentes palavras, situações ou ações significam a mesma coisa. Siga as convenções de plataforma para que o usuário não tenha dúvidas.

Prevenção de erros

Melhor que um sistema com boas mensagens de erro é um projeto cuidadoso que impede que um erro ocorra. O sistema deve eliminar as condições passíveis de erros ou verificá-las e apresentar aos usuários uma opção de confirmação da ação.

Reconhecimento em vez de memorização

O usuário não deve ter de se lembrar de todas as funções, informações e de todos os objetos para poder interagir corretamente com o sistema. As instruções para utilização do sistema devem estar visíveis ou facilmente acessíveis sempre que necessário.

Flexibilidade e eficiência de uso

O sistema deve permitir que usuários possam personalizar ou configurar suas ações mais frequentes. Teclas aceleradoras (de atalho) devem ser implementadas para serem usadas por usuários mais experientes.

Projeto estético e minimalista

Os diálogos não devem conter informações irrelevantes. Toda a informação apresentada ao usuário (em caixa de diálogo, por exemplo) deve ser objetiva, de forma que não confunda ou tire a atenção do usuário.

Suporte para o usuário no reconhecimento, no diagnóstico e na recuperação de erros

As mensagens de erro devem ser expressas em linguagem simples (sem códigos), indicar com precisão o problema e sugerir uma solução construtiva.

Ajuda e documentação

Uma documentação de consulta fácil e com linguagem simples e objetiva deve estar à disposição do usuário. A documentação não pode ser muito extensa e deve ter o foco nas tarefas do usuário.

8.4 Mais princípios de projeto de interface com o usuário

Ian Sommerville (2007) apresenta uma lista de princípios de projeto de interface com o usuário que podem ser aplicáveis a todos os projetos de interface.

Quadro 8 – Princípios de projeto de interface com o usuário

Princípio	Descrição
Familiaridade de usuário	A interface deve usar termos e conceitos relacionados à experiência dos usuários que farão mais uso do sistema. Os usuários não devem ser forçados a se adaptar a uma interface malprojetada.
Consistência	A interface deve manter a consistência, ou seja, manter os mesmos formato e modo de funcionamento de menus, comandos etc. Assim, sempre que possível, as operações comparáveis devem ser ativadas da mesma maneira, reduzindo o tempo de aprendizagem dos usuários.
Surpresa mínima	Os usuários nunca devem ser surpreendidos pelo comportamento de um sistema. Os projetistas devem assegurar que as interfaces sempre se comportem de acordo com o modelo mental do usuário.

Facilidade de recuperação	A interface deve incluir mecanismos que permitam aos usuários se recuperarem de erros. Partindo do princípio de que é inevitável que o usuário cometa algum erro em algum momento, a interface deve minimizar esses erros e, quando acontecer, disponibilizar recursos para que os usuários se recuperem dos erros cometidos. Esses recursos podem ser: confirmação de ações destrutivas; disponibilidade de recurso do tipo "refazer"; <i>checkpointing</i> (gravar o estado do sistema periodicamente e permitir que o sistema seja reiniciado a partir de um determinado <i>checkpoint</i>).
Guia de usuário	A interface deve fornecer <i>feedback</i> significativo quando ocorrerem erros e fornecer recursos sensíveis ao contexto para ajudar o usuário. A interface deve ter a opção de "ajuda ao usuário" integrada ao sistema, com diferentes níveis de ajuda e orientação (das mais simples às mais complexas).
Diversidade de usuário	A interface deve fornecer recursos de interação adequados para tipos diferentes de usuários de sistema. Os usuários casuais precisam de um guia de utilização. Os mais experientes precisam de atalhos para interagir mais rapidamente.

Adaptado de: Sommerville (2007)

Dennis, Wixom e Roth (2014) reuniram alguns princípios do projeto de interface com o usuário. De acordo com os autores, o maior problema enfrentado por projetistas experientes é utilizar o espaço de forma eficiente, ou seja, apresentar a elevada quantidade de informações num espaço restrito. O desafio para os projetistas é equilibrar a necessidade de uma interface com aspecto simples e agradável em relação à necessidade de apresentar as informações em diversas páginas ou telas, o que reduz a simplicidade. O quadro a seguir apresenta alguns princípios fundamentais do *design* da interface comuns para o *design* de navegação, de entrada e de saída.

Quadro 9 – Princípios de projeto de interface com o usuário

Princípio	Descrição
<i>Layout</i>	A interface deve ser uma série de áreas na tela usada consistentemente para diferentes finalidades — por exemplo, uma área na parte superior destinada a comandos e navegação, uma área central destinada a entrada e saída de informações e uma área na parte inferior destinada a informações sobre o <i>status</i> .
Conhecimento do conteúdo	Os usuários devem saber sempre onde estão no sistema e que informações estão sendo exibidas.
Estética	As interfaces devem ser funcionais e atraentes para os usuários por intermédio do uso cuidadoso de espaços em branco, cores e tipos de letras. Com frequência, há um equilíbrio entre a inclusão de espaços em branco, suficientes para tornar agradável o aspecto da interface, e a perda de muito espaço, de modo que informações importantes deixem de ser apresentadas na tela.
Experiência do usuário	Embora a facilidade de uso e a facilidade de aprendizado levem frequentemente a decisões de <i>design</i> semelhantes, algumas vezes há uma compensação entre as duas. Os usuários novatos ou esporádicos de <i>software</i> preferirão a facilidade de aprendizado, ao passo que os usuários frequentes preferirão a facilidade de uso.
Consistência	A consistência no <i>design</i> da interface permite que os usuários prevejam o que acontecerá antes de eles realizarem uma função. É um dos elementos mais importantes da facilidade de aprendizado, da facilidade de uso e da estética.
Minimização do esforço do usuário	A interface deve ser simples de usar. A maioria dos <i>designers</i> pretende que não sejam realizados mais de três cliques do <i>mouse</i> desde o menu inicial até ser executada alguma ação

Adaptado de: Dennis, Wixom e Roth (2014).



Lembrete

Diretrizes para projetos de interface com o usuário foram desenvolvidas para ajudar os projetistas no desenvolvimento de interfaces de usuário com elevados níveis de usabilidade. Devem ser entendidas como um conjunto de recomendações e princípios norteadores do projeto e devem ser aplicadas de forma contextualizada.

8.5 Design responsivo

Atualmente, uma variedade muito grande de dispositivos possui acesso à internet. Esses dispositivos, como *desktops*, *notebooks*, *tablets*, *smartphones*, televisores etc., possuem características diferentes, como o tamanho de tela.

A utilização de uma página *web* não adaptativa nestes dispositivos pode ser algo muito desagradável e frustrante para o usuário. Para resolver essa questão, os especialistas devem desenvolver aplicações com interfaces adaptáveis a diversos tamanhos de telas, proporcionando uma boa experiência ao usuário.

O *design* responsivo combina um conjunto de técnicas e tecnologias para melhorar a experiência do usuário independentemente do dispositivo, apresentando as páginas *web* com flexibilidade e com suporte aos diversos tamanhos de tela e resolução.

A figura a seguir representa o conceito de *design* responsivo. Ela apresenta a ideia de uma mesma página *web* em diversos dispositivos e tamanhos de tela sem deixar de apresentar as informações de forma adequada ao usuário.

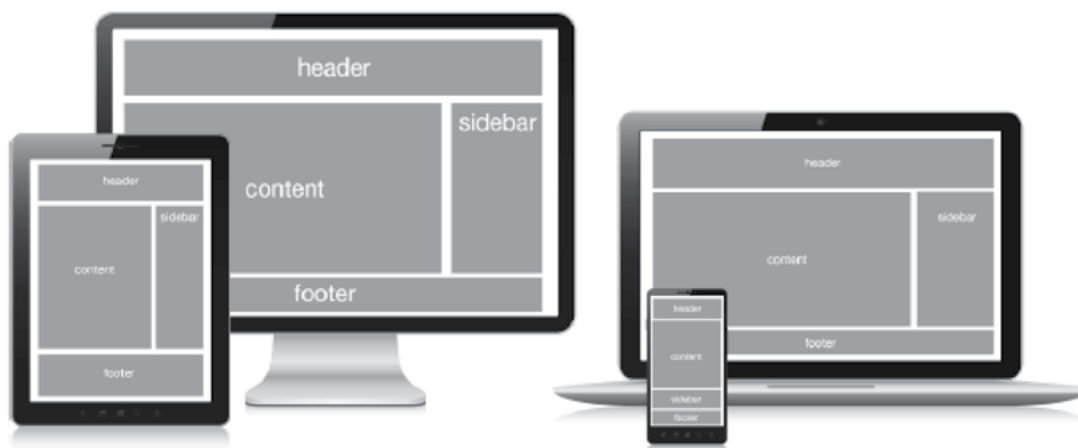


Figura 31 – Design responsivo: uma mesma página *web* em vários tamanhos de tela

O *design* responsivo está fundamentado em três tecnologias (ZEMEL, 2013):

- *Layout* fluido.
- Imagens e recursos flexíveis.
- *Media Queries*.



Saiba mais

Não faz parte do objetivo deste livro-texto o desenvolvimento de *design* responsivo. Entretanto, para um estudo mais aprofundado sobre as técnicas para se chegar a um *design* responsivo, leia:

ZEMEL, T. *Web design responsivo: páginas adaptáveis para todos os dispositivos*. São Paulo: Casa do Código, 2013.

Um *layout* fluido (ou *grid* flexível) requer que o desenho das páginas não tenha medidas fixas. Portanto, a construção dos projetos é realizada em uma grade para que o *layout* possa se ajustar a diferentes ambientes e tamanhos de tela. O resultado é uma adaptação "natural" e automática do que se apresenta na tela, evitando, portanto, barras de rolagem e/ou conteúdo "cortado" (apenas parte do conteúdo visível).

Do mesmo modo, as imagens (e outros recursos) precisam ser flexíveis. As imagens podem se adaptar (aumentar ou diminuir) dentro de uma coluna de grade flexível.

Por meio de *Media Queries*, que têm por finalidade servir uma folha de estilo específica para uma determinada mídia mediante consulta e identificação das características da mídia para a qual a aplicação está sendo servida (SILVA, 2014), é possível alterar, ocultar, fazer aparecer e reposicionar elementos conforme a resolução da tela ou à medida que o navegador é redimensionado. Adicionada no CSS3, *Media Queries* permite aplicar diferentes regras de estilo CSS, podendo adaptar a exibição da página para diferentes resoluções de tela.



Saiba mais

Media Queries são essenciais para a criação de *design* responsivo. O documento de recomendação do W3C (*World Wide Web Consortium*) com especificação CSS3 para *Media Queries* pode ser encontrado em:

W3C (WORLD WIDE WEB CONSORTIUM). *Media Queries: W3C recommendation*. São Paulo: W3C, 2012. Disponível em: <<http://www.w3.org/TR/css3-mediaqueries/>>. Acesso em: 24 fev. 2015.

PROJETO DE INTERFACE COM O USUÁRIO

As figuras a seguir apresentam exemplos de páginas web com *design* responsivo. Elas evidenciam a diferença quando as páginas são acessadas por outros dispositivos com resoluções de tela diferentes.



Figura 32 – Mr. Simon Collison: página web responsiva

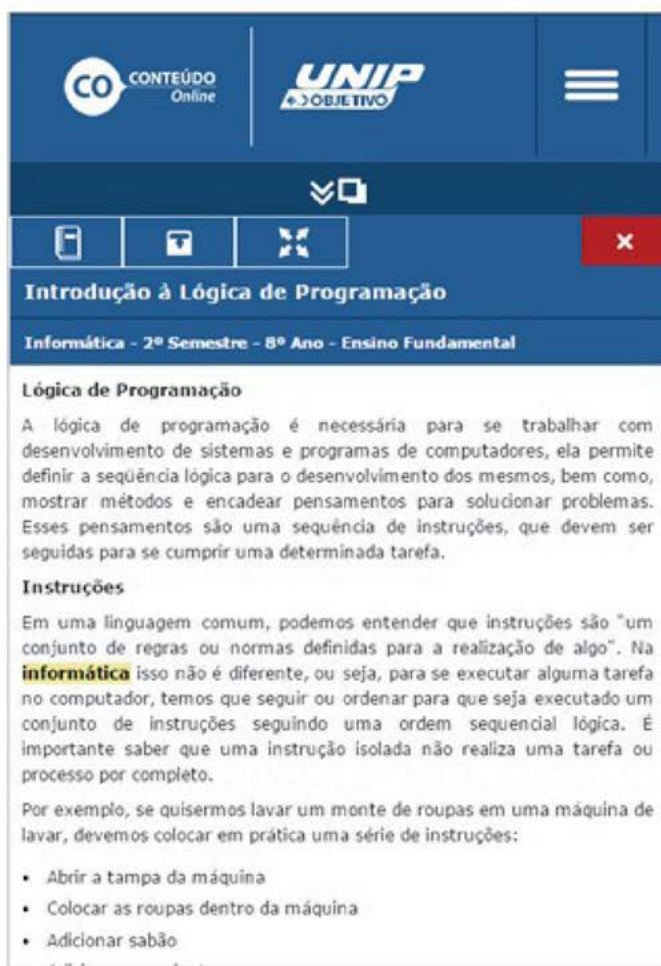
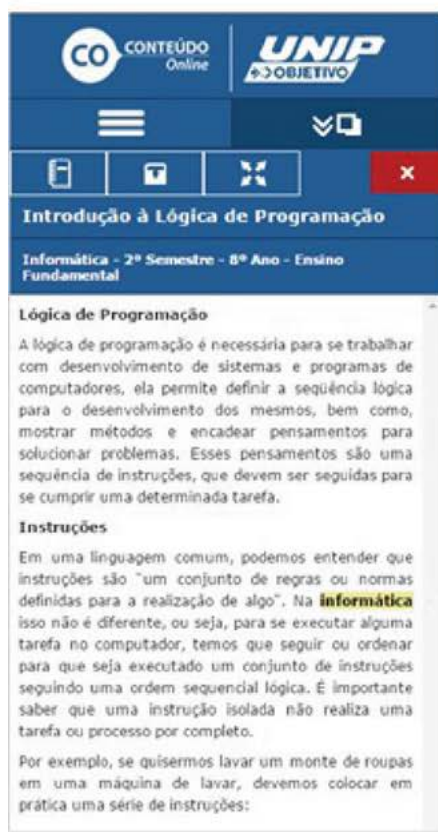
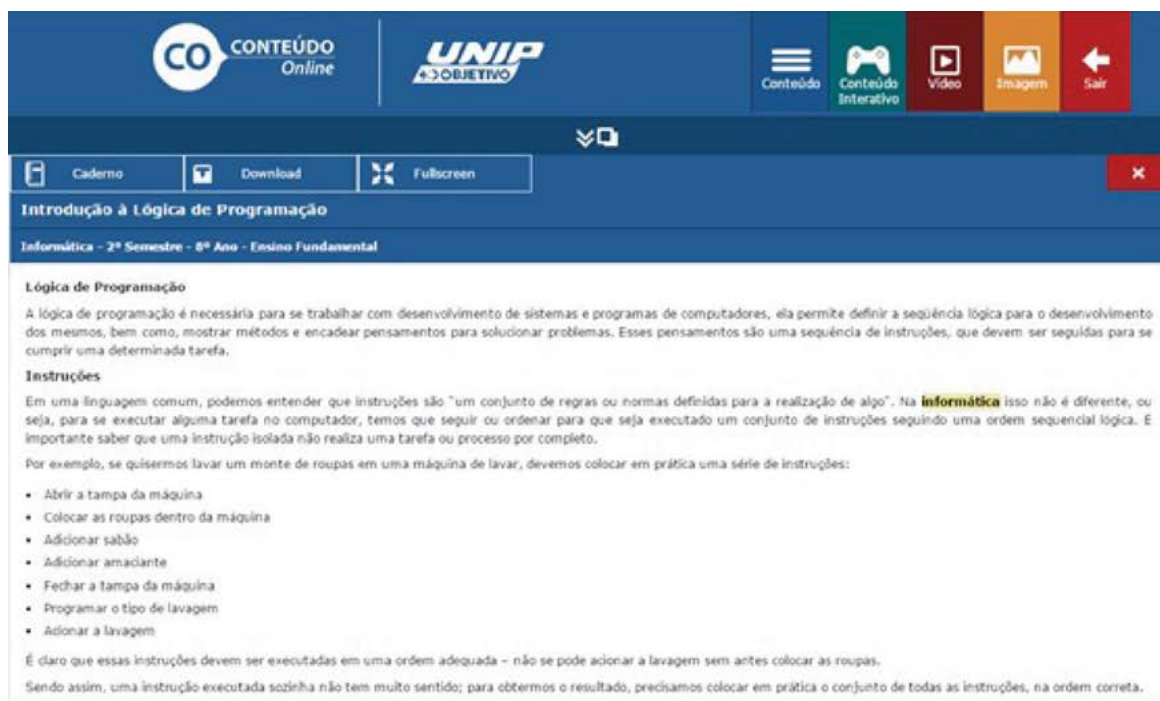


Figura 33 – Conteúdo on-line: página web responsiva



Resumo

Vimos que não basta o projetista seguir *guidelines* para projetos de interface, tampouco devem fazer avaliações apenas na fase final do processo de projeto de interface, "se der tempo". Avaliações podem ser realizadas durante todo o ciclo de desenvolvimento da interface, desde as fases iniciais do projeto, em que são produzidos os primeiros esboços, até os estágios mais avançados, já com o *software* ou protótipo executável.

Aprendemos que os principais objetivos das avaliações são analisar a funcionalidade do sistema, avaliar o efeito da interface junto ao usuário e identificar problemas específicos do sistema.

Vimos também que inspeção de usabilidade é um conjunto de métodos utilizados para avaliação de aspectos relacionados à usabilidade de uma interface e não envolve usuários. Pode ser realizada em qualquer fase do desenvolvimento do sistema e normalmente é realizada por avaliadores especialistas em usabilidade.

Em seguida, abordamos a avaliação heurística é um método de inspeção de usabilidade utilizavam para encontrar problemas de projeto de interfaces. Tem como base princípios de usabilidade amplamente reconhecidos e aceitos denominados heurísticas. É um método considerado econômico, de fácil aprendizagem e fácil de ser aplicado. Entretanto, não deve substituir os testes de usabilidade.

Aprendemos ainda que o teste de usabilidade é utilizado para avaliação de um produto ou serviço envolvendo os usuários. É obrigatório em casos críticos para os usuários, em situações que envolvam riscos a vidas humanas, ao meio ambiente, dentre outros. O teste de usabilidade permite identificar problemas antes de a interface ser implantada, tornando menores os custos das correções. Para aplicar um teste de usabilidade, é necessário desenvolver um consistente plano de teste, que tem o objetivo de documentar o que será feito, como será conduzido o teste, quais métricas serão capturadas, o número de participantes no teste e quais os cenários que serão utilizados.

Vimos que depois de a interface atingir os níveis definidos de usabilidade, deverá ser integrada à aplicação. Uma avaliação adicional pode ser realizada para assegurar todas as questões que se relacionem ao uso do sistema. Depois que o sistema computacional for liberado para os usuários, uma avaliação do seu uso em contextos reais poderá ser altamente benéfica.

Vimos também que existem diversas diretrizes para projetos de interface com o usuário. Diretrizes para projetos de interface são conjuntos de regras com informações e recomendações que visam padronizar decisões de projeto de interface com o intuito de tornar as interfaces mais consistentes, contribuindo, portanto, para a melhoria do nível de usabilidade. Foram desenvolvidas para ajudar os projetistas no desenvolvimento de interfaces de usuários com usabilidade. Esse conjunto de regras, também conhecido como *guidelines*, não deve ser visto como uma receita de projeto. Deve ser entendido como um conjunto de recomendações e princípios norteadores do projeto e deve ser aplicado de forma contextualizada.

Aprendemos que diversos pesquisadores publicaram importantes princípios de projeto e heurísticas para projetos de interface que são aplicáveis à maioria dos sistemas interativos. Algumas "regras de ouro" formam uma base consolidada para um conjunto de princípios para o projeto de interfaces de usuário. Essas regras devem ser interpretadas, refinadas e estendidas para cada domínio específico. Têm as suas limitações, mas fornecem um bom ponto de partida para projetos de interface de usuário de computadores, celulares e *websites*.

Por fim, vimos que, diante de todos os desafios no projeto de interface com o usuário, a grande variedade disponível atualmente de dispositivos com acesso à internet e diferentes tamanhos de tela, como *desktops*, *notebooks*, *tablets*, *smartphones*, televisores etc., gera outro desafio aos projetistas. A utilização de uma página *web* não adaptativa nestes dispositivos pode ser algo muito desagradável e frustrante para o usuário. O *design* responsivo pode resolver essa questão. Combina um conjunto de técnicas e tecnologias para melhorar a experiência do usuário independentemente do dispositivo, apresentando as páginas *web* com flexibilidade e com suporte aos diversos tamanhos de tela e resolução.



Exercícios

Questão 1. (Enade 2014) Vivemos em um mundo de produtos de alta tecnologia e praticamente todos requerem interação humana. Para que um produto de software seja bem-sucedido, deve apresentar boa usabilidade. Se os mecanismos de interface tiverem sido bem projetados, o usuário flui suavemente através da interação usando um ritmo cadenciado que permite que o trabalho seja realizado sem grandes esforços. Entretanto, se a interface for mal concebida, o usuário se move aos trancos e barrancos, e o resultado será frustração e baixa eficiência no trabalho.

Três regras de ouro são a base para um conjunto de princípios para o projeto de interfaces do usuário:

- 1 – deixar o usuário no comando;

2 – reduzir a carga de memória do usuário;

3 – tornar a interface consistente.

PRESMAN, R.S. Engenharia de software: uma abordagem profissional. 7. ed. McGraw Hill, 2011. p. 287-288 (adaptado).

Com base nessas três regras, avalie as afirmativas a seguir.

I – Um sistema que permita ao usuário desfazer qualquer ação respeita a regra de ouro 1.

II – Um sistema de pagamento de contas que usa a imagem de um cartão de crédito para orientar o usuário pelo processo de pagamento de uma conta respeita a regra de ouro 2.

III – Um conjunto de aplicações ou produtos que implementam as mesmas regras de projeto de modo padronizado respeita a regra de ouro 3.

É correto o que se afirma em:

A) I, apenas.

B) II, apenas.

C) I e III, apenas.

D) II e III, apenas.

E) I, II e III.

Resposta correta: alternativa E.

Análise das afirmativas

I – Afirmativa correta.

Justificativa: permitir que o usuário desfça qualquer ação respeita a regra de ouro 1: "deixar o usuário no comando".

II – Afirmativa correta.

Justificativa: ao apresentar a imagem do cartão de crédito para orientar o usuário, vai-se ao encontro da regra de ouro 2: "reduzir a carga de memória do usuário".

III – Afirmativa correta.

Justificativa: a implementação das mesmas regras de modo padronizado reforça a consistência da interface, apresentada pela regra de ouro 3.

Questão 2. (Enade 2014) A verificação e a validação de uma interface de usuário ocorre em três pontos distintos: análise, projeto e teste. Considerando um cenário de uma aplicação web, tal verificação pode ser realizada através de testes de interface, testes de usabilidade e testes de compatibilidade.

PRESSMAN, R. Engenharia de software: uma abordagem profissional. 7. ed. Mc Graw Hill, 2011 (adaptado).

Nesse contexto, avalie as afirmativas a seguir.

I – O teste de interface experimenta mecanismos de interação e valida aspectos estéticos da interface do usuário, apontando erros específicos de interface e erros na maneira como a interface implementa as semânticas de navegação, funcionalidade ou exibição de conteúdo.

II – O teste de usabilidade avalia o grau com o qual os usuários podem interagir efetivamente com a aplicação e o grau em que a aplicação dirige as ações do usuário.

III – O primeiro passo no teste de compatibilidade é definir uma série de configurações típicas encontradas do lado cliente e suas respectivas variantes, identificando características como plataforma, sistema operacional e navegador.

É correto o que se afirma em:

- A) I, apenas.
- B) III, apenas.
- C) I e II, apenas.
- D) II e III, apenas.
- E) I, II e III.

Resolução desta questão na plataforma.

FIGURAS E ILUSTRAÇÕES

Figura 1

SOUZA, C. S. *et al.* Interação Humano-Computador: perspectivas cognitivas e semióticas. In: FUKS, H. (Org.). In: JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA, 1999, Rio de Janeiro. *Anais...* Rio de Janeiro: Edições EntreLugar, 1999. p. 420-70. Adaptada.

Figura 2

PRESSMAN, R. S. *Engenharia de software*. 6. ed. São Paulo: McGraw-Hill, 2011. p. 39.

Figura 3

SCHMIDT, M. E. C. *Implementing the IEEE software engineering standards*. Indianapolis: Sams, 2000. Adaptado.

Figura 4

SOMMERVILLE, I. *Engenharia de software*. 6. ed. São Paulo: Prentice Hall Brasil, 2003. p. 43.

Figura 5

SOMMERVILLE, I. *Engenharia de software*. 6. ed. São Paulo: Prentice Hall Brasil, 2003. p. 45.

Figura 6

DENNIS, A.; WIXOM, B. H.; ROTH, R. M. *Análise e projeto de sistemas*. 5. ed. Rio de Janeiro: LTC, 2014. Vitalbook file. Capítulo 2.

Figura 7

DENNIS, A.; WIXOM, B. H.; ROTH, R. M. *Análise e projeto de sistemas*. 5. ed. Rio de Janeiro: LTC, 2014. Vitalbook file. Capítulo 2.

Figura 8

SHARP, H.; ROGERS, Y.; PREECE, J. *Design de interação: além da interação homem-computador*. Porto Alegre: Bookman, 2005. p. 213.

Figura 9

ROCHA, H. V.; BARANAUSKAS, M. C. C. *Design e avaliação de interfaces humano-computador*. Campinas: NIED/UNICAMP, 2003. p. 123. Adaptado.

Figura 11

ABNT (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS). *NBR 9241-11*: requisitos ergonômicos para trabalho de escritório com computadores: parte 11 – orientação sobre usabilidade. Rio de Janeiro, 2002. p. 4.

Figura 12

ABNT. *NBR ISO/IEC 9126-1*: engenharia de software: qualidade de produto: parte 1 – modelo de qualidade. Rio de Janeiro, 2003. p. 11.

Figura 13

ISO (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION). *ISO 13407*: human-centred design process for interactive systems. Genebra, 1999. p. 9.

Figura 15

SHARP, H.; ROGERS, Y.; PREECE, J. *Design de interação*: além da interação homem-computador. Porto Alegre: Bookman, 2005. p. 253.

Figura 17

NORMAN, D. A. *The Design of Everyday Things*. Nova Iorque: Basic Books, 2002. p. 16. Adaptado.

Figura 18

MILETTO, E. M.; BERTAGNOLLI, S. C. *Desenvolvimento de Software II*: introdução ao desenvolvimento web com HTML, CSS, JavaScript e PHP. Porto Alegre: Bookman, 2014. p. 56.

Figura 20

DAVIES, M. Paper prototyping as a core tool in the design of cell phone user interfaces. *Id-Book.com*, Jun. 2011. Disponível em: <http://www.id-book.com/downloads/casestudy_11point3.pdf>. Acesso em: 5 mar. 2015. p. 9.

Figura 21

AMBLER, S. W. *Modelagem ágil*: práticas eficazes para a programação extrema e o processo unificado. Porto Alegre: Bookman, 2004. p. 112.

Figura 22

AMBLER, S. W. *Modelagem ágil*: práticas eficazes para a programação extrema e o processo unificado. Porto Alegre: Bookman, 2004. p. 113.

Figura 23

AMBLER, S. W. *Modelagem ágil: práticas eficazes para a programação extrema e o processo unificado*. Porto Alegre: Bookman, 2004. p. 114.

Figura 25

WALKER, M.; TAKAYAMA, L.; LANDAY, J. A. High-Fidelity or Low-Fidelity, Paper or Computer? Choosing Attributes When Testing Web Prototypes. In: HUMAN FACTORS AND ERGONOMICS SOCIETY ANNUAL MEETING, 46., 2002, Baltimore. *Proceedings...* Baltimore: HFES, 2002. p. 663.

Figura 26

WALKER, M.; TAKAYAMA, L.; LANDAY, J. A. High-Fidelity or Low-Fidelity, Paper or Computer? Choosing Attributes When Testing Web Prototypes. In: HUMAN FACTORS AND ERGONOMICS SOCIETY ANNUAL MEETING, 46., 2002, Baltimore. *Proceedings...* Baltimore: HFES, 2002. p. 663.

Figura 27

GREENBERG, S. et al. *Sketching user experiences: the workbook*. Waltham: Elsevier, 2011. p. 170.

Figura 28

MICROSOFT. Designing the Hilo User Experience. In: _____. Hilo: Developing C++ Applications for Windows 7. Seattle: Microsoft, 2010. p. 32-41. Disponível em: <<https://msdn.microsoft.com/en-us/library/windows/desktop/ff800706.aspx>>. Acesso em: 15 dez. 2014. p. 34.

Figura 31

OVERFIELD, E. et al. *Pro SharePoint 2013 Branding and Responsive Web Development*. Berkeley: Apress, 2013. p. 59.

Figura 32

ZEMEL, T. *Web design responsivo: páginas adaptáveis para todos os dispositivos*. São Paulo: Casa do Código, 2013. p. 6.

REFERÊNCIAS

Textuais

ABNT (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS). *NBR 9241-11: requisitos ergonômicos para trabalho de escritório com computadores: parte 11 – orientação sobre usabilidade*. Rio de Janeiro, 2002.

____. *NBR ISO/IEC 9126-1: engenharia de software: qualidade de produto: parte 1 – modelo de qualidade*. Rio de Janeiro, 2003.

ACM SIGCHI (ASSOCIATION FOR COMPUTING MACHINERY SPECIAL INTEREST GROUP ON COMPUTER-HUMAN INTERACTION). Curriculum Development Group. *ACM SIGCHI curricula for human-computer interaction*. Nova Iorque: ACM, 1992.

AGNER, L. Em busca de um olhar interdisciplinar sobre a arquitetura de informação, a usabilidade e a metacomunicação em dispositivos móveis com interfaces gestuais. In: SIMPÓSIO NACIONAL DA ABCIBER, 5., 2011, Florianópolis. *Anais...* Florianópolis: UDESC/UFSC, 2011. Disponível em: <<http://abciber.org.br/simposio2011/anais/Trabalhos/artigos/Eixo%202/3.E2/258-398-1-RV.pdf>>. Acesso em: 25 fev. 2015.

AMBLER, S. W. *Modelagem ágil: práticas eficazes para a programação extrema e o processo unificado*. Porto Alegre: Bookman, 2004.

BARBOSA, S. D. J.; SILVA, B. S. *Interação humano-computador*. Rio de Janeiro: Elsevier, 2011.

BEVAN, N. International standards for HCI and usability. *International Journal of Human Computer Studies*, Londres, v. 55, n. 4, p. 533-52, out. 2001. Disponível em: <http://www.nigelbevan.com/papers/HCI-Usability_standards.pdf>. Acesso em: 2 mar. 2015.

____. Quality in use: meeting user needs for quality. *Journal of Systems and Software*, Nova Iorque, v. 49, n. 1, p. 89-96, dez. 1999.

BEVAN, N., BOGOMOLNI, I. Incorporating user quality requirements in the software development process. In: INTERNATIONAL SOFTWARE & INTERNET QUALITY WEEK CONFERENCE, 4., 2000, Bruxelas. *Proceedings...* São Francisco: Software Research, 2000. p. 1192-204.

BIAS, R. G.; MAYHEW, D. J. (Ed.). *Cost-justifying usability*. São Francisco: Morgan Kaufmann, 1994.

BOEHM, B. A spiral model for software development and enhancement. *IEEE Computer*, Los Alamitos, v. 21, n. 5, p. 61-72, maio 1988.

BRASIL. Presidência da República. Casa Civil. *Decreto nº 5.296 de 2 de dezembro de 2004*. Regulamenta as Leis nºs 10.048, de 8 de novembro de 2000, que dá prioridade de atendimento às pessoas que especifica, e 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida, e dá outras providências. Brasília, 2004. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2004/decreto/d5296.htm>. Acesso em: 24 fev. 2015.

BRIGHT, L.; GILG, R.; NODLER, H. *Usability test plan*. Austin: University of Texas, 2005. Disponível em: <http://www.rachaelgilg.com/docs/testplan_final.pdf>. Acesso em: 6 mar. 2015.

- BROOKE, J. *SUS: a quick and dirty usability scale*. Earley: Redhatch Consulting, [s.d.]. Disponível em: <http://cui.unige.ch/isi/icle-wiki/_media/ipm:test-suschapt.pdf>. Acesso em: 6 mar. 2015.
- BROWN, J. *Methodologies for the creation of interactive software*. Wellington: Victoria University of Wellington, 1996. Technical Report CS-TR-96/1.
- COSTA, I. et al. *Qualidade em tecnologia da informação: conceitos de qualidade nos processos, produtos, normas, modelos e testes de software no apoio às estratégias empresariais*. São Paulo: Atlas, 2013.
- COSTABILE, M. F. Usability in the software life cycle. In: CHANG, S. K. (Ed.). *Handbook of software engineering and knowledge engineering*. World Scientific, Cingapura, v. 1, p. 179-92, 2001.
- DAVIES, M. Paper prototyping as a core tool in the design of cell phone user interfaces. *Id-Book.com*, Jun. 2011. Disponível em: <http://www.id-book.com/downloads/casestudy_11point3.pdf>. Acesso em: 5 mar. 2015.
- DENNIS, A.; WIXOM, B. H.; ROTH, R. M. *Análise e projeto de sistemas*. 5. ed. Rio de Janeiro: LTC, 2014. Vitalbook file.
- DIX, A. et al. *Human-computer interaction*. 3. ed. Harlow: Pearson, 2004.
- FAULKNER, X.; CULWIN, F. Enter the usability engineer: integrating HCI and software engineering. In: ANNUAL SIGCSE/SIGCUE ITICSE CONFERENCE ON INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION, 5., 2000, Helsinki. *Proceedings...* Nova Iorque: ACM Press, 2000. p. 61-64.
- FERRÉ, X. et al. Usability basics for software developers. *IEEE Software*, Los Alamitos, v. 18, n. 1, p. 22-9, Jan./Feb. 2001. Disponível em: <http://lucio.ls.fi.upm.es/xavier/papers/usability_b.pdf>. Acesso em: 24 fev. 2015.
- FOINA, P. *Tecnologia de informação: planejamento e gestão*. 3. ed. Atlas, 2013.
- GRANOLLERS, T. User centred design process model: integration of usability engineering and software engineering. In: IFIP TC13 INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION – INTERACT, 9., 2003, Zurique. *Proceedings...* Zurique: Ifip, 2003.
- GREENBERG, S. et al. *Sketching user experiences: the workbook*. Waltham: Elsevier, 2011.
- HOLTZBLATT, K. Customer-centered design for mobile applications. *Personal Ubiquitous Comput*, Londres, v. 9, n. 4, p. 227-37, jul. 2005.
- IBM. *Documento de visão*. [s.d.]. Disponível em: <http://www-01.ibm.com/support/knowledgecenter/SSCP65_4.0.7/com.ibm.rational.rrm.help.doc/topics/r_vision_doc.html?lang=pt-br>. Acesso em: 3 mar. 2015.

____. *Understanding acessibility*. [s.d.]. Disponível em: <http://www-03.ibm.com/able/access_ibm/disability.html>. Acesso em: 24 fev. 2015.

IEEE (INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS). *Std 610.12-1990 (R2002)*: IEEE Standard Glossary of Software Engineering Terminology. Nova lorque, 1990.

ISO (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION). *ISO 13407*: human-centred design process for interactive systems. Genebra, 1999.

LEITE, J. C. *Modelos e formalismos para a engenharia semiótica de interfaces usuário*. 1998. 205 f. Tese (Doutorado em Ciências em informática)– Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio, Rio de Janeiro, 1999. Disponível em: <ftp://ftp.inf.puc-rio.br/pub/docs/theses/98_PhD_leite.pdf>. Acesso em: 24 fev. 2015.

LIESENBERG, H. *Processo áureo: um guia para o projeto de interfaces de usuários*. Campinas: Unicamp, 2005.

MANDEL, T. *Elements of user interface design*. Nova lorque: John Wiley & Sons, 1997.

MARCUS, A. Dare we define user-interface design? *Interactions*, Nova lorque, v. 9, n. 5, p. 19-24, set. 2002.

MICROSOFT. Designing the Hilo User Experience. In: _____. *Hilo: Developing C++ Applications for Windows 7*. Seattle: Microsoft, 2010. p. 32-41. Disponível em: <<https://msdn.microsoft.com/en-us/library/windows/desktop/ff800706.aspx>>. Acesso em: 15 dez. 2014.

MILETTO, E. M.; BERTAGNOLLI, S. C. *Desenvolvimento de Software II: introdução ao desenvolvimento web com HTML, CSS, JavaScript e PHP*. Porto Alegre: Bookman, 2014.

NIELSEN, J. 10 Usability Heuristics for User Interface Design. *Nielsen Norman Group*, Freemont, jan. 1995. Disponível em: <<http://www.nngroup.com/articles/ten-usability-heuristics/>>. Acesso em: 9 mar. 2015.

____. *Return on investment for usability*. *Nielsen Norman Group*, Freemont, jan. 2003. Disponível em: <<http://www.nngroup.com/articles/return-on-investment-for-usability/>>. Acesso em: 24 fev. 2015.

____. The usability engineering life cycle. *IEEE Computer*, Los Alamitos, v. 25, n. 3, p. 12-22, Mar. 1992.

____. *Usability engineering*. San Francisco: Morgan Kaufmann, 1993.

NORMAN, D. A. *The design of everyday things*. Nova lorque: Basic Books, 2002.

ORTH, A. I. *Interface homem-máquina*. Porto Alegre: AIO, 2005.

OVERFIELD, E. et al. *Pro SharePoint 2013 Branding and Responsive Web Development*. Berkeley: Apress, 2013.

PAULA FILHO, W. P. *Engenharia de Software: fundamentos, métodos e padrões*. 3. ed. São Paulo: LTC, 2008.

PAULA, M. G.; BARBOSA, S. D. J.; LUCENA, C. J. P. Conveying human-computer interaction concerns to software engineers through an interaction model. In: CONFERENCIA LATINOAMERICANA DE INTERACCIÓN HUMANO-COMPUTADORA, 2., 2005, Cuernavaca. *Proceedings...* Nova Iorque: ACM Press, 2005. p. 109-19.

PETERS, J. F.; PEDRYCZ, W. *Engenharia de software: teoria e prática*. Rio de Janeiro: Campus, 2001.

PFLIEGER, S. L. *Engenharia de software: teoria e prática*. 2. ed. São Paulo: Prentice Hall, 2004.

PRESSMAN, R. S. *Engenharia de software*. 6. ed. São Paulo: McGraw-Hill, 2006.

____. _____. 7. ed. Porto Alegre: AMGH, 2011.

PRESSMAN, R. S.; LOWE, D. B. *Engenharia web*. Rio de Janeiro: LTC, 2009.

ROCHA, H. V.; BARANAUSKAS, M. C. C. *Design e avaliação de interfaces humano-computador*. Campinas: Nied/Unicamp, 2003.

ROZANSKI, E. P.; HAAKE, A. R. The many facets of HCI. In: CONFERENCE ON INFORMATION TECHNOLOGY CURRICULUM, 4., 2003, Lafayette. *Proceedings...* Nova Iorque: ACM Press, 2003. p. 180-5.

SÁ, M.; CARRIÇO, L. Low-fi prototyping for mobile devices. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 6., 2006, Montreal. *Proceedings...* Nova Iorque: ACM, 2006. p. 694-9.

SCHMIDT, M. E. C. *Implementing the IEEE software engineering standards*. Indianapolis: Sams, 2000.

SEFFAH, A.; METZKER, E. The obstacles and myths of usability and software engineering. *Communications of the ACM*, Nova Iorque, v. 47, n. 12, p. 71-6, dez. 2004.

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

____. _____. 3. ed. Porto Alegre: Bookman, 2013.

SHNEIDERMAN, B. *Designing the user interface: strategies for effective human-computer interaction*. 3. ed. Berkeley: Addison Wesley Longman, 1998.

SHNEIDERMAN, B.; PLAISANT, C. *Designing the user interface: strategies for effective human-computer interaction*. 5. ed. Reading: Addison-Wesley, 2010.

SILVA, A. C. *et al.* Aplicabilidade de padrões de engenharia de software e de IHC no desenvolvimento de sistemas interativos. In: CONGRESSO BRASILEIRO DA COMPUTAÇÃO, 4., 2004, Itajaí. *Anais...* Itajaí: Ufscar, 2004. p. 118-23.

SILVA, M. S. *Web design responsivo*. São Paulo: Novatec, 2014.

SMITH, P. G.; REINERSTEN, D. G. *Developing products in half the time*. Reinhold, Nova Iorque: Van Nostrand, 1991.

SOMMERVILLE, I. *Engenharia de software*. 6. ed. São Paulo: Prentice Hall Brasil, 2003.

_____. _____. 8. ed. São Paulo: Pearson, 2007.

SOUZA, C. S. et al. Interação Humano-Computador: perspectivas cognitivas e semióticas. In: FUKS, H. (Org.). In: JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA, 1999, Rio de Janeiro. *Anais...* Rio de Janeiro: Edições EntreLugar, 1999. p. 420-70.

SOUZA, L. S.; COSTA, I.; SPINOLA, M. M. Projetos de interface com o usuário de software de dispositivos móveis: requisitos de usabilidade e fatores impactantes. In: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 13., 2006, Bauru. *Anais...* Bauru: Simpep, 2006.

SWEBOK. *Guide to the software engineering body of knowledge: a project of the IEEE Computer Society Professional Practices Committee*. California: IEEE Computer Society, 2004.

TORRES, E. F.; MAZZONI, A. A. Multimedia digital contents: an approach on usability and accessibility. *Ci. Inf.*, Brasília, v. 33, n. 2, p. 152-60, maio/ago. 2004.

USA. U.S. Department of Health & Human Services. Planning a usability test. *Usability.gov*, [s.d.]a. Disponível em: <<http://www.usability.gov/how-to-and-tools/methods/planning-usability-testing.html>>. Acesso em: 6 mar. 2015.

_____. System Usability Scale (SUS). *Usability.gov*, [s.d.]b. Disponível em: <<http://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>>. Acesso em: 6 mar. 2015.

_____. Usability testing. *Usability.gov*, [s.d.]c. Disponível em: <<http://www.usability.gov/how-to-and-tools/methods/usability-testing.html>>. Acesso em: 6 mar. 2015.

W3C (WORLD WIDE WEB CONSORTIUM). *Diretrizes de acessibilidade para conteúdo web (WCAG) 2.0*. Tradução Everaldo Bechara. São Paulo: W3C, 2014. Disponível em: <<http://www.w3.org/Translations/WCAG20-pt-br/>>. Acesso em: 24 fev. 2015.

_____. *Media Queries: W3C recommendation*. São Paulo: W3C, 2012. Disponível em: <<http://www.w3.org/TR/css3-mediaqueries/>>. Acesso em: 24 fev. 2015.

WALKER, M.; TAKAYAMA, L.; LANDAY, J. A. High-Fidelity or Low-Fidelity, Paper or Computer? Choosing Attributes When Testing Web Prototypes. In: HUMAN FACTORS AND ERGONOMICS SOCIETY ANNUAL MEETING, 46., 2002, Baltimore. *Proceedings...* Baltimore: HFES, 2002. p. 661-5.

WEISS, S. *Handheld usability*. Hoboken: John Wiley & Sons, 2002.

WERNKE, R. *Gestão financeira: ênfase em aplicações e casos nacionais*. São Paulo: Saraiva, 2008.

ZEMEL, T. *Web design responsivo: páginas adaptáveis para todos os dispositivos*. São Paulo: Casa do Código, 2013.

Sites

<<http://emag.governoeletronico.gov.br/>>.

<<http://www.w3.org/WAI/>>.

Exercícios

Unidade I – Questão 1: INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA (INEP). Exame Nacional de Desempenho dos Estudantes (ENADE) 2014: Sistemas de Informação. Questão 28. Disponível em: <http://download.inep.gov.br/educacao_superior/enade/provas/2014/39_sistemas_informacao.pdf>. Acesso em: 7 dez. 2018.

Unidade I – Questão 2: INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA (INEP). Exame Nacional de Desempenho dos Estudantes (ENADE) 2008: Computação. Questão 15. Disponível em: <http://download.inep.gov.br/download/Enade2008_RNP/COMPUTACAO.pdf>. Acesso em: 7 dez. 2018.

Unidade II – Questão 1: INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA (INEP). Exame Nacional de Desempenho dos Estudantes (ENADE) 2008: Computação. Questão 77. Disponível em: <http://download.inep.gov.br/download/Enade2008_RNP/COMPUTACAO.pdf>. Acesso em: 7 dez. 2018.

Unidade II – Questão 2: INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA (INEP). Exame Nacional de Desempenho dos Estudantes (ENADE) 2011: Tecnologia em Análise e Desenvolvimento de Sistemas. Questão 9. Disponível em: <http://download.inep.gov.br/educacao_superior/enade/provas/2011/ANALISE_E_DESENVOLVIMENTO_DE_SISTEMAS.pdf>. Acesso em: 7 dez. 2018.

Unidade III – Questão 1: INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA (INEP). Exame Nacional de Desempenho dos Estudantes (ENADE) 2014: Ciência da Computação – Bacharelado. Questão 9. Disponível em: <http://download.inep.gov.br/educacao_superior/enade/provas/2014/03_computacao_bacharelado.pdf>. Acesso em: 7 dez. 2018.

Unidade III – Questão 2: INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA (INEP). Exame Nacional de Desempenho dos Estudantes (ENADE) 2011: Tecnologia em Análise e Desenvolvimento de Sistemas. Questão 15. Disponível em: <http://download.inep.gov.br/educacao_superior/enade/provas/2011/ANALISE_E_DESENVOLVIMENTO_DE_SISTEMAS.pdf>. Acesso em: 7 dez. 2018.

Unidade IV – Questão 1: INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA (INEP). Exame Nacional de Desempenho dos Estudantes (ENADE) 2014: Tecnologia em Análise e Desenvolvimento de Sistemas. Questão 32. Disponível em: <http://download.inep.gov.br/educacao_superior/enade/provas/2014/40_tecnologia_analise_desenv_sistemas.pdf>. Acesso em: 7 dez. 2018.

Unidade IV – Questão 2: INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA (INEP). Exame Nacional de Desempenho dos Estudantes (ENADE) 2014: Tecnologia em Análise e Desenvolvimento de Sistemas. Questão 28. Disponível em: <http://download.inep.gov.br/educacao_superior/enade/provas/2014/40_tecnologia_analise_desenv_sistemas.pdf>. Acesso em: 7 dez. 2018.

This image shows a full page of blank white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page, providing a template for writing or drawing. There are no margins, text, or other markings present.



Handwriting practice lines consisting of 30 horizontal rows. Each row is defined by two thin grey lines, with a slightly thicker grey line in the middle of each row to guide letter height.



Lined writing area with horizontal lines.



Handwriting practice lines consisting of 30 horizontal rows. Each row is defined by two thin grey lines, with a slightly thicker grey line in the middle of each row to guide letter height.



Handwriting practice lines consisting of 30 horizontal blue lines. Each line is preceded by a small blue dot, serving as a guide for letter height and placement.



Interativa

Informações:
www.sepi.unip.br ou 0800 010 9000