

Unipê

Especialização em Desenvolvimento Web

Princípios de Design e Usabilidade em Aplicações Web (User Experience - UX)

Módulo II

Testes de Usabilidade

Prof. Ms João Paulo Fachine Sette
fachine@gmail.com

<http://lattes.cnpq.br/3477271592797947>

I. Sobre este documento

Dicas importantes para usar o material

Este material foi especialmente desenvolvido para acompanhamento da Disciplina **Princípios de Design e Usabilidade em Aplicações Web (UX)**, da Especialização de Desenvolvimento Web do Centro Universitário Unipê. Este material tem características de "misto" no que se refere apresentação, isto quer dizer que ele servirá como indicador em apresentação de slides, bem como uma leitura mais detalhada. No decorrer do curso, muitos conteúdos serão aprofundados com textos à parte onde a literatura indicativa será exposta. Para que este material funcione de forma adequada, **todos os alunos devem abrir este documento e acompanhá-lo em seu próprio monitor.**

Iconografias



Isso quer dizer que você terá que deixar a preguiça de lado e trabalhar!



Um vídeo ou animação deverá ser visualizada na pasta da disciplina



A "corrente" significa que temos um link para averiguar



Este ícone remete a necessidade de interação em grupo

Drive Virtual

Todo o material utilizado nesta disciplina é disponibilizado em drive virtual, versionado, dessa forma você pode sempre pegar o conteúdo mais atualizado.



< <https://drive.google.com/open?id=0B80ZHtVwE3fGQjJwbDNkNExfSEU> >

II. Estrutura do ensino

Como construir seu conhecimento

É com satisfação que estou participando e aprendendo com todos vocês sobre o desenvolvimento web. Sabemos como a tecnologia evolui constantemente e como nós precisamos estar atualizados, dessa forma, esta especialização mais que se justifica, se torna necessária! **Espero contar com todos participando e trocando informações e experiências.**

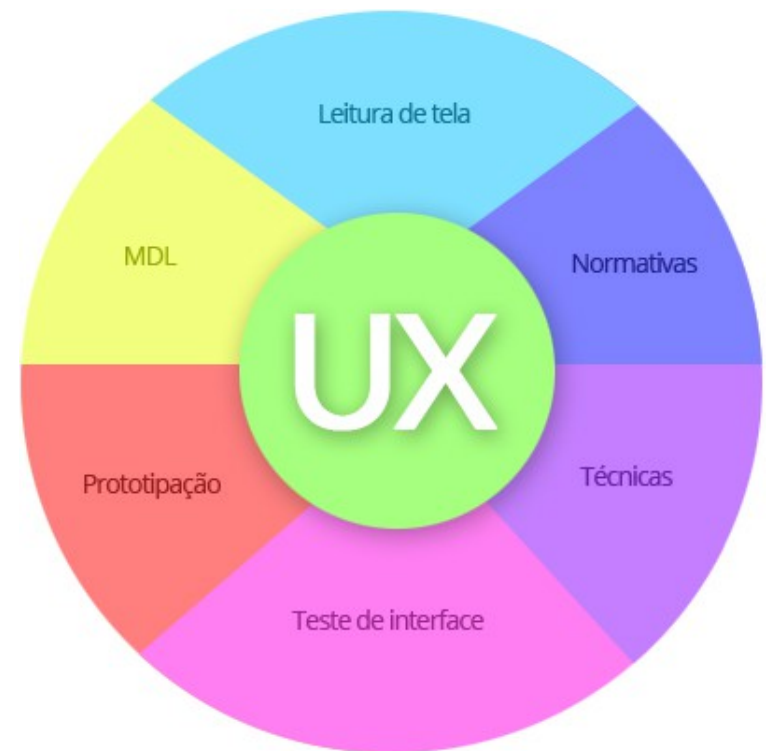
Os Pilares da disciplina

A ementa da disciplina indica os seguintes pontos:

- Introdução à usabilidade;
- Usabilidade para sistemas web;
- Critérios, especificações e recomendações de usabilidade;
- Prototipação de interfaces web;
- Técnicas de avaliações de usabilidade em interfaces web.

Dessa forma temos como foco capacitar os alunos sobre a tematização de **Experiência de Usuário (UX)** para que construam e critiquem de forma racional interfaces e sistemas Web. Para tal, vamos distribuir nosso conteúdo em três módulos:

- Módulo I - Fundamentação e normativas de usabilidade;
- **Módulo II - Ferramentas de teste de usabilidade;**
- Módulo III - Material Design (MDL).



Vamos lá!



COMO FUNCIONA
ESTE **SITE?**

1. Impactos da (falta de) usabilidade

No módulo anterior já falamos sobre ferramentas de **eye tracking** e **mouse tracking** que são, sem dúvida, excelentes ferramentas de teste de usabilidade. Eye tracking descobre para onde seus olhos estão indo e assim, identificar a atenção do usuário. Mouse tracking determina como está seu comportamento de interação, para onde sua ação está sendo direcionada, dessa forma fica fácil perceber erros e falhas de interface. Porém, outras ferramentas podem e devem ser utilizadas para testar interface.

Neste módulo veremos o impacto da Usabilidade no software, aprenderemos como testar usabilidade em três fases distintas do projeto (fase inicial de requisitos com **protótipos de fluxo**, fase intermediária de desenvolvimento com **heurísticas** e pós implantação avaliando a **experiência do usuário UX**). Antes disso, veremos o impacto da falta de usabilidade.



Me digam pelo menos 1 sistema que vocês odeiam. Por que?

Sistema com baixa usabilidade:

- **Requer treinamento excessivo;**
- **Desmotiva a exploração do aplicativo por parte do usuário;**
- **Confunde os usuários;**
- **Induz os usuários ao erro;**
- **Gera insatisfação;**
- **Diminui a produtividade.**

A baixa usabilidade gera ao usuário uma sobrecarga que segundo **Walter Cybis (2012)**, pode ser em três aspectos...

Cognitiva

Não entende

Perceptiva

Não consegue encontrar

Física

Não consegue acionar
(complexidade ou impedimento)



Ganhos com Usabilidade

A usabilidade já é considerada um investimento de retorno no desenvolvimento de software, isto porque seu impacto é percebido da produção até a venda. Embora a usabilidade já seja mapeada como importante para todo o projeto, muitas equipes negligenciam a fase de UX por incapacidade técnica ou limitação de equipe.

Impacto da usabilidade durante o ciclo de desenvolvimento

- **Menores custos de desenvolvimento:** pensar em facilitar a vida do usuário em fases iniciais do projeto faz com que os problemas detectados sejam de mais fácil correção;
- **Redução do tempo de projeto:** o cuidado com a usabilidade em todo o ciclo de desenvolvimento do aplicativo traz a redução dos tempos de teste e controle de qualidade;
- **Menores custos de manutenção e suporte:** os produtos com uma boa usabilidade são mais fáceis de instalar e de aprender a usar. Isto leva a uma redução em custos de aprendizagem e de suporte;
- **Vantagem competitiva:** o mais fácil de usar é mais simples de vender;
- **Fidelização do cliente:** é natural que um cliente satisfeito com a usabilidade do aplicativo adquirido volte a procurar os desenvolvedores do mesmo, caso precise de outra solução em software.

Dessa forma é fácil perceber a necessidade de utilizar técnicas de avaliação de usabilidade.

2. Técnicas de Avaliação de Usabilidade

As técnicas de avaliação podem ser divididas em **Empíricas** e **Diagnósticas**

Técnicas Empíricas (conhecimento a partir da experiência vivenciada)

- Pesquisa de satisfação
- Busca de opinião de usuários
- Simulação de interações

Técnicas Diagnósticas (através de técnicas consolidadas)

- Checklist
- Heurística
- Percurso cognitivo
- Semiótica
- Coloração



Pergunte ao colega do seu lado quais são os 3 softwares que ele mais gosta e anote. Responda para seu colega a mesma pergunta.



Agora pergunte o motivo pelo qual ele (e vc) gosta deste software. Pense, é um motivo técnico ou pessoal ?

Geralmente o software que mais gostamos é listado de forma empírica, ou seja, baseado na nossa experiência. O motivo de gostar do software pode estar ligado a fatores sentimentais, mas também à funcionalidades específicas de um software.



3. Processo de Teste de Usabilidade

O processo de teste de usabilidade se resume a **Planejar, Executar e Compilar** os dados. Não deixa de ser um PDCA!

O complexo de testar usabilidade é **definir bem, executar sem erros e compilar os dados da forma correta**. Com isso quero dizer que na teoria é bem claro testar usabilidade porém, na prática, é extremamente difícil e melindroso conseguir resultados fidedígnos. O papel ideal para teste de usabilidade é do testador (não é do desenvolvedor ou do analista).

Quem testa é o TESTADOR (não é o desenvolvedor/analista)



A Fase de Planejamento (Plan)

Esta é a **fase mais delicada** no teste de usabilidade. Nesta etapa definimos em que fase do ciclo produtivo iremos testar a interface. Além da definição das fases, deve-se determinar que tipos de teste serão feitos em cada fase. Os testes devem ser pensados com cautela e se possível, validados com o dono do produto.

Geralmente os testes de usabilidade planejados são (e vamos focar nestes):

- Exploratórios de protótipo (fase de análise);
- Heurísticos e checklists (fase de desenvolvimento) e ;
- Questionários (pós implantado).



Nesta fase é importante definir:

- Em que fases testarão a usabilidade;
- Quais tipos de testes serão feitos (empíricos, diagnósticos? etc);
- Quais funcionalidades ou casos de uso serão testados;
- Quais são os critérios de teste (Relevância, Eficácia, Eficiência, Retorno financeiro...)
- Qual o peso de cada um dos critérios de teste;
- A quantidade de usuários para cada teste;
- Local, estrutura física do teste;
- Tempo do teste;
- Assistência aos participantes do teste... entre outros.



A Fase de Execução (Do)

A execução do teste **deve seguir à risca o planejado**. O ambiente deve ser propício e a quantidade de testadores têm que ser apropriada para o teste. Considerando testes de Heurística, o número de usuários melhora os teste exponencialmente, uma vez que, quanto mais testadores, mais erros são conhecidos.

Nesta fase é importante:

- Evitar ruídos nos testes (ambientes impróprios etc);
- Testar em quantidade adequada;
- Utilizar pessoas capacitadas (testadores) ou baseada em escopo (podem ser leigos);
- Executar o plano de teste corretamente registrando eventuais mudanças;
- Em alguns casos, gravar o teste.



A Fase de Compilação de dados (Check)

Esta é a **fase mais importante** dos teste de usabilidade. Aqui vamos tratar os dados e usá-los como métricas para melhorar o software. Geralmente para analisar os dados de um teste de usabilidade, planilhas são montadas e gráficos são interpretados, deixando claro os resultados do teste. As decisões tomadas a partir do teste são de responsabilidade do analista e do dono do projeto.

Não tem mistério, é analisar dados e saber gerar informações. Exemplo:

Analise abaixo qual item testado deve ter prioridade para o próximo release de software?

Caso de uso 1

Grau de importância: 5 (1 a 10)

Eficácia: 80% (realizar a atividade)

Eficiência: 50% (realizar em 1 minuto)

Caso de uso 2

Grau de importância: 8 (1 a 10)

Eficácia: 90% (realizar a atividade)

Eficiência: 50% (realizar em 1 minuto)

Caso de uso 3

Grau de importância: 2 (1 a 10)

Eficácia: 90% (realizar a atividade)

Eficiência: 90% (realizar em 1 minuto)

A forma de avaliar os testes de usabilidade serão montados durante o planejamento. A importância dada a um caso de uso ou a um critério como Eficácia ou Eficiência depende das definições impostas no planejamento.

Dessa forma, considerando que cada teste tem um peso de importância definido pelo Dono do Projeto (de 1 a 10) e que cada índice negativo de Eficácia e Eficiência tem peso 1 para 1, podemos chegar as seguintes conclusões:

Analisar abaixo qual item testado deve ter prioridade para o próximo release de software?

Caso de uso 1

Grau de importância: 5 (1 a 10)

Eficácia: 80% (realizar a atividade)

Eficiência: 50% (realizar em 1 minuto)

Uma forma seria: $(20 + 50) * 5 = 350$

Caso de uso 2

Grau de importância: 8 (1 a 10)

Eficácia: 90% (realizar a atividade)

Eficiência: 50% (realizar em 1 minuto)

$(10 + 50) * 8 = 480$

Caso de uso 3

Grau de importância: 2 (1 a 10)

Eficácia: 90% (realizar a atividade)

Eficiência: 90% (realizar em 1 minuto)

$(10 + 10) * 2 = 40$

Analise abaixo qual item testado deve ter prioridade para o próximo release de software?

Enviar foto para contato

Grau de importância: multiplicador final **x5** (1 a 10)

Eficácia: **80%** (realizar a atividade) (1 pra 1)

Eficiência: **30%** (realizar em 1 minuto) (1 pra 1)

Selecionar amigos para compartilhar dados

Grau de importância: multiplicador final **x8** (1 a 10)

Eficácia: **60%** (realizar a atividade) (1 pra 1)

Eficiência: **50%** (realizar em 1 minuto) (1 pra 1)

Excluir mensagens inapropriadas

Grau de importância: multiplicador final **x2** (1 a 10)

Eficácia: **40%** (realizar a atividade) (1 pra 1)

Eficiência: **90%** (realizar em 1 minuto) (1 pra 1)

Visualizar perfil de amigo

Grau de importância: multiplicador final **x5** (1 a 10)

Eficácia: **70%** (realizar a atividade) (1 pra 1)

Eficiência: **50%** (realizar em 1 minuto) (1 pra 1)

Postar texto na linha do tempo

Grau de importância: multiplicador final **x7** (1 a 10)

Eficácia: **40%** (realizar a atividade) (1 pra 1)

Eficiência: **40%** (realizar em 1 minuto) (1 pra 1)

4. Ferramentas de Teste de Usabilidade

Fase de Planejamento

- Avaliação de protótipos de tela e fluxo
 - Mockups (Balsamiq, Adobe XD, Figma etc)
- Teste com cartões

Fase de Desenvolvimento

- Checklist com Heurísticas de Nielsen
- Teste com cartões

Fase pós Implantação

- Questionário de satisfação
- Teste AB

Testes remotos e testes online



<http://www.testaisso.com.br/>

Um erro clássico [em projetos] é
estruturar a arquitetura de informação
baseando-se apenas no ponto de vista
do projetista

Nielsen (2004)

Testes com cartões "Cards"

Teste de cartões (CARDS) é um **experimento simples para auxiliar a organização do conteúdo**. Pode ser usado na fase de planejamento como também na fase de desenvolvimento. Existem duas vertentes, os CARDS ABERTOS, onde os usuários irão agrupá-los e categorizá-los e os CARDS FECHADOS, onde algumas categorias são definidas e os participantes montam as pilhas previamente rotuladas.

CARDS ABERTOS: Entregue a grupos aproximadamente 30 cards que devem ser empilhados e rotulados (categoria) como considerar melhor.

CARDS FECHADOS: Entregue a grupos aproximadamente 30 cards que devem ser empilhados em categorias já existentes.



Individualmente, cada aluno terá 30 minutos para pesquisar na internet sobre o tema que escolheu para o projeto final. Após pesquisar, os alunos deverão escrever em Post-it as seções mais comuns nos sites (ou APP's) para o tema escolhido (pelo menos 10 seções). Em seguida, cada aluno irá elencar os cards de acordo com a importância (o mais importante na parte superior da pilha). Tire uma foto, esta imagem será anexada no PROJETO FINAL.

Exemplo: Tema "Futebol"

1) Tabela da rodada; 2) Resultado dos últimos jogos; 3) Palpite; 4) Produtos relacionados à venda; 5) Artilharia; 6) Posição dos clubes no campeonato; 7) Notícias; 8) Ao-vivo; 9) Comentário do público.



Company merchandise

Staff directory

Buy & swap area

Company news

CEO blog

Company profile

List of departments

Share price

Company vision

Expense form

HR forms

Vehicle

Testes A/B

Com essa técnica de teste você irá testar componentes específicos e mudanças únicas. Ou seja, este teste é indicado para mensurar se uma determinada mudança será boa para o contexto. É importante realizar em apenas uma mudança, pois, com mais de uma, ficaria difícil identificar qual das duas trouxe mais resultados.

Ideal para testar mudanças de comportamento em itens específicos

Em resumo, esta técnica é bem simples. Parte do público irá acessar a página original e parte irá acessar uma página modificada. Esta divisão de público será feita diretamente na programação da página ou do servidor. As páginas serão rastreadas, de forma que, determinadas ações serão contadas e ao fim um relatório será gerado.



Vamos realizar um TESTE A/B.

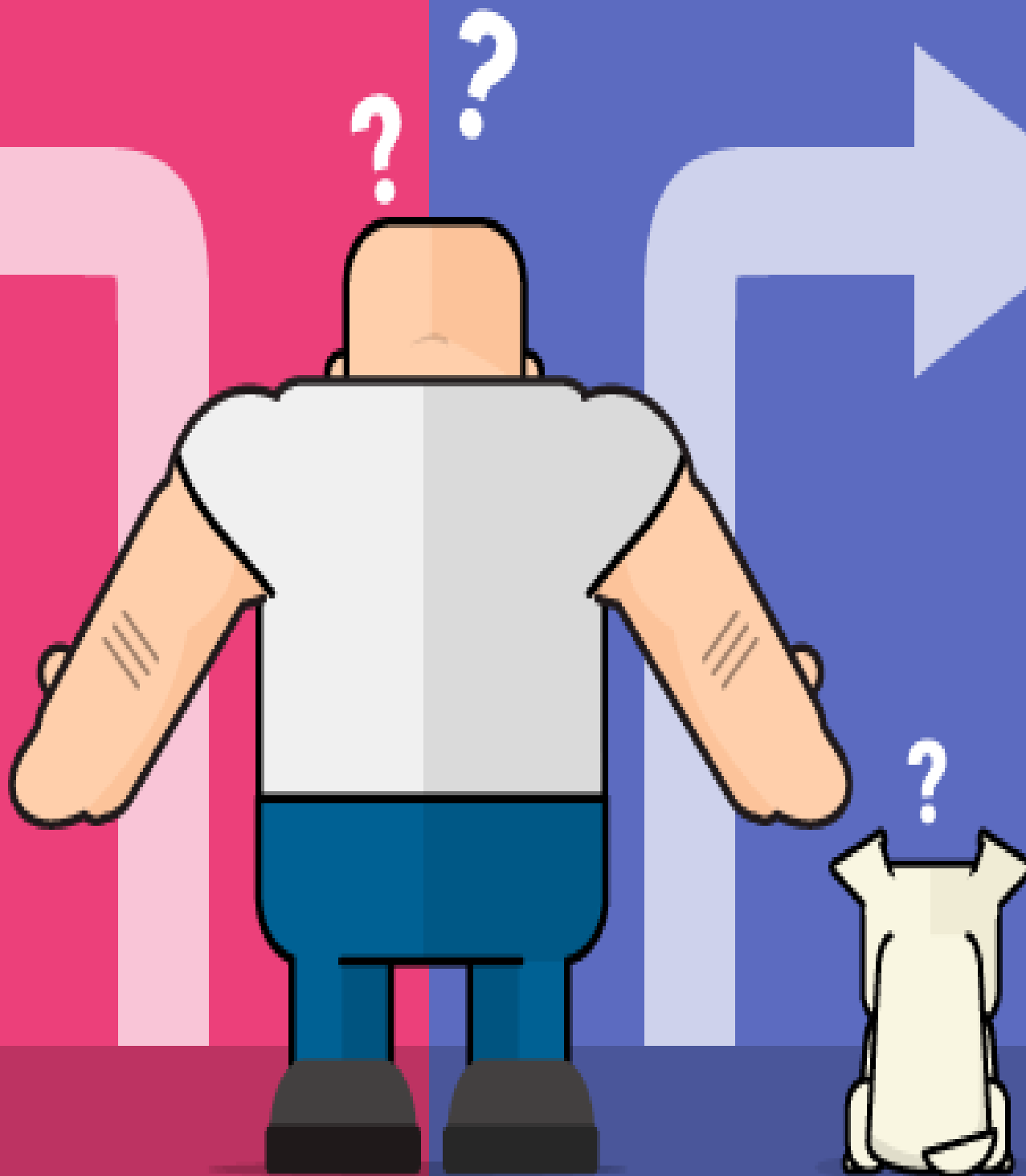
Na pasta "Experimentos", acesse "TESTE_AB.html". Algumas instruções serão passadas.
Ao fim, anote o tempo em que realizou a tarefa e o tipo de teste (se foi "A" ou "B").
Vamos comparar os resultados.



Lib free emPHP para teste A/B: <https://phpab.github.io/>

A

B



Testes com Heurísticas de Nielsen

Este é o **principal método proposto nesta disciplina para avaliação de interfaces**, uma vez que é de fácil implementação e traz bons resultados. Indica-se que a revisão de interface utilizando as Heurísticas deve ser feita por muitos avaliadores uma vez que é muito difícil e improvável uma mesma pessoa encontrar todos os erros.

Como proceder

Individualmente, o avaliador irá verificar página a página em busca de um dos 10 problemas citados na heurística. Assim que um problema é percebido, este deve ser registrado e detalhado. Após verificar toda a página e acreditar ter listados todos os erros das Heurísticas, o avaliador vai para a próxima página. Importante dizer que essa mudança de página deve ser vista com cautela, principalmente para **analisar padrões entre as páginas** (ou a falta deles).

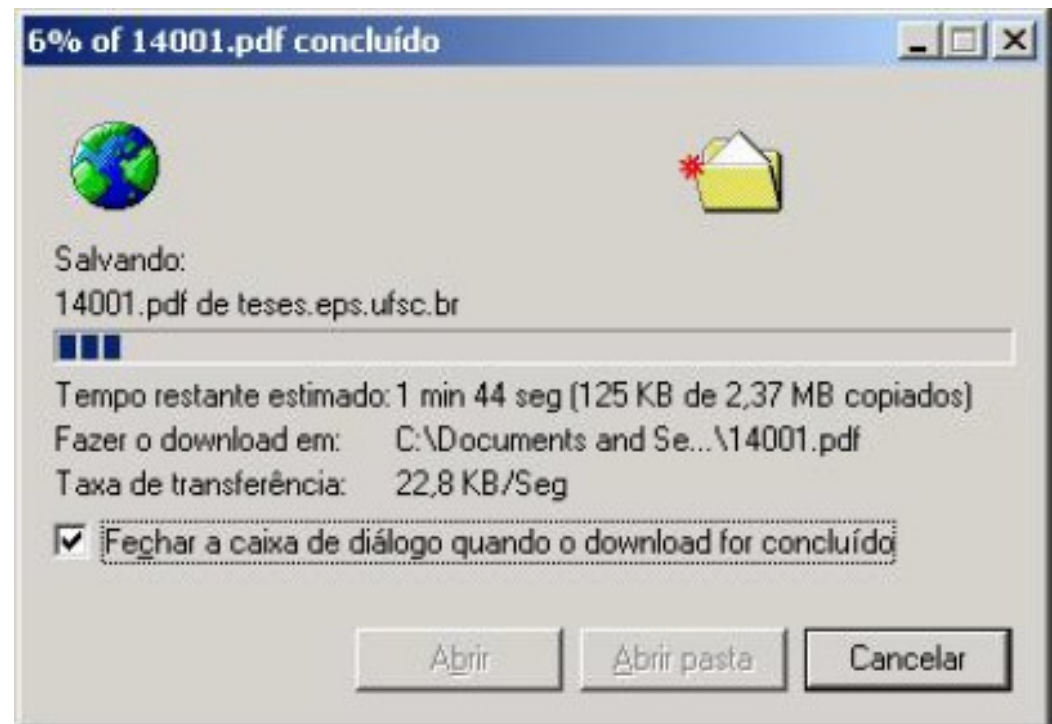
As 10 heurísticas

- Visibilidade do status do sistema
- Compatibilidade do sistema com o mundo real
- Controle do usuário e liberdade
- Consistência e padrões
- Prevenção de erros
- Reconhecimento em lugar de lembrança
- Flexibilidade e eficiência de uso
- Estética e design minimalista
- Ajudar o usuário a reconhecer, diagnosticar e corrigir erros
- Facilita documentar

1. Visibilidade do Status do sistema

Isso significa que você precisa se certificar de que a interface sempre informe ao usuário o que está acontecendo, ou seja, todas as ações precisam de feedback instantâneo para orientá-lo.

Corretos:



1. Visibilidade do Status do sistema

Isso significa que você precisa se certificar de que a interface sempre informe ao usuário o que está acontecendo, ou seja, todas as ações precisam de feedback instantâneo para orientá-lo.

Incorretos:

Detalhes do produto, simulação de preço e remarcação

ANTENA AQUARIO DTV-3000 C/CABO DIGITAL 50282

Preços Atuais Conforme Última entrada:

Preço de Compra:	59,52	Preço Atacado 03:	178,84
Custo de Compra:	65,81	Preço Atacado 04:	178,84
Custo de Venda:	79,29		
Preço de Venda:	178,84	Última Remarcação:	27/07/2013
Preço Líquido:	178,84	Preço de Venda Remarcado:	178,84
Preço Atacado:	178,84	Margem Normal R\$:	99,55 45,20 %
Preço Composto:	178,84	Margem Flutuante R\$:	99,55 55,66 %
Preço Promoção:	178,84	Preço Atacado Remarcado:	178,84
Preço Atacado 02:	178,84		

Fornecedor: KIDASEN INDUSTRIA E COMERCIO DE ANTE

Preço de Venda: 0,00 Preço Promoção: 0,00 Preço Atacado 02: 0,00 Preço Atacado 04: 0,00

Preço Composto: 0,00 Preço Atacado 03: 0,00

Utilizar Desconto na Remarcação? %Desconto Máximo: 33,00

Configuração dos preços para aceitar remarcação igual a zero:

Preço de venda: ☐ Preço promoção: ☐ Preço atacado 02: ☐ Preço atacado 04: ☐ Preço composto: ☐ Preço atacado 03: ☐

Preços Antes da Última Entrada:

Preço Compra:	Cus. Compra:	Cus. Venda:	Últ. Remarcação:	Últ. Remarcação:	Preço Venda:
90,00	74,70	90,00	16/07/2013	183,09	183,09

Fornecedor: KIDASEN INDUSTRIA E COMERCIO DE ANTE

Data, Hora e Loja Atual

10/09/2013 14:32:30 | Loja 1 - ELETROPECAS

softcom

Cadastros Obras Imóveis Compras Administrativo Financeiro Relatórios

Imóveis

Empresa: Quartos: e

Tipo de Imóvel: Suite: e

Situação: Garagem: e

Negociação: Área: e

Status: Valor: e

Categoria: Aluguel: e

Posição: Data de Cadastro: e

Captado por: Unidade:

Desativado: Não ☐ Bloco:

Proprietário: Código Imóvel:

ID: Imóvel:

Endereço: Bairro:

Cidade: UF:

Zona: Código Anúncio:

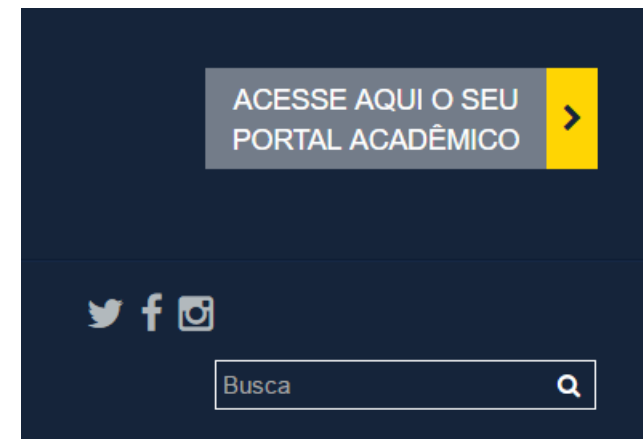
Buscar Novo Cadastro Mapa de Unidades Tabela de Preços Comparar Selecionados Excluir Selecionados

<input type="checkbox"/>	ID	Imóvel	Unidade	Ponto de Referência	Situação	Status
<input type="checkbox"/>	13	100000	0101		PRONTO	DISPONÍVEL
<input type="checkbox"/>	13	100000	102		PRONTO	DISPONÍVEL
<input type="checkbox"/>	13	100000	103		PRONTO	DISPONÍVEL

2. Compatibilidade do sistema com o mundo real

Não usar palavras que não fazem sentido pro usuário. Toda a comunicação do sistema precisa ser contextualizada ao usuário, e ser coerente com o chamado modelo mental do usuário.

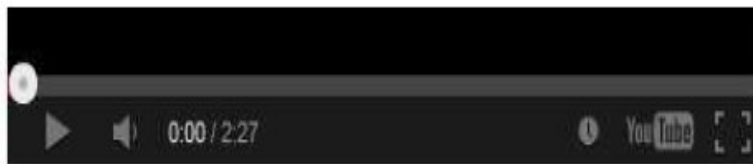
Corretos:



2. Compatibilidade do sistema com o mundo real

Não usar palavras que não fazem sentido pro usuário. Toda a comunicação do sistema precisa ser contextualizada ao usuário, e ser coerente com o chamado modelo mental do usuário.

Incorretos:



Dados do Filme:

Tamanho: 299 MB ou 790 MB

Formato: TS -> CINEMA/Rmvb ou Avi

Idioma: Portugues ou Portugues/Ingles

Classificação Etária: 14 Anos

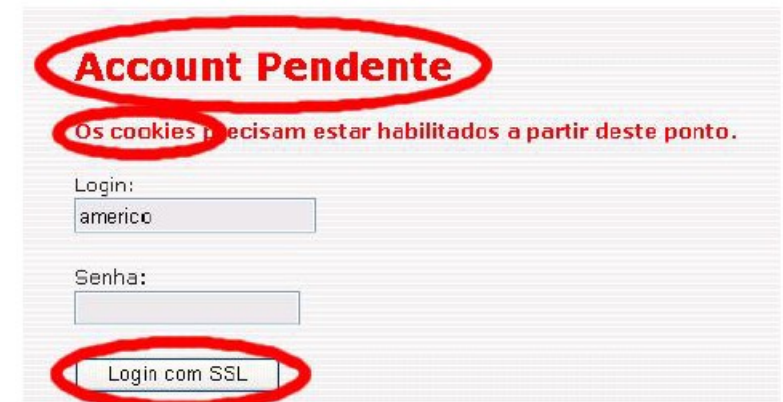
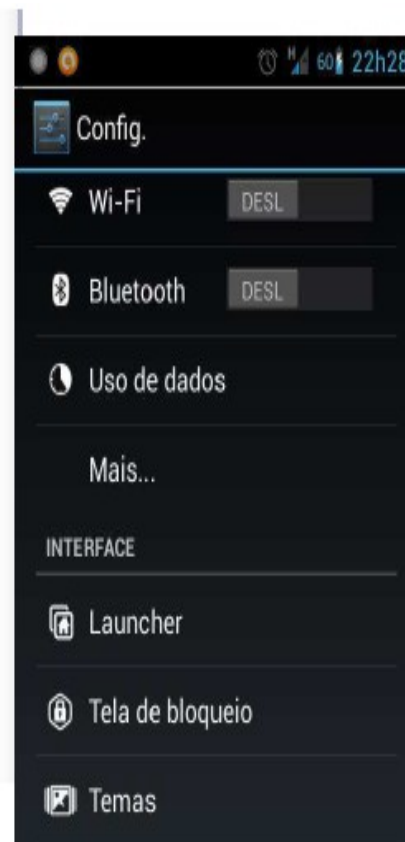
Lançamento: 2013

Qualidade Audio: 8

Qualidade Video: 8

Servidores: Uploaded/Crocko/QueenShare/FreakShare/NetLoad/Turbobit
/PutLocker/FileGag/BitShare/RapidGator

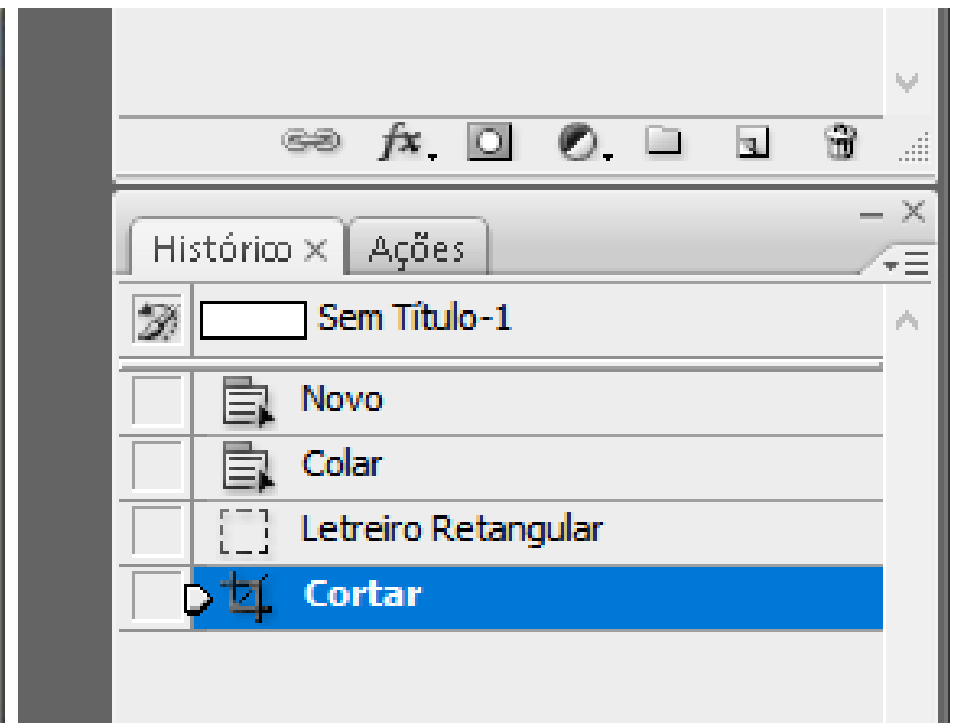
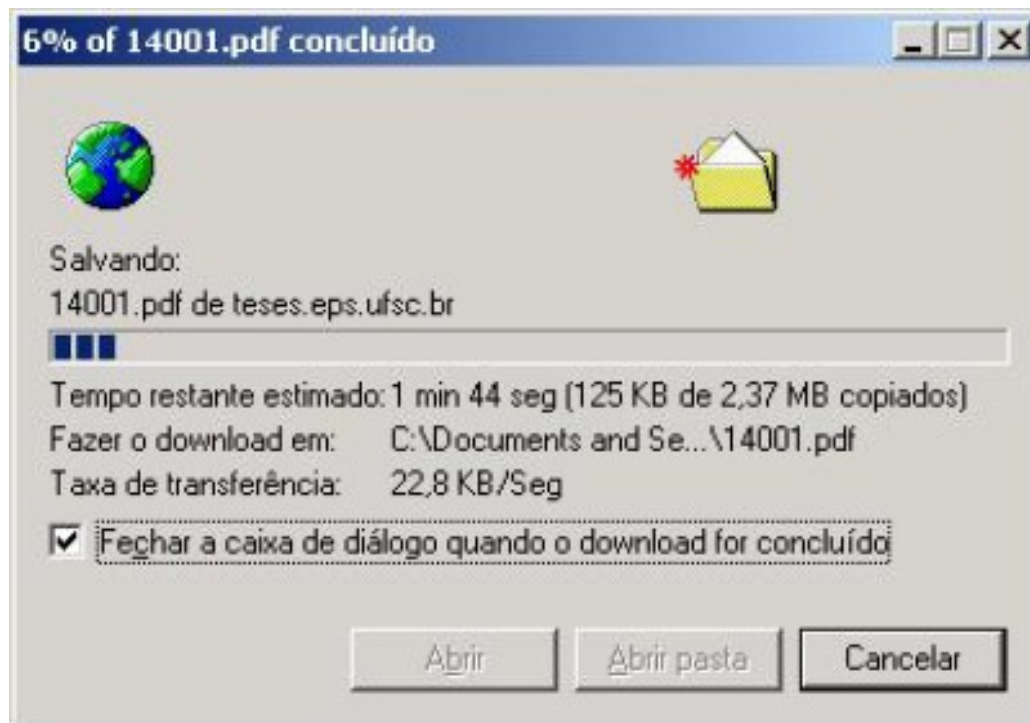
Rmvb - Dublado



3. Liberdade e Controle do usuário

Facilita as “saídas de emergência” para o usuário, permitindo desfazer ou refazer a ação no sistema e retornar ao ponto anterior, quando estiver perdido ou em situações inesperadas.

Corretos:



3. Liberdade e Controle do usuário

Facilita as “saídas de emergência” para o usuário, permitindo desfazer ou refazer a ação no sistema e retornar ao ponto anterior, quando estiver perdido ou em situações inesperadas.

Incorretos:

link do mega não funciona porque suspenderam minha conta

Cosme Diz:
setembro 10th, 2013 at 1:03

assisti no cinema, é mais do mesmo.

Aluno Diz: Your comment is awaiting moderation.
setembro 10th, 2013 at 4:17

Um bom filme!

Deixe Seu Comentário

Regras:
Não use palavras de baixo calão.
Não tenha vergonha, pode perguntar a vontade.
Comente! Diga o que achou da postagem!
Se gostou, agradeça :D

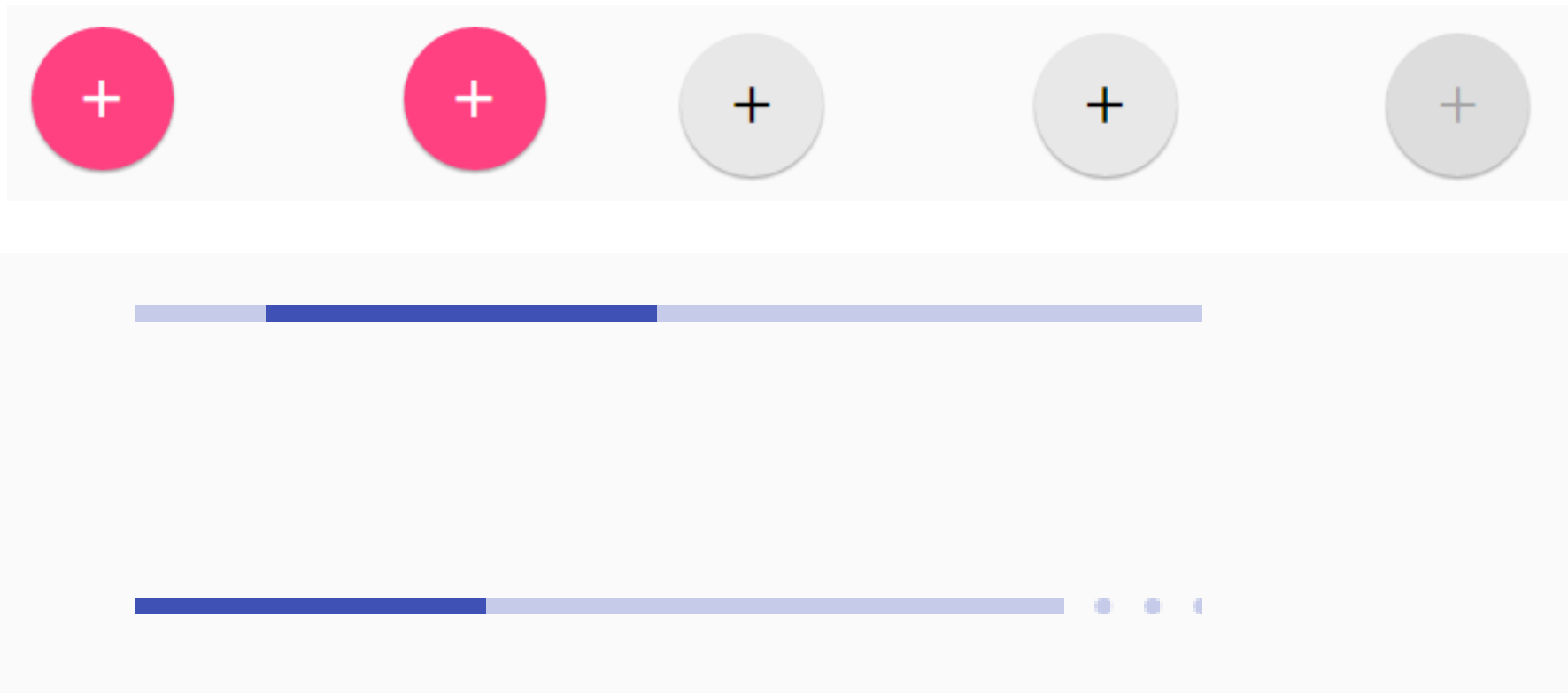
<input type="text" value="Aluno"/>	Nome (Obrigatório)
<input type="text" value="Aluno@unipe.com"/>	Email (Não será publicado) (Obrigatório)
<input type="text"/>	Site/Blog



4. Consistência e padrões

Fale a mesma língua o tempo todo, e nunca identifique uma mesma ação com ícones ou palavras diferentes. Trate coisas similares, da mesma maneira, facilitando a identificação do usuário.

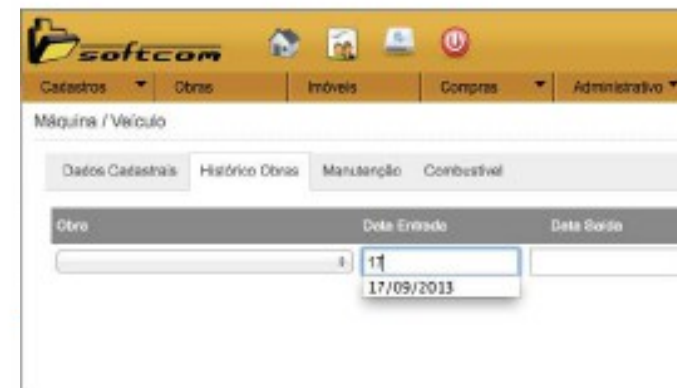
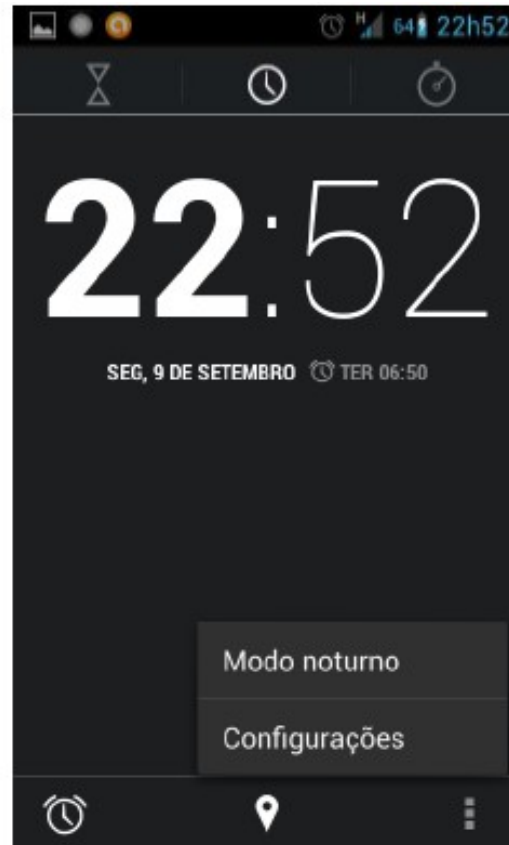
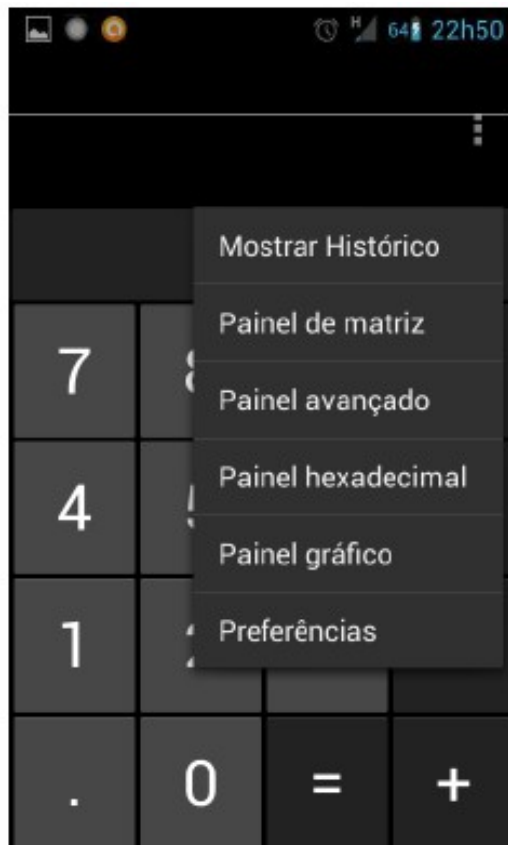
Corretos:



4. Consistência e padrões

Fale a mesma língua o tempo todo, e nunca identifique uma mesma ação com ícones ou palavras diferentes. Trate coisas similares, da mesma maneira, facilitando a identificação do usuário.

Incorretos:



5. Prevenção de erros

Na tradução livre das palavras do próprio Nielsen “Ainda melhor que uma boa mensagem de erro é um design cuidadoso que possa prevenir esses erros”. Por exemplo, ações definitivas, como deleções ou solicitações podem vir acompanhadas de um checkbox ou uma mensagem de confirmação.

Corretos:

CADASTRO PESSOA FÍSICA

☒ Pessoa Física ☐ Pessoa Jurídica

Nome completo *

CPF *

Sexo *

Data de Nascimento *

Telefone Preferencial * -

Telefone Alternativo -

Telefone Comercial -

(DDD - TELEFONE) (OPCIONAL)

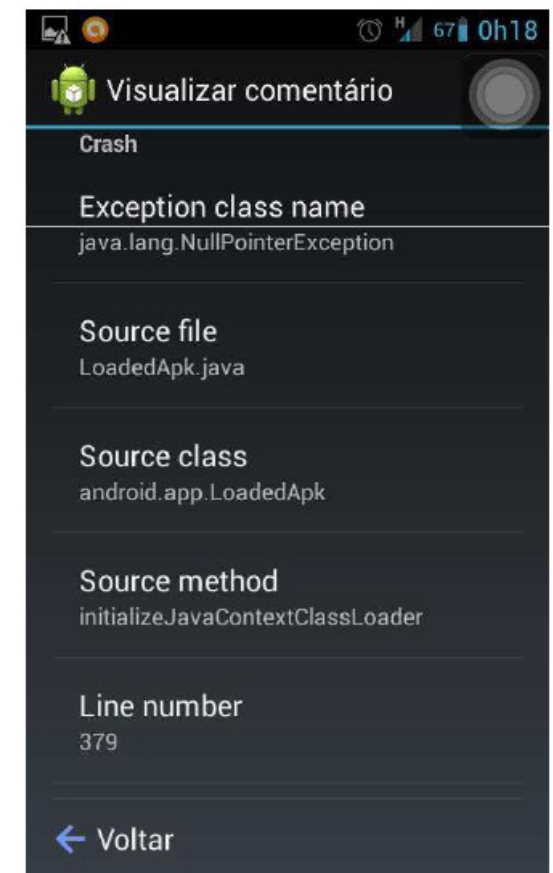
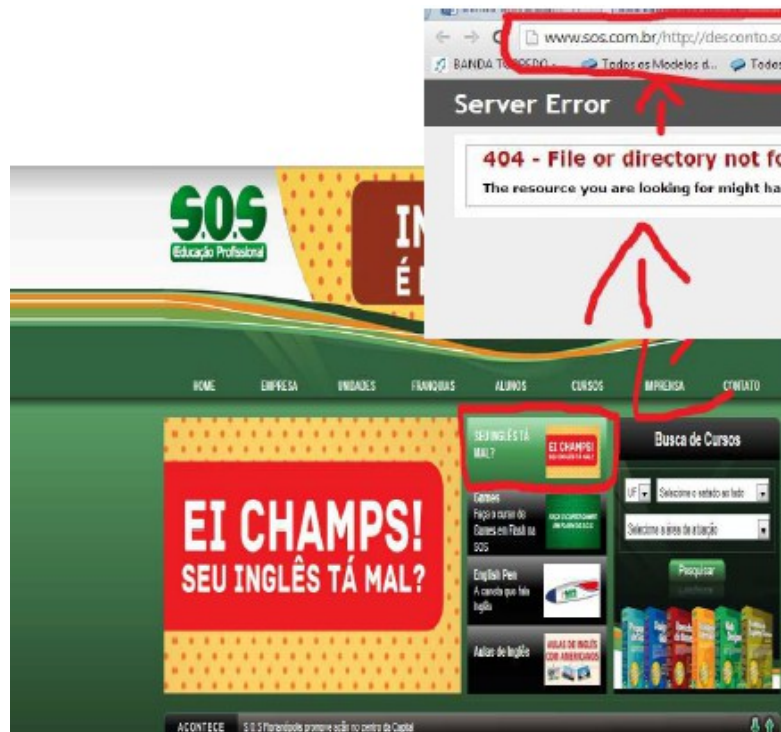
Apenas números, sem pontos ou traços.
Exemplo: 00000000000
O CPF deve ser da pessoa cadastrada

Pesquisar	
Palavras-chave	<input type="text" value="informática"/>
Tipo	<input type="text" value="Todos (E)"/>
Pesquisar em	<input checked="" type="checkbox"/> TinrDN <input checked="" type="checkbox"/> Notícias <input checked="" type="checkbox"/> SmartPartner <input checked="" type="checkbox"/> WF-Downloads
Regra de pesquisa	As palavras-chave com menos de 5 caracteres serão ignoradas
<input type="button" value="Pesquisar"/>	

5. Prevenção de erros

Na tradução livre das palavras do próprio Nielsen “Ainda melhor que uma boa mensagem de erro é um design cuidadoso que possa prevenir esses erros”. Por exemplo, ações definitivas, como deleções ou solicitações podem vir acompanhadas de um checkbox ou uma mensagem de confirmação.

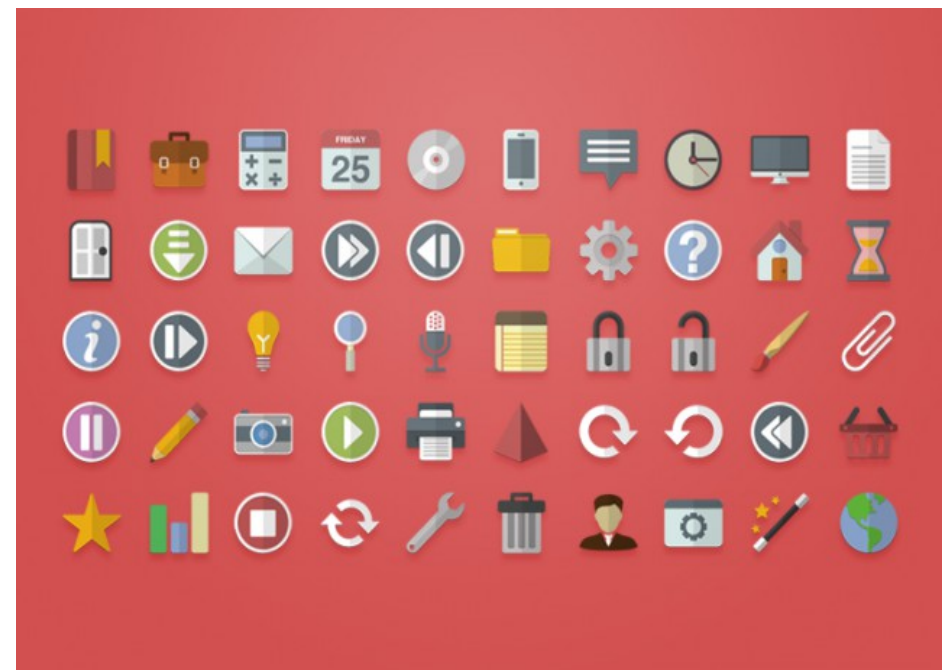
Incorretos:



6. Reconhecimento ao invés de lembrança

Evite acionar a memória do usuário o tempo inteiro, fazendo com que cada ação precise ser revista mentalmente antes de ser executada. Permita que a interface ofereça ajuda contextual, e informações capazes de orientar as ações do usuário – ou seja – que o sistema dialogue com o usuário.

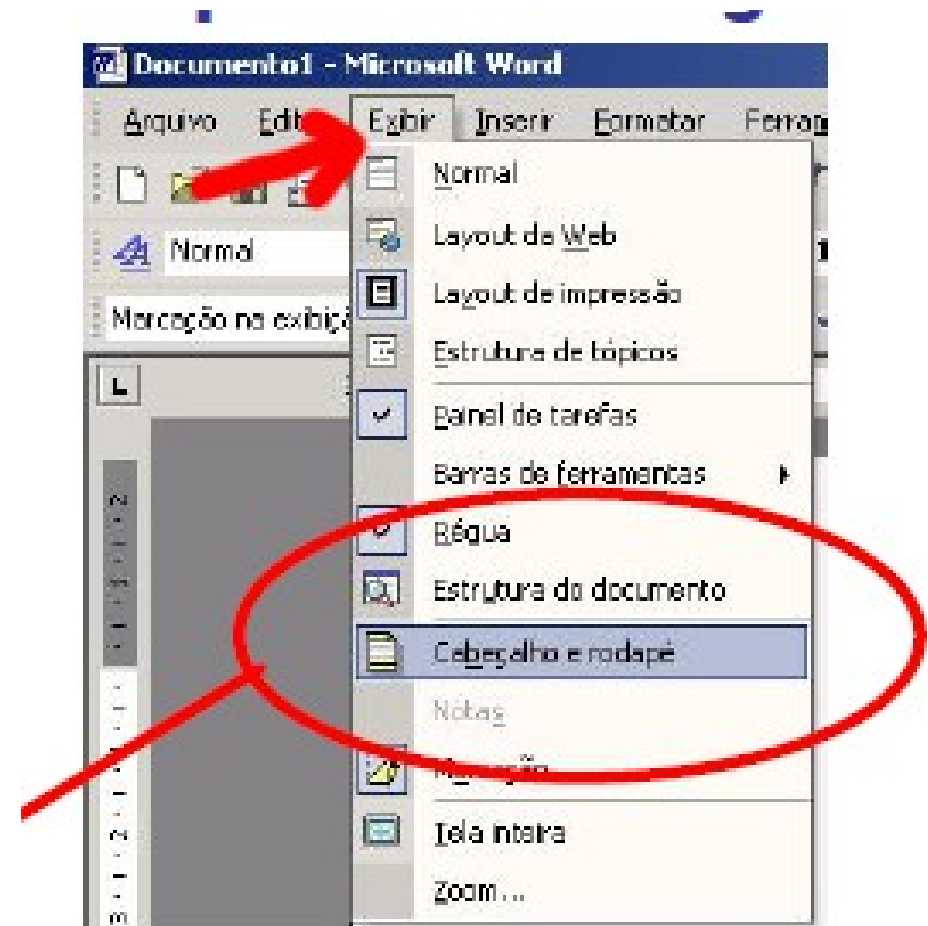
Corretos:



6. Reconhecimento ao invés de lembrança

Evite acionar a memória do usuário o tempo inteiro, fazendo com que cada ação precise ser revista mentalmente antes de ser executada. Permita que a interface ofereça ajuda contextual, e informações capazes de orientar as ações do usuário – ou seja – que o sistema dialogue com o usuário.

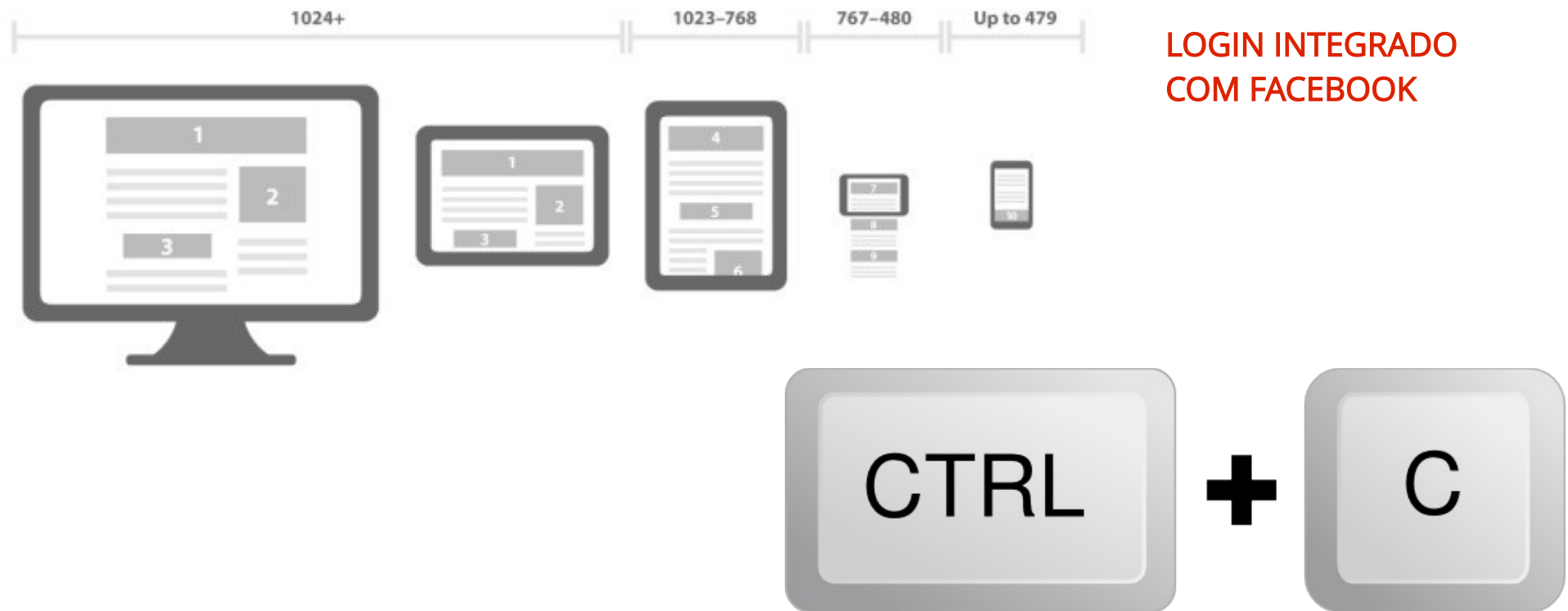
Incorretos:



7. Flexibilidade e eficiência de uso

O sistema precisa ser fácil para usuários leigos, mas flexível o bastante para se tornar ágil à usuários avançados. Essa flexibilidade pode ser conseguida com a permissão de teclas de atalhos, por exemplo. No caso de websites, uso de máscaras e navegação com tab em formulários são outros exemplos.

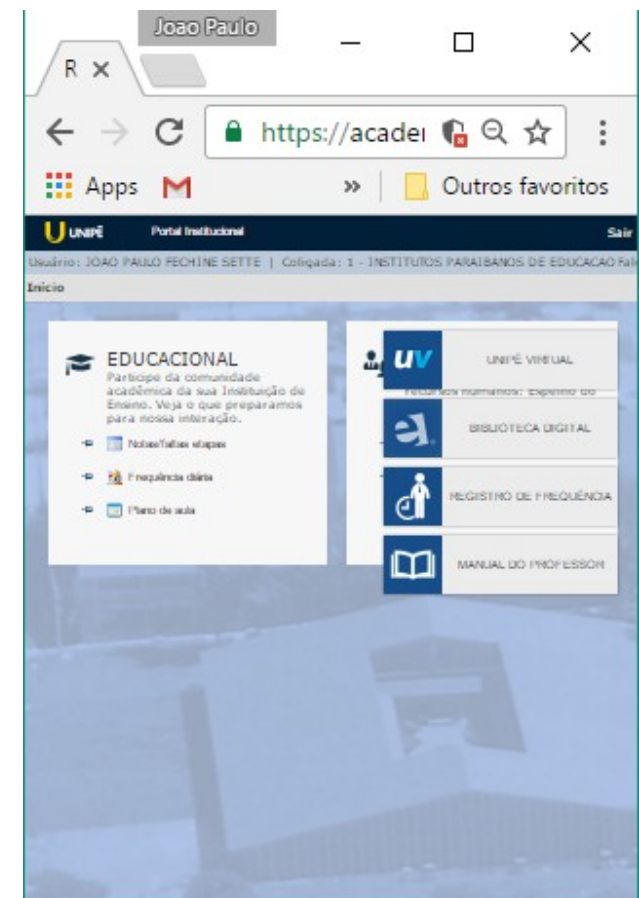
Corretos:



7. Flexibilidade e eficiência de uso

O sistema precisa ser fácil para usuários leigos, mas flexível o bastante para se tornar ágil à usuários avançados. Essa flexibilidade pode ser conseguida com a permissão de teclas de atalhos, por exemplo. No caso de websites, uso de máscaras e navegação com tab em formulários são outros exemplos.

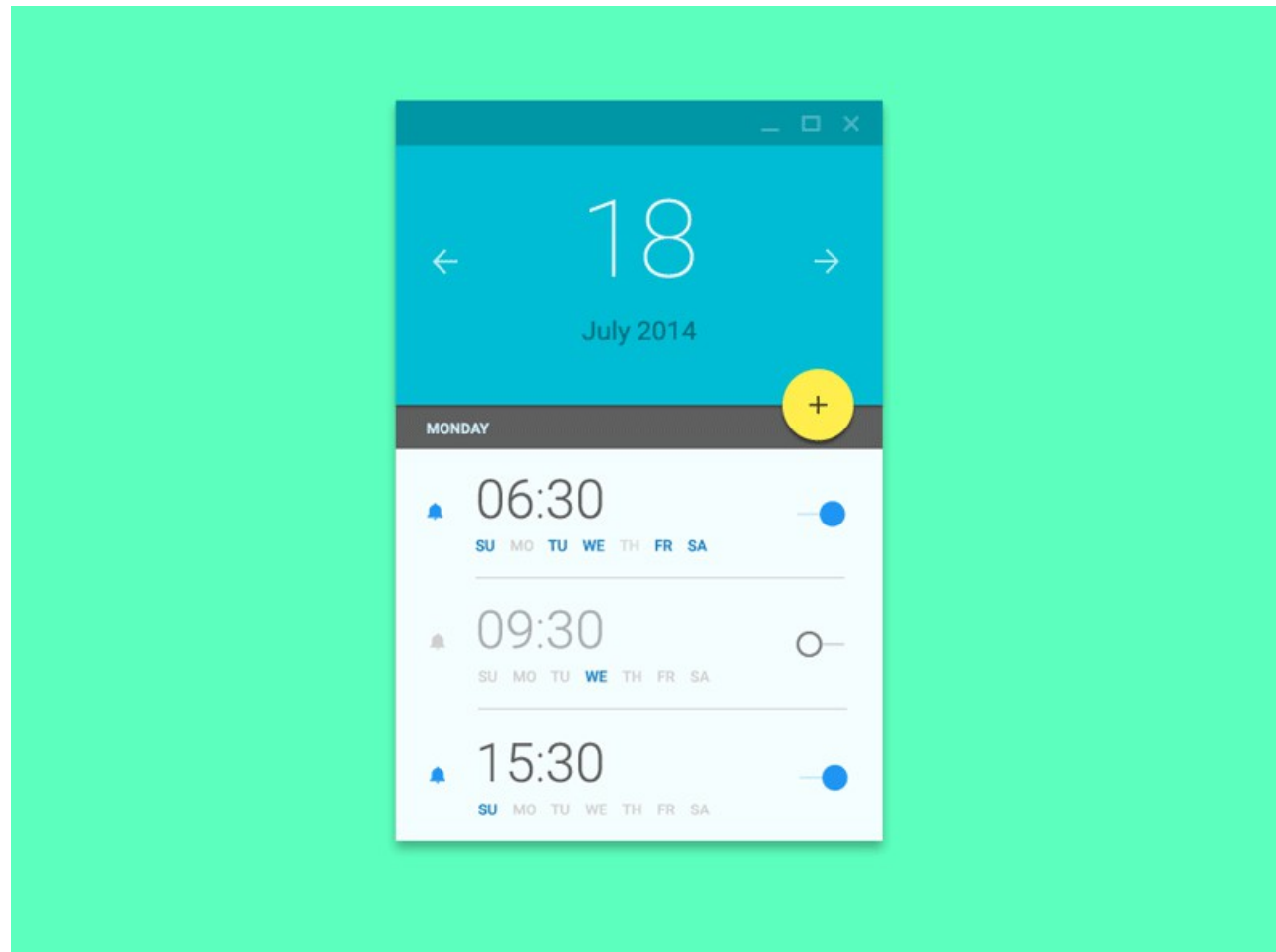
Incorretos:



8. Estética e Design Minimalista

Evite que os textos e o design fale mais do que o usuário necessita saber. Os “diálogos” do sistema precisam ser simples, diretos e naturais, presentes nos momentos em que são necessários.

Corretos:



8. Estética e Design Minimalista

Evite que os textos e o design fale mais do que o usuário necessita saber. Os “diálogos” do sistema precisam ser simples, diretos e naturais, presentes nos momentos em que são necessários.

Incorretos:



9. Ajude os usuários a reconhecer, diagnosticar e sanar erros

As mensagens de erro do sistema devem possuir uma redação simples e clara que ao invés de intimidar o usuário com o erro, indique uma saída construtiva ou possível solução.

Corretos:



9. Ajude os usuários a reconhecer, diagnosticar e sanar erros

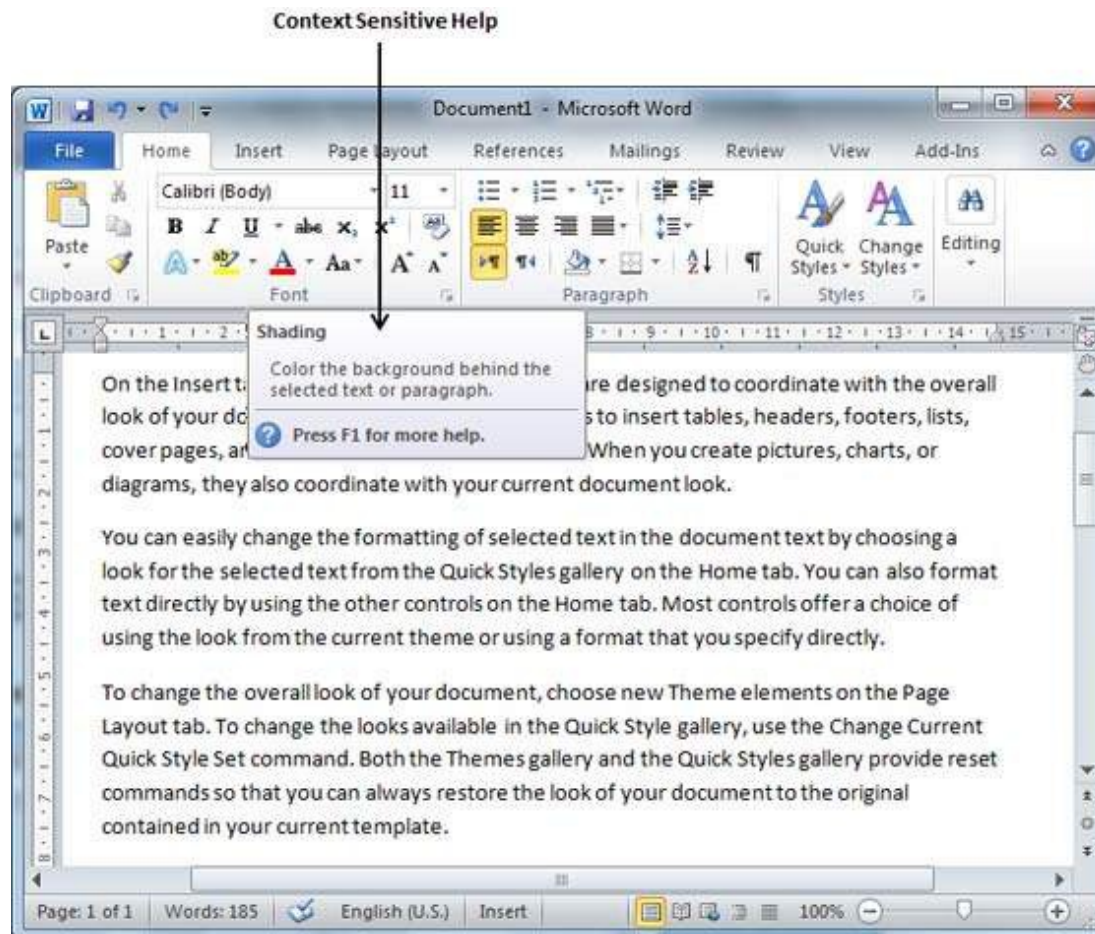
As mensagens de erro do sistema devem possuir uma redação simples e clara que ao invés de intimidar o usuário com o erro, indique uma saída construtiva ou possível solução.

Incorretos:



10. Ajuda e documentação

Um bom design deveria evitar ao máximo a necessidade de ajuda na utilização do sistema. Ainda assim, um bom conjunto de documentação e ajuda deve ser utilizado para orientar o usuário em caso de dúvida. Deve ser visível, facilmente acessada, e com oferecer uma ferramenta de busca na ajuda.



Resumo das Heurísticas de Nielsen

Abaixo, uma lista de resumo, mas não absoluta, para avaliação das Heurística de Nielsen.

- **Visibilidade do status do sistema:**
 - Url mostra a página atual?
 - Página possui título claro e visível?
 - As alterações de estado são representadas?
 - Mudanças de informação são claras?
- **Compatibilidade do sistema com o mundo real:**
 - Os textos são condizentes com as pessoas que o utilizarão?
 - As imagens são claras para o público-alvo?
 - O tamanho dos textos e dos botões são possíveis de leitura para a maior parte do público?
- **Controle do usuário e liberdade:**
 - Como usuário posso desistir de uma ação?
 - Como usuário posso cancelar uma ação?
 - As minhas ações podem ser alteradas livremente?
- **Consistência e padrões:**
 - Os botões estão sempre alinhados e coloridos de forma coerente?
 - Os textos, títulos e links possuem cores bem definidas e estão padronizados durante todo o site?
 - As telas possuem estrutura visual semelhante entre si?
 - As telas usam os padrões da Web para sua distribuição de conteúdo?

- **Prevenção de erros:**
 - Botões de ações diferentes estão a uma distância segura?
 - Botões de múltiplas exclusões necessitam confirmação?
 - Campos com máscaras informam adequadamente como preencher o campo?
 - Páginas oferecem contexto e informações para facilitar o preenchimento
- **Reconhecimento em lugar de lembrança:**
 - Software deixa claro onde estão seus itens e a estrutura facilita o entendimento do uso?
 - Iconografia é clara e funcional?
 - As affordances são claras?
 - As cores corroboram com o entendimento global (azul/verde, amarelo, vermelho)?
- **Flexibilidade e eficiência de uso:**
 - Existem atalhos para otimizar o uso?
 - Existem ferramentas para agilizar determinadas ações?
 - O site apresenta responsividade?
- **Estética e design minimalista:**
 - O site possui a quantidade de elementos aceitável, para não gerar dúvida e exaustão cognitiva?
 - A maior parte dos elementos do layout são úteis?
- **Ajudar o usuário a reconhecer, diagnosticar e corrigir erros:**
 - Textos contextuais são apresentados?
 - Mensagem de erros são claras e ajudam o usuário?
- **Facilita documentar**
 - O software é de fácil explicação para documentação?

5. Conclusões do Módulo II

Embora existam várias formas de teste de usabilidade as de mais fácil implantação são questionários e testes A/B (pós desenvolvimento), Heurísticas (na fase de desenvolvimento) e CARDS (planejamento). Das três ferramentas indicadas, acredito que as Heurísticas de Nielsen possuem um resultado mais sólido, auxiliando a adequação do software às normas ISO e NBR.

Enfim, testar usabilidade é bastante subjetivo, pois, não existem ferramentas que façam todo o trabalho automaticamente. **Um bom testador e um plano de teste bem definido é o que realmente irá garantir bons testes.** Ainda assim algumas ferramentas de automação de testes de interface podem auxiliar o teste de usabilidade, como é o caso do SELENIUM entre outros.

No nosso próximo encontro iremos focar no modelo de interface da Google (Material Design), e utilizar ferramentas de prototipação (Adobe XD, Figma etc).

Até lá!