4633H-02 PROJETO DE INTERFACES – FACIN/PUCRS
Trabalho - AVALIAÇÃO HEURÍSTICA

[Repositórios de Código]
= TURMA 168 =

Objetivo

Estudar o método de avaliação heurística através de sua aplicação em um **Repositório de Código** (por exemplo, Github, Assembla, Bitbucket, Google Code, etc.).

Tarefa

Definir – no mínimo – **duas** tarefas a serem analisadas, lembrando que estas tarefas **devem ser interativas** (não apenas ler informações em uma página). IMPORTANTE: as tarefas devem ser validadas com a professora antes de se iniciar a avaliação.

Etapas do trabalho

- Definir a equipe: 3 a 5 componentes.
- Reunir a equipe e escolher o sistema e as tarefas a analisar.
- Cada componente da equipe deverá avaliar o sistema **individualmente**:
  - Cada avaliador deve percorrer a interface pelo menos duas vezes. Na primeira vez deve se concentrar no fluxo e na segunda nos componentes individuais do diálogo.
  - A interface deve ser inspecionada com base nas heurísticas vistas em aula e **todos os problemas devem ser justificados e detalhados o máximo possível** (devem ser **incluídas** telas do sistema mostrando os problemas quando de sua ocorrência. No final deste documento há um exemplo de descrição de problemas na Avaliação Heurística).
  - Cada avaliador deverá gerar um relatório contendo o detalhamento de todos os problemas por **ele** encontrados e, para cada problema, associar a figura que o ilustra e a heurística violada.
- Consolidação dos resultados **pela equipe**:
  - Discussão e listagem dos problemas encontrados (listagem geral).
  - Definição dos graus de severidade dos problemas encontrados a fim de priorizar sua correção.
  - Elaboração de sugestão(ões) para o *redesign* de cada problema.

O QUE DEVE SER ENTREGUE POR CADA EQUIPE DE AVALIAÇÃO

Relatório **contendo**:

1. **Introdução** (descrição do objetivo da avaliação e do método de avaliação utilizado)
2. **Descrição das Tarefas** (descrição do sistema e das tarefas avaliadas)
3. **Consolidação dos Resultados** (consolidação feita pela equipe, com listagem geral dos problemas encontrados e, para cada problema: descrição detalhada do problema, incluindo as imagens; heurística associada; grau de severidade atribuído pela equipe e sugestão de *redesign* feita pela equipe*).*
4. **Considerações Finais** (apreciação do grupo quanto à aplicação do método de avaliação utilizado – dificuldades, vantagens, desvantagens, etc)

Datas Importantes

27/03 (LAB 309): detalhamento do trabalho, definição das equipes, sistema e tarefas, início da avaliação individual.

Até 10/04 (*extra-classe*): finalização da avaliação **individual**. **IMPORTANTE**: o relatório das avaliações individuais será utilizado para a consolidação pelo grupo (e será verificado, em aula, pela professora).

10/04 (LAB 309): de posse do relatório das avaliações individuais, as equipes devem se reunir e consolidar seus dados. Cada aluno **deverá trazer seu relatório individual** neste dia para discussão com seu grupo e, também, para apresentá-lo para a professora (a avaliação individual não deverá ser incluída no Relatório Final).

17/04 (LAB 310): entrega do trabalho. O trabalho deverá ser entregue no **início da aula** (esta aula será destinada ao estudo de novo conteúdo), **impresso**, um por equipe, com os itens detalhados na seção anterior.

Avaliação do Trabalho

AVALIAÇÃO INDIVIDUAL (apresentada para a professora na aula do dia 10/04)	2,0
AVALIAÇÃO DO GRUPO (Relatório)	
Introdução e Descrição das Tarefas	0,5
Consolidação dos Resultados Individuais	
Descrição do Problema	2,5
Ilustração	1,0
Heurística Associada	2,0
Severidade	0,5
Sugestão de *Redesign*	1,0
Considerações Finais	0,5
Total	**10,0**

ANEXOS

Resumo das Heurísticas [Nielsen 1993]

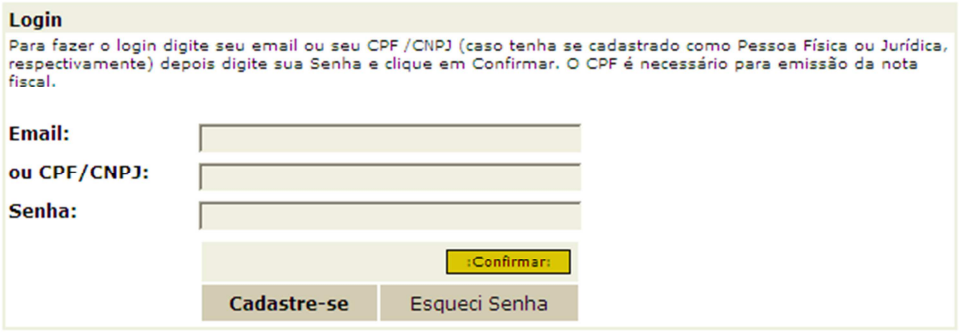
1. **visibilidade do estado do sistema:** O sistema deve sempre manter os usuários informados sobre o que está acontecendo através de *feedback* adequado e no tempo certo.
2. **correspondência entre o sistema e o mundo real:** O sistema deve falar a língua do usuário, com palavras, expressões e conceitos que lhe são familiares, em vez de utilizar termos orientados ao sistema. O projetista deve seguir as convenções do mundo real, fazendo com que a informação apareça em uma ordem natural e lógica.
3. **controle e liberdade do usuário:** Os usuários freqüentemente escolhem funções do sistema por engano e precisam de uma "saída de emergência" claramente marcada para sair do estado indesejado sem ter que percorrer um diálogo extenso. A interface deve permitir que o usuário desfaça ou refaça suas ações.
4. **consistência e padronização:** Os usuários não devem ter que se perguntar se palavras, situações ou ações diferentes significam a mesma coisa. O projetista deve seguir as convenções da plataforma ou ambiente.
5. **prevenção de erros:** Melhor do que uma boa mensagem de erro é um projeto cuidadoso que evite que um problema ocorra.
6. **reconhecimento em vez de lembrança:** O projetista deve tornar os objetos, ações e opções visíveis. O usuário não deve ter que se lembrar de informação de uma parte do diálogo para uma outra. As instruções de uso do sistema devem estar visíveis ou facilmente acessíveis sempre que necessário.
7. **flexibilidade e eficiência de uso:** Aceleradores — imperceptíveis aos usuários novatos — podem tornar a interação do usuário mais rápida e eficiente, permitindo que o sistema consiga servir igualmente bem os usuários experientes e inexperientes. O projetista pode prover mecanismos a serem utilizados pelos usuários para customizar ações freqüentes.
8. **projeto estético e minimalista:** Os diálogos não devem conter informação que seja irrelevante ou raramente necessária. Cada unidade extra de informação em um diálogo compete com as unidades relevantes de informação e reduz sua visibilidade relativa.
9. **auxilia os usuários a reconhecerem, diagnosticarem e se recuperarem de erros:** As mensagens de erro devem ser expressas em linguagem simples (sem códigos), indicar precisamente o problema e sugerir uma solução de forma construtiva.
10. **ajuda e documentação:** O sistema deve prover ajuda e documentação. Este tipo de informação deve ser fácil de ser encontrado, focado na tarefa do usuário, enumerar passos concretos a serem realizados, e não ser muito grande.

Graus de Severidade [Nielsen 1993]

0 **Não concordo** que isto seja um problema de usabilidade.
1 **Cosmético** - não precisa ser consertado a menos que haja tempo extra.
2 **Pequeno** - o conserto deve receber baixa prioridade.
3 **Grande** – importante de ser consertado; deve receber alta prioridade.
4 **Catastrófico** – é imperativo consertar antes de se lançar o produto.

Exemplo de Avaliação Heurística [Barbosa e Silva 2010]

Avaliação da seção de login do site da Livraria Galileu, disponível em https://www.livrariagalileu.com.br/home/login.asp?origem=0&uid=554491138



Violação das seguintes heurísticas:

- **Visibilidade do estado do sistema, prevenção de erros**
  Problema: O elemento secundário *Cadastre-se* tem mais destaque do que o elemento *Confirmar*. Isso pode levar o usuário a acionar o botão errado ou se perguntar se entrou corretamente na tela de login, e até mesmo voltar para a página anterior e repetir a operação de acesso a essa página.
  - Local: abaixo do formulário, apenas nessa tela.
  - Severidade: 3 (problema grande), pois o usuário pode acreditar que precisa se cadastrar a cada compra, ou que o sistema está com defeito, e com isso pode desistir de efetuar a compra através desse site.
  - Recomendação: destacar o botão primário (*Confirmar*) e reduzir a ênfase dos botões secundários (*Cadastrar-se* e *Esqueci Senha*). Considere modificar os botões secundários para links, mais afastados do botão primário do formulário.

- **Controle e liberdade do usuário**
  Problema: Os usuários não têm a opção, através do Web site, de voltar à página anterior. Para isso, precisam utilizar o botão de voltar do próprio navegador.
  - Local: ausência de um botão de volta em todos os formulários do site.
  - Severidade: 2 (problema pequeno). O usuário está acostumado a utilizar o botão de volta do navegador em outros sites, e perceberá que pode fazer isso sem perder o que tenha feito no site (e.g., itens colocados no carrinho de compras).
  - Recomendação: incluir um botão *Voltar* como botão secundário do formulário.

- **Consistência e padronização, prevenção de erros**
  Problema: Os campos de preenchimento alternativo (“Email:” e ou “CPF/CNPJ:”) não estão claramente marcados, como de costume, por botões de opção (*radio buttons*). Como os usuários costumam seguir dicas visuais melhor do que instruções textuais, muitos preencherão os dois campos.
  - Local: formulário de login, campos “Email:” e “ou CPF/CNPJ:”.
  - Severidade: 2 (problema pequeno). Apesar de ineficiente, o preenchimento dos dois campos não impede o usuário de efetuar o login.
  - Recomendação: identificar os campos alternativos por botão de opção, que devem ser automaticamente selecionados quando o usuário inicia a digitação no campo correspondente.

- **Flexibilidade e eficiência de uso, consistência e padronização**
  Problema: O usuário não tem a opção de pedir para o sistema se lembrar do seu e-mail ou mesmo manter seu login ativo, como ocorre em boa parte dos sites de comércio eletrônico.
  - Local: formulário de login, ausência de botões de seleção (checkboxes).
  - Severidade: 2 (problema pequeno) para usuários ocasionais; 3 (problema grande) para usuários freqüentes, que provavelmente darão preferência a Web sites que se lembrem “deles”.
  - Recomendação: oferecer um checkbox *Lembrar dos meus dados* e/ou um checkbox *Manter meu login ativo por 15 dias*.