

# Análise e Desenvolvimento de Sistemas

- Interação Humano-Computador
  - *Profa. Dra. Daniela Gibertoni*

# Análise e Desenvolvimento de Sistemas

- Técnicas de Avaliação
  - *Inspeção de Usabilidade*
  - *Avaliação Heurística*
  - *Percurso Cognitivo*

## Testes de Usabilidade

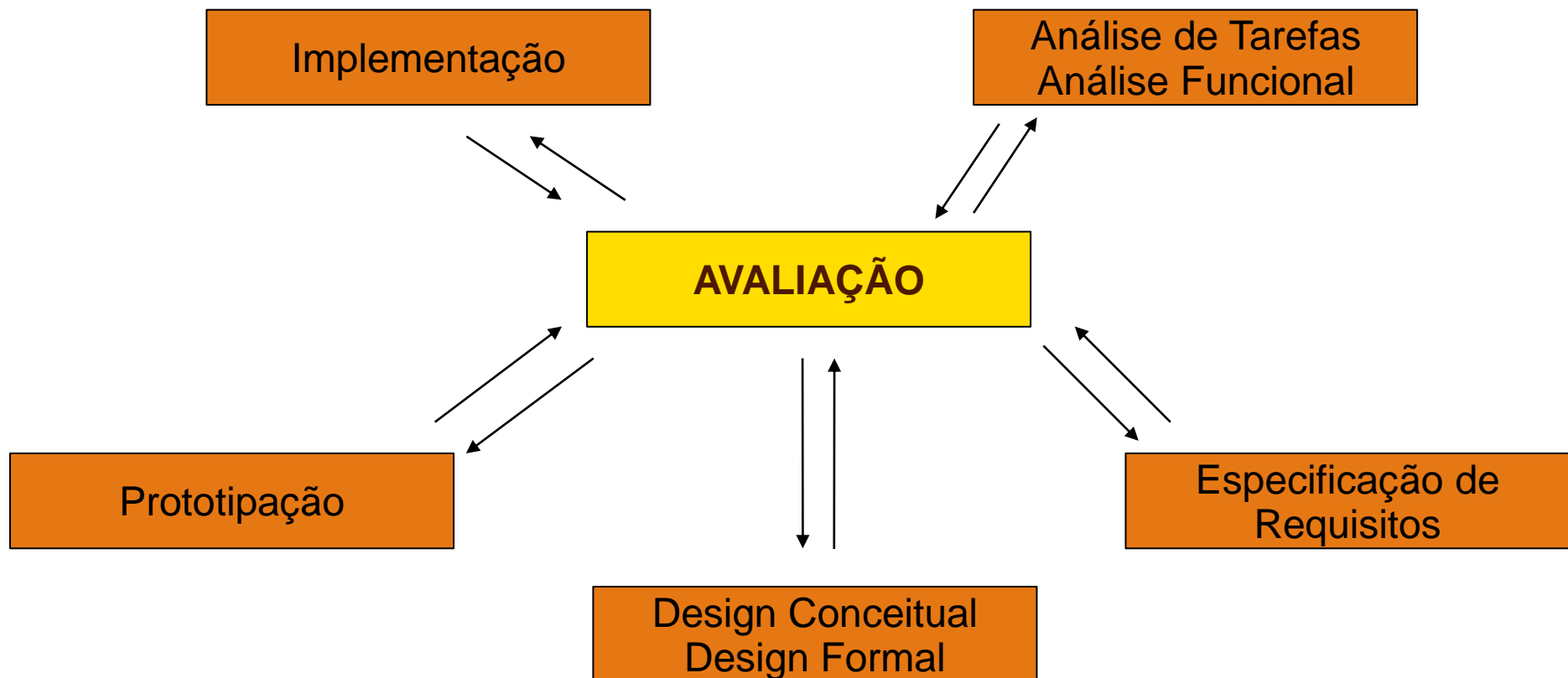
- *Think Aloud*
- *Medidas de Performance*

# Análise e Desenvolvimento de Sistemas

- Por quê?
- Avaliar adequação de designs
- Testar a **usabilidade** e a **funcionalidade** do sistema
  - Requisitos propostos estão sendo atendidos?
  - Sistema comporta-se como esperado?
  - Usuário fica satisfeito e consegue realizar suas tarefas?

# Análise e Desenvolvimento de Sistemas

- Pode ou não envolver usuários finais
- Avaliação não deve ser uma fase única no final do processo de design



# Análise e Desenvolvimento de Sistemas

- Local de avaliação:



# Análise e Desenvolvimento de Sistemas

## Em laboratório

- Experimental: formal e objetiva
- Com designers
- Com usuários (testes de usabilidade)
  - Facilidade de registro
  - Computadores “preparados”
  - “Espelho” de duas faces
- Situação artificial
  - Sem interrupção, sem contexto, sem conversa, sem interferência do ambiente

# Análise e Desenvolvimento de Sistemas

## Em laboratório

- Indicações
  - Uso real oferece perigo ou distância;
  - Avaliação de tarefas restritas;
  - A manipulação do contexto é desejada;
    - Comparação entre alternativas de design;
    - Detecção de problemas/aspectos específicos;
    - Observação de procedimentos pouco usados;

# Análise e Desenvolvimento de Sistemas

## Em laboratório

Algumas críticas ao teste de hipóteses científicas em um ambiente fechado

- Laboratório não é o mundo real;
- Não pode controlar todas as variáveis;
- Contexto é negligenciado;
- Tarefas testadas são artificiais e curtas;



# Análise e Desenvolvimento de Sistemas

## *In loco*

- Designer/avaliador vai ao ambiente de trabalho do usuário para observá-lo em ação com o sistema
- Situação de uso real, “quase natural”:
  - Interrupções, ruído, influência da presença do ‘observador’, humano ou não...
- Certas tarefas são difíceis de observar em laboratório ou *in loco*
- Custo é um aspecto relevante
  - Tirar usuário do trabalho x montar uma infra-estrutura de observação no local (e talvez atrapalhar...)

# Análise e Desenvolvimento de Sistemas

## Métodos de Avaliação

### Inspeção de Usabilidade

- Pode ou não envolver usuários
- Aplicável a qualquer fase do desenvolvimento de um sistema (implementado ou não)

### Testes de Usabilidade

- Métodos de avaliação centrados no usuário
  - Métodos experimentais ou empíricos
  - Métodos observacionais
  - Métodos de questionamento (consultivos)

# Análise e Desenvolvimento de Sistemas

## Inspeção de Usabilidade

- Métodos empíricos
  - Usam a habilidade e experiência do avaliador
- Métodos não mutuamente exclusivos
  - Percurso Cognitivo (*Cognitive Walkthrough*)
  - Avaliação Heurística
  - Revisão de *Guidelines*
- Estudos demonstram que são um bom complemento aos testes de usabilidade realizados com o usuário

# Análise e Desenvolvimento de Sistemas

## Inspeção de Usabilidade

### Objetivos

- Encontrar problemas de usabilidade no design de uma interface
- Fazer recomendações no sentido de melhorar a usabilidade do design

### Trabalho de inspeção

- Identificar, classificar e contar o número de problemas de usabilidade

Problema de usabilidade: qualquer aspecto do design que possa causar usabilidade reduzida ao usuário final do sistema.

# Análise e Desenvolvimento de Sistemas

## **Inspeção de Usabilidade**

Um único problema pode se manifestar de várias formas:

- pode dificultar o aprendizado
- atrasar o aprendizado (torná-lo lento)
- causar erros no sistema
- simplesmente ser feio/desagradável aos olhos do usuário

# Análise e Desenvolvimento de Sistemas

## **Inspeção de Usabilidade**

A identificação do problema é APENAS PARTE do processo

A equipe deve:

- Propor um redesign da interface
- Procurar corrigir o maior número de erros
- Erros devem ser priorizados de acordo com a gravidade

# Análise e Desenvolvimento de Sistemas

## Inspeção de Usabilidade – Vantagens

- Métodos facilmente integráveis a processos convencionais de desenvolvimento de software
- Fornecem evidências concretas de quais aspectos do design devem ser aperfeiçoados
- Não exigem muita experiência e longo treinamento para que possam ser realizados
- Boa experiência educacional para designers novatos

# Análise e Desenvolvimento de Sistemas

## Métodos para Inspeção de Usabilidade

- Avaliação Heurística
  - Inspeção da interface tendo como base uma pequena lista de heurísticas de usabilidade
- Percurso Cognitivo (Cognitive Walkthrough )
  - Avaliador simula um usuário típico 'percorrendo' a interface para executar tarefas típicas
  - Tarefas mais frequentes
  - Tarefas críticas



# Análise e Desenvolvimento de Sistemas

## Avaliação Heurística

- Nielsen, 1993: Discount Usability Engineering
- [http://www.useit.com/papers/guerrilla\\_hci.html](http://www.useit.com/papers/guerrilla_hci.html)
- Avaliação heurística é o principal método
- Métodos baratos, rápidos e fáceis de serem usados

# Análise e Desenvolvimento de Sistemas

## Avaliação Heurística

- Fácil: pode ser ensinada em 4 hs
- Rápida: maioria das avaliações requer cerca de um dia
- Barata: tanto quanto se deseje
- Pequeno conjunto de avaliadores examina a interface e julga suas características em face de reconhecidos princípios de usabilidade (heurísticas)

# Análise e Desenvolvimento de Sistemas

## Avaliação Heurística

- Deve ser feita por mais de uma pessoa
  - Experiência indica que diferentes pessoas encontram diferentes problemas
- Resultados melhoram significativamente utilizando múltiplos avaliadores
- Estudos mostram que a relação direta entre número de avaliadores, a quantidade de problemas identificados e a relação custo/benefício.

# Análise e Desenvolvimento de Sistemas

## Avaliação Heurística

### Procedimento

1. Obter entradas
2. Realizar avaliação independente
3. Discussão/Coleta
4. Atribuição de taxa de severidade

# Análise e Desenvolvimento de Sistemas

## Avaliação Heurística

### Procedimento

#### *Obter entradas*

- Time de avaliação (treinado no método)
- Familiaridade com o domínio
- Storyboards/cenários/protótipos que mostram detalhes suficientes
- Heurísticas!!!

# Análise e Desenvolvimento de Sistemas

## Avaliação Heurística

### Procedimento

*Realizar avaliação independente*

- Julgar protótipo relativamente ao conjunto de heurísticas
- Quais heurísticas?
  - Shneidermann: Oito Regras de Ouro
  - Nielsen: 10 heurísticas
  - Outras...

# Análise e Desenvolvimento de Sistemas

## Avaliação Heurística

### Procedimento

#### *Discussão/Coleta*

- Organizar todos os problemas encontrados

#### *Atribuição de taxa de severidade*

- Filtrar quais são problemas importantes

# Análise e Desenvolvimento de Sistemas

- **Heurísticas de Nielsen**

- 1) Visibilidade do status do sistema

○ sistema deve sempre manter os usuários informados sobre o que está acontecendo, através de feedback apropriado dentro de um tempo de resposta razoável

- 2) Compatibilidade do sistema com o mundo real

○ sistema deve falar a linguagem do usuário, com palavras, frases e conceitos familiares ao usuário (em vez de termos do sistema);

○ projeto deve seguir convenções do mundo real numa ordem natural e lógica;



# Análise e Desenvolvimento de Sistemas

- **Heurísticas de Nielsen**
- 3) Controle do usuário e liberdade

O sistema deve oferecer saídas claras para situações nas quais usuários se encontram por terem escolhido funções erradas do sistema;

Não deve ser necessário um diálogo extenso;

O sistema deve suportar *undo* e *redo*;

# Análise e Desenvolvimento de Sistemas

- **Heurísticas de Nielsen**
- 4) Consistência e padrões

O sistema deve apresentar informações de modo consistente e padronizado;

- 5) Prevenção de erros

Evitar, sempre que possível, que o erro aconteça, informando o usuário sobre as conseqüências de suas ações ou, se possível, impedir ações que levariam a uma situação de erro

# Análise e Desenvolvimento de Sistemas

- **Heurísticas de Nielsen**

- 6) Reconhecimento ao invés de lembrança (memorização)

O sistema deve fazer com que objetos, ações e opções estejam claramente visíveis

O usuário não deve ser obrigado a lembrar informações de uma parte do diálogo para outra

Instruções para uso do sistema devem estar visíveis ou facilmente acessíveis sempre que apropriado

# Análise e Desenvolvimento de Sistemas

- **Heurísticas de Nielsen**

- 7) Flexibilidade e eficiência de uso

O sistema deve oferecer “aceleradores” ao usuário especialista, os quais são invisíveis aos novatos

O sistema deve permitir que usuários customizem ações frequentes

# Análise e Desenvolvimento de Sistemas

- **Heurísticas de Nielsen**
- 8) *Design estético e minimalista*

Os diálogos usuário-sistema não devem conter informações que são irrelevantes ou raramente utilizadas

Cada unidade de informação no diálogo compete com as unidades que são de fato relevantes e, assim, diminuem a sua visibilidade relativa

# Análise e Desenvolvimento de Sistemas

- **Heurísticas de Nielsen**
- 9) Ajudar os usuários a reconhecer, diagnosticar e corrigir erros

Mensagens de erro deve ser expressas em linguagem simples, sem códigos, indicando precisamente o problema e sugerir uma solução de modo construtivo

# Análise e Desenvolvimento de Sistemas

- **Heurísticas de Nielsen**
- 10) Ajuda e documentação

Apesar de ser melhor se o sistema puder ser utilizado sem documentação alguma, há situações em que ajuda e documentação se fazem necessários

Tal apoio deve ser fácil de ser encontrado, estar focado na tarefa do usuário, listar passos concretos a serem realizados, e não serem grandes

# Análise e Desenvolvimento de Sistemas

- **8 Regras de Ouro**
- Shneiderman, 1994.
- Perseguir a consistência;
- Fornecer atalhos;
- Fornecer feedback informativo;
- Marcar o final dos diálogos;



# Análise e Desenvolvimento de Sistemas

- **8 Regras de Ouro**
- Shneiderman, 1994.
- Fornecer prevenção e manipulação simples de erros;
- Permitir o cancelamento das ações;
- Fornecer controle e iniciativa ao usuário;
- Reduzir a carga de memória de trabalho.

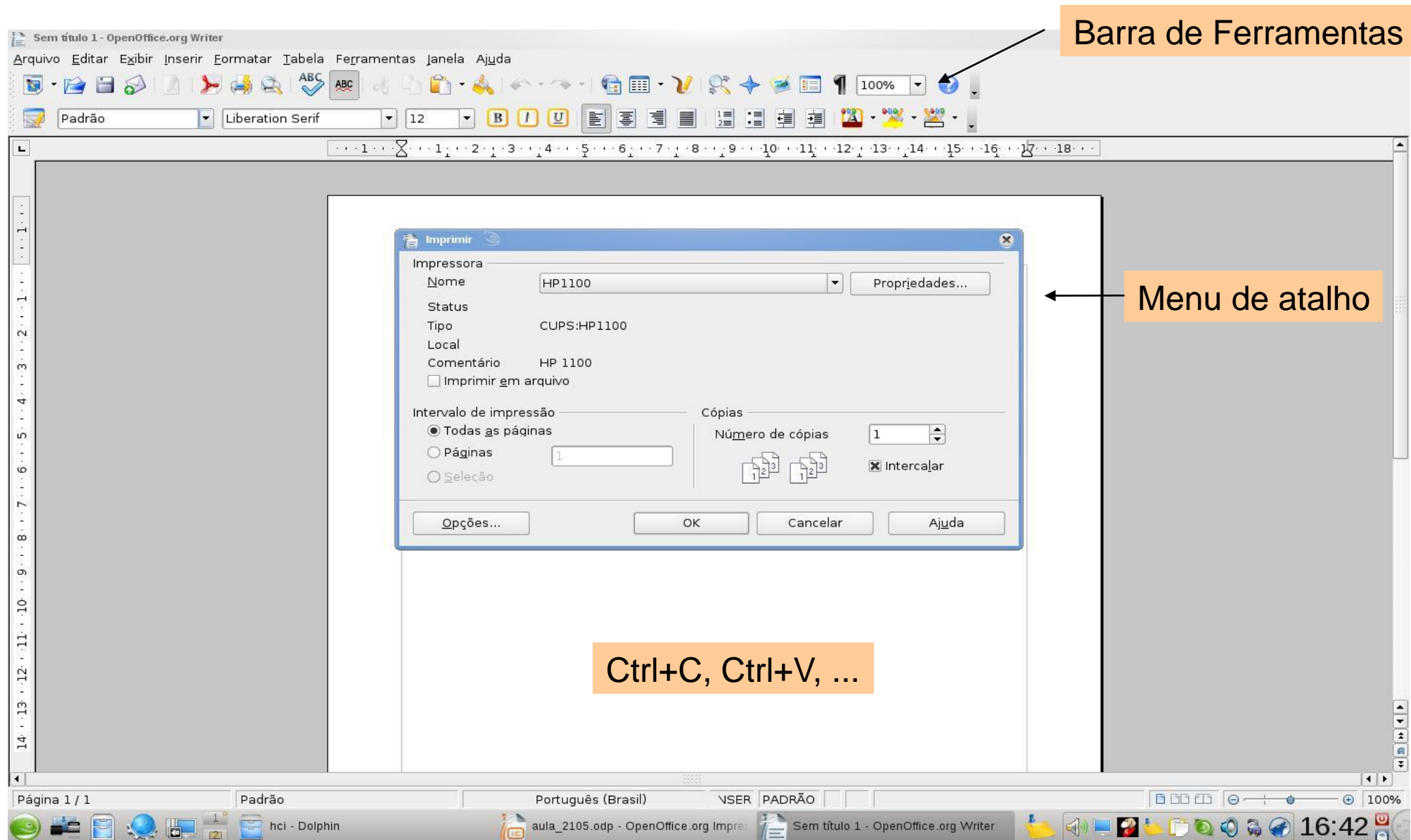
# Análise e Desenvolvimento de Sistemas

- **Oferecer Atalhos**
- Usuários experientes devem ser capazes de realizar tarefas mais rapidamente
- Flexibilidade

## Exemplos

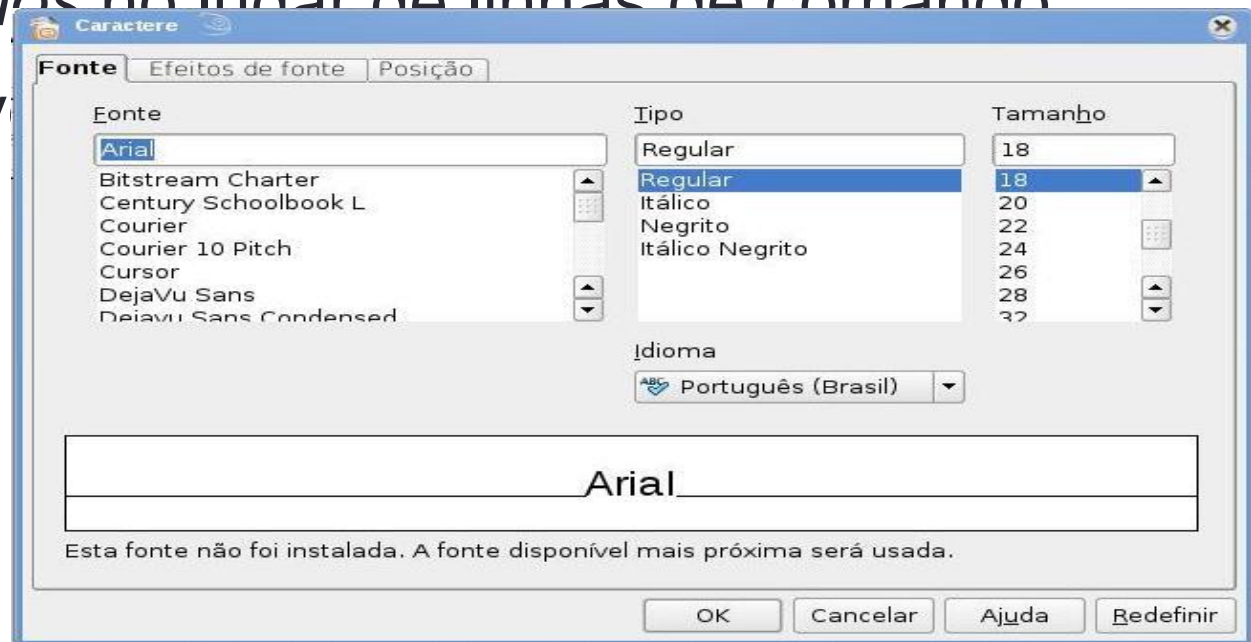
- Teclas de atalho, barra de ferramentas, preenchimento automático de campos, etc.

# Análise e Desenvolvimento de Sistemas



# Análise e Desenvolvimento de Sistemas

- **Minimizar a Memorização**
- Para pessoas, reconhecer é mais fácil que lembrar
- Para computadores, memorizar é mais fácil que reconhecer
- Usar *menus* e *atalhos* no lugar de linhas de comando
- Deixar objetos visíveis



# Análise e Desenvolvimento de Sistemas

- Exercício
- Avaliar uma interface com base nas 10 heurísticas de Nielsen
- Comparar identificação de problemas para priorizar os mais importantes

# Material Consultado

Rocha, H.V.; Baranauskas, M.C.C. *Design e Avaliação de Interfaces Humano-Computador*, NIED/INICAMP, 2003.