

# Uma Plataforma de Software para o Estudo Interativo de Métodos e Algoritmos Econométricos

Carlos Duarte do Nascimento

Instituto de Matemática e Estatística  
Universidade de São Paulo

4 de Março de 2009

# Introdução

# Apresentações

## Banca Avaliadora

- Prof Cicely Moitinho Amaral (orientador)
- Prof. Claudio Possani
- Prof. Sergio Muniz Oliva Filho

# O que este trabalho *não* é

- Análise de um Problema Matemático
- Um Estudo Profundo de Métodos Numéricos
- Apologia (ou Crítica) do Ensino à Distância

# O Problema

# O Ensino de Econometria

- Teoria + Prática: É importante experimentar!
- Opções para experimentar:
  - Softwares específicos de Estatística/Econometria (EViews, SPSS, Stata)
  - Pacotes Matemáticos "puros"(Mathematica, Gnu R, Matlab, Octave)
  - Linguagens de Programação (C, Java, Pascal, Fortran, etc.)
    - Desenvolvimento pelo Professor
    - Desenvolvimento pelo Aluno

# Um Problema Econométrico

(Judge) Estimação de Parâmetros no Modelo:

$$y_t = \theta_1 + \theta_2 x_{t2} + (\theta_2)^2 x_{t3} + e_t, t = 1, 2, \dots, 20$$

$$y = f(\theta) + e$$

$$f(\theta) = \begin{pmatrix} \theta_1 + \theta_2 x_{12} + \theta_2^2 x_{13} \\ \theta_1 + \theta_2 x_{22} + \theta_2^2 x_{23} \\ \vdots \\ \theta_1 + \theta_2 x_{20,2} + \theta_2^2 x_{20,3} \end{pmatrix}$$

Função objetivo (soma quadrática do erro):

$$H(\theta) = [y - f(\theta)]'[y - f(\theta)]$$

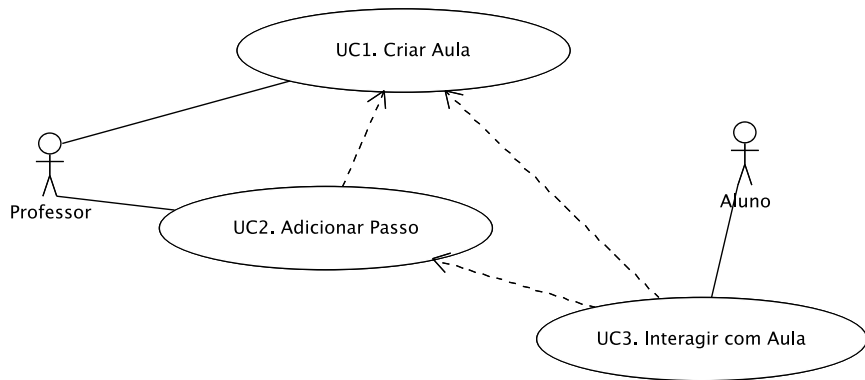
# Modelo de Solução



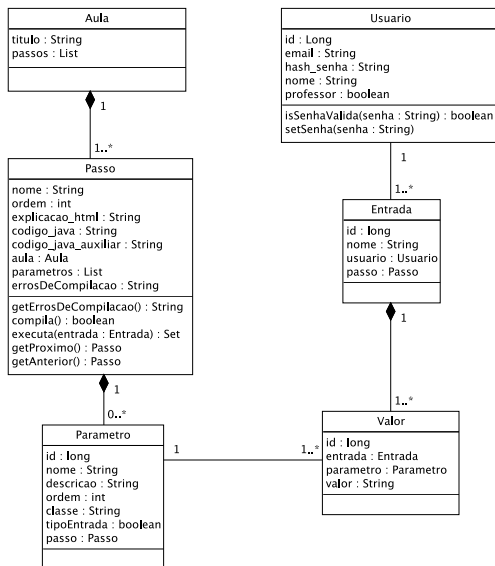
# Proposta Funcional

- Duas Categorias de Usuários:
  - Professores (cadastram aulas)
  - Alunos (interagem com aulas)
- Aulas Divididas em Passos
- Passos Divididos em:
  - Parte Teórica: Texto/HTML
  - Parte Prática: Algoritmo interativo

# Casos de Uso



# Diagramas de Classe



# Arquitetura de Software

# Escolha da Linguagem

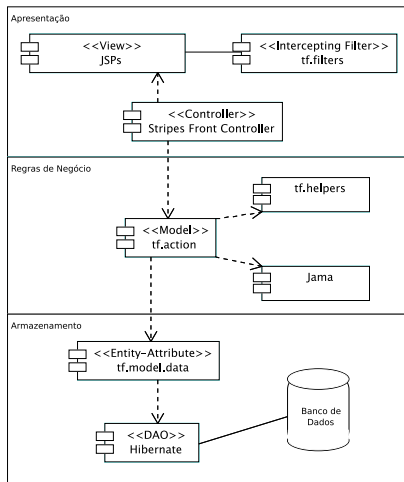
Cada uma apresenta suas vantagens:

- C/C++: Performance
- Pascal: Simplicidade
- Fortran: Material Acadêmico
- Java: Equilíbrio destes fatores; facilidade para compilação dinâmica; JAMA

# Padrões de Projeto

- Mapeamento Objeto-Relacional
- Model / View / Controller
- Inversão de Controle / Injeção de Dependências

# Componentes



# Compilação Dinâmica de Algoritmos

- Idéia: usar a própria linguagem para executar os algoritmos
- Cadastro do Algoritmo (ex.: Cálculo de Juros):
  - Parâmetros de Entrada  
(*taxa, valor*)
  - Parâmetros de Saída  
(*juros*)
  - Algoritmo  
( $juros = valor * (1 + taxa)$ ;  $valor = valor + juros$ )
  - Código Auxiliar (opcional)  
(*função para Tabela Price*)
- O sistema monta o código em tempo real usando estes elementos, e o *javac* se encarregad o resto.



# Demonstração

# Uma Aula Prática

# Um Problema Econométrico (retomando)

Modelo:

$$y_t = \theta_1 + \theta_2 x_{t2} + (\theta_2)^2 x_{t3} + e_t, t = 1, 2, \dots, 20$$

$$y = f(\theta) + e$$

Função-Objetivo:

$$H(\theta) = [y - f(\theta)]'[y - f(\theta)]$$

# Métodos Numéricos

## Idéia Geral

$$\theta_{n+1} = \theta_n - t_n P_n \gamma_n$$

- $P_n$ : direção
- $t_n$ : "distância"
- $\gamma_n$ : gradiente de  $H$

## Condições de Parada

1.  $(\theta_{n+1} - \theta_n)'(\theta_{n+1} - \theta_n) < \epsilon$
2.  $H(\theta_n) - H(\theta_{n+1}) < \epsilon$
3.  $[\frac{\partial H}{\partial \theta} |_{\theta_n}]' [\frac{\partial H}{\partial \theta} |_{\theta_n}] < \epsilon$

# Newton-Rhapson

# Gauss-Newton

# Demonstração

# Conclusão



# Conclusões e Continuidade