

# **GERAÇÃO DE CÓDIGO PHP A PARTIR DA FERRAMENTA CASE RATIONAL ROSE**

Siegfried Kreuzfeld Neto

Orientador: Everaldo Artur Grahl

# Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
  - PHP
  - UML e WAE
  - Rational Rose e Add-in
- Implementação
- Conclusão

# Introdução

- Engenharia de software para web
- WAE (Web Application Extension)
- Rational Rose incorpora os estereótipos básicos da WAE
- Geração de código
- PHP

# Objetivos do trabalho

Desenvolver uma extensão para a ferramenta CASE Rational Rose, que permita através da modelagem UML de um sistema *web*, gerar código em nível estrutural na linguagem PHP.

# Objetivos secundários

- pesquisar a extensão WAE da UML, que permite a modelagem de sistemas *Web*;
- pesquisar a ferramenta Rational Rose, verificando a melhor forma de estender suas funcionalidades

# PHP

- Aplicações Web
- Classificação
  - Web magro
  - Web gordo
- Hypertext Preprocessor (Pré-processador de Hipertexto)

# PHP

- Script embutido entre o arquivo HTML
- Simples, portátil
- Código-fonte aberto

# UML

- “UML é uma linguagem destinada a visualizar, especificar, construir e documentar os artefatos de um sistema complexo de software” (BOOCH, 2000).
- extensibilidade e especialização



# UML para *Web*

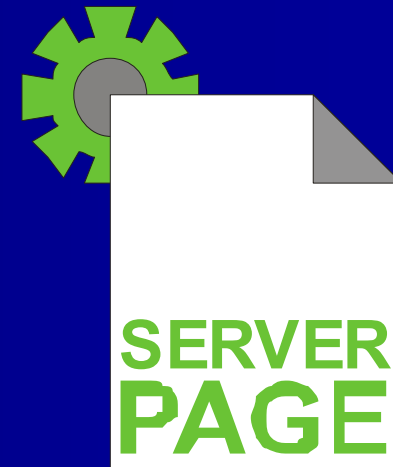
- *Web Application Extension (WAE)*
- Permite representar páginas *web* e outros elementos significativos do ponto de vista arquitetônico

# WAE – Estereótipos de associação

- Link
- Build
- Submit
- Redirect
- Forward
- Object
- Include

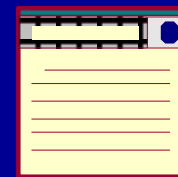
# WAE – Estereótipos de classe

- PÁGINA SERVIDOR
- É uma página *Web* dinâmica que contém o conteúdo executado no lado do servidor.



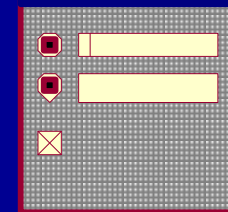
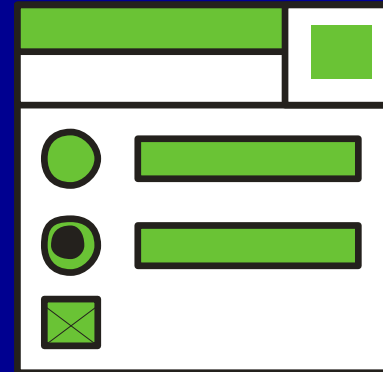
# WAE – Estereótipos de classe

- PÁGINA CLIENTE
- É uma página *Web* formatada em HTML com uma mistura de dados, apresentação e lógica.

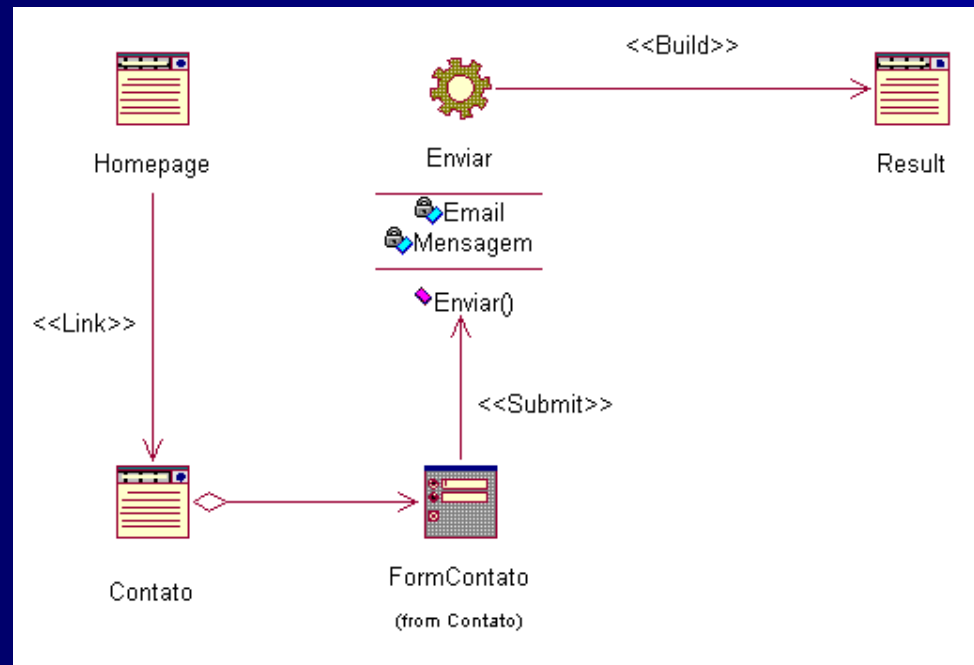


# WAE – Estereótipos de classe

- FORMULÁRIO
- É uma coleção de campos de entrada que são parte de uma página do cliente.



# Exemplo de modelagem



# A ferramenta Rational Rose

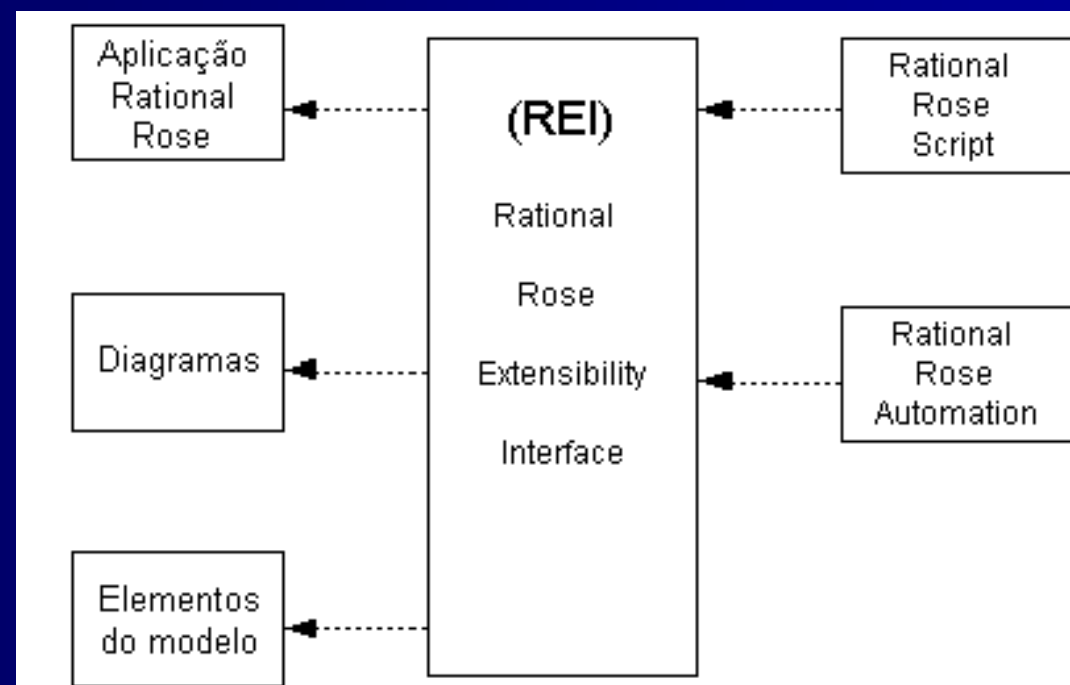
- Ferramenta de modelagem visual
- Auxilia o processo de desenvolvimento
- Web Modeler (ASP e JSP)

# Estendo o Rose

- customizar os menus;
- automatizar funções manuais do Rational Rose através de *scripts*;
- Rational Rose Automation;
- acessar as classes do Rational Rose, propriedades e métodos através de *scripts* ou da inclusão da Biblioteca de Extensão do Rational Rose;
- ativar *add-ins* utilizando o Gerenciador de *Add-in*.



# REI



# Add-in

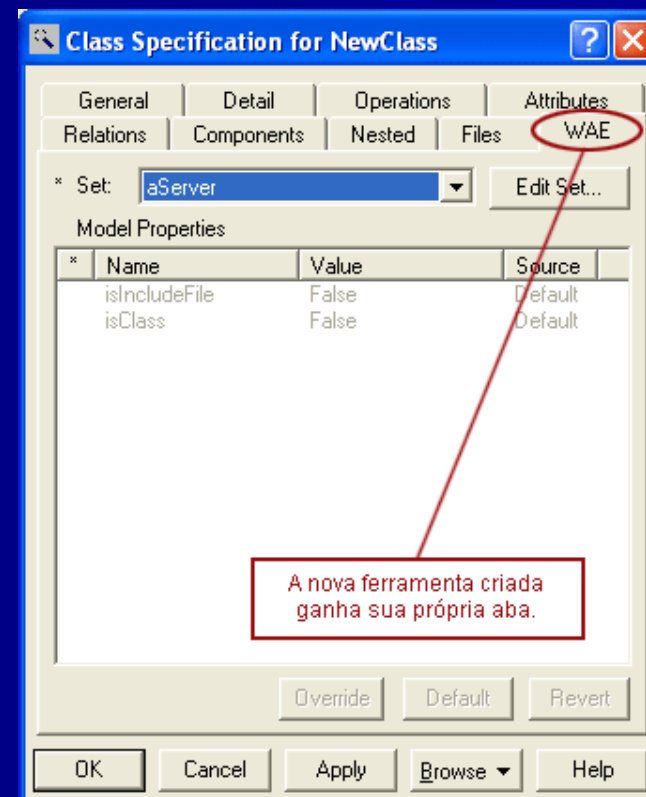
- ítems de menu;
- ítems de atalho;
- propriedades;
- especificação de tipos de dados;
- esteriótipos;
- arquivos de ajuda (help);
- tratamento de eventos;
- funções através de *scripts*.

# Add-in

- Permitem customizações e automatizações de várias funções do Rational Rose através da Interface de Extensão (REI) em um único pacote.

# Add-in

Aba de  
propriedades



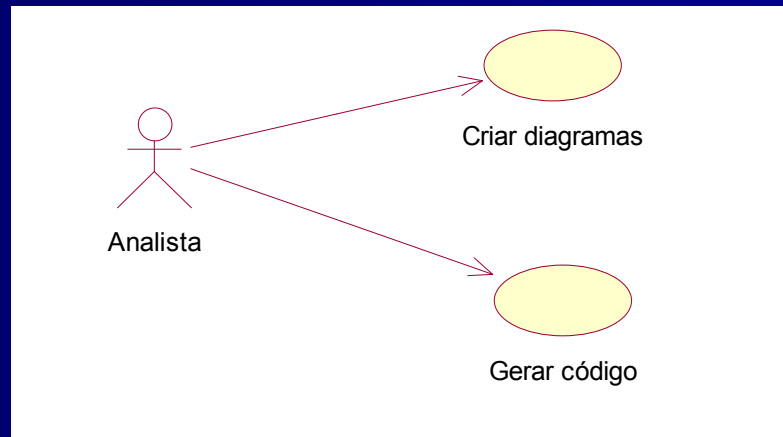
# Desenvolvimento do trabalho

- Ferramenta CASE Rational Rose Enterprise;
- Scripts;
- Arquivos de configuração.

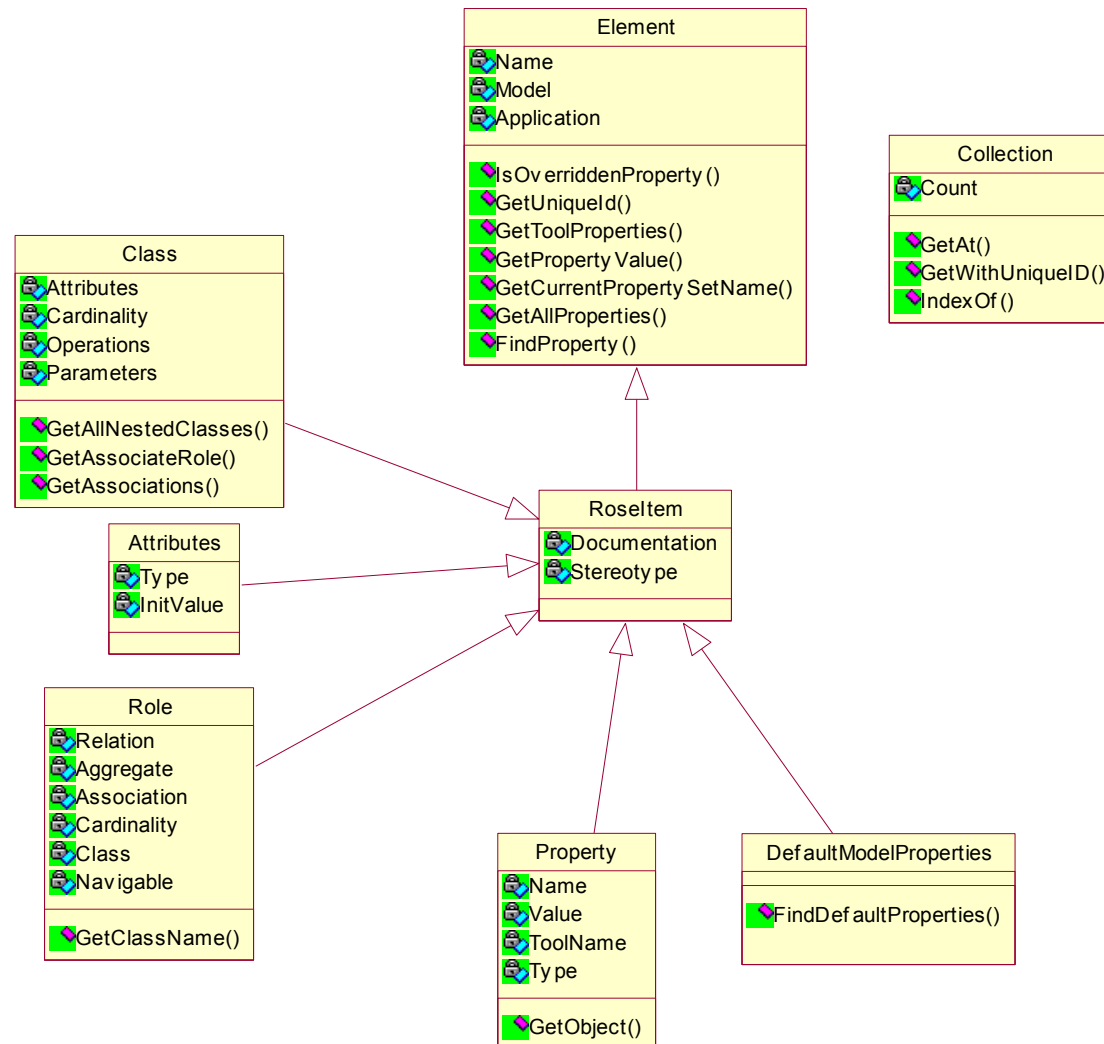
# Requisitos

- interagir com a ferramenta Rational Rose Enterprise através da interface fornecida por ela, utilizando a modelagem desenvolvida;
- permitir gerar código PHP a partir da UML;
- gerar código para a versão 4 do PHP;
- utilizar a notação WAE;

# Diagramas de caso de Uso



# Diagrama de Classes





```
PARA cada classe do modelo FAÇA
  CASO estereótipo da classe
    "página servidor" :
      criar cabeçalho
      SE existir propriedades ENTÃO
        criar código propriedades
      SE existir atributos ENTÃO
        criar código métodos
      //se for uma classe em PHP, deve gerar o construtor
      SE classe ENTÃO
        criar código construtor
      criar código do arquivo
      criar arquivo
    Fim
  "página cliente"
    criar cabeçalho HTML
    SE existir propriedades ENTÃO
      criar código propriedades
    SE existir links ENTÃO
      criar código link
    SE existir formulário ENTÃO
      criar código do formulário
    criar código do arquivo
    criar arquivo
  Fim
Fim CASO
FIM PARA
```

# Implementação

- Scripts desenvolvidos no Rational Rose;
- OLE server desenvolvido no Visual Basic;
- Ícones criados utilizando as ferramentas Adobe Photoshop e Corel Draw;

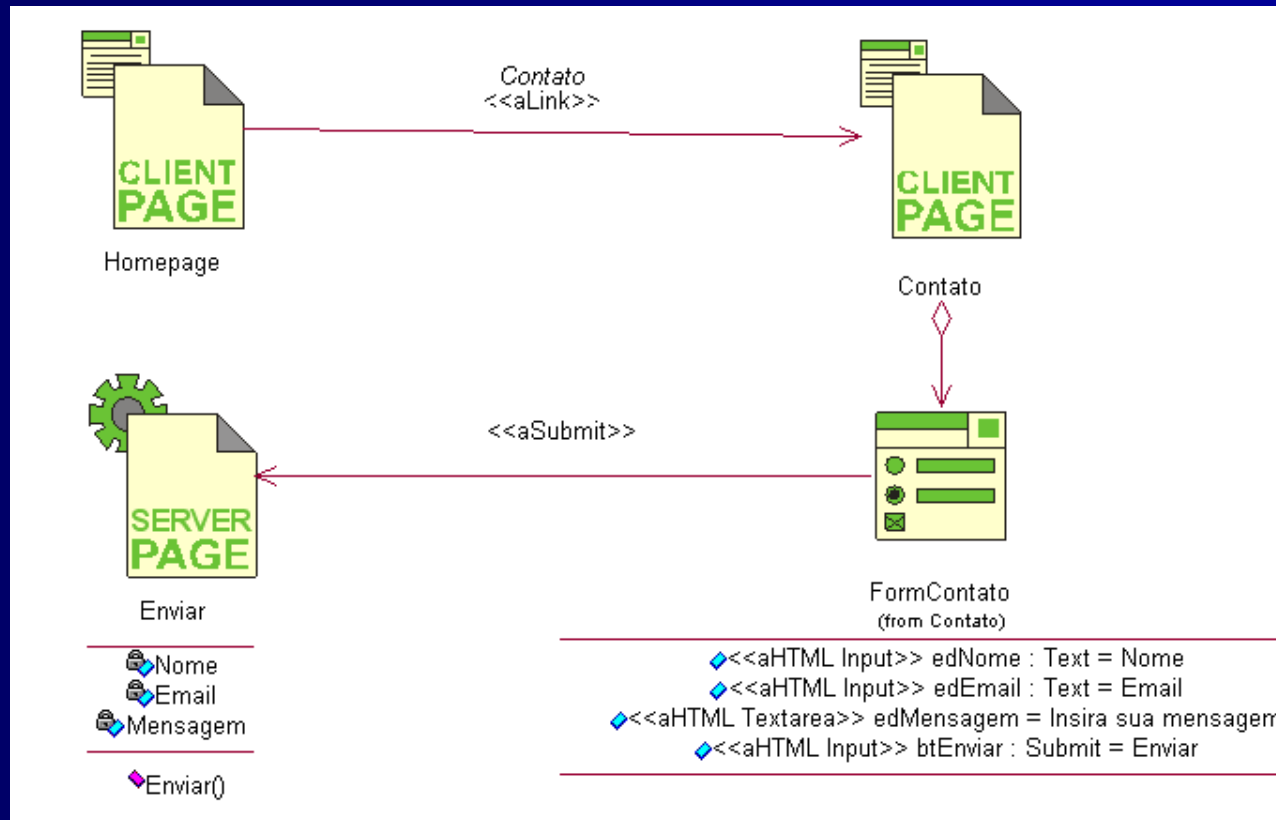
# Implementação

- Arquivo de propriedades:
  - Default;
  - aClientPage;
  - aServerPage;
  - aForm HTML.

# Implementação

- Arquivo de configuração (.ini);
- Servidor OLE trata os eventos de menu;
- Os scripts são responsáveis por:
  - Auxiliar a modelagem;
  - Gerar código;

# Implementação



# Implementação

```
<!-- Página inicial do website -->
<HTML>
  <BODY >
    <!-- Link para a classe: Contato -->
    <a href="contato.htm">Contato</a>
  </BODY>
</HTML>
<!--Página de contato do website -->
<HTML>
  <BODY >
    <FORM Method="post" Action="enviar.php" >
      <input type="text" Name="Nome" Value="Nome">
      <input type="text" Name="Email" Value="Email">
      <select name="Destino" > </select>
      <textarea name="Mensagem" >Mensagem</textarea>
      <input type="submit">
    </FORM>
  </BODY>
</HTML>
```

# Implementação

```
<?
$Nome;
$email;
$Opcao;
$mensagem;

function Enviar() {

}

?>
```

# Limitações

- classes de associação: <<*link*>>, <<submit>>, <<build>>.
- Visão lógica.



# Resultados

- Código gerado facilita a codificação final;
- Aplicações de pequena complexidade;
- Necessidade de excessivas alterações;
- Resultados semelhantes ao WebModeler.

# Conclusão

- Aproveitar ferramentas = vantagem;
- Especificação de aplicações web;
- Geração de código garante a especificação;
- WAE está em constante desenvolvimento;
- Objetivos atingidos.

# Sugestões

- Implementar a visão de componentes;
- Ampliar a quantidade de estereótipos suportados pelo add-in;
- Engenharia reversa.