

Componentes gráficos para prototipagem e documentação rápida em Delphi

Aluno: Reinoldo Krause Junior

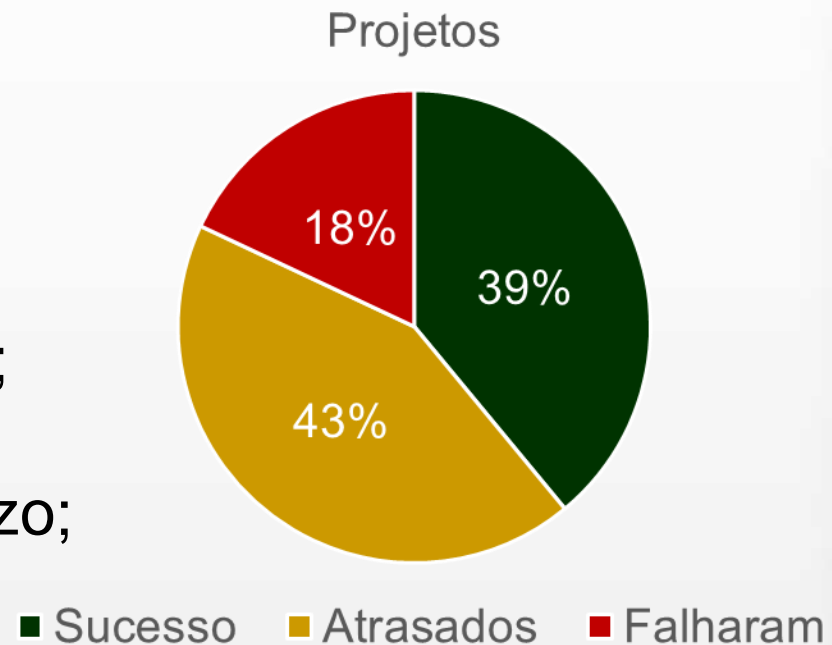
Orientadora: Luciana Pereira de Araújo

Roteiro

- Introdução
- Objetivos
- Fundamentação Teórica
- Trabalhos Correlatos
- Requisitos
- Especificação
- Implementação
- Operacionalidade da Implementação
- Resultados e Discussões
- Conclusões e Sugestões
- Demonstração

Introdução

- Dificuldade no envolvimento dos usuários;
- Necessidade de otimização;
- Necessidade de documentação;
- Necessidade de entrega no prazo;
- Prototipagem.



Objetivo Geral

Desenvolver componentes gráficos para auxiliar na prototipagem de telas e implementar a transformação das informações da prototipagem realizada em diagramas de casos de uso e de classes da UML, compatíveis às ferramentas de modelagem de software.

Objetivos Específicos

- Desenvolver componentes gráficos para a prototipagem;
- Implementar a geração dos diagramas de casos de uso da UML;
- Implementar a geração dos diagramas de classes da UML;
- Integrar os componentes gráficos desenvolvidos na ferramenta de Delphi 10 Seattle;
- Validar se os artefatos gerados pelos componentes gráficos desenvolvidos condizem com a prototipagem realizada.

Fundamentação Teórica

- Documentação
- Prototipagem
- UML
- XMI
- Componentes gráficos em Delphi

Documentação

- Satisfazer as necessidades dos usuários;
- Garantir a qualidade interna do software;
- Respeitar as restrições de custo e prazo do projeto.

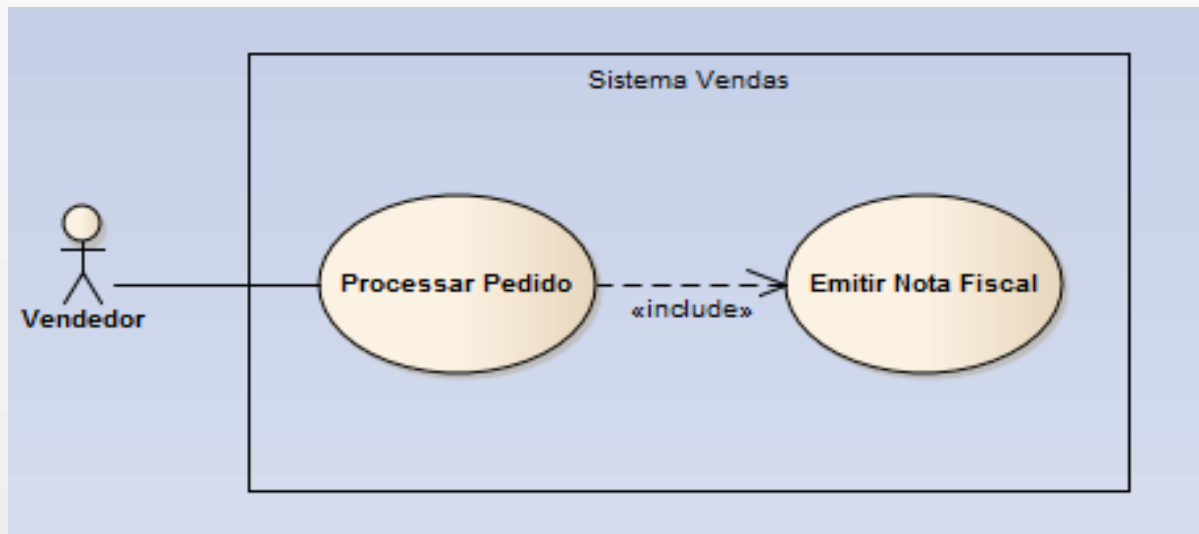
Prototipagem

- A chave é definir-se as regras logo no começo;
- Mecanismo a fim de definir os requisitos;
- Prototipagem baixa, média e alta fidelidade.

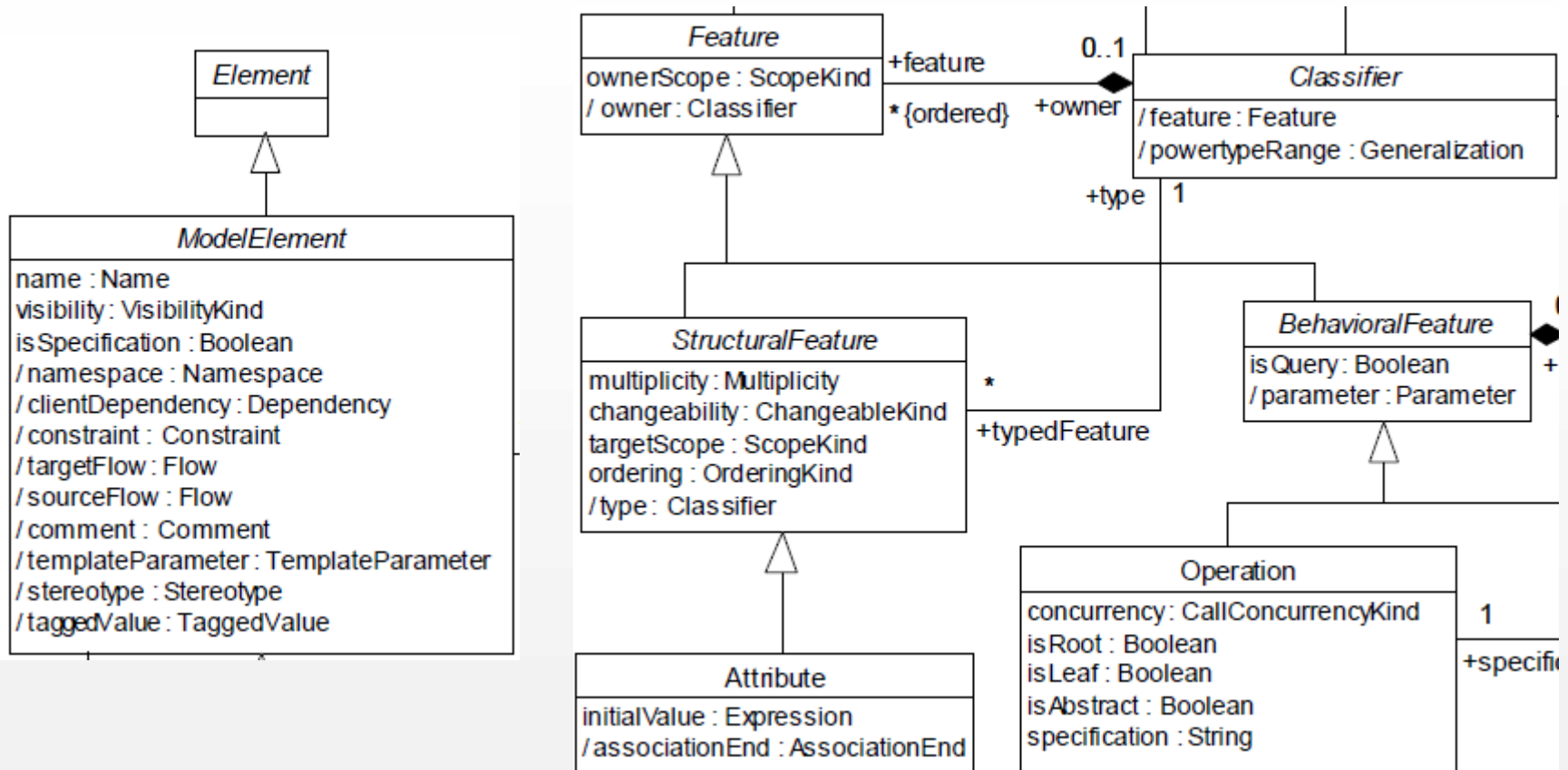
UML

- Diagrama de classes:
- Diagrama de casos de uso:

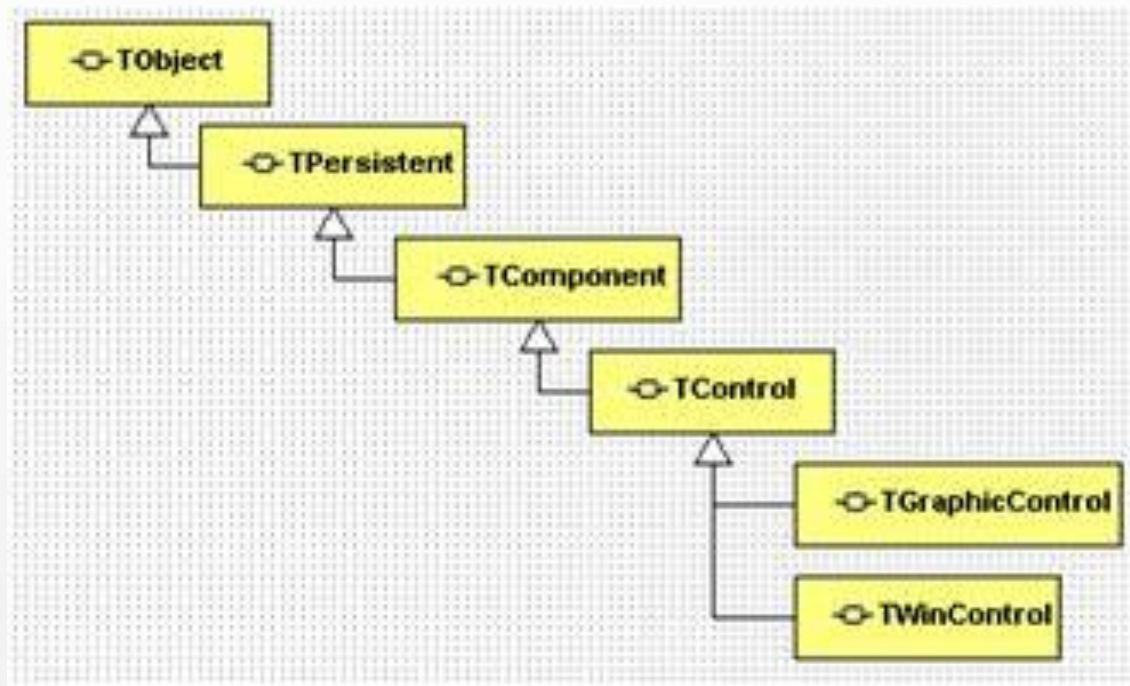
| |
|--|
| «utility» NomeClasse |
| -atributo1 : String -atributo2 : Integer -atributoN : Date |
| +operacao1() +operacao2() +operacaoN() |



XMI



Componentes gráficos em Delphi



Trabalhos Correlatos

- Atlântico
- Elicitar
- QEA
- ESS-Model

Atlântico

- Entrada: elicitación dos requisitos.
- Saída: prototipagem.
- Tecnologias: Microsoft Visio e Componentes visuais da web.

The image shows a screenshot of a web form titled "Formulário de Requisitos". The form is structured with several sections, each containing input fields and labels. The sections are labeled "NOME", "ENDEREÇO", "TELEFONE", "E-MAIL", "CATEGORIA", "VALOR", "DATA", "LOCAL", "OBSERVAÇÕES", and "COMENTÁRIOS". The "OBSERVAÇÕES" and "COMENTÁRIOS" sections are large text areas. The form is enclosed in a rectangular border and has a title bar at the top.

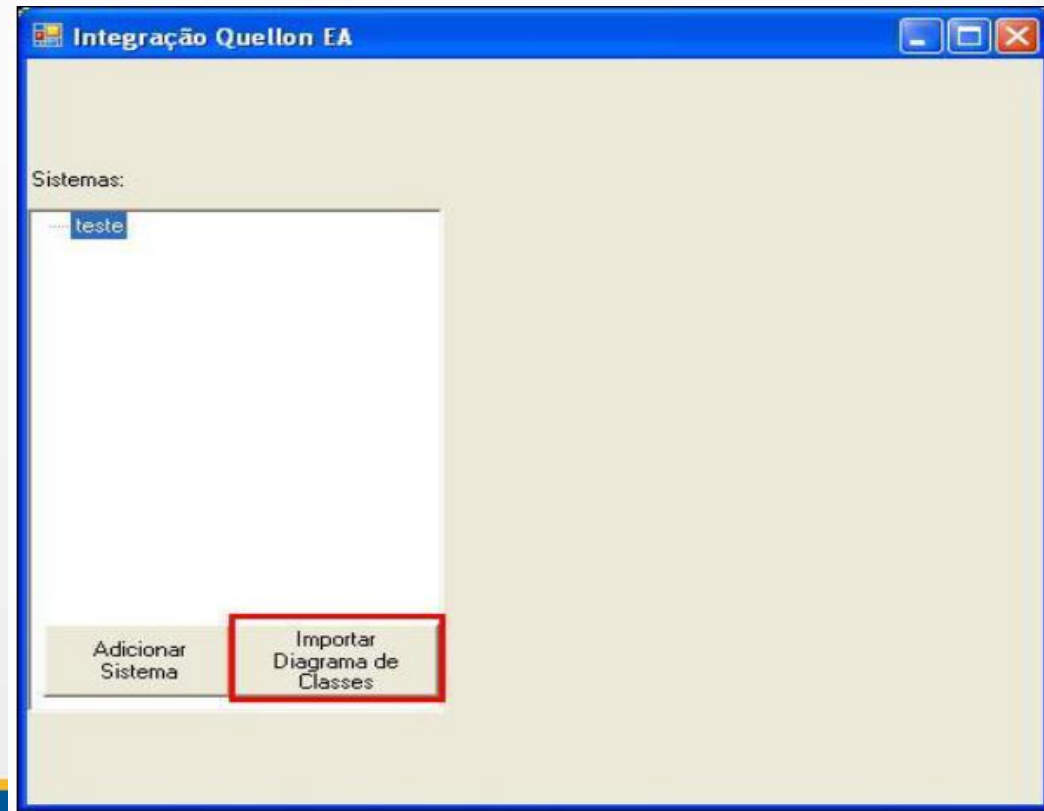
Elicitar

- Entrada: requisitos escritos em língua portuguesa.
- Saída: código para o protótipo da tela correspondente.
- Tecnologias: GoGrOO, Java, Primefaces, Eclipselink e Velocity.



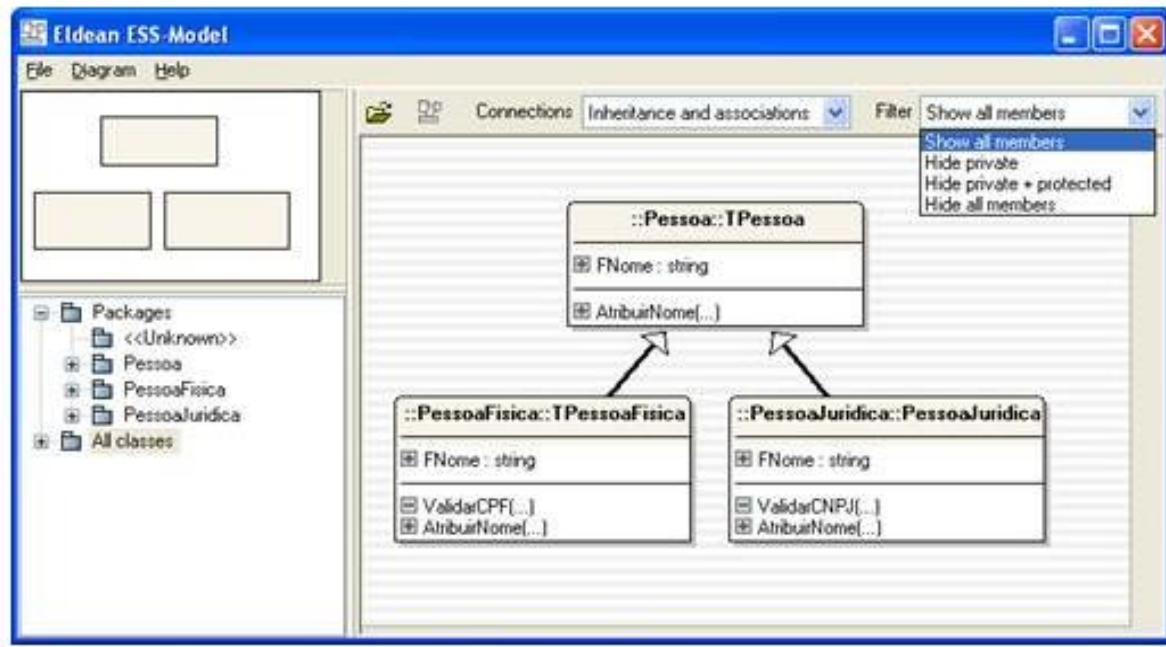
QEA

- Entrada: diagrama de classes
- Saída: estrutura para banco de dados
- Tecnologias:
EA, C#, ASP, .NET e XMI;



ESS-Model

- Entrada: código fonte Object Pascal e Java.
- Saída: diagramas de classe.
- Tecnologias:
Delphi e XMI.



Quadro comparativo dos trabalhos correlatos

| Características | Atlântico | Elicitar | QEA | ESS-Model |
|--|-----------|----------|-----|-----------|
| utiliza a geração de prototipagem de telas | Sim | Sim | Não | Não |
| utiliza o padrão XMI para exportação/leitura de diagramas da UML | Não | Não | Sim | Sim |
| ferramenta é dependente de aplicações comerciais | Sim | Não | Sim | Não |
| busca otimizar o processo de desenvolvimento de softwares | Sim | Sim | Sim | Sim |

Requisitos Funcionais

RF01 - RF02: o sistema deverá permitir o uso de componentes gráficos pré-definidos para auxiliar na criação da prototipagem e extração das informações para geração da documentação. Os componentes gráficos são:

- área de texto, botão, botão de seleção, caixa de combinação, caixa de seleção, caixa de texto, grade, imagem, rótulo, estereótipo e ator;

Requisitos Funcionais

RF03 – RF04: o sistema deverá permitir a exportação das informações da prototipagem das telas em padrão XMI com configurações que resultem na diagramação de classes e casos de uso para importar em ferramentas de modelagem de software;

Requisitos Não Funcionais

RNF01: o sistema deverá ser integrado com a ferramenta RAD Studio Delphi 10 Seattle em forma de *plugin*;

RNF02: o sistema deverá utilizar a linguagem de programação Object Pascal

Diagrama de casos de uso da aplicação

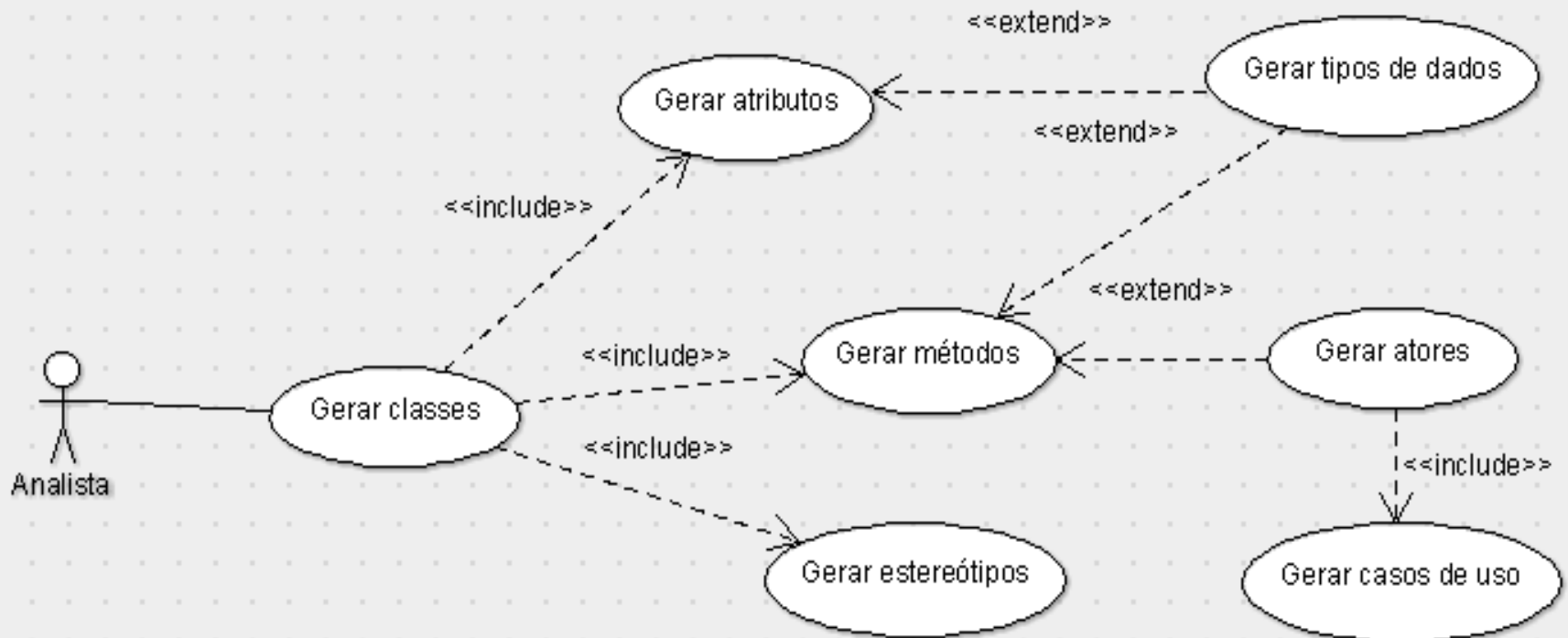
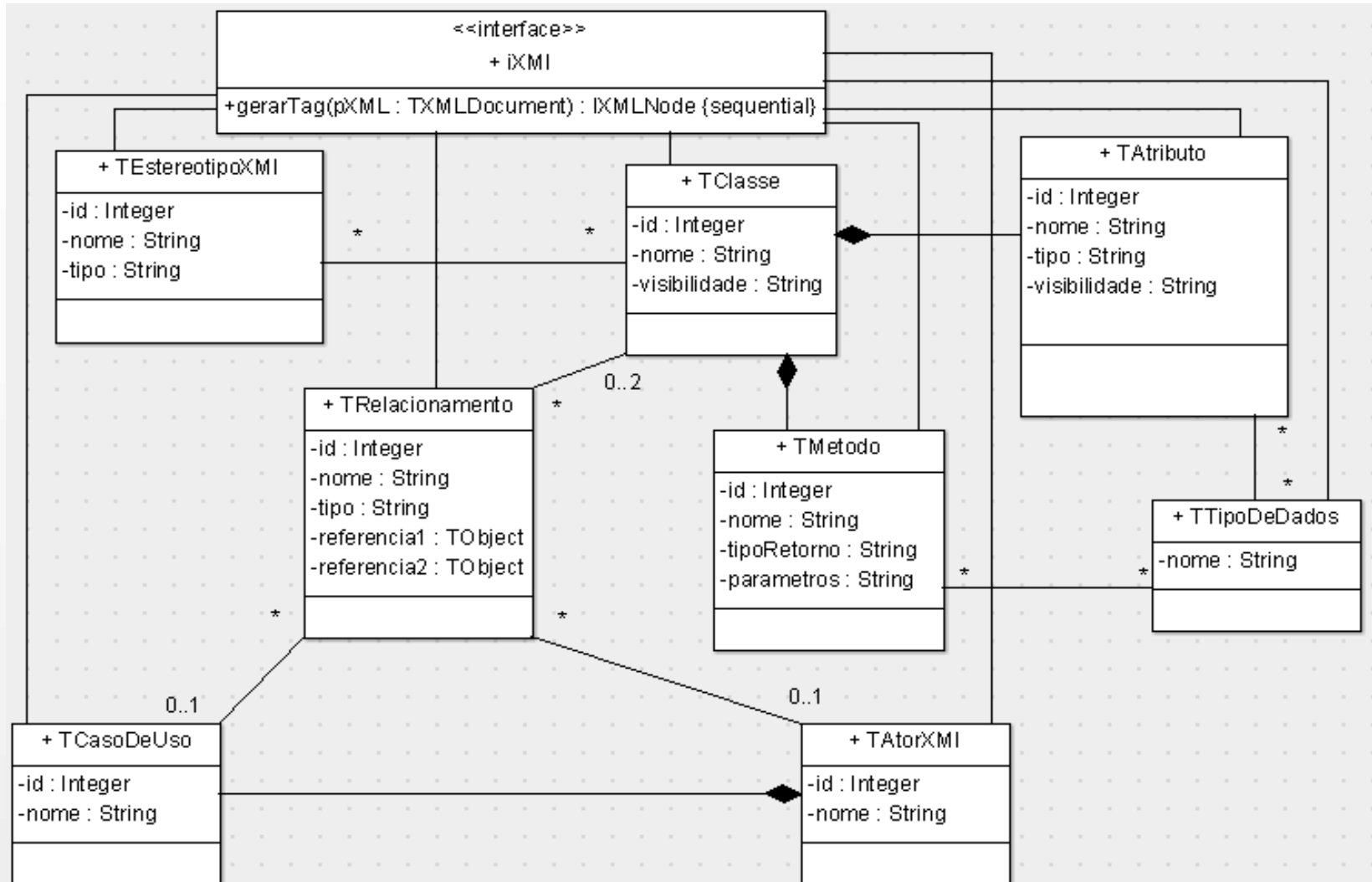


Diagrama de classes da geração do XMI



Quadro dos componentes gráficos

| Nome | Significado | Herança |
|-------------------------|---|--------------|
| TAreaTexto | Permite a entrada de textos com várias linhas. | TMemo |
| TBotao | Permite a execução de uma ação. | TButton |
| TBotaoSelecao | Representa uma opção a ser marcada. Quando utilizado em conjunto com outros botões de seleção só pode ser marcado um deles. | TRadioButton |
| TCaixaCombinacao | Representa uma opção que pode ser marcada ou não. | TCheckBox |
| TCaixaSelecao | Permite a seleção de uma única opção dentre uma lista de opções. | TComboBox |
| TCaixaTexto | Permite a entrada de um texto em uma linha | TEdit |
| TGrade | É uma tabela formada por linhas e colunas, permitindo a entrada de vários dados. | TStringGrid |
| TImagem | Representa uma imagem. | TImage |
| TRotulo | Descrição de alguma informação. Componente visual para o usuário. | TLabel |
| TAtor | Define um ator para o caso de uso. | TComponent |
| TEstereotipo | Define um estereótipo para a classe. | TComponent |

Implementação

- Object Pascal
- Delphi 10 Seattle
- XMI 1.2
- UML 1.4
- ArgoUML 0.34
- Enterprise Architect 12

Classe TDocumentacao

```
TDocumentacao = class(TPersistent)
private
  { Protected declarations }
  fVisibilidade: TVisibilidade;
  fTipo: String;
  fClasse: Boolean;
  fAtributo: Boolean;
  function TipoIsStored:Boolean;
protected
  { Protected declarations }
public
  { Public declarations }
  constructor create(AOwner: TPersistent);
  property Tipo: String read fTipo write fTipo stored TipoIsStored;
  property Classe: Boolean read fClasse write fClasse;
  property Atributo: Boolean read fAtributo write fAtributo;
published
  { Published declarations }
  property Visibilidade: TVisibilidade read fVisibilidade write fVisibilidade;

end;
```

Uso da Classe TDocumentacao

```
TDocOwner = class(TDocumentacao)
published
  property Tipo;
  property Atributo;
end;

TCaixaTexto = class(TEdit)
private
  fDoc : TDocOwner;
protected
  { Protected declarations }
public
  { Public declarations }
  constructor Create (AOwner: TComponent); override;
  destructor Destroy; override;

published
  { Published declarations }
  property Documentacao : TDocOwner read fDoc write fDoc ;
end;
```

Criação das tags

```
// Criando outro nodulo na RAIZ (um node filho de RAIZ)
cabecalho := XMLDocument.CreateNode('XMI.header', ntElement);
Raiz.ChildNodes.Add(cabecalho);

documentacao := XMLDocument.CreateNode('XMI.documentation', ntElement);
cabecalho.ChildNodes.Add(documentacao);

nodoElemento := XMLDocument.CreateNode('XMI.exporter', ntElement);
nodoElemento.Text := 'TCC Reinoldo';
documentacao.ChildNodes.Add(nodoElemento);

nodoElemento := XMLDocument.CreateNode('XMI.metamodel', ntElement);
cabecalho.ChildNodes.Add(nodoElemento);

nodoAtributo := XMLDocument.CreateNode('xmi.version', ntAttribute);
nodoAtributo.Text := '1.4';
nodoElemento.AttributeNodes.Add(nodoAtributo);

nodoAtributo := XMLDocument.CreateNode('xmi.name', ntAttribute);
nodoAtributo.Text := 'UML';
nodoElemento.AttributeNodes.Add(nodoAtributo);
```

Tags no XMI

```
<XMI.header>  
  <XMI.documentation>  
    <XMI.exporter>TCC Reinoldo</XMI.exporter>  
  </XMI.documentation>  
  <XMI.metamodel xmi.version="1.4" xmi.name="UML"/>  
</XMI.header>
```

Operacionalidade da Implementação

The screenshot displays the Delphi IDE interface for a project named 'Testes - Delphi 10 Seattle - Unit1'. The central workspace shows a form with several input fields and labels: 'Pedido', 'Cliente', 'Previsão', 'Situação', 'Total do Pedido', 'Status', and 'Valor Total'. Below these is a table with columns for 'Produto', 'Quantidade', and 'Valor unitário'. The 'Help' menu is open, with 'Documentação' highlighted in blue. The 'Tool Palette' on the right is also highlighted, showing a list of components including 'Prototipagem', 'TCaixaTexto', 'TRotulo', 'TCaixaSelecao', 'TBotao', 'TAreaTexto', 'TImagem', 'TCaixaCombinacao', 'TBotaoSelecao', 'TAtor', 'TEstereotipo', and 'TGrade'. The 'Object Inspector' on the left shows the properties of the selected 'Form1' component.

Structure

- Form1
 - Ator1
 - Estereotipo1
 - Estereotipo2
 - Panel1
 - Panel2

Object Inspector

Form1 TForm1

Properties Events

Search

Action

- ActiveControl
- BiDiMode: bdLeftToRight
- Border: (TFormBorder)
- BorderIcons: [biSystemMenu, biMini]
- BorderStyle: Sizeable
- Caption: Form1
- ClientHeight: 467

Help

- Delphi Help
- RAD Studio Docwiki
- Third-Party Help
- Platforms SDK Help
- Help Wizards
- Embarcadero Home Page
- Embarcadero Developer Support Page
- Embarcadero Community Site
- Delphi Home Page
- License Manager...
- About Embarcadero® Delphi
- Documentação

Testes.dproj - Project Manager

File

- ProtoDoc
- Prototipagem.bpl
- Documentacao.bpl

D:\Development\GitHub\Testes\Unit1.pas

Testes.d... Model ... Data Ex... Multi-D...

Tool Palette

Search

Prototipagem

- TCaixaTexto
- TRotulo
- TCaixaSelecao
- TBotao
- TAreaTexto
- TImagem
- TCaixaCombinacao
- TBotaoSelecao
- TAtor
- TEstereotipo
- TGrade

Operacionalidade da Implementação

RX Gerador de documentação

Geração da prototipagem das telas para XMI

Caminho das telas

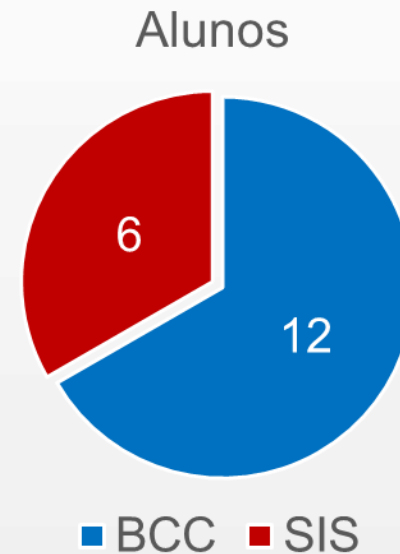
Local para salvar arquivo XMI

C:\Documentacao.xmi

Gerar

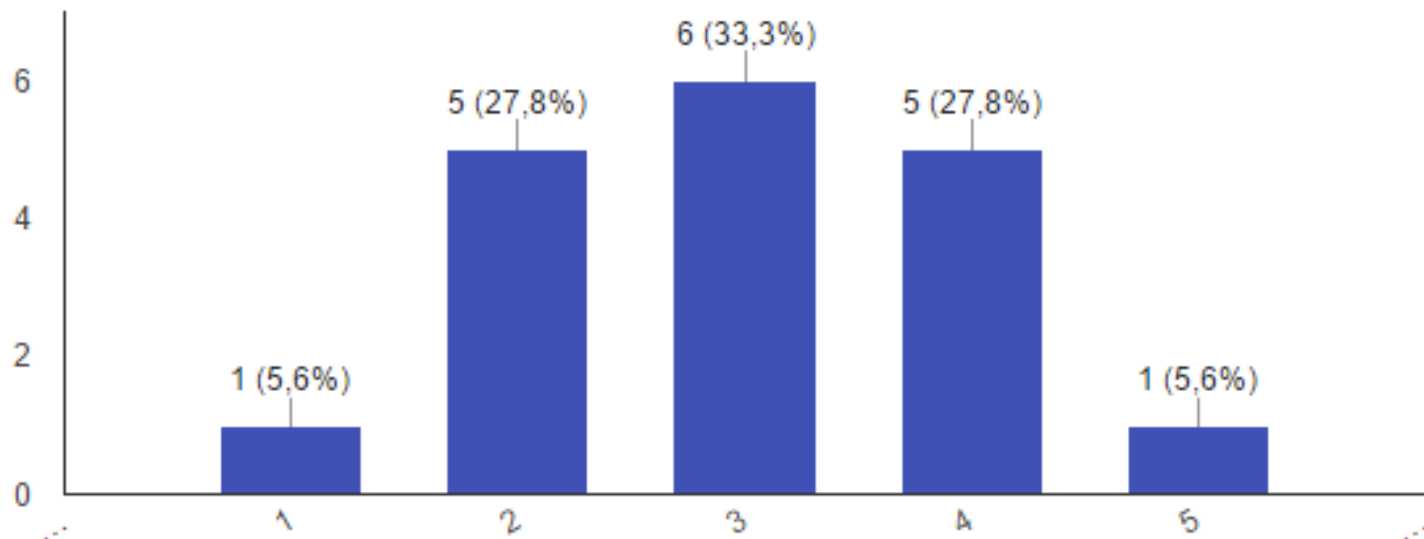
Resultados e Discussões

- Pesquisa de campo baseada nos métodos SUS e SUMI;
- Termo de consentimento livre e esclarecido;
- Tutorial;
- Validar os diagramas;
- Testar a usabilidade;
- 18 alunos da FURB.



Resultados e Discussões

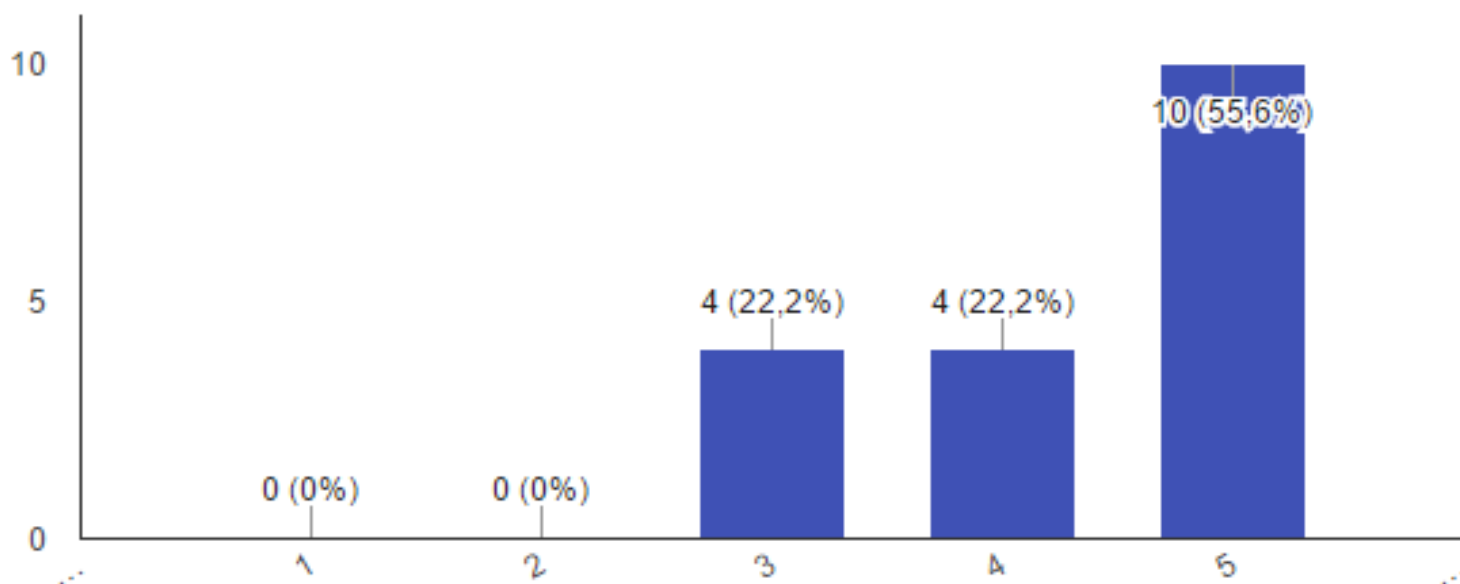
1. Eu acho que gostaria de usar esse sistema com frequência. (18 respostas)



Resultados e Discussões

11. Eu achei que o digrama de classes ficou compatível com minha prototipagem.

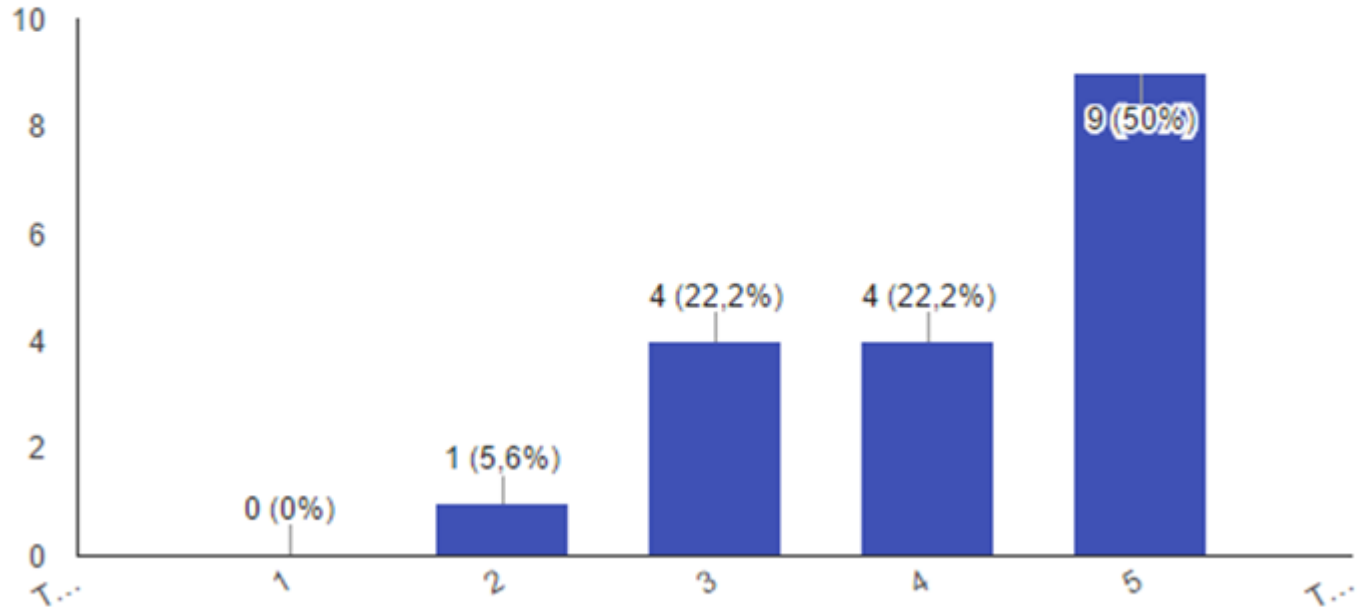
(18 respostas)



Resultados e Discussões

12. Eu achei que o digrama de casos ficou compatível com minha prototipagem.

(18 respostas)



Resultados e Discussões

- 1/3 dos usuários usariam a ferramenta frequentemente;
- 78% dos diagramas de casos de uso foram compatíveis com a prototipagem;
- 77% dos diagramas de classes foram compatíveis com a prototipagem;

Resultados e Discussões

| Características | Atlântico | Elicitar | QEA | ESS-Model | Trabalho desenvolvido |
|--|-----------|----------|-----|-----------|-----------------------|
| utiliza a geração de prototipagem de telas | Sim | Sim | Não | Não | Sim |
| utiliza o padrão XMI para exportação/leitura de diagramas da UML | Não | Não | Sim | Sim | Sim |
| ferramenta é dependente de aplicações comerciais | Sim | Não | Sim | Não | Sim |
| busca otimizar o processo de desenvolvimento de softwares | Sim | Sim | Sim | Sim | Sim |

Conclusões

- Todos os objetivos foram atingidos;
- A ferramenta apresenta algumas limitações para os diagramas de classes e caso de uso;
- É uma nova alternativa para documentar sistemas de forma rápida.

Sugestões

- Implementar os tipos de relacionamentos para o diagrama de classes e diagramas de casos de uso;
- Implementar a exportação para demais diagramas da UML;
- Implementar a gravação de tipos de dados digitados pelo usuário;
- Implementar uma propriedade que indique se o componente será um método.

Demonstração