

UNIVERSIDADE REGIONAL DE BLUMENAU  
CENTRO DE CIÊNCIAS EXATAS E NATURAIS  
DEPARTAMENTO DE CIÊNCIAS DA COMPUTAÇÃO  
ALUNO: CLAUDIO SCHVEPE  
ORIENTADORA: JOYCE MARTINS

GERADOR DE CÓDIGO JAVA A PARTIR DE ARQUIVOS  
DO ORACLE FORMS 6i

# Converte Forms<sup>6i</sup>



[www.furb.br](http://www.furb.br)



# Roteiro

 Introdução

 Objetivos

 Contextualização

 Desenvolvimento da ferramenta

 Conclusões

 Extensões



## Introdução

- 🦋 Migração sistemas legados
- 🦋 Reengenharia de software
- 🦋 Tradução de código
- 🦋 Oracle Forms
- 🦋 Java



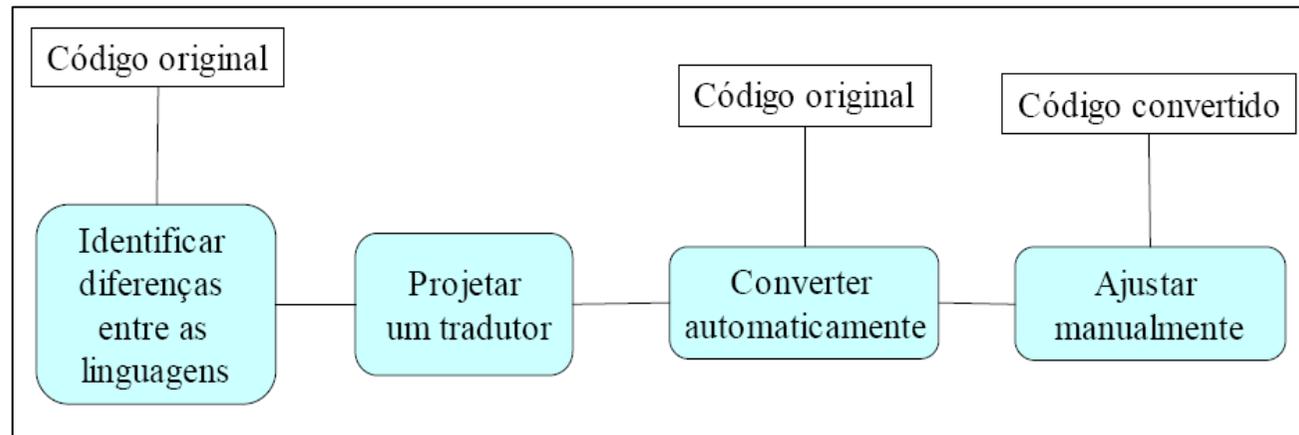
## Objetivos

- ✿ Migrar aplicações desenvolvidas em Oracle Forms 6i para Java;
- ✿ Traduzir um subconjunto de código PL/SQL, incluindo apenas as construções procedurais;
- ✿ Permitir a conversão de alguns objetos visuais (Canvas, CheckBox, Frame, Image, Line, ListItem, PushButton, RadioGroup, Tab, Text, TextItem, Window);
- ✿ Mapear eventos (gatilhos ou *triggers*) de controle de objetos visuais para eventos Java;
- ✿ Utilizar *templates* para configurar o formato do código de saída.



# Contextualização

🌿 Tradução de código fonte





# Contextualização

## Geradores de código

-  Geradores de código são programas que geram programas.
  
-  Vantagens:
  - ✓ Qualidade
  - ✓ Produtividade
  - ✓ Abstração
  
-  Etapas de desenvolvimento:
  - ✓ Escrever o código de saída manualmente;
  - ✓ Projetar o gerador;
  - ✓ Analisar a entrada;
  - ✓ Definir os *templates*;
  - ✓ Gerar a saída usando *templates*.



## Contextualização

 Código de saída em Java

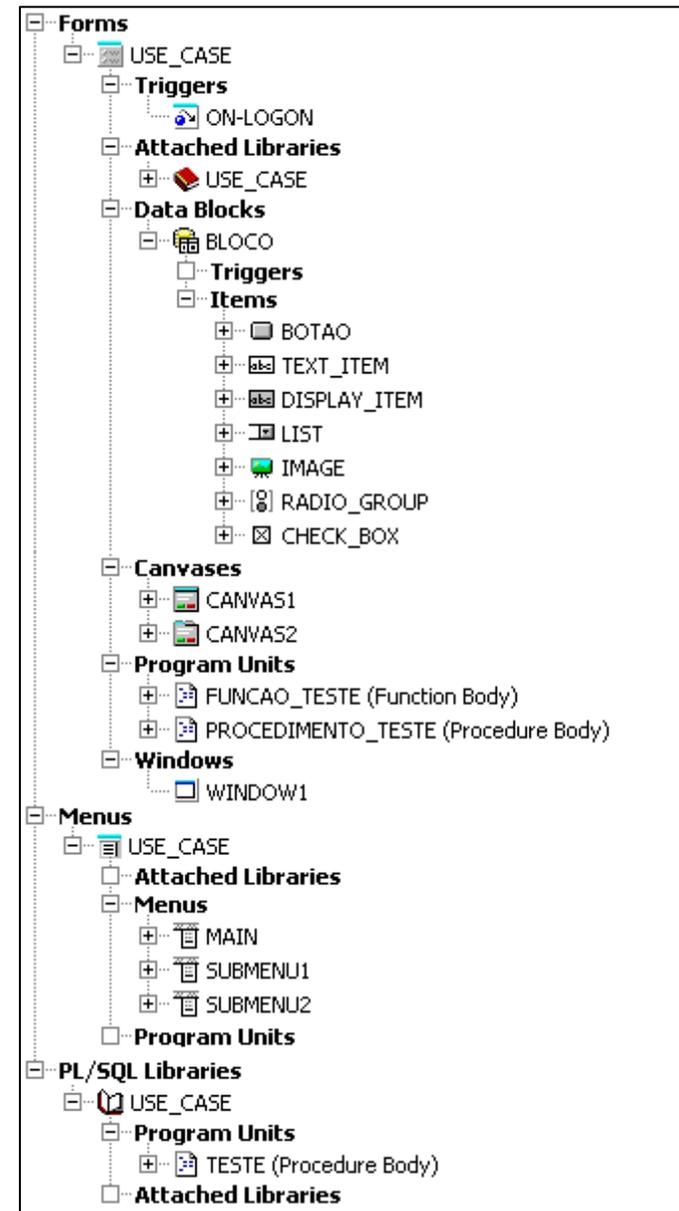
 Bibliotecas gráficas Swing e AWT.

 Mecanismo de tratamento de eventos em Java.



# Contextualização

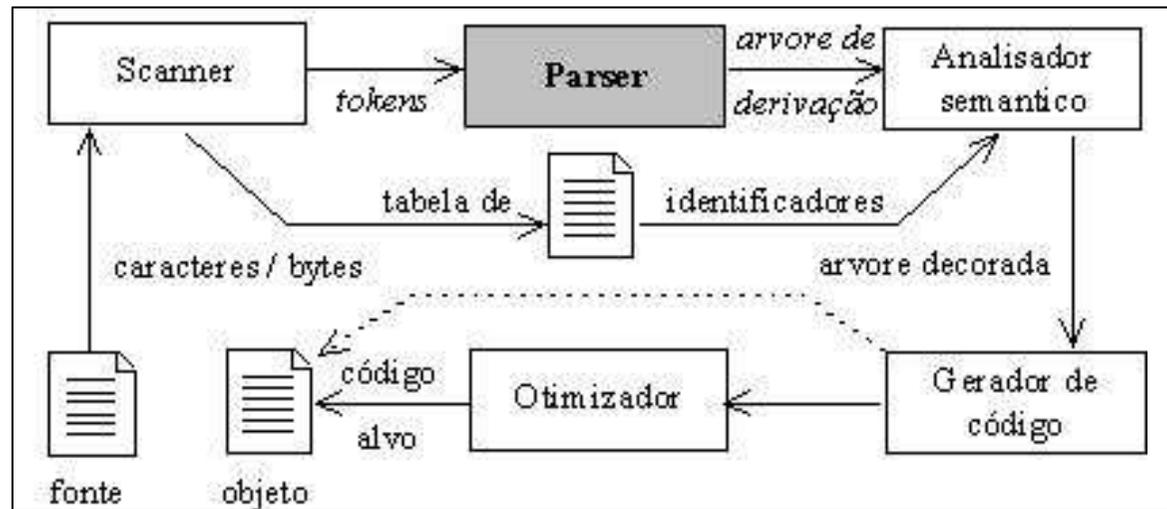
 Código de entrada Oracle Forms





# Contextualização

## 🌿 Análise de entrada





## Contextualização

### Motor de *templates*

-  Mecanismo que permitem desenvolver geradores de código independentes do código alvo.
-  Código estático.
-  Código dinâmico.
-  Linguagem de *templates*.



# Contextualização

## Motor de *templates* eNITL

```

<#·MainModule·#>
<#·setoutput·file("MD"$env.module.name·$·".java")·#>
package·view;
import·java·awt·event.*;
<#if·env.module.init_mnu·!="#>
import·view.MB<#env.upper(env.module.init_mnu)#>;
<#endif#>

public·class·MD<#env.module.name#>·extends·javax.swing.JFrame
{
    //Construtor·default
    public·MD<#env.module.name#>()
    {
        theDesktop·=·new·JDesktopPane();
        <#if·env.module.init_mnu·!="#>
        MB<#env.upper(env.module.init_mnu)#>·bar·=·new·MB<#env.upper(env.module.init_mnu)#>();
        setJMenuBar(bar);
        <#endif#>
        getContentPane().add(theDesktop);

        <#·····win:=·env.windows#>
        <#·····while·win·!=·0·,·win:=·env.object(win).next#>
        <#·······tpl.CreateWin(win)#>
        <#·····endwhile#>
        <#·····tpl.CreateBlocks#>

        }

    public·static·void·main(String[]·args)
    {
        MD<#env.module.name#>·application·=·new·MD<#env.module.name#>();
        application.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }
}
<#·endsetoutput·#>
<#·tpl.CreateEvents("MD"$env.module.name)#>

<#endtpl#>

```



## Contextualização

🌈 Reestruturação de sistemas procedurais em orientados a objetos

- 🌈 Identificação das construções sintáticas.
- 🌈 Elaboração das regras de mapeamento.

CÓDIGO PROCEDURAL	CÓDIGO ORIENTADO A OBJETOS
arquivo ou tabela de banco de dados e seus campos	classe de persistência com respectivos atributos
estrutura de dados e seus campos	classe com respectivos atributos
interface de interação e seus componentes	classe de interface com respectivos atributos, um para cada componente
unidade de programa que faz referência a um arquivo ou a uma tabela de banco de dados	método da classe correspondente ao arquivo ou à tabela de banco de dados
unidade de programa associada a um componente de interface	método da classe de interface correspondente

- 🌈 Geração automática do código orientado a objeto.



# Especificação

## Enterprise Architect

-  Conceitos de orientação a objetos
-  Diagrama de casos de uso
-  Diagrama de classes
-  Diagrama de seqüência



## Desenvolvimento da ferramenta

- 🌸 Elicitação dos requisitos
- 🌸 Análise das linguagens PL/SQL e Java
- 🌸 Especificação da saída
- 🌸 Análise da entrada
- 🌸 Especificação
- 🌸 Implementação
- 🌸 Elaboração de *templates*
- 🌸 Testes



## Desenvolvimento da ferramenta

- 🌸 Elicitação dos requisitos: requisitos funcionais
  - 🔧 Processar arquivos fontes do Oracle Forms (Form, Menu, PL/SQL Libray);
  - 🔧 Analisar léxica e sintaticamente os arquivos de entrada;
  - 🔧 Produzir, para cada arquivo de entrada, a tradução especificada nos *templates* correspondentes;
  - 🔧 Traduzir um subconjunto de código PL/SQL (procedimentos, funções e blocos, com atribuições, testes condicionais, declarações de variáveis, laços de repetição, blocos de exceções e expressões);
  - 🔧 Traduzir objetos visuais (Canvas, CheckBox, Frame, Image, Line, ListItem, PushButton, RadioGroup, Tab, Text, TextItem e Window), implementando suas principais funcionalidades;
  - 🔧 Mapear gatilhos de controle de objetos visuais para eventos.



## Desenvolvimento da ferramenta

- 🌸 Elicitação dos requisitos: requisitos não funcionais
  - ✚ Implementar a ferramenta utilizando o ambiente de desenvolvimento Microsoft Visual C++ 6.0;
  - ✚ Implementar a ferramenta utilizando a linguagem de programação C++;
  - ✚ Utilizar a ferramenta Flex para gerar o analisador léxico;
  - ✚ Utilizar a ferramenta Bison para gerar o analisador sintático;
  - ✚ Utilizar o motor de *templates* eNITL;
  - ✚ Utilizar a biblioteca Forms API para manipulação de arquivos do Oracle Forms 6i;
  - ✚ Utilizar a biblioteca wxWidgets-2.6.2 na implementação da interface da ferramenta.



# Desenvolvimento da ferramenta

## Mapeamento Oracle Forms – Java: objetos visuais

OBJETO VISUAL ORACLE FORMS	COMPONENTE EM SWING
Canvas	JLayeredPane
CheckBox	JCheckBox
Frame	Graphics2D
Graphics	Graphics2D
Image	ImageIcon
Line	Graphics2D
ListItem	JComboBox, JList
Menu	JMenu
PushButton	JButton
RadioButton	JButton
RadioGroup	ButtonGroup
Tab	JTabbedPane
Text	JTextArea
TextItem	JTextArea, JPasswordField, JTextField
Window	JInternalFrame





# Desenvolvimento da ferramenta

## 🌸 Mapeamento Oracle Forms – Java: gatilhos

<b>GATILHO (TRIGGER)</b>	<b>LISTENER</b>	<b>MÉTODO</b>
KEY-F1 a KEY-F12 KEY-OTHERS	KeyListener	keyPressed
WHEN-BUTTON-PRESSED	ActionListener	actionPerformed
WHEN-CHECKBOX-CHANGED WHEN-LIST-CHANGED	ItemListener	itemStateChanged
WHEN-MOUSE-CLICK WHEN-MOUSE-DOUBLECLICK	MouseListener	mouseClicked
WHEN-MOUSE-DOWN	MouseListener	mousePressed
WHEN-MOUSE-ENTER	MouseListener	mouseEntered
WHEN-MOUSE-LEAVE	MouseListener	mouseExited
WHEN-MOUSE-MOVE	MouseListener	mouseMoved
WHEN-MOUSE-UP	MouseListener	mouseReleased
WHEN-RADIO-CHANGED	ItemListener	itemStateChanged
WHEN-WINDOW-ACTIVATED	InternalFrameListener	internalFrameActivated
WHEN-WINDOW-CLOSED	InternalFrameListener	internalFrameClosed
WHEN-WINDOW-DEACTIVATED	InternalFrameListener	internalFrameDeactivated
WHEN-WINDOW-RESIZED	InternalFrameListener	InternalFrameListener



# Desenvolvimento da ferramenta

## Mapeamento Oracle Forms – Java: gatilhos

```
//-- método responsável pelos eventos disparados na tela
private void criarEventos() {
    java.awt.event.ItemListener cancelarItemListener=
        new java.awt.event.ItemListener(){
        public void itemStateChanged (java.awt.event.ItemEvent event ) {
            blbloco.cancelarItemSelectable(event);
        }
    };
    screen.getblbloco().getcancelar_1().addItemListener(cancelarItemListener);
    screen.getblbloco().getcancelar_2().addItemListener(cancelarItemListener);
    screen.getblbloco().getcancelar_3().addItemListener(cancelarItemListener);
    screen.getblbloco().getcancelar_4().addItemListener(cancelarItemListener);
    screen.getblbloco().getcancelar_5().addItemListener(cancelarItemListener);
}

public class BLBLOCO {
    public BLBLOCO () {
    }
    private void cancelarItemSelectable(java.awt.event.ItemEvent event) {
    }
}
```



## Desenvolvimento da ferramenta

### 🌸 Mapeamento Oracle Forms – Java: código PL/SQL

ESTRUTURA SINTÁTICA EM ORACLE FORMS	ESTRUTURA SINTÁTICA EM JAVA
<pre>If &lt;expressão&gt; then   ...   [ Elsif &lt;expressão&gt; then     ... ]*   [ Else     ... ]? End if</pre>	<pre>if (&lt;expressão&gt;) {   ... } [ else if (&lt;expressão&gt;) {   ... } ]* [ else {   ... } ]?</pre>
<pre>Loop   ... End loop</pre>	<pre>while (true) {   ... }</pre>
<pre>Exit when &lt;expressão&gt; ;</pre>	<pre>if (&lt;expressão&gt;) break ;</pre>
<pre>identificador := &lt;expressão&gt; ;</pre>	<pre>identificador = &lt;expressão&gt; ;</pre>
<pre>For I in 1..10 loop   ... End loop</pre>	<pre>for (long i=1; i&lt;=10; i++) {   ... }</pre>
<pre>While (&lt;expressão&gt;) loop   ... End loop</pre>	<pre>while (&lt;expressão&gt;) {   ... }</pre>



# Desenvolvimento da ferramenta

## 🌸 Mapeamento Oracle Forms – Java: código PL/SQL

CÓDIGO FONTE EM ORACLE FORMS	CÓDIGO FONTE EM JAVA
<pre> PROCEDURE TESTE IS   str  VARCHAR2(40);   nreal NUMBER(5,2);   nint  INTEGER;   bx    BOOLEAN; BEGIN   str:= 1;   str:= str + 3;   nreal:= nint * 5 + str;   str:= str + 1    str;   nreal:= nint + 5 + 1;   nreal:= str;   if str = 4 then     nreal:= (nint + 5);    elsif str = 5 then     str:= str + 3;     nint:= str;    elsif (str = 6) and (str = 5)   then     str:= str + 3;   else     str:= str + 3;   end if; END; </pre>	<pre> public void teste(){   String str;   double nreal;   long nint;   boolean bx;    str = String.valueOf(1);   str = String.valueOf(Long.parseLong(str) + 3);   nreal = nint * 5 + Long.parseLong(str);   str = String.valueOf(Long.parseLong(str)+1)+str;   nreal = nint + 5 + 1;   nreal = Double.parseDouble(str);   if (String.valueOf(4).equals(str)) {     nreal = (nint + 5);   }   else if (String.valueOf(5).equals(str)) {     str = String.valueOf(Long.parseLong(str)+3);     nint = Long.parseLong(str);   }   else if ((String.valueOf(6).equals(str)) &amp;&amp;     (String.valueOf(5).equals(str))) {     str = String.valueOf(Long.parseLong(str) + 3);   }   else {     str = String.valueOf(Long.parseLong(str) + 3);   } } </pre>



# Desenvolvimento da ferramenta

🌸 Especificação da saída

✚ Geração de classes Java

✚ Pacote view

✚ Pacote Controller

✚ Pacote model



# Desenvolvimento da ferramenta

🌸 Análise da entrada

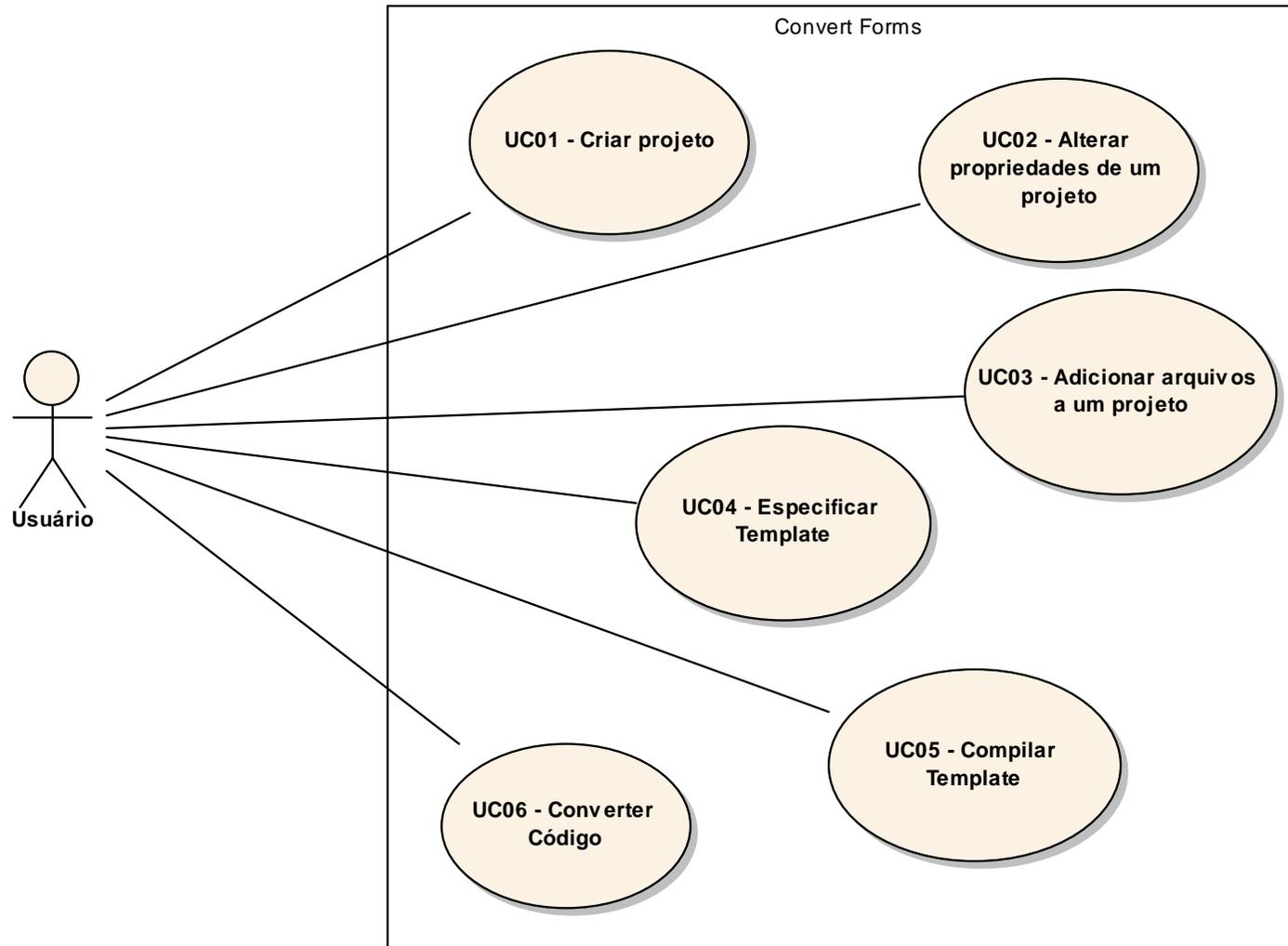
✚ Arquivos Oracle Forms

✚ Código PL/SQL



# Desenvolvimento da ferramenta

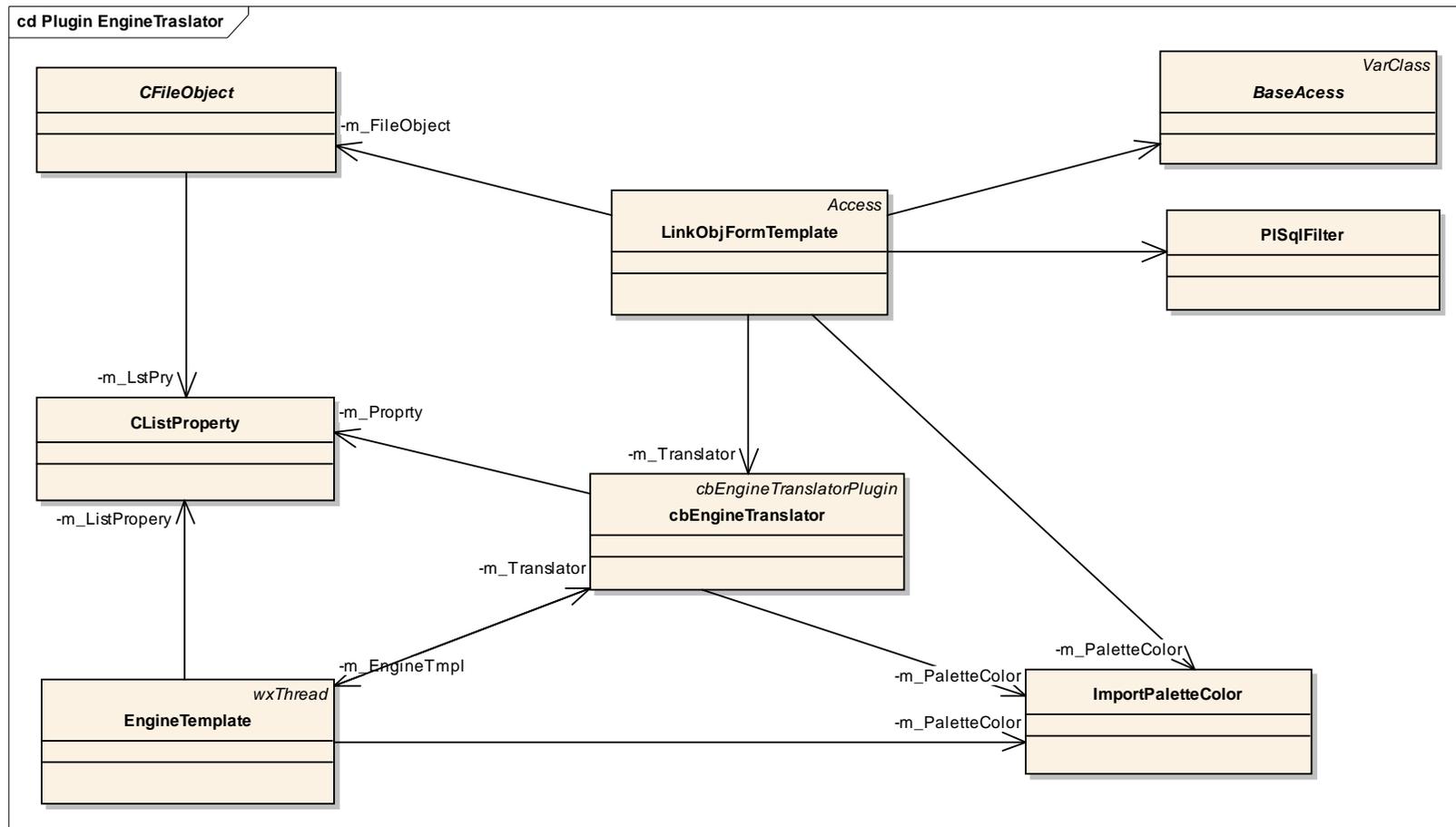
🌸 Especificação: casos de uso





# Desenvolvimento da ferramenta

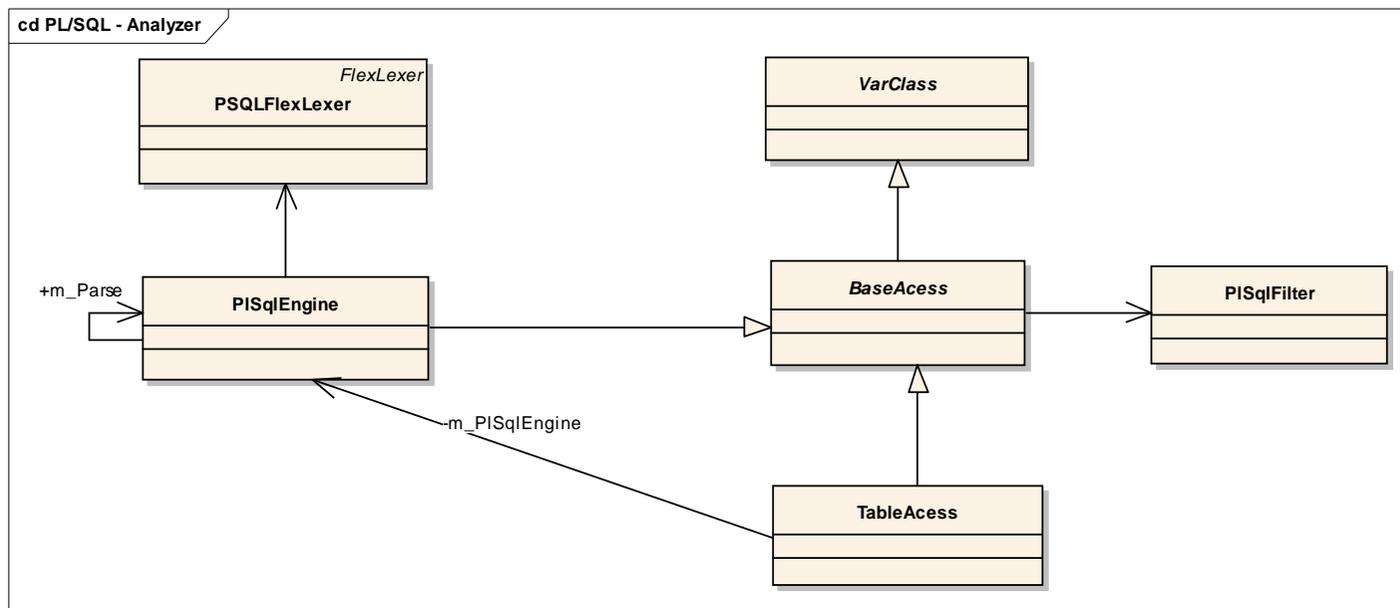
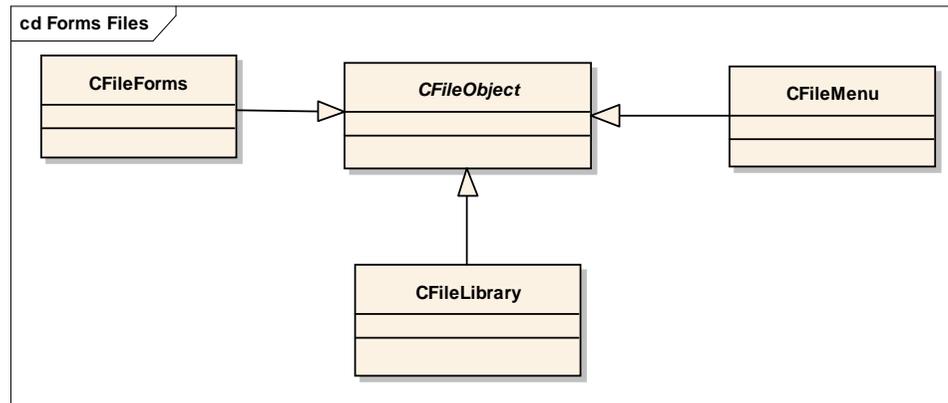
🌸 Especificação: diagrama de classe





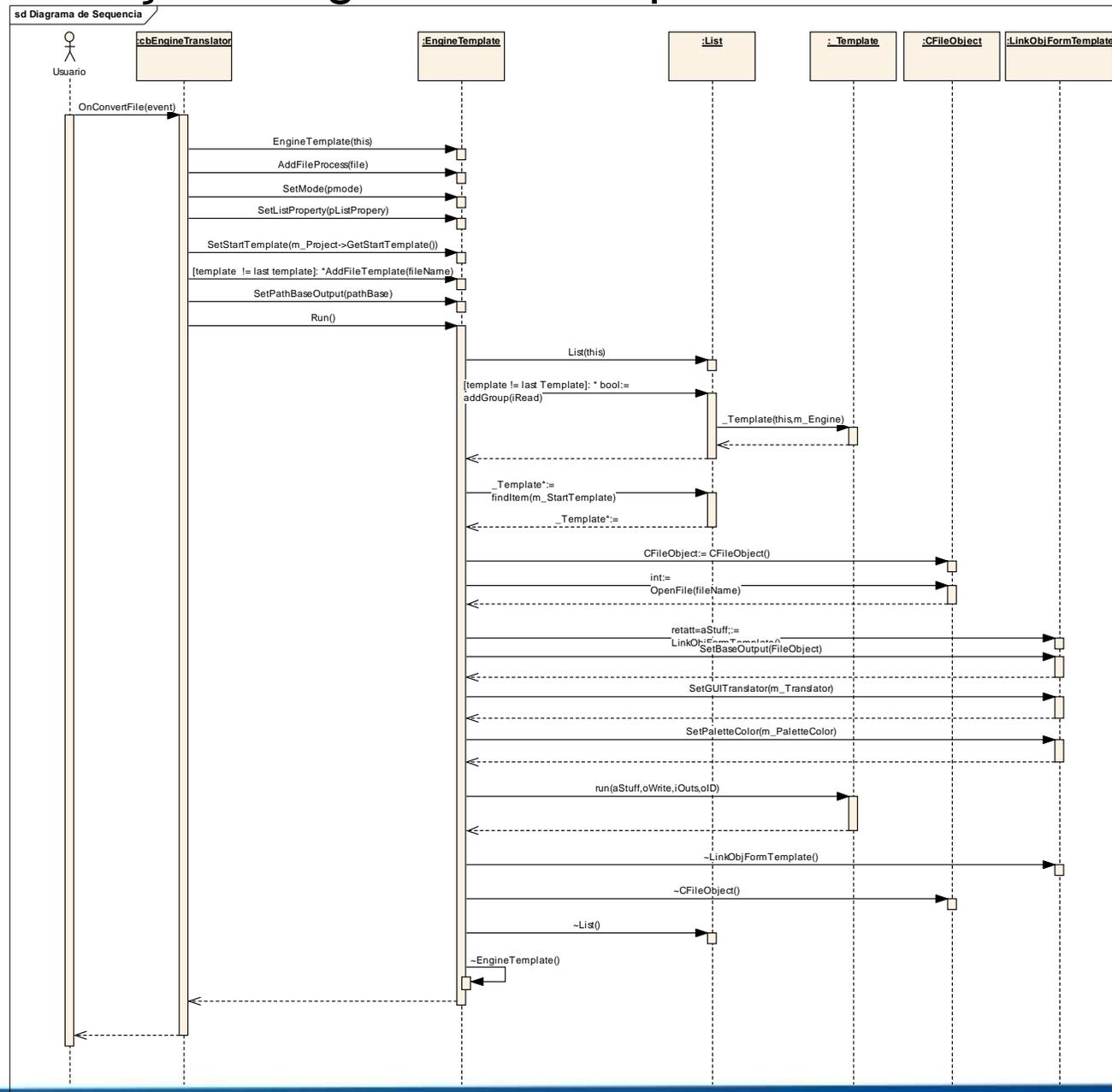
# Desenvolvimento da ferramenta

🌸 Especificação: diagrama de classe



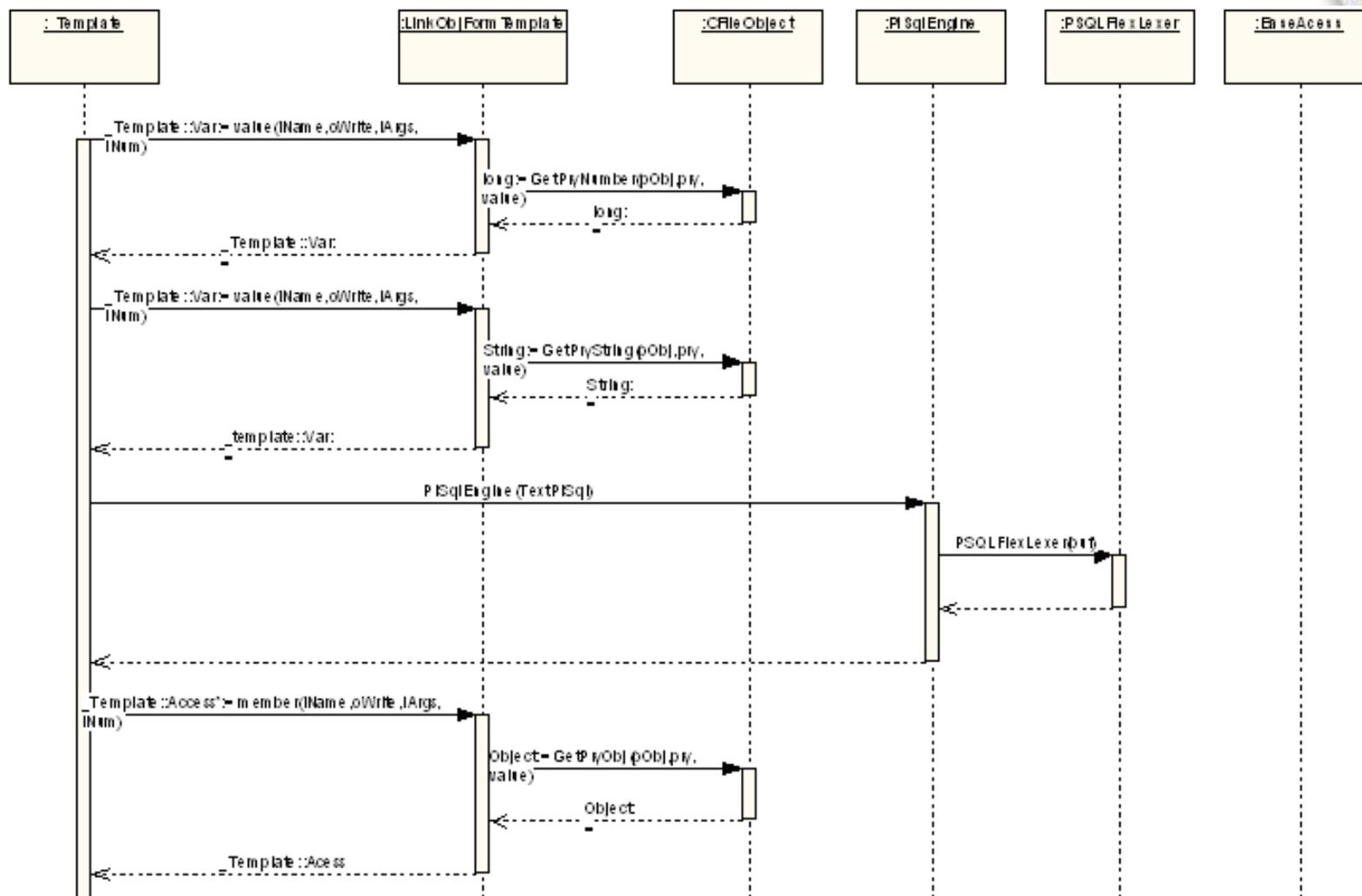


## Especificação: diagrama de seqüência



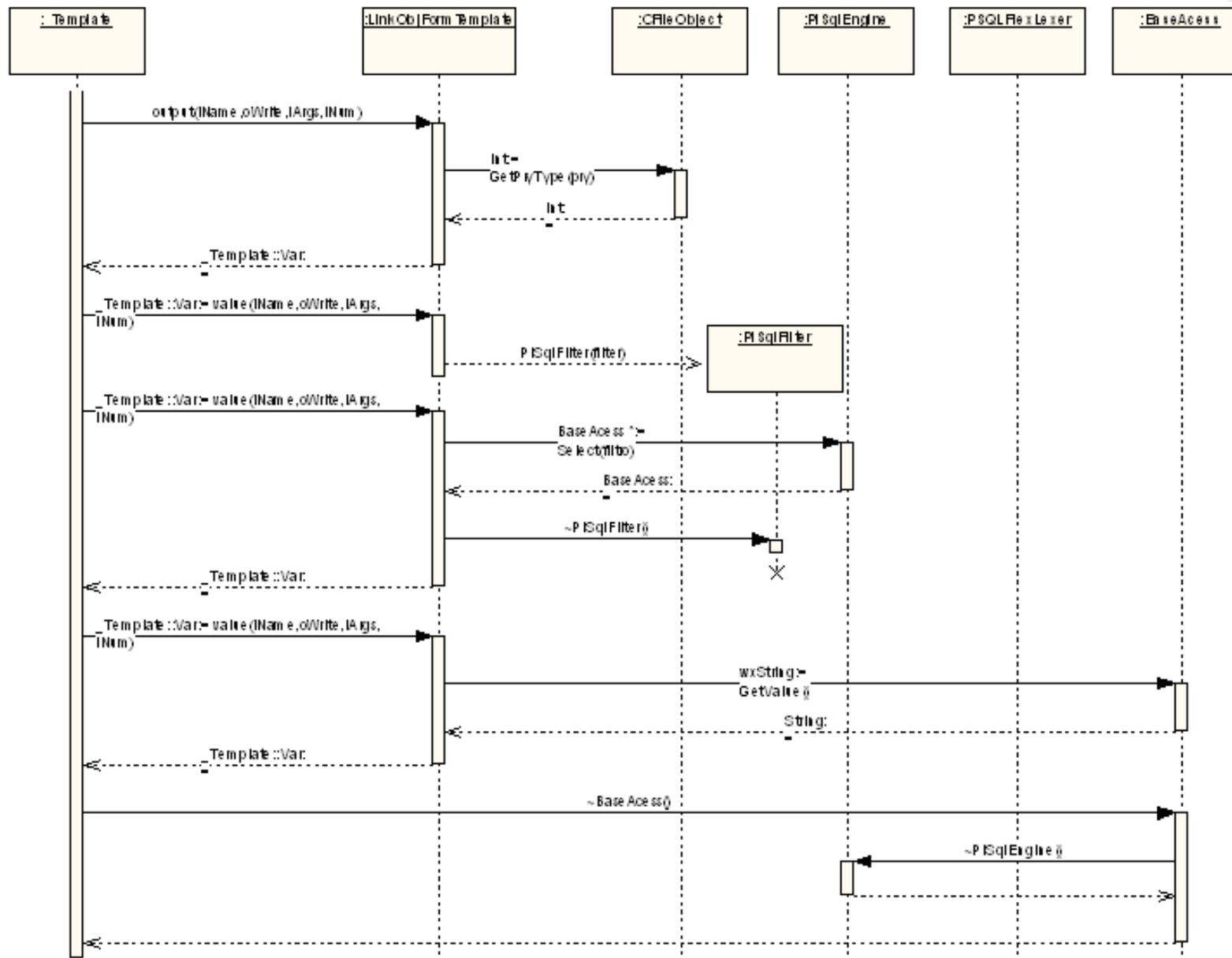


## 🌸 Especificação: diagrama de seqüência





## 🌸 Especificação: diagrama de seqüência





## Desenvolvimento da ferramenta

### Técnicas e ferramentas utilizadas

- Interface
- Plugin
- Forms API
- Flex e Bison
- eNITL



## Desenvolvimento da ferramenta

🌸 Implementação da ferramenta: projetos criados no ambiente de desenvolvimento Microsoft Visual C++ 6

📁 Sdk

📁 TinyXML

📁 wxScintilla

📁 PISqlEngine

📁 ManegerConvert\_Forms

📁 wxDocket

📁 EngineTranlator

📁 Convert\_Forms



## Comparativo

Ferramenta	Exodus	SwisSQL	Forms2 Net	Converte Forms
<b>Característica</b>				
Código gerado baseado em API open source	Sim	Sim	Não	Sim
Suporte a todas as construções PL/SQL	Sim	Sim	Não	Instruções PL
Conversão de componentes visuais	Sim	Não	Sim	Sim
Migração para J2EE	Sim	Não mencionado	Não	Não
Suporte a JDBC	Sim	Não mencionado	Não mencionado	Não
Uso de <i>templates</i> para a geração de código	Não	Não	Não	Sim
Linguagem alvo	Java	Java	ASP.NET	Java



## Conclusões

- 🌸 Reengenharia de software
- 🌸 Motor de *templates*
- 🌸 Oracle Forms API
- 🌸 Restrições



## Extensões

- ✿ Efetuar a conversão dos demais componentes visuais e de todo o código PL/SQL;
- ✿ Implementar classes Java para prover comportamentos idênticos ao objeto Block do Oracle Forms;
- ✿ Implementar conectividade com banco de dados para as classes geradas;
- ✿ Permitir traduzir todas as funções nativas da linguagem Oracle Forms ;
- ✿ Possibilitar o acesso à propriedade coordinate system dos módulos Forms;
- ✿ Exportar imagens contidas nos módulos Forms;
- ✿ Especificar *templates* para a geração de código para outra linguagem, como por exemplo, C#;
- ✿ Implementar *plugin* para gerar a especificação a partir dos arquivos do Oracle Form.