

Universidade Federal de Minas Gerais  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação

ALEX ANDRE DE SOUZA CARMO

**USO DO FLEX PARA SISTEMAS DESNVOLVIDOS EM PROGRESS**

Belo Horizonte  
2011

Universidade Federal de Minas Gerais  
Instituto de Ciências Exatas  
Departamento de Ciências da Computação  
Especialização em Informática: Ênfase: Engenharia de Software

**USO DO FLEX PARA SISTEMAS DESNVOLVIDOS EM PROGRESS**

por

ALEX ANDRE DE SOUZA CARMO

Monografia de Final de Curso  
CEI-ES-0352-T6-2007-01

Prof. Roberto da Silva Bigonha  
Orientador

Belo Horizonte  
2011

ALEX ANDRE DE SOUZA CARMO

## **USO DO FLEX PARA SISTEMAS DESNVOLVIDOS EM PROGRESS**

Monografia apresentada ao Curso de Especialização em Informática do Departamento de Ciências Exatas da Universidade Federal de Minas Gerais, como requisito parcial para a obtenção do certificado de Especialista em Informática.

Área de concentração: Ênfase em Engenharia de Software

Orientador(a): Prof. Roberto da Silva Bigonha

Belo Horizonte  
2011

*Dedico este trabalho primeiramente à minha noiva Hedirlaila, pela sua paciência e por estar ao meu lado todo este tempo, me dando forças e acreditando no meu sucesso. E dedico a todos aqueles que contribuíram direta ou indiretamente para a conclusão desta obra.*

"E aqueles que foram vistos dançando foram julgados insanos por aqueles que não podiam escutar a música." (Friedrich Nietzsche)

## RESUMO

O principal objetivo do presente trabalho é demonstrar os benefícios do Adobe Flex como ferramenta de front-end para a linguagem de desenvolvimento Progress, mais especificamente o Webspeed, que é uma cópia do Progress para desenvolvimento de produtos Web, que possui poucos recursos disponíveis para o desenvolvimento de interfaces amigáveis e criativas de interação com o usuário. O Flex acabou se tornando uma das melhores alternativas, porque, além de gratuito, possui todos os recursos necessários para comunicação com diversas outras linguagens de back-end. Sua fácil estrutura se resume a linguagem MXML, responsável pelo desenvolvimento do layout das aplicações, e o ActionScript que permite a codificação lógica do sistema. No fim do trabalho iremos apresentar uma empresa que faz uso de tais tecnologias e possui sistemas que são “cases” de sucesso e foram desenvolvidos usando o Flex e o Progress Webspeed, permitindo comunicação direta com o ERP Datasul sem a necessidade de outras ferramentas e nem codificações complexas para tal feito.

**Palavras-chave:** Front-end, Back-end, interfaces, ERP.

## **ABSTRACT**

The main objective of this study is to demonstrate the benefits of Adobe Flex as a front-end tool for the Progress development language, specifically Webspeed which is a copy of Progress for development of Web products, which has few resources available for development of friendly interfaces and creative interaction with the user. Flex has become one of the best alternatives because, in addition of being free, has all the resources necessary for communication with several other languages' backend. Its easy structure boils down to MXML responsible for developing the layout and ActionScript applications that allow the encoding system logic. At the end of the work we will present a company that makes use of such technologies, and has produced systems that are "cases" of success and were developed using Flex and Progress Webspeed allowing direct communication with the ERP Datasul without the need for other tools and even complex encodings for such a feat.

**Keywords:** Front-end, Back-end, interfaces, ERP.

## LISTA DE FIGURAS

Figura 1: Exemplo de erp feito em flex.....	17
figura 2: Exemplo de código mxml .....	19
figura 3: Modelo de configuração .....	23
figura 4: Exemplo de código progress .....	24
figura 5: Modelo de banco progress .....	25
figura 6: Página principal do webtools.....	27
figura 7: Exemplo de código httpservice.....	30
figura 8: Exemplo de código usando resultformat .....	30
figura 9: Exemplo código webservices .....	31
figura 10: Exemplo de código remoteobject .....	32
figura 12: Sistema integrado de vendas (siv) .....	36



## LISTA DE TABELAS

Tabela 1: Características do flex.....	34
--	----

## LISTA DE SIGLAS

CSS	.....	<i>Cascading Style Sheets</i>
ERP		<i>Enterprise Resource Planning</i>
HTML		<i>HyperText Markup Language</i>
Web		<i>World Wide Web</i>
Xml	....	<i>eXtensible Markup Language</i>



## SUMÁRIO

<b>RESUMO .....</b>	<b>VI</b>
<b>ABSTRACT .....</b>	<b>VII</b>
<b>LISTA DE FIGURAS .....</b>	<b>VIII</b>
<b>LISTA DE TABELAS.....</b>	<b>IX</b>
<b>LISTA DE SIGLAS .....</b>	<b>X</b>
<b>1. INTRODUÇÃO.....</b>	<b>13</b>
1.1. Objetivo geral.....	13
1.2. Objetivo específico.....	13
<b>2. GESTÃO DO CONHECIMENTO.....</b>	<b>15</b>
2.1. Adobe flex.....	15
2.2. Usando ria para motivar o usuário.....	16
2.3. Estrutura do adobe flex.....	17
2.4. A linguagem mxml .....	18
2.5. A linguagem actionscript.....	19
<b>3. PROGRESS 4GL.....</b>	<b>22</b>
3.1. História e objetivo do progress.....	22
3.2. O banco progress .....	24
3.3. Webspeed e progress.....	26
<b>4. USO DO FLEX EM SISTEMAS PROGRESS.....</b>	<b>29</b>
4.1. Configuração httpservice .....	29
4.2. Comunicação com webservices.....	30
4.3. Remoteobjects.....	31
4.4. Porque usar adobe flex? .....	33
<b>5. RESULTADOS .....</b>	<b>35</b>
<b>6. CONCLUSÃO.....</b>	<b>37</b>
6.1. Contribuições da monografia .....	37
<b>REFERÊNCIAS .....</b>	<b>38</b>

## **1. INTRODUÇÃO**

Com os variados tipos de linguagens e a exigência do mercado para melhores sistemas com qualidade e design sofisticados, as empresas responsáveis pela produção de ferramentas de desenvolvimento estão investindo muito em produtos que atendam essas expectativas, como é o caso da Adobe, que responsável pelo Flex Build, ferramenta que, além da facilidade no desenvolvimento de programas, possui vários recursos pré-programados para dar um aparência profissional aos sistemas.

Muitas empresas optam por utilizar no desenvolvimento de seus produtos linguagens mais antigas, mas que, até hoje, garantem um alto grau de segurança e estabilidade dos seus softwares, uma delas é o progress, que é uma linguagem mais avançada e produtiva do que as suas concorrentes de época, como Cobol, Abadas, Natural, Clipper, etc. Apesar de ser uma linguagem robusta e de qualidade, não oferece uma interface amigável, com isto devemos fazer uso de outras ferramentas que podem cobrir esta brecha e transformar o produto desenvolvido altamente competitivo.

Hoje as linguagens de programação alcançaram um alto grau de evolução, que duas linguagens podem interagir umas com outras e formar uma ferramenta completa para produção de qualquer software de qualidade, seja Web, Desktop ou para servidores. Com tanta variedade, pode parecer complicado aprender a lidar com tantas opções, apesar da lógica binária, as instruções de condições serem as mesmas, as sintaxes podem mudar.

### **1.1.OBJETIVO GERAL**

Apresentar fundamentação teórica para apresentar o Adobe Flex como ferramenta de frente-end para sistemas desenvolvidos com progress 4GL

### **1.2.OBJETIVO ESPECÍFICO.**

- Pesquisar e apresentar fundamentação teórica do Flex como ferramenta mais apropriada para desenvolvimento de grandes interfaces.
- Pesquisar e apresentar fundamentação teórica para utilização do progress como uma das ferramentas desenvolvimento mais robustas do mercado
- Pesquisar e apresentar fundamentação teórica para construção de produtos usando Flex e Progress 4GL.
- Apresentar modelos de sucesso de produtos desenvolvidos com Adobe Flex e Progress 4GL.

## 2. GESTÃO DO CONHECIMENTO

### 2.1. ADOBE FLEX

O Adobe Flex é uma estrutura de código aberto, criada pelo Adobe, específica para desenvolvimento de aplicativos Web por sua grande facilidade na codificação para este tipo de ambiente e pela grande quantidade de efeitos herdados do Flash, fazem do Flex uma linguagem completa para o desenvolvimento de aplicativos Web.

O flex, antes conhecido como Macromédia Flex, foi lançado em março de 2004 pela Macromédia com uma estrutura baseada em aplicações para Internet, ele possui recursos do Flash, pois, foi baseado na plataforma Macromédia Flash.

A versão inicial possuía um SDK, um IDE e uma integração com o J2EE também conhecido como Flex Data Services. Desde que a Adobe adquiriu a Macromedia em 2005, as versões subsequentes do Flex começaram a requerer uma licença para o Flex Data Services, que era inicialmente um produto separado e que posteriormente foi rebatizado como LiveCycle Data Services. (WIKPEDIA)<sup>1</sup>

O Flex possui algumas facilidades que você não encontra no HTML, apesar de ter características baseadas no Flash, ele possui certa facilidade no desenvolvimento de aplicativos. Contando com MXML e XML, que facilitam a construção de *lay-out*, a interação com o usuário é feita com o uso do ActionScript, linguagem padrão do Flash, que se baseia em ECMAScript<sup>2</sup>.

O Flex vem com várias ferramentas prontas, que você não encontra, por exemplo no HTML, tais como: botões, list Box, árvores de acesso, etc.

Com o Flex, uma sigla ficou muito famosa, RIA, Rich Internet Application ou Aplicações Ricas para Internet. Este conceito veio trazer para o Flex e para internet características e funcionalidades de Desktop, isto é, desenvolver sistemas para web usando recursos visuais avançados e novos conceitos de interação com o usuário.

Conforme SCHMITZ (2009), as Aplicações Ricas para Internet não deixam a Web simplesmente com semelhanças desktop, vai muito além, com maior interatividade

---

<sup>1</sup> [http://pt.wikipedia.org/wiki/Adobe\\_Flex](http://pt.wikipedia.org/wiki/Adobe_Flex)

<sup>2</sup> **ECMAScript** é uma linguagem de programação baseada em *scripts*, padronizada pela Ecma International na especificação ECMA-262

com o usuário, permitindo-se explorar algo que ainda não tinha sido feito. O RIA mudar a forma como os usuários utilizam o sistema, em vez de simplesmente ficar preenchendo formulários, e gerando relatórios, é possível utilizar todos os recursos disponíveis para a interação com o usuário.

## **2.2. USANDO RIA PARA MOTIVAR O USUÁRIO**

Hoje os desenvolvedores sabem que, um pecado mortal, é desenvolver um sistema que não atenda todas as especificações relacionadas ao fator “visual” da aplicação. De acordo com SCHMITZ (2009), uma aplicação mal desenhada, sem usabilidade, pode acarretar um fracasso no projeto, mesmo que todas suas funcionalidades estejam corretas e sem erros.

Os usuários finais que iram utilizar o sistema prezam muito pela aparência e pela facilidade de acesso e digitação dos dados. Como mostra SCHMITZ (2009), para se obter um sistema que tenha uma usabilidade de qualidade e um design elegante, são necessárias horas trabalhadas para se encontrar as cores, ícones e fontes necessárias que iram agradar o usuário.

Com aplicações RIA, vários destes conceitos citados, se tornaram muito mais simples. Se você é um desenvolvedor que trabalha com HTML, sabe das dificuldades que esta linguagem tem em proporcionar um ambiente atraente para o usuário, seja para posicionar um elemento na tela, ou apresentar uma mensagem de aviso. Diferentemente do adobe Flex, que possui recursos que irão facilitar e agilizar toda a construção de recursos visuais necessários para um software de qualidade, de acordo com a figura 1, podemos ver em detalhes como um sistema web feito em Flex possui um visual mais sofisticado e profissional.



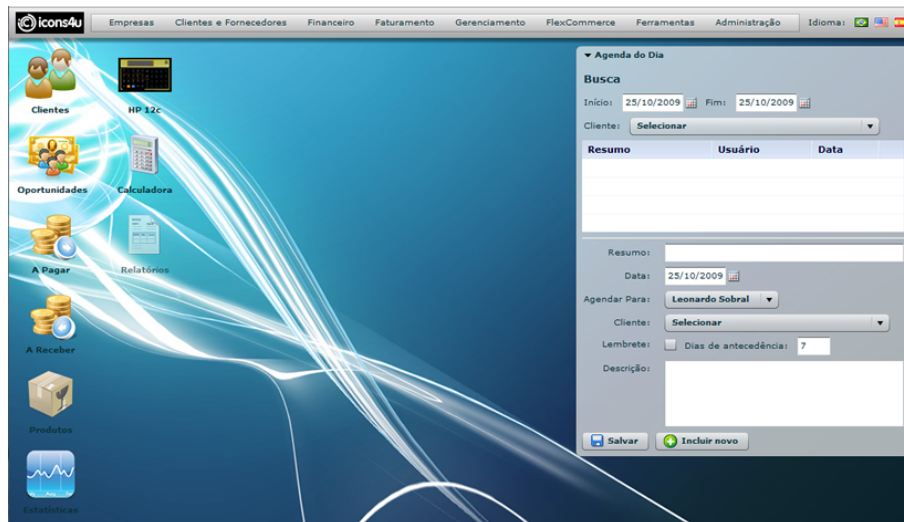


Figura 1: Exemplo de ERP feito em Flex.

Fonte: <http://demo.flasherp.com.br/#>

De frente a tais características, encontramos sua definição oficial, encontrada em <http://www.adobe.com/products/flex/>:

*“Flex is a highly productive, free open source framework for building and maintaining expressive web applications that deploy consistently on all major browsers, desktops, and operating systems”*

O Flex como uma ferramenta open source para desenvolvimento de aplicações Web poderá ser acessada por meio de um navegador, seu diferencial é que as aplicações criadas podem usar interfaces ricas, com recursos e componentes, além de funcionalidades como “arrastar e soltar”, mover objetos, zoom, transparência, etc, com o objetivo de tornar a experiência com o usuário a melhor possível.

### 2.3. ESTRUTURA DO ADOBE FLEX

De acordo com SCHMITZ (2009), o FLEX possui, em sua estrutura de desenvolvimento, duas linguagens de programação. Elas são base para desenvolvimento de todas as aplicações e são chamadas de MXML e ActionScript.

- **MXML:** Uma linguagem com características semelhantes ao XML por usar marcação de texto, seu objetivo principal é desenhar o layout da aplicação. Sua estrutura de desenvolvimento é idêntica a estrutura

HTML, mas, por possuir padrões do XML, é bem mais estruturada. Como no HTML, utiliza tags, que são comandos que se iniciam com o sinal de menor e terminam com o sinal de maior, assim como HTML.

- **ActionScript:** É a linguagem responsável pelo desenvolvimento de toda a lógica da aplicação. Na versão 3.0, o ActionScript se tornou ainda mais poderoso, com suporte a orientação a objetos, tratamento de exceções, eventos, pacotes, expressões regulares, etc.

Existem vários outros arquivos presentes no Flex como: folhas de estilo (css), imagens, arquivos de configuração, etc.

O compilador do flex é responsável por verificar toda a codificação e armazenar em um arquivo com a extensão swf.

A extensão swf foi introduzida pela antiga Macromédia no desenvolvimento do Flash, que é um plug-in responsável pelo funcionamento das aplicações feitas em Flex e encontra-se na versão 10, de acordo com o site oficial da ADOBE<sup>3</sup>, “o flash encontra-se Instalado em mais de 98% dos computadores conectados à Internet”, devido ao seu sucesso, é responsável por identificar o arquivos swf e fazer com que sua aplicação funcione.

## 2.4. A LINGUAGEM MXML

Foi apresentado que a linguagem MXML é também utilizada para o desenho das aplicações. E foi apresentado também que o MXML é uma linguagem de marcação de texto como o HTML e com estrutura semelhante ao XML, na figura 2, é apresentado um exemplo de código MXML.

---

<sup>3</sup> <http://www.adobe.com/br/products/flex/overview/>

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
  layout="absolute">

  <mx:Panel id="pnlLogin" layout="absolute" title="Sistema de Login - Imasters"
    x="10" y="10"
    width="250" height="141"
    horizontalCenter="0" verticalCenter="0">

    <mx:Label text="Login:"
      x="10" y="10" />
    <mx:TextInput id="tfLogin"
      x="50" y="8"/>

    <mx:Label text="Senha: "
      x="5" y="36"/>
    <mx:TextInput id="tfSenha"
      x="58" y="34"
      displayAsPassword="true"/>

    <mx:Button x="84" y="64"
      label="Entrar"/>

  </mx:Panel>
</mx:Application>

```

**Figura 2:** Exemplo de código MXML

**Fonte:** [http://imasters.com.br/artigo/11822/flex/sistema\\_de\\_login/](http://imasters.com.br/artigo/11822/flex/sistema_de_login/)

De acordo com a Figura 2, temos um código MXML completo e com ele podemos identificar sua estrutura, que obedece às mesmas regras do XML.

- Toda tag deve ter seu fechamento;
- As tags devem estar alinhadas;
- Todo documento MXML deve ter uma única tag raiz, neste caso a `<mx:Application>`;
- A propriedade `id` não é obrigatória em MXML;
- Campos booleanos somente aceitam `true` ou `false` como valores.

Essas são algumas características do MXML que ainda apresenta uma variedade grande de componentes prontos para Adobe Flex, tais componentes reunidos irão formar sua aplicação.

## 2.5.A LINGUAGEM ACTIONSCRIPT

De acordo com SCHIMTZ (2009), quando concluímos a inserção de componentes na tela que irão formar o layout da aplicação, será necessária a inserção de

uma lógica que permitirá que a aplicação faça algo mais elaborado. E toda esta lógica é criada por meio da linguagem ActionScript, que baseada no padrão ECMAScript, permite inserir funcionalidades que diversas outras linguagens mais tradicionais como Java, C++ e PHP possuem.

Com o ActionScript, você pode inserir várias funcionalidades de orientação a objetos, que tornará sua aplicação mais poderosa e profissional como:

- **Tratamento de exceções:** No ActionScript versão 3.0, temos um tratamento de erros mais estruturado que permite capturar informações mais precisas dos erros. Podemos obter informações sobre o *stack trace*<sup>4</sup>
- **Tipos:** Na versão 3.0 do ActionScript, os tipos são mantidos trazendo uma melhora na otimização de memória.
- **Classes Seladas:** Classes criadas no ActionScript 3.0, definindo suas propriedades e métodos, dizemos que esta classe estará “selada” em runtime, ou seja, não será possível adicionar novas propriedades ou métodos permitindo o compilador realizar análises mais detalhadas no código prevenindo possíveis erros.
- **ECMAScript:** Uma forma mais fácil de manipular arquivos XML.
- **Expressões regulares:** O ActionScript tem total suporte a elas.
- **NameSpaces:** Para separar melhor a lógica da aplicação, podemos dividir nossas classes usando namespaces em um conjunto de pacotes.
- **Novos tipos primitivos:** Os tipos primitivos estão presentes em todas as linguagens, são eles, Inteiros, Strings, booleanos, etc. No ActionScript temos novos tipos como int e uint, que ampliam a manipulação de dados.
- **Carregamento dinâmico de dados:** É possível carregar dados, imagens de forma dinâmica, durante o processamento da aplicação.
- **Acesso “baixo nível”:** É possível acessar dados em baixo nível para manipular por exemplo, bytes de uma resposta de requisição de um servidor.

---

<sup>4</sup> Stack trace: rastreamento de pilha

- **Melhorias na manipulação de textos:** O ActionScript possui API's que poderiam manipular textos de várias maneiras como, obter uma palavra em um texto, alterar a fonte do texto, etc.

### 3. PROGRESS 4GL

#### 3.1. HISTÓRIA E OBJETIVO DO PROGRESS

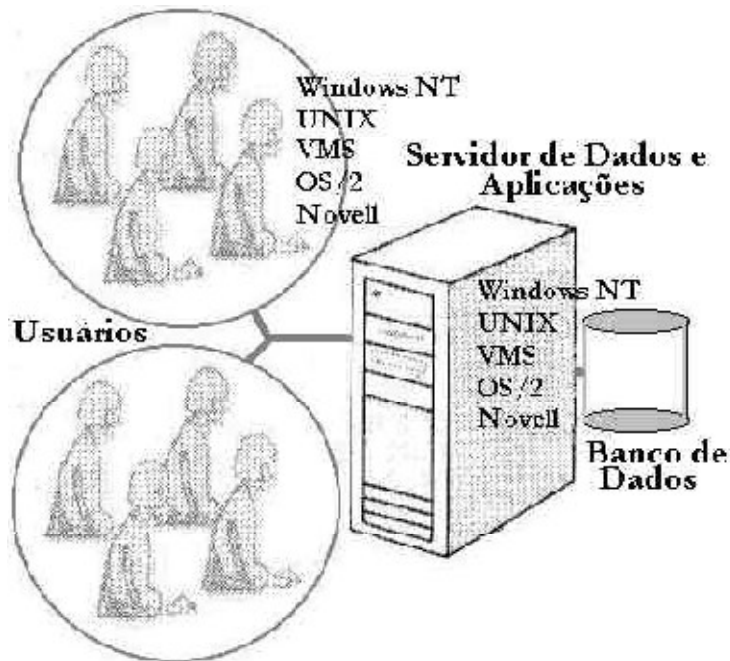
De acordo com COSTA (2000), o progress foi desenvolvido pela Progress Software Corporation em 1984, com o intuito de usá-lo em sistemas Unix com uso em mainframe. Foi uma alternativa para o Cobol, Abadas, Clipper, etc, onde exigia-se uma grande quantidade de linhas de código para qualquer aplicação. Seu objetivo é focado em sistemas que pudessem trabalhar com uma grande quantidade de informações. Seu maior sucesso é a portabilidade e ser livre de plataforma, podendo funcionar em praticamente qualquer sistema operacional hoje no mercado.

Possui um banco de dados relacional de altíssimo desempenho, podendo ser comparado ou até melhor que muitos outros existentes no mercado como: Oracle, DB/2 IBM, My SQL Server, entre outros.

Sendo uma linguagem de quarta geração, por isto o nome Progress 4GL, baseada em eventos e proprietária, possui integrados em um único produto Banco de Dados e Linguagem de desenvolvimento, fazendo progress se tornar uma excelente ferramenta de desenvolvimento para aplicações comerciais. Esse fato ocorre porque as camadas de desenvolvimento como: regras de negócio, dados e interface estão totalmente unidas em uma única ferramenta, isso evita redundância, ou retrabalho em qualquer camada da aplicação.

A configuração típica e mais comum é a instalação e armazenamento centralizado do Progress servidor e cliente, bancos de dados e aplicações em um servidor disponível para acesso de diversos clientes em rede. Costa Marcio, 2000, p.21.

Há vários outros tipos de configurações para o progress como: múltiplos servidores, banco de dados distribuídos, servidores ou plataformas distintas, a Figura 3 mostra com mais detalhes o modelo de configuração de uma instalação.



**Figura 3:** Modelo de Configuração  
**Fonte:** (COSTA, 2000)

A linguagem Progress, é usada especificamente para desenvolvimento de Sistemas de Gestão (ERP) que são sistemas integrados para controle de uma empresa. Seu maior uso é pelas empresas de médio e grande porte, por ter um custo elevado de licença. Um sistema ERP muito famoso desenvolvido em Progress 4GL é o Datasul, que possui como dono hoje a Totvs. Na Figura 2, temos um modelo de código progress.

```

DEFINE VARIABLE i          AS INTEGER INIT 0 NO-UNDO.
DEFINE VARIABLE j          AS INTEGER INIT 0 NO-UNDO.
DEFINE VARIABLE soma       AS INTEGER INIT 0 NO-UNDO.
DEFINE VARIABLE num-achados AS INTEGER INIT 1 NO-UNDO.
DEFINE VARIABLE qtd-procurada AS INTEGER INIT 0 NO-UNDO.
DEFINE VARIABLE num-testado AS INTEGER INIT 0 NO-UNDO.

```

```

ASSIGN num-perf.
ASSIGN qtd-procurada = num-perf.
ASSIGN i = 0.

```

```

DO WHILE (num-achados <= qtd-procurada):
  ASSIGN i = i + 2.
  IF (i MOD 10 = 6 OR i MOD 10 = 8) THEN DO:
    ASSIGN
      soma = 0
      j = 1.

```

```

DO WHILE (j <= (i / 2)):

```

```

IF (i MOD j = 0) THEN
  ASSIGN
    soma = soma + j.
  j = j + 1.
END.
IF (i = soma) THEN DO:
  MESSAGE i ' é Numero Perfeito'
  VIEW-AS ALERT-BOX INFO BUTTONS OK.
  ASSIGN num-achados = num-achados + 1.
END.
END.
END.

```

**Figura 4:** Exemplo de código progress

**Fonte:** wikipedia<sup>5</sup>

De acordo com a Figura 4, temos um exemplo de código Progress e podemos perceber que ele é semelhante, em sua lógica, a outras linguagens, temos os comandos de laços, IF, ELSE, WHILE, apresentação de mensagens para o usuário usando MESSAGE, definição de variáveis, etc. Apesar de ser uma ferramenta robusta e segura, o Progress é relativamente simples no uso de desenvolvimento de aplicativos por ter uma linguagem prática e objetiva.

### 3.2. O BANCO PROGRESS

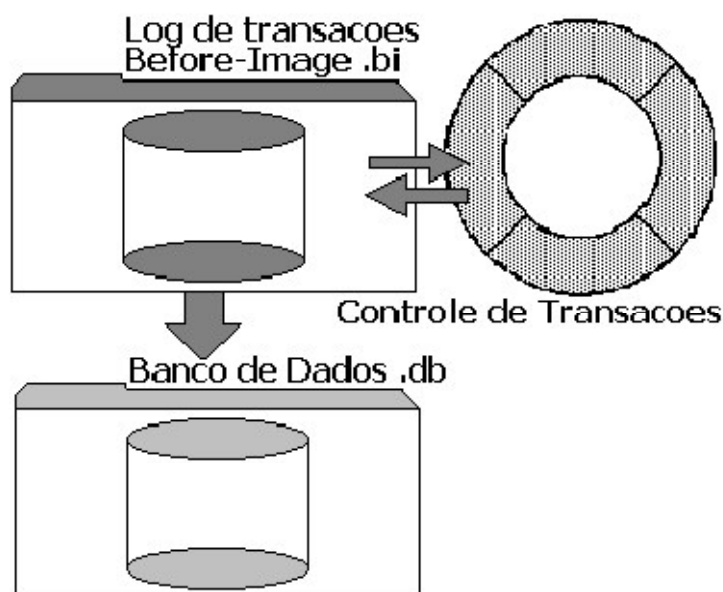
BRENER (2000) apresenta como é a estrutura do Progress em relação ao banco de dados, a forma como ele é constituído por dois arquivos básicos, o próprio banco de dados com extensão BD e o controle de transações com extensão BI (Before-image).

Como foi apresentado por BRENER (2000), esses dois arquivos juntos, conseguem gerenciar todas as solicitações dos aplicativos que fazem ligação a eles, de acordo com a Figura 5, podemos ver uma prévia de como é o funcionamento do banco.

---

<sup>5</sup> [http://pt.wikipedia.org/wiki/Progress\\_4GL](http://pt.wikipedia.org/wiki/Progress_4GL)





**Figura 5:** Modelo de banco progress  
**Fonte:** Brenner (1998, p. 52)

Um dos arquivos de maior importância é o arquivo de “Imagem Anterior” (Before Image), sua importância é confirmada, por assegurar a total e completa integridade física e referencial dos dados, como por exemplo, índices, validações, tipos e formatos através do “*Engine Database*”, ela confirma estas transações para gravação ou deleção no banco de dados, tudo vai depender de uma completa consistência dos dados existentes, tornando a segurança do processo mais eficaz no momento de tratar com as informações.

Os métodos para criação de um banco de dados mostrados se aplicam apenas para Bancos de Volume Único ou Simples, existem técnicas em Progress mais sofisticadas para criação de bancos em Múltiplos Volumes, conforme será mostrado no capítulo Performance. (BRENER, 2000, p. 54).

EDELSBRUNNER (1998) nos fala que para as demandas especiais de aplicações “Críticas”, muitas empresas escolhem o Sistema de Gerenciamento de Base de Dados Relacional (RDBMS Relational Database Management System) Progress. O progress apesar de ser uma linguagem de desenvolvimento como também um RDBMS, é totalmente integrado à arquitetura Progress DataServer. Acabou se tornando uma aplicação com uma base de dados de alta performance para softwares do mundo real. As aplicações para atenderem a grande quantidade de informação e processamento, devem ter grande capacidade para suportar um alto volume de dados e também alto rendimento

e baixo tempo de resposta para transações mais complexas que podem interagir várias aplicações de negócio, para estas situações, o progress está completo.

O progress RDBMS possui um conjunto de recursos com o objetivo específico de suportar tais ambientes onde a execução de um alto volume de transações, integridade de dados garantida e alta disponibilidade são necessárias. (BRENER, 2000, p. 54).

Com tais especificações, podemos apresentar algumas capacidades como:

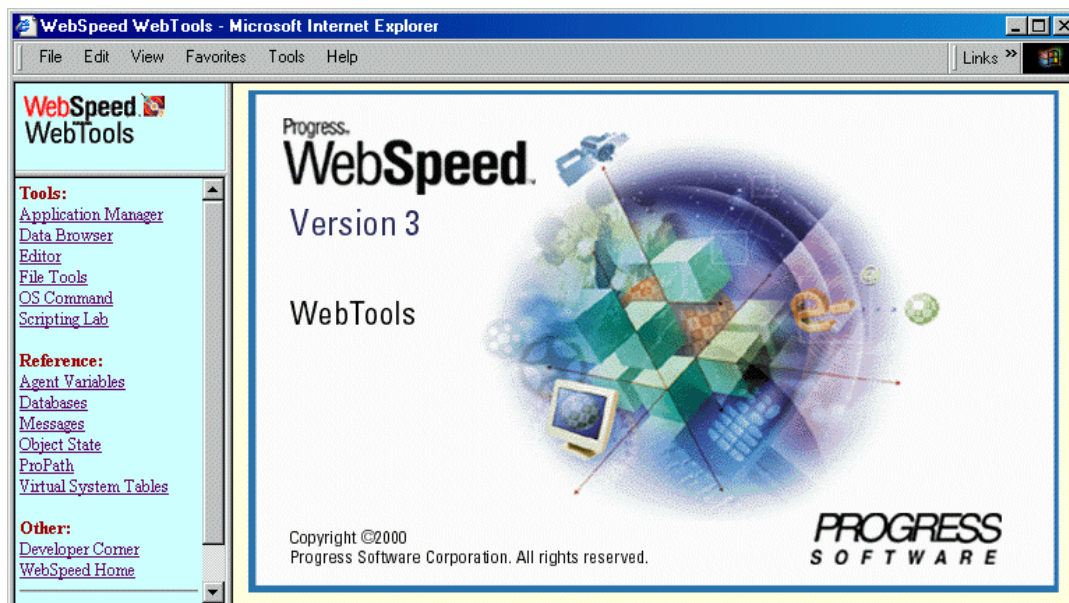
- *Lock* a nível de registro, que minimiza a contenção de dados;
- estratégias de otimização de consultas que melhoram o tempo para recuperação de dados;
- uma inteligente buferização de registros que reduz o tráfego da rede.

No Progress, a integridade de segurança dos dados é dada através das características de *recovery*, projetado com flexibilidade para tirar o máximo proveito das características únicas de desempenho de diferentes plataformas e sistemas operacionais. Quando é utilizado no ambiente cliente/servidor, com configuração baseada em rede, as aplicações Progress podem comunicar-se com o RDBMS, utilizando qualquer protocolo padrão de rede.

### 3.3. WEBSPEED E PROGRESS

O Progress apesar de ser uma linguagem que tem como principal finalidade o desenvolvimento de sistemas ERP, ou programas baseados em desktop, também possui uma ferramenta criada pela Progress, que permite usar recursos para desenvolvimento de sistemas Web, usando a linguagem Progress. Tal ferramenta recebe o nome de Webspeed. O Progress e o Webspeed são ferramentas praticamente iguais no modo de desenvolver, só mudando algumas funcionalidades, já que com webspeed iremos desenvolver programas que irão rodar em um servidor enviando informações para o cliente, bem diferente do Progress normal, a Figura 6 nos mostra uma tela inicial do Webspeed que contém links para diversas configurações do Webspeed. Esta tela recebe o nome de WebTools, que são uma coleção de utilitários baseados em navegador, que permitem executar tarefas de desenvolvimento e acesso a informações do sistema. Cada

ferramenta é acessada diretamente clicando no seu link correspondente onde será apresentada sua tela de configuração.



**Figura 6:** Página principal do WebTools  
**Fonte:** Progress Software Corporation (2000, p. 32)

De acordo com a Progress Software Corporation (2000), o Progress WebSpeed WorkShop é uma suíte integrada de desenvolvimento para construir aplicações ITP (Internet Transaction Processing) aplicações que rodam sobre o WebSpeed Transaction Server, que é um componente do OpenEdge Application Server. O OpenEdge Application é um servidor robusto para desenvolvimento de aplicações com alta disponibilidade, escalabilidade, e taxas de resposta rápida.

Por ser uma ferramenta criada para desenvolver aplicações Web, o Webspeed utiliza uma linguagem Web especializada chamada SpeedScript, que permite combinar lógica com o uso da interface HTML. WebSpeed WorkShop é integrado no sentido que você desenvolve suas aplicações conectando-se a um WebService e um WebSpeed Transaction Server. Isto significa que você está desenvolvendo suas aplicações no mesmo ambiente que sua aplicação irá rodar, aumentando a performance, pois a parte pesada do sistema irá rodar direto no servidor.

Com o Webspeed você pode desenvolver utilizando todos os recursos Web, como HTML, CSS, JavaScript, etc. As aplicações terão acesso facilitado ao banco Progress, pois, a linguagem por ser praticamente a mesma, permite acesso direto, sem precisar de configurações que outras linguagens necessitam para acessar determinados

bancos, com ele você tem acesso simplesmente fazendo referência a uma determinada tabela que está na base de dados.

Por ser uma ferramenta bastante robusta, temos que tomar alguns cuidados, apesar das facilidades, temos diferentes comportamentos dependendo do navegador, tudo tem que ser bem pensado.

A linguagem Webspeed, que será responsável pela comunicação entre o Flex e o Progress, será um back-end da interface Flex e responsável pelo acesso as informações do banco progress. Abaixo são apresentados alguns benefícios do WebSpeed.

- Rápida criação e apresentação de aplicações comerciais em intranets corporativas, extranets, e a internet;
- Uso de um ambiente de desenvolvimento integrado para desenvolver e distribuir aplicações lógicas com SpeedScript;
- Criação de aplicações corporativas robustas com o quite de ferramentas que inclui um compilador, um debugger, um dicionário de dados, e um utilitário de administração de dados.

## 4. USO DO FLEX EM SISTEMAS PROGRESS

Como foi apresentado, o Progress, apesar de ser uma ferramenta robusta utilizada para o desenvolvimento de aplicações comerciais ERP, tem como ponto fraco o layout das aplicações desenvolvidas, principalmente em se tratando de sistemas web, por não possuir muitos recursos disponíveis para esta finalidade.

Com base nesta fragilidade, podemos contar com o uso do Adobe Flex, que é rico em ferramentas que auxiliam o desenvolvimento de interfaces tornando a aparência dos programas amigáveis e agradáveis a qualquer usuário que for utilizá-lo.

De acordo com SCHMITZ (2009), o flex é uma tecnologia que será executada no lado cliente, ou seja, os arquivos swf que contêm a aplicação flex é baixada para a máquina do cliente e copiada para arquivos temporários do navegador.

Com isto, o flex será responsável pela camada de apresentação de dados. Ele ficará encarregado do desenho das telas, menus, formulários, gráficos, etc, enquanto que o progress fará a parte de acesso ao servidor, ou seja, ele será executado no lado servidor, e seu código não ficará aparente no navegador do cliente.

O Flex trabalha com requisições ao servidor, que, por sua vez, processa essa informação e retorna em forma de resposta ao Flex. Por exemplo, quando temos uma tela de login no Flex, este fica responsável em obter os dados do cliente e de enviá-los ao servidor web. No lado do servidor, é verificado se o login está correto e, em caso positivo, é retornada uma mensagem ao Flex dizendo que a senha digitada está correta e que o usuário pode entrar no sistema.

Para que seja feita a comunicação com o Flex e o Servidor, temos três modos distintos, onde cada um deles tem seus lados positivos e negativos.

- HTTPService
- WebService
- RemoteObject

### 4.1. Configuração HTTPService

HTTPService é uma das formas mais simples de conexão entre o Flex e o WebSpeed Progress. Sem necessidade de configurações extras, podendo ser aplicado em qualquer projeto Flex. A requisição Flex acessa um endereço do site, por exemplo:

<http://localhost/seuprojeto/exibirDados.html>

Após esta requisição, o servidor WebSpeed deve imprimir dados pré-formatados na tela, para que o Flex possa lê-los. Estes dados podem estar em três formatos distintos: XML, JSON<sup>6</sup>, ou texto comum, em nosso caso iremos trabalhar com XML.

A primeira forma de conexão seria via HTTP ou HTTPS, o Flex oferece a classe HTTPService para esse fim. Com ela podemos ler dados de arquivos locais ou remotos e trazer esses resultados de formas diferentes como mostrado na figura 7 temos um exemplo de código HTTPService fazendo requisição a um servidor WebSpeed que contém um arquivo XML.

```
<s:HTTPService id="hs" url="arquivo.xml">
s:request>
<dados>{RequestObject}</dados>
</s:request>
</s:HTTPService>
```

**Figura 7:** Exemplo de código HTTPService.

**Fonte:** Autoria própria

O HTTPService também consegue ler dados no formato flashvars através da propriedade resultFormat, sendo uma ótima opção para ler arquivos preparados para o Adobe Flash. A estrutura do flashvar segue o esquema:variavel1=valor&variavel2=valor&variavel3=valor. Na Figura 8 temos um exemplo de acesso usando a propriedade resultFormat.

```
<s:HTTPService id="hs" url="dados.txt" resultFormat="flashvars">
</s:HTTPService>
```

**Figura 8:** Exemplo de código usando resultFormat

**Fonte:** Autoria própria

## 4.2. Comunicação com WebServices

---

<sup>6</sup> Acesso <http://www.json.org>

Os WebServices são mundialmente conhecidos como uma forma de prover dados em Sistemas multiplataforma, como é explicado por SCHMITZ (2009). Quando criamos um Webservice, estamos oferecendo a qualquer tipo de tecnologia ou linguagem de programação a possibilidade de consumir dados.

Por exemplo, existem WebServices para previsão do tempo, conversão de medidas, validação de dados etc. Com isso, conseguimos estabelecer que tecnologias completamente diferentes possam trocar informações entre si, por meio de um protocolo conhecido, geralmente XML.

Uma das formas mais comuns de acesso a dados no Flex é através de SOAP (Simple Object Access Protocol). SOAP provê uma estrutura padronizada de XML para enviar e receber dados de um Webservice através da internet. Usualmente você envia mensagens em SOAP usando HTTP, mas pode também usar SMTP e outros protocolos. Abaixo de acordo com a figura 9, temos um exemplo de código Webservice para acesso a dados.

```
<s:WebService id="ws" wsdl=http://www.site.com.br/websevice?wsdl>
  <s:operation name="listaCategorias"/>
  <s:operation name="listaProdutos">
    <s:request>
      <categoria>{categoria}</categoria>
    </s:request>
  </s:operation>
</s:WebService>
```

**Figura 9:** Exemplo código webservices

**Fonte:** Autoria própria

O objeto *s:WebService* contém o endereço WSDL do Webservice que será acessado.

Todo Webservice é formado por métodos (operações), e no objeto *s:operation* dizemos que os métodos serão chamados.

### 4.3. RemoteObjects

Conforme é apresentado por SCHMITZ (2009), RemoteObjects foram criados inicialmente para uma comunicação muito rápida entre Java e Flash, por meio de um formato de dados chamado AMF<sup>7</sup> (Action Message Format). Esses dados são transmitidos por meio do protocolo SOAP.

O RemoteObject popularizou-se bastante quando passou a suportar outras linguagens de programação, como o PHP e o Ruby on Rails. Esse suporte deve-se à criação de produtos de terceiros. Para o PHP, um dos produtos mais usados é o AMFPHP.

Existem algumas características importantes dos RemoteObjects que facilitam a sua integração com algumas linguagens.

A primeira delas é a forma como os dados são transmitidos entre o servidor e o cliente. Essa forma é completamente binária, muito mais rápida do que o HTTPService e o WebService. O retorno de dados também é binário, aumentando a velocidade de resposta quando estamos preenchendo DataGrids e grandes estrutura em formato de texto.

A segunda característica é a facilidade com que podemos enviar e receber um objeto entre WebSpeed e o Adobe Flex. É possível criar um objeto no Adobe Flex e enviá-lo diretamente ao WebSpeed, mantendo toda a sua composição intacta.

O que basicamente o AMF faz é transformar as requisições do actionscript em requisições para a linguagem destino, e devolver os resultados em um formato compatível, através de um processo chamado serialização. Na figura 10 é apresentado um exemplo de código RemoteObject.

```
<s:RemoteObject id="ro" destination="coldfusion" source="componente">
  <s:method name="ListaProdutos">
    <s:arguments>
      <categoria>novos</categorias>
    </s:arguments>
  </s:method>
</s:RemoteObject>
```

**Figura 10:** Exemplo de código RemoteObject

**Fonte:** Autoria própria

---

<sup>7</sup> Acesse em: [http://en.wikipedia.org/wiki/action\\_message\\_format](http://en.wikipedia.org/wiki/action_message_format)



#### 4.4. Porque usar Adobe Flex?

Como já vimos anteriormente, o Flex é uma ferramenta preparada para ser utilizada como FrontEnd de diversas linguagens de desenvolvimento, umas com alguns recursos próprios de design e outras apresentando recursos escassos para criar uma interface agradável e profissional para o usuário.

Devido a estes fatores, o Flex que é conhecido como uma evolução do Flash tornou-se uma ferramenta poderosa para desenvolver interfaces de muita qualidade e profissionalismo, e tem suas características apresentadas na tabela 1 mostrada abaixo.

<b>Tecnologias</b>	Adobe Flex 4
<b>Runtime</b>	Flash Player 10.1
<b>Aplicações Multi-Plataforma</b>	<ul style="list-style-type: none"> <li>- Windows</li> <li>- MacOS</li> <li>- Linux</li> <li>- Solaris</li> <li>- Google Android</li> </ul>
<b>Animação – Timeline</b>	- Frame-by-Frame
<b>3D</b>	- Baseado em simulação de perspectiva X, Y e Z.
<b>Aplicações baseadas em Desktop 2.0</b>	- Adobe Air
<b>Modelo de programação</b>	- Orientado a Objetos
<b>Linguagem de interfaces</b>	- Baseado em XML, o MXML
<b>Linguagem base – Client</b>	- ActionScript 3
<b>Linguagens utilizadas com frequência – Server/Side</b>	<ul style="list-style-type: none"> <li>- Java</li> <li>- PHP</li> <li>- ColdFusion</li> <li>- C#</li> <li>- Webspeed</li> </ul>
<b>Arquivo executável binário</b>	- Arquivo SWF

<b>Característica da plataforma</b>	<ul style="list-style-type: none"> <li>- Data Binding</li> <li>- Event Handling</li> <li>- Layout</li> <li>- Navigation</li> <li>- Controls</li> <li>- DataGrid</li> <li>- Data Visualization</li> <li>- Styling</li> <li>- Câmera e Microfone</li> <li>- Suporte Offline DRM</li> <li>- Peer-assisted P2P</li> <li>- Print Support</li> <li>- Right-to-Left Text</li> <li>- HTML hosting</li> <li>- Aceleração de Hardware</li> </ul>
<b>Ferramentas de desenvolvimento</b>	<ul style="list-style-type: none"> <li>- Flash Builder 4</li> </ul>
<b>Ferramentas de design</b>	<ul style="list-style-type: none"> <li>- Adobe Flash CS5</li> <li>- Adobe Catalyst CS5</li> </ul>
<b>Integração com outras ferramentas de design</b>	<ul style="list-style-type: none"> <li>- Adobe PhotoShop CS5</li> <li>- Adobe Illustrator CS5</li> <li>- Adobe Fireworks CS5</li> </ul>
<b>Linguagens baseadas em estilos</b>	<ul style="list-style-type: none"> <li>- CSS</li> </ul>
<b>Suporte a Imagens</b>	<ul style="list-style-type: none"> <li>- Todos os formatos</li> </ul>
<b>Acesso remoto</b>	<ul style="list-style-type: none"> <li>- HTTP</li> <li>- Socket</li> <li>- Web Services</li> <li>- Remoting</li> </ul>
<b>Data Services</b>	<ul style="list-style-type: none"> <li>- Flash Remoting</li> <li>- Bynary e Soap Services</li> </ul>
<b>Manipulação de vídeo</b>	<ul style="list-style-type: none"> <li>- Tipos de arquivos(FLV, F4V)</li> <li>- Tool: Adobe Media Encoder</li> <li>- Suporte a DRM</li> </ul>
<b>Distribuição de vídeo</b>	<ul style="list-style-type: none"> <li>- Adobe Flash Media Server</li> </ul>

Tabela 1: Características do Flex

Fonte: <http://www.riasoftware.com.br/blog/?p=1233>

De acordo com a tabela apresentada acima, podemos observar as várias características que o Flex possui e sua facilidade em interagir com outras ferramentas de desenvolvimento.

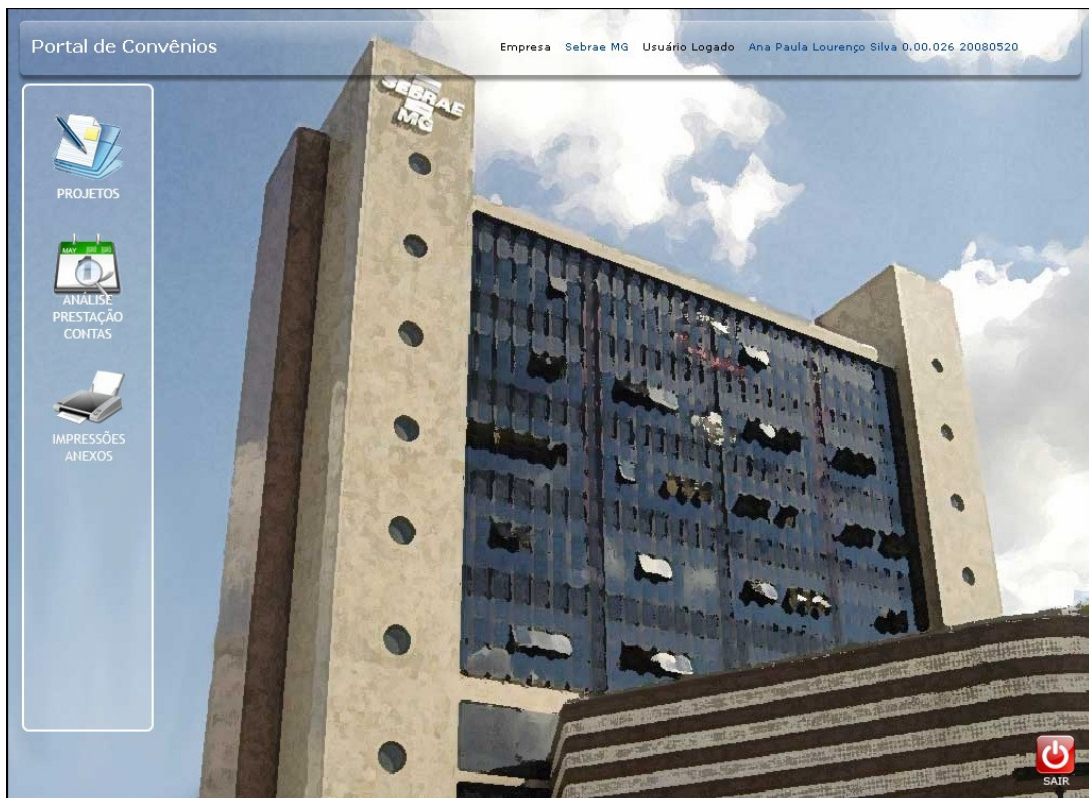
## 5. RESULTADOS

Conforme TAPPER, Jeff, LABRIOLA, Michael e BOLES, Matthew (2009), a Macromédia resolveu criar o termo *Rich Internet Application*, foi em 2002 que este termo que para aquela época parecia um pouco futurístico, tornou-se a realidade do que é hoje, no desenvolvimento de sistemas Web.

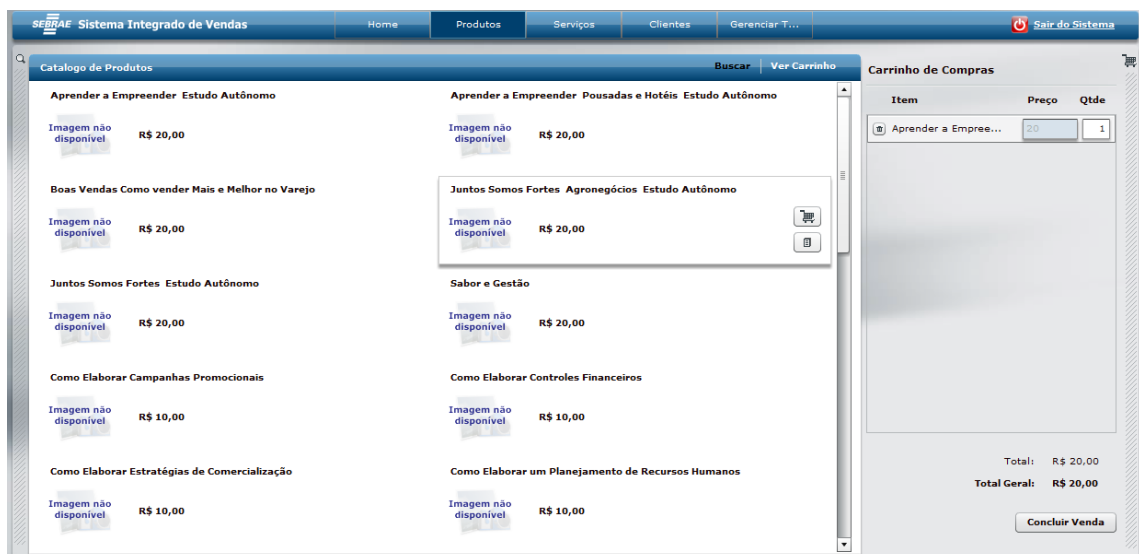
A macromédia apresentou o Flex, em 2004, para possibilitar que desenvolvedores criassem aplicações para a quase onipresente plataforma Flash. Estas aplicações podem beneficiar-se de design aperfeiçoado, usabilidade e portabilidade, mudando dramaticamente a experiência do usuário para aplicações web. Esta é a pedra fundamental do conceito de Web 2.0, uma nova geração de aplicações de Internet focadas em criatividade e colaboração. (TAPPER, Jeff, LABRIOLA, Michael e BOLES, Matthew, 2009, p.16)

Com tantas facilidades e especialidades na criação de interfaces Web, os sistemas desenvolvidos usando o Progress ou Webspeed que não tem como característica principal o desenvolvimento de telas envolvidas em recursos de design como: animação, recursos de arrastar e soltar com o mouse, grids dinâmicos, etc, hoje possuem como recurso o Adobe Flex que irá funcionar como um Front-end destas aplicações.

Várias empresas estão adotando essas duas ferramentas para complementar a criação de suas aplicações que não perderão a segurança e a robustez que a tecnologia progress proporciona e a facilidade de interação entre usuário e sistema que o Flex permite. Como exemplo, temos o Sebrae – MG, que tem como Sistema de gestão empresarial o Datasul, desenvolvido em Progress 4gl e também possui alguns projetos web que usam a linguagem progress para web, como vimos acima, chamada de WebSpeed permitindo fácil integração com os módulos do Datasul, de acordo com as figuras 11 e 12, são apresentadas as telas de dois projetos desenvolvidos pelo Sebrae MG que utilizam WebSpeed e Flex, sendo hoje cases de sucesso da empresa, o Portal de Convênios e o SIV (Sistema Integrado de Vendas).



**Figura 11:** Tela principal do portal de convênios  
**Fonte:** Sebrae - MG



**Figura 12:** Sistema Integrado de Vendas (SIV)  
**Fonte:** Sebrae - MG

## 6. CONCLUSÃO

Devido à grande exigência nos dias de hoje por programas interativos, com recursos diversos de *interface* com o usuário, que não percam a qualidade em sua usabilidade, precisamos de uma ferramenta poderosa em recursos de design, principalmente para complementar linguagens que, apesar de robustas, não tem como foco a criação de interfaces poderosos, como é o caso do Progress que vimos acima.

Devido a estes fatores, podemos concluir que o Adobe Flex, como foi apresentado, é a melhor alternativa para prover tais recursos. O Flex possui todas as características e facilidades necessárias para interagir com qualquer outra linguagem de desenvolvimento, possibilitando criar ambientes profissionais com o mínimo de esforço possível. Por possuir recursos poderosos para conexão com diversas linguagens de back-end, torna-se uma poderosa ferramenta de front-end tornando a aplicação criada ainda mais poderosa de recursos.

### 6.1. Contribuições da monografia

Esta monografia apresenta uma revisão do Adobe Flex como ferramenta de *front-end* para linguagens de *back-end*. Como contribuição adicional, porém não menos importante, através do estudo das referências bibliográficas, buscou-se desenvolver uma pesquisa para provar o uso do Flex e sua interação com o Progress.

## REFERÊNCIAS

BOAS, Anne. DEABAY, Gary. **4GL Essentils – Student Guide Progress**. Version 1.1. USA. 2001, 940 p.

COSTA, Marcio. **Dominando o Progress**. São Paulo: Ano, 2000. Disponível em: <<http://www.geocities.com/marcio-brener>> Acesso em: 01 de Agosto 2010.

PROGRESS, WebSpeed WorkShop, 2011. Disponível em: <<http://web.progress.com/en/openedge/webspeed-worshop.html> >. Acesso em 10 fev. 2011.

SCHMITZ, Daniel. **Desenvolvendo Sistemas com Flex e PHP**. 1º Edição. São Paulo: Novatec. 2009, 294 p.

SILVA, NAYARA, OLIVEIRA, FÁBIO, *Entendendo WebSpeed*, 2010. Disponível em: <[http://www.devmedia.com.br/articles/viewcomp\\_forprint.asp?comp=7848](http://www.devmedia.com.br/articles/viewcomp_forprint.asp?comp=7848)>. Acesso em 16 fev. 2011.

TAPPER, Jeff. LABRIOLA, Michael e BOLES, Matthew. **Adobe Flex 3 Treinamento Direto da Fonte**. 1 Edição. São Paulo: Alta Books. 2009, 554 p.

WENGER, Robson, *Características do Adobe Flex 4*, 2010. Disponível em: <<http://www.riasoftware.com.br/blog/?p=1233> >. Acesso em 16 fev. 2011.

WIKIPÉDIA. **Progress 4GL**. Disponível em: <<http://pt.wikipedia.org/wiki/Ultra-sonografia>>. Acesso em 01 de Agosto 2010.

This document was created with Win2PDF available at <http://www.win2pdf.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.  
This page will not be added after purchasing Win2PDF.