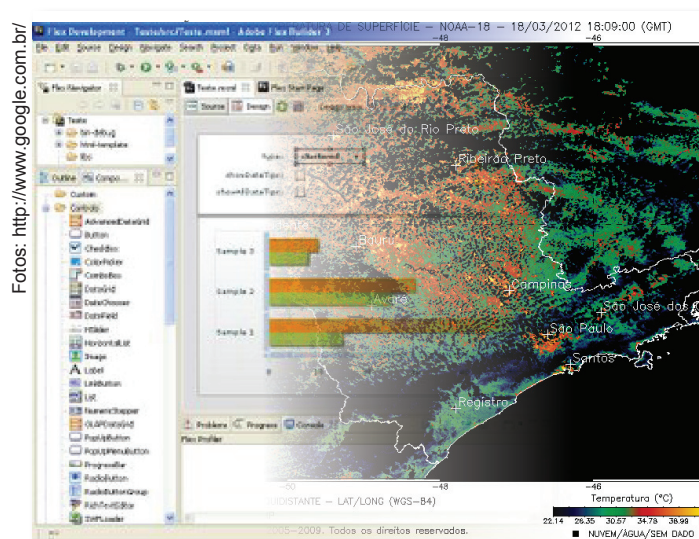


# Comunicado 110

## Técnico

ISSN 1677-8464  
Dezembro, 2011  
Campinas, SP



## Desenvolvendo aplicações web mais interativas com o Flex Builder

Gabriel Garcia do Nascimento<sup>1</sup>  
Luciana Alvim Santos Romani<sup>2</sup>  
Adriano Franzoni Otavian<sup>3</sup>

O Agritempo é um sistema web de monitoramento agrometeorológico que fornece informações para o usuário por meio de mapas e gráficos (COLTRI et al., 2007; EVANGELISTA et al., 2003). Em sua nova versão, o Agritempo pretende prover mais flexibilidade e autonomia para seus usuários. Dessa forma, está sendo desenvolvida uma nova ferramenta para geração de gráficos mais flexíveis, integrados à base de dados do sistema.

Diversas soluções foram analisadas, como a biblioteca *javascript* do Dojo (DOJO, 2011) que embora esteja sendo utilizada no desenvolvimento do sistema Agritempo, seu uso na geração de gráficos não se mostrou adequado por possuir uma interface menos flexível com codificação complexa. Com isso, buscou-se outra alternativa e a solução adotada foi a ferramenta *Flex Builder*.

A usabilidade da interface do software de desenvolvimento, seus gráficos com design moderno e flexível, somados à linguagem declarativa e fácil de usar, bem como a existência de uma comunidade grande de desenvolvedores e total compatibilidade, tanto com os

computadores clientes como o servidor, fizeram do Flex a melhor escolha para o projeto Agritempo. Este artigo tem por objetivo apresentar a solução adotada de uso da ferramenta *Flex Builder* no desenvolvimento de uma interface mais interativa para manipulação de gráficos.

O restante deste artigo está organizado da seguinte forma: a próxima seção apresenta as características básicas da ferramenta *Flex Builder*, a seção 2 descreve o uso da ferramenta *Flex Builder* no sistema Agritempo e, finalmente, são apresentadas as conclusões.

### O Adobe Flex Builder

O *Adobe Flex Builder* é um *framework* de desenvolvimento baseado no Eclipse, disponível para MacOS e Windows, que permite a codificação e edição para *Macromedia eXtensible Markup Language* (MXML) e *ActionScript*, além de possibilitar a projeção visual do *layout* de interface, aparência e comportamento dos

<sup>1</sup> Estagiário da Embrapa Informática Agropecuária, Campinas, SP, [mjga@terra.com.br](mailto:mjga@terra.com.br)

<sup>2</sup> Doutora em Ciência da Computação, Pesquisadora da Embrapa Informática Agropecuária, Campinas, SP, [luciana@cnptia.embrapa.br](mailto:luciana@cnptia.embrapa.br)

<sup>3</sup> Bacharel em Engenharia de Computação, Analista da Embrapa Informática Agropecuária, Campinas, SP, [adriano@cnptia.embrapa.br](mailto:adriano@cnptia.embrapa.br)

componentes (ADOBE..., 2010). Utiliza uma ampla biblioteca de componentes incorporados que podem ser modificados ou mesmo criados de acordo com a necessidade da aplicação.

O MXML é uma linguagem de marcação baseada em XML que permite dispor declarativamente a interface de aplicativos. A linguagem *ActionScript* é responsável pela parte lógica e funcional do aplicativo, ou seja, seus eventos e procedimentos. As duas linguagens trabalham em conjunto para criar aplicações mais interativas.

O *Flash Player* é necessário no navegador para visualizar a aplicação gerada em MXML. Ele funciona como uma máquina virtual encarregada de interpretar o objeto binário do flash, o *ShockWave Flash* (SWF). A compilação do MXML para o SWF é feita por meio do próprio *Flex Builder*. Isso facilita a execução da aplicação, uma vez que o *plugin* do Flash é leve e compatível com praticamente todos os navegadores no mercado. Além disso, a linguagem é rodada em uma máquina virtual, o que torna a aplicação portátil para qualquer sistema operacional. A Figura 1 ilustra o funcionamento do *Flex Builder*.

O grande diferencial do *Flex Builder* está no modo de exibição de interface. Nela é possível modelar

visualmente o *layout* da aplicação, criar novos objetos como botões, *canvas*, adicionar imagens, gráficos, *datagrids*, caixas de texto, e ainda editar os eventos de cada elemento. Todas essas funcionalidades podem ser usadas sem que haja a necessidade de editar o código fonte do programa, uma vez que o *Flex* se encarrega de sua geração conforme a interface é modificada – “What You See Is What You Get” (WYSIWYG). Isso torna a experiência com o programador mais confortável, pois parte do código é gerada diretamente pelo *Flex Builder* com somente alguns cliques do *mouse*. Editar o código é necessário apenas quando algum tipo de alteração fina no programa for necessária, como definir uma função que é chamada de acordo com um evento de interface. A Figura 2 mostra uma tela com o ambiente de desenvolvimento da ferramenta *Flex Builder*.

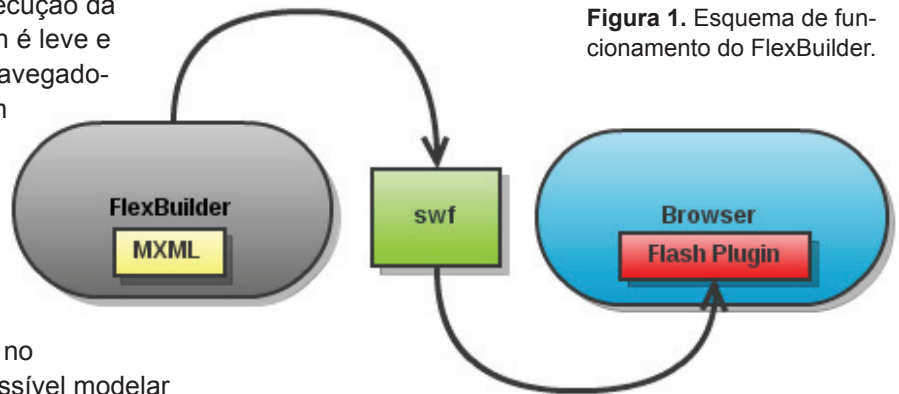


Figura 1. Esquema de funcionamento do FlexBuilder.

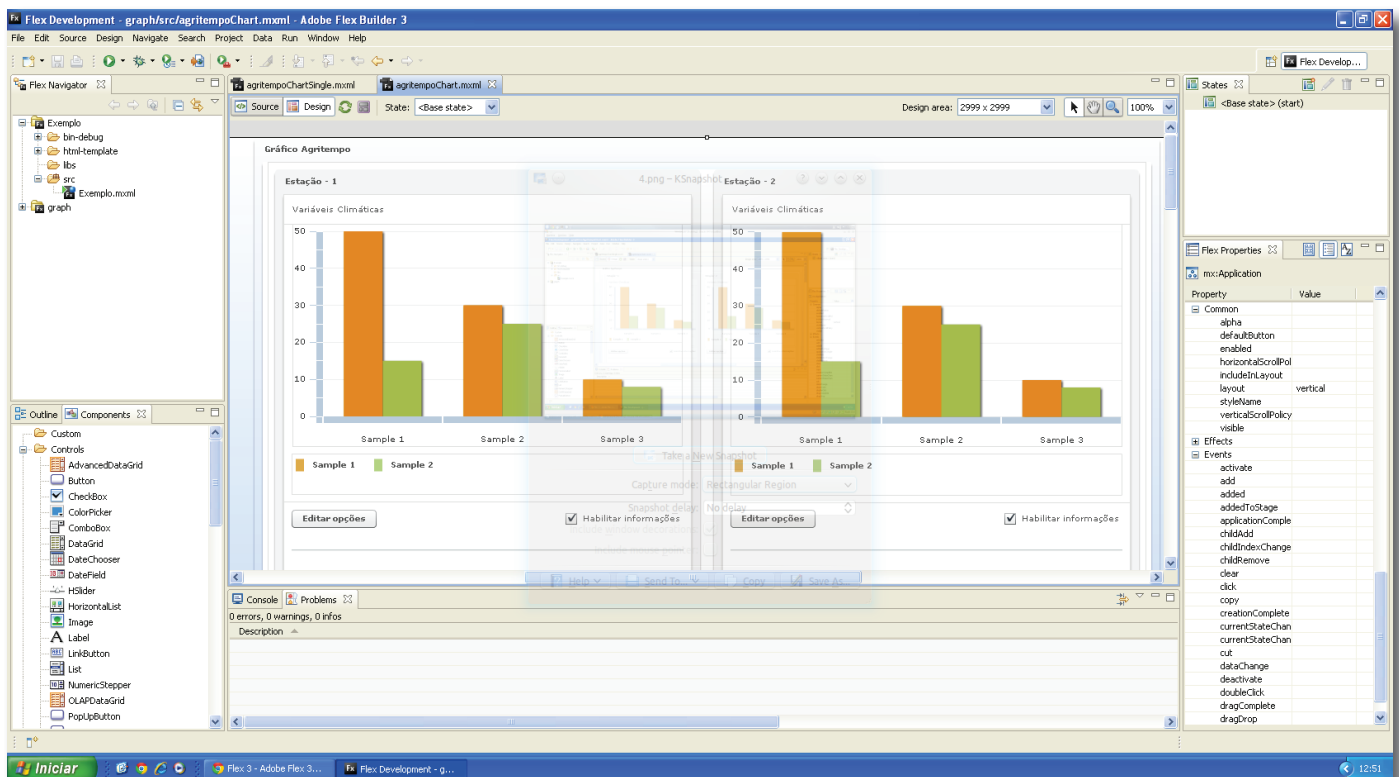


Figura 2. Ambiente de desenvolvimento do FlexBuilder.

De forma geral, a ferramenta *FlexBuilder* tem muitas vantagens como design intuitivo e o fato de permitir tratar as aplicações em desenvolvimento da maneira mais simples possível por meio de seu editor gráfico no estilo WYSIWYG. Além disso, possui uma grande comunidade de desenvolvedores, manuais e tutoriais na internet.

Outra vantagem, é o fato de a ferramenta manter conceitos do *Adobe Flash*, um dos principais programas responsáveis por criar animações na web, o que torna o *Flex Builder* muito fácil de ser manuseado por usuários que já tenham experiência com essa outra ferramenta. No entanto, não prende o desenvolvedor no conceito de timeline, e o deixa livre para criar não somente animações, mas todo e qualquer tipo de aplicação web. O *FlexBuilder* cria tanto aplicações web como desktop, porém seu principal foco é o desenvolvimento on-line.

A integração com outras ferramentas, tanto dentro como fora da Adobe é total, isso porque o *FlexBuilder* faz uso de diversas bibliotecas que tratam da conversão entre linguagens sem que isso afete o desempenho de nenhum dos lados.

Uma das desvantagens da ferramenta é a licença profissional individual, que pode chegar a custar US\$ 800,00, o que a torna viável somente para grandes empresas, e não para uso doméstico.

## Uso do *FlexBuilder* no projeto Agritempo

Tendo sido desenvolvido em 2001, o Agritempo não segue o novo paradigma que envolve a internet, denominado de Web 2.0, onde espera-se mais interatividade para os sistemas permitindo que os usuários tenham mais flexibilidade de uso. Com o objetivo de tornar a interface de geração e manipulação de gráficos do sistema Agritempo mais flexível, foram analisadas diversas soluções, como a biblioteca *javascript* do Dojo e o *Flex Builder*. Embora a biblioteca Dojo esteja sendo utilizada no desenvolvimento

do sistema e também suporte a geração de gráficos, a interface gerada com Dojo é menos flexível, tem uma codificação ainda complexa e visualmente é inferior à interface gerada com o *Flex Builder*.

Diante disso, a solução adotada foi usar a ferramenta *Flex Builder*. A usabilidade da interface do software de desenvolvimento, seus gráficos com *design* moderno e flexível, somados à linguagem declarativa fácil de usar, bem como a existência de uma comunidade grande de desenvolvedores e total compatibilidade tanto com os computadores clientes como o servidor, fizeram do Flex a melhor escolha para o projeto Agritempo.

O grande desafio foi implantar o componente dos gráficos no projeto, alterando o lado servidor o mínimo possível, ou seja, aproveitando ao máximo as funções já existentes, sem impactar nas demais funcionalidades. A aplicação do lado servidor foi escrita em linguagem de programação orientada a objeto, Java, onde houve a necessidade de utilizar *JavaScript Object Notation* para fazer (JSON) a comunicação entre os componentes Javascript, Dojo, do lado cliente, que foi a solução encontrada para a implementação dos gráficos.

No contexto do Agritempo, o componente dos gráficos elaborado no *Flex Builder*, requisita ao servidor um objeto JSON do tipo 'ClimaDiario', porém, para que seja utilizado em linguagem MXML, o objeto JSON tem de ser decodificado para XML, que é o formato padrão de entrada de dados da linguagem MXML. A biblioteca 'corelib' do *Flex* possui todas as funções de codificação e decodificação de objetos JSON, sendo necessário apenas importá-la e utilizar o método *JSONDecoder* para que o objeto seja convertido para XML. Uma ilustração desse esquema é apresentada na Figura 3.

O módulo do gráfico foi gerado com rapidez, leveza e *design* moderno e flexível. O modo de edição visual do *Flex Builder*, somado à sua vasta biblioteca de funções, reduziu muito o número de alterações no sistema Agritempo. Foi necessária apenas a implementação do

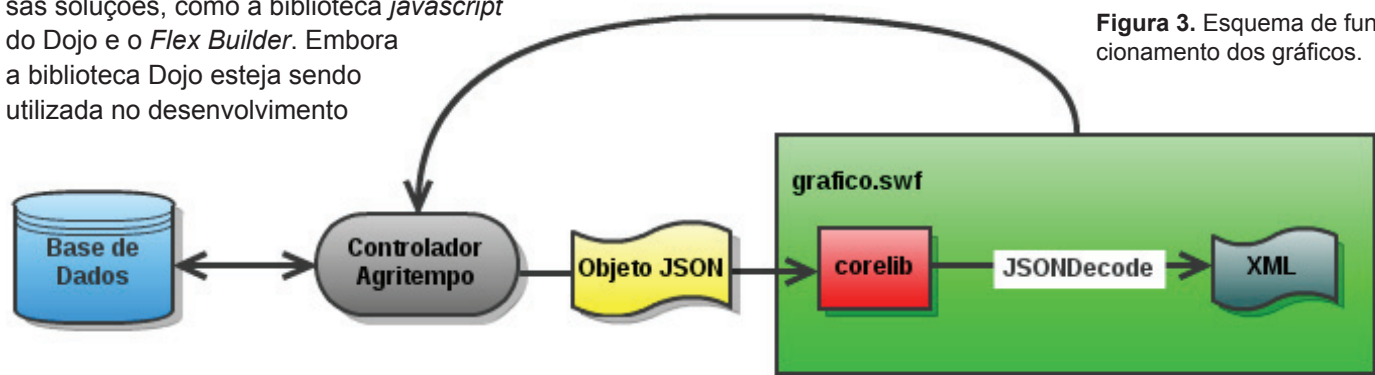


Figura 3. Esquema de funcionamento dos gráficos.

controlador para selecionar as estações a serem analisadas nos gráficos. Os objetos que são requisitados e enviados pelo servidor permaneceram os mesmos.

## Exemplo de codificação para o sistema Agritempo

A vasta biblioteca do *Flex* (TOUR DE FLEX, 2010) permite a criação de diversos elementos, entre eles os menus interativos, a visualização de imagens, a organização por *containers* e *canvas*, e, até mesmo, os elementos mais complexos como *datagrids* e gráficos dinâmicos. Para gerar um gráfico é necessário declarar uma aplicação, em seguida o tipo de gráfico a ser gerado, com suas respectivas séries de dados:

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="vertical"
verticalAlign="top"
horizontalAlign="center"
backgroundColor="#FFFFFF" width="1110" height="925">

  <mx:AreaChart id="graficodearea" height="270" width="473">
    <mx:series>
      <mx:AreaSeries id="areal" yField="y"
displayName="Area 1" form="curve"/>
      <mx:AreaSeries id="area2" yField="y"
displayName="Area 2" form="curve"/>
    </mx:series>
  </mx:AreaChart>

</mx:Application>
```

Aplicações como gráficos e *datagrids* necessitam de um *datasource* externo que, por padrão, deve ser um XML. No entanto, é possível também usar outro formato, desde que este seja convertido para XML antes de ser utilizado. Por exemplo, a função *ActionScript* "onJSONResult()" recebe um objeto serializável JSON do servidor e o decodifica para XML para servir de *datasource* para o gráfico "graficodearea", como exemplificado no trecho de código a seguir:

```
private function onJSONResult(
event:ResultEvent ) : void {
  //Declara as variaveis
  var data:String = event.result.toString();

  //Converte de JSON para XML
  var jd:JSONDecoder = new JSONDecoder(data);

  //Define os Data Sources
  graficodearea.dataProvider = jd.getValue();
}
```

O objeto JSON antes da conversão para o XML deve seguir um formato padrão. No exemplo a seguir o objeto descreve 3 dados 'y' com 2 séries 'area' em cada um:

```
{
  {
    "y":1,
    "areal": 25,
    "area2":10
  },
  {
    "y":2,
    "areal": 24,
    "area2": 90
  },
  {
    "y":3,
    "areal": 30,
    "area2": 30
  },
}
```

Após a conversão em XML o formato esperado deve ser:

```
<data>
  <result>
    <y>1</y>
    <areal>25</areal>
    <area2>10</area2>
  </result>
  <result>
    <y>2</y>
    <areal>24</areal>
    <area2>90</area2>
  </result>
  <y>3</y>
  <areal>30</areal>
  <area2>30</area2>
</result>
</data>
```

A Figura 4 exemplifica um gráfico gerado seguindo a estrutura proposta no sistema Agritempo e descrita anteriormente.

## Conclusão

A adoção de uma ferramenta moderna como o Flex Builder permite que o redesign do sistema Agritempo possa ter uma interface mais próxima do conceito web 2.0. Esse novo modelo de interface mais interativa permite mais flexibilidade e deixa o poder sobre a interação nas mãos do usuário durante sua interação com o sistema. Dessa forma, o Agritempo consolida-se como

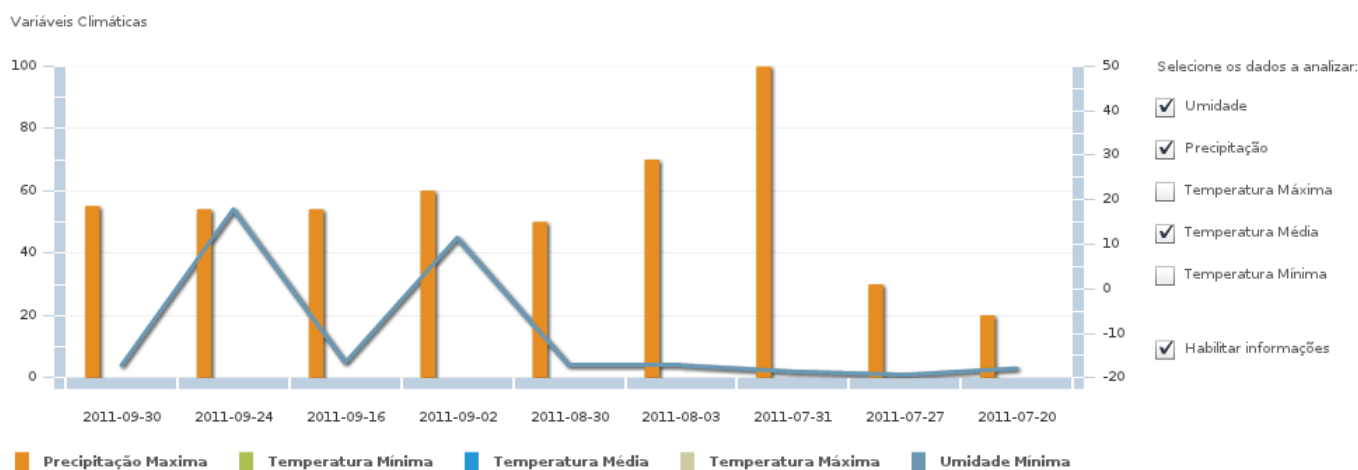


Figura 4. Exemplo de gráfico gerado no *Flex Builder*.

um sistema de monitoramento agrometeorológico que objetiva atingir, de forma adequada, uma vasta gama de clientes.

## Referências

ADOBE flash Builder 4.5 Premium. 2010. Disponível em: <<http://www.adobe.com/products/flash-builder.html>>. Acesso em: 21 out. 2011.

COLTRI, P. P.; PINTO, H. S.; FERREIRA, N. J.; CECCARELLI, M.; CORAL, G. Sistema de monitoramento e previsão agrometeorológica: Agritempo. In: CONGRESSO BRASILEIRO DE AGROMETEOROLOGIA, 15., 2007, Aracaju. **Anais...** Aracaju: Sociedade Brasileira de Agrometeorologia, 2007. 2007. 5 p.1 CD-ROM.

EVANGELISTA, S. R. M.; TERNES, S.; SANTOS, E. H. dos; ASSAD, E. D.; ROMANI, L. A. S.; OTAVIAN, A. F. Agroclima sistema de monitoramento agroclimatológico.. In: CONGRESSO BRASILEIRO DE AGROMETEOROLOGIA, 2003, Santa Maria. **Anais...** Santa Maria: Unifra, SBA, UFSM, 2003. v. 1. p. 603-604.

TOUR de flex. 2010. Disponível em: <<http://www.adobe.com/devnet/flex/tourdeflex.html>>. Acesso em: 21 out. 2011.

### Comunicado Técnico, 110

**Embrapa Informática Agropecuária**  
 Endereço: Caixa Postal 6041 - Barão Geraldo  
 13083-886 - Campinas, SP  
 Fone: (19) 3211-5700  
 Fax: (19) 3211-5754  
<http://www.cnptia.embrapa.br>  
 e-mail: [sac@cnptia.embrapa.com.br](mailto:sac@cnptia.embrapa.com.br)

**Embrapa**

Ministério da  
 Agricultura, Pecuária  
 e Abastecimento

GOVERNO FEDERAL  
**BRASIL**  
 PAÍS RICO E PAÍS SEM POBREZA

1ª edição on-line - 2011

Todos os direitos reservados.

### Comitê de Publicações

**Presidente:** *Sílvia Maria Fonseca Silveira Massruhá*  
**Membros:** *Poliana Fernanda Giachetto, Roberto Hiroshi Higa, Stanley Robson de Medeiros Oliveira, Maria Goretti Gurgel Praxedes, Neide Makiko Furukawa, Adriana Farah Gonzalez, Carla Cristiane Osawa (secretária)*  
**Suplentes:** *Alexandre de Castro, Fernando Attique Máximo, Paula Regina Kuser Falcão*

### Expediente

**Supervisão editorial:** *Stanley Robson de Medeiros Oliveira, Neide Makiko Furukawa*  
**Normalização bibliográfica:** *Maria Goretti Gurgel Praxedes*  
**Revisão de texto:** *Adriana Farah Gonzalez*  
**Editoração eletrônica:** *Neide Makiko Furukawa*