

## Desenvolvimento de *portlets* para a plataforma Liferay Portal

Bruno Luis Silveira Andretta<sup>1</sup>

Glauber José Vaz<sup>2</sup>

A plataforma Liferay Portal vem sendo amplamente utilizada para a construção de portais. A Embrapa adotou a tecnologia para a implantação de seu novo portal, de sua nova Intranet (que ainda será lançada) e para outras iniciativas, como a Agropedia brasilis, uma plataforma para a construção de ambientes virtuais voltados aos grupos de pesquisa, desenvolvimento e inovação da empresa. Um portal é projetado para ser um ambiente único baseado na web, a partir do qual todas as aplicações de um usuário possam ser executadas. Os *portlets* são essas aplicações que rodam em uma porção de uma página web e são integradas de maneira sistemática e consistente. Assim, um portal pode ser considerado um *container* de *portlets*, cuja função é administrar as páginas do portal e agregar o conjunto de *portlets* que aparecem em cada página (SEZOV JUNIOR, 2012). Apesar de o Liferay Portal oferecer inúmeros *portlets* já prontos, normalmente, eles não atendem adequadamente às demandas dos usuários. Assim, é necessário um grande esforço para desenvolver novos *portlets* e customizar os já existentes, uma vez que essa tarefa não requer apenas um bom conhecimento de *portlets*, mas também uma maior compreensão sobre como essa tecnologia pode ser usada com outras para desenvolver portais web no mundo real (SARIN, 2012). Este trabalho apresenta o conjunto dos principais conceitos, linguagens e tecnologias para o adequado desenvolvimento de *portlets* para Liferay Portal, a fim de tornar mais clara a dimensão da complexidade desta tarefa. Também sugere um roteiro de estudo para aqueles que estão começando a atuar nesta área. Para que os *portlets* sejam facilmente adaptáveis e de rápida manutenção, é recomendada a utilização do *Model-View-Controller* (MVC), um padrão de

---

<sup>1</sup> Universidade Estadual de Campinas (Unicamp)

<sup>2</sup> Embrapa Informática Agropecuária

projetos que possibilita uma maneira fácil de se estruturar uma aplicação web dividindo-a em três partes conectadas:

- Model*, encarregado pelos dados da aplicação e as regras de negócio que permitem a manipulação desses dados;
- View*, responsável por apresentar os dados aos usuários;
- Controller*, que faz a interação entre as duas outras partes: recebe informações oriundas da *View* e requisita processamento para o *Model*, e passa os dados manipulados pelo *Model* para a apresentação na *View*.

Em Java, um portlet deve ser uma classe que implementa, direta ou indiretamente, a interface `javax.portlet.Portlet`. A Portlet API ainda tem uma classe abstrata `GenericPortlet` que implementa a interface `Portlet` e fornece implementações padrões para seus métodos. Os desenvolvedores normalmente criam *portlets* a partir de `GenericPortlet` sobrescrevendo seus métodos. No entanto, aqueles que utilizam Liferay Portal podem utilizar o *framework* MVC da Liferay, o `MVCPortlet`, que é uma extensão de `GenericPortlet` e torna o desenvolvimento de *portlets* mais simples e mais rápido. A Figura 1 representa as relações de herança entre a interface `Portlet`, `GenericPortlet` e `MVCPortlet`.

Para se desenvolver *portlets* adotando boas práticas de programação, é necessário conhecer muito mais do que o *framework* MVC da Liferay. A Figura 2 apresenta o que é essencial para esse desenvolvimento mais

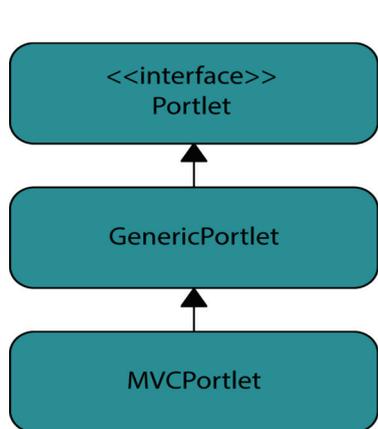


Figura 1. Implementação do MVC-Portlet.

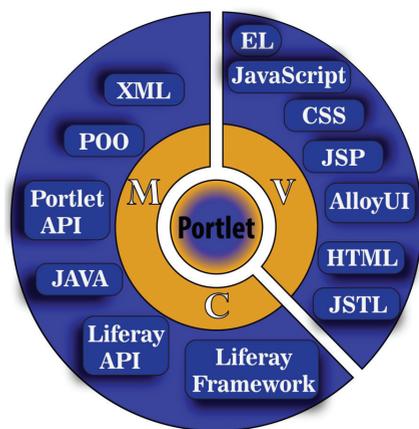


Figura 2. Tecnologias usadas no desenvolvimento de *portlets*.

adequado de *portlets*, conforme cada parte do MVC. O *framework* MVC da plataforma Liferay é baseado em Java, que, por sua vez, é uma linguagem de programação orientada a objetos (POO). Portanto, é fundamental o conhecimento de POO para a utilização desta linguagem que é a tecnologia base para o *Model* e o *Controller* dos *portlets*. Além disso, para desenvolver *portlets* específicos para a plataforma Liferay Portal, são utilizadas várias APIs da própria Liferay, que fornecem ao desenvolvedor flexibilidade e acesso a serviços exclusivos e usam dados de entidades específicas do Liferay Portal. Como estamos tratando de aplicações Web, o desenvolvedor também precisa conhecer Java Server Pages (JSP) para a geração dinâmica de conteúdo, funcionando portanto como um componente *View* da aplicação. Os JSPs normalmente entregam para a exibição das páginas web documentos HTML, que, com CSS e JavaScript formam o conjunto de linguagens básicas para a web. Enquanto *Cascading Style Sheets* (CSS) facilita a formatação e a padronização de páginas web, JavaScript possibilita programar o comportamento dessas páginas. Assim, temos alguns dos elementos essenciais para o desenvolvimento de *portlets*. A Figura 3 mostra um roteiro para quem pretende trabalhar nessa área. A porção superior da Figura 3 apresenta a necessidade de conhecer os conceitos de POO para que seja possível entender Java e JSP. A porção inferior da Figura 3 mostra que o uso adequado do Liferay MVC *framework* depende da compreensão dos conceitos envolvidos em MVC e em *portlets*. Além da Portlet API, a Liferay API também é essencial para o desenvolvimento na plataforma Liferay Portal.

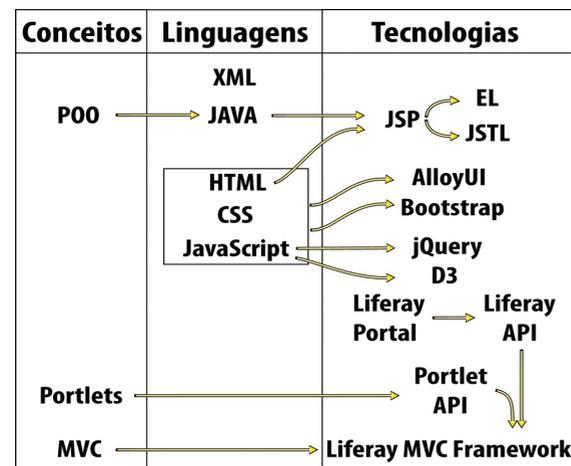


Figura 3. Roteiro para o desenvolvimento de *portlets* no Liferay Portal.

O MVC auxilia na adoção de boas práticas de programação. No entanto, para que as partes *Model*, *View* e *Controller* fiquem efetivamente separadas, há outros recursos que precisam ser utilizados, como a *JSP Standard Tag Library* (JSTL) e a *Expression Language* (EL), muito importantes na separação entre as páginas JSP, responsáveis pela *View*, e os códigos Java, inerentes a *Model* e *Controller*.

À medida que se deseja melhorar a apresentação das aplicações, novos recursos podem ser utilizados. As bibliotecas AlloyUI, por exemplo, auxiliam na criação de formulários, tabelas de dados, botões, visualizadores de imagens e uma série de outros componentes. Ao utilizar estas bibliotecas, os componentes são gerados de forma completa, já com CSS e JavaScript implementados, facilitando muito na criação dos *portlet*s (SEZOV JUNIOR, 2012). Bibliotecas JavaScript, como D3 e jQuery, e *frameworks*, como o Bootstrap, também podem ajudar muito na construção de novos *portlet*s.

Por fim, o eXtensible Markup Language (XML) é capaz de descrever documentos com dados organizados e de forma hierárquica. No contexto de desenvolvimento de *portlet*s para Liferay Portal, esta linguagem é utilizada para estabelecer uma grande variedade de configurações.

Portanto, o tempo necessário para se começar a desenvolver *portlet*s pode ser bem elevado, dependendo do conhecimento e da experiência do desenvolvedor. O presente trabalho mostra que o desenvolvimento de *portlet*s requer o conhecimento de muitos conceitos, linguagens e tecnologias, e apresenta um roteiro que auxilia quem pretende trabalhar nessa área.

Palavras-chave: Portal, *portlet*, MVC, desenvolvimento web.

## Referências

SARIN, A. **Portlets in action**. Shelter Island: Manning, 2012. 612 p.

SEZOV JUNIOR, R. **Liferay in action: the official guide to Liferay Portal development**. Shelter Island: Manning, 2012. 351 p.