

Uma plataforma livre para padronização do desenvolvimento de sistemas no Governo Federal

Antonio Carlos Tiboni, Flávio Gomes da Silva Lisboa, Luciana Campos Mota

CETEC Curitiba - Serviço Federal de Processamento de Dados
CEP 80520-170 – Curitiba – PR – Brazil

{antonio.tiboni, flavio.lisboa, luciana.mota}@serpro.gov.br

***Abstract.** This article presents the Demoiselle Framework project, a platform for Java development that aims to be the standard for all new development hires for the Brazilian federal government. It will ensure interoperability and ease the maintenance of systems in several ministries and local authorities through the adoption of open standards. Built as a free software, it will be maintained in community, benefiting the whole Brazilian society, which can take part in the technological directions of the government as any person on the planet, interested in helping and improving the tool.*

***Resumo.** Este artigo apresenta o projeto do Demoiselle Framework, uma plataforma de desenvolvimento em Java que visa ser o padrão para todas as novas contratações de desenvolvimento de programas do governo federal. Ela garantirá interoperabilidade e facilidade de manutenção dos sistemas de diferentes ministérios e autarquias graças a adoção de padrões abertos. Construída como software livre, será mantida em comunidade, beneficiando a toda a sociedade brasileira, que poderá participar do direcionamento tecnológico do governo, bem como de qualquer indivíduo do planeta, interessado em contribuir e melhorar a ferramenta.*

1. Introdução

A ampliação do uso de software livre pelo governo federal surgiu de uma necessidade de segurança ampliada, ao utilizar softwares nacionais, da não dependência tecnológica, da redução de custos com pagamento de licenças e, em contrapartida, investimento na inteligência nacional. Daí verificou-se a necessidade da criação de um produto que permitisse o reuso de estruturas pré-existentes, que facilitasse a manutenção das aplicações e, principalmente, que permitisse a padronização dos códigos.

O framework Demoiselle para Java é uma ferramenta de código-aberto e totalmente livre, que visa garantir a interoperabilidade e facilidade de manutenção dos sistemas das diferentes instituições do governo federal. A padronização é o cerne do Demoiselle, pois visa facilitar a integração dos sistemas. A ideia é que, a partir de um framework e de uma arquitetura de referência, um conjunto de requisitos gere uma aplicação que possa ser mantida por qualquer um que conheça os dois primeiros.

Como framework integrador, o Demoiselle constitui-se de uma camada de alto nível denominada Framework Arquitetural, que estabelece interfaces padronizadas para serem usadas pelas aplicações. Nas camadas inferiores encontram-se os frameworks de base e de fundação, amplamente utilizados pelo mercado, e para a aplicação, as mudanças de infraestrutura são completamente transparentes. A ferramenta foi

construída sob as premissas de ser extensível, fácil de usar, estável, configurável, confiável e ter sua documentação publicada.

O Demoiselle pode oferecer uma maior produtividade, redução na curva do aprendizado, bem como a simplificação dos processos. O framework integrador será mantido em comunidade, totalmente aberto e compartilhado, o que permitirá que diferentes entidades e instituições contribuam e sejam beneficiadas pelo reuso das estruturas existentes.

O desenvolvimento orientado a componentes com ciclo de vida independente do framework arquitetural permite que as aplicações não fiquem dependentes dos módulos, que podem ser construídos colaborativamente.

2. Objetivos

Todos os sistemas especialmente desenvolvidos para órgãos federais deverão utilizar a plataforma Demoiselle. As linguagens e outros aspectos que balizaram a elaboração da plataforma são constituídas de padrões, o que facilita sua adoção, não só por parte dos responsáveis pelas áreas de TI nos órgãos federais, como também pelas próprias empresas fornecedoras de soluções para o Governo.

A implementação do Demoiselle pretende automatizar e acelerar a integração de sistemas, aumentar a produtividade e eliminar o retrabalho. Um dos benefícios esperados é a economia financeira, pois não há necessidade de arcar com custos de licenças de software. Outro benefício é o de qualquer pessoa poder baixar a plataforma e usar livremente, fato este que incentiva a utilização de software livre em todas as esferas da sociedade, quer sejam empresas de Governo, quer sejam empresas privadas.

3. Arquitetura

O modelo arquitetural do Demoiselle é apresentado na figura 1. Os componentes na verdade não fazem parte do *framework* arquitetural, pois possuem um ciclo de vida independente. Por este fato, eles não geram dependência obrigatória nas aplicações utilizadoras e podem ser construídos colaborativamente.

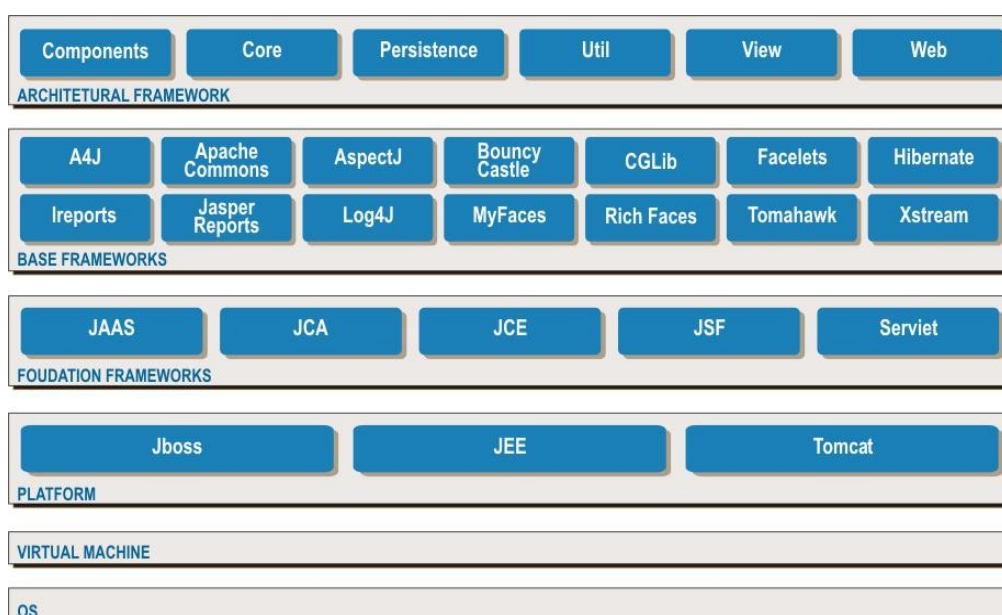


Figura 1. Arquitetura do Demoiselle Framework

A arquitetura de referência proposta para o Demoiselle é baseada em camadas. Além das camadas clássicas do modelo MVC (Modelo, Visão e Controlador), elas se distinguem como camadas de persistência, transação, segurança, injeção de dependência e mensagem.

3.1. Módulos

- Core: conjunto de especificações que dão base estrutural ao framework possibilitando padronização, extensão e integração entre as camadas das aplicações nele baseadas.
- Persistence: realiza a integração do sistema com outros sistemas gerenciadores de dados garantindo eficiência para recuperar, armazenar e tratar informações.
- Util: contém componentes utilitários que facilitam o trabalho de outras funcionalidades do framework e seus módulos lógicos.
- View: contém implementações de componentes específicos de interface com usuário baseados na especificação JSF.
- Web: implementação do módulo lógico Core para aplicações Web (J2EE), provê utilitários comuns de aplicações web que facilitam tratamento de sessões de usuário e suas requisições.

3.2. Contextos

Enquanto os módulos dentro da arquitetura podem ser vistos como camadas verticais, existem camadas transversais, que são os contextos. Os contextos referem-se a características cuja operação transcende as camadas verticais, como transações, troca de mensagens, segurança da aplicação e aspectos que devem ser injetados no código. A figura 2 ilustra o relacionamento do contexto com a implementação de MVC do Demoiselle.

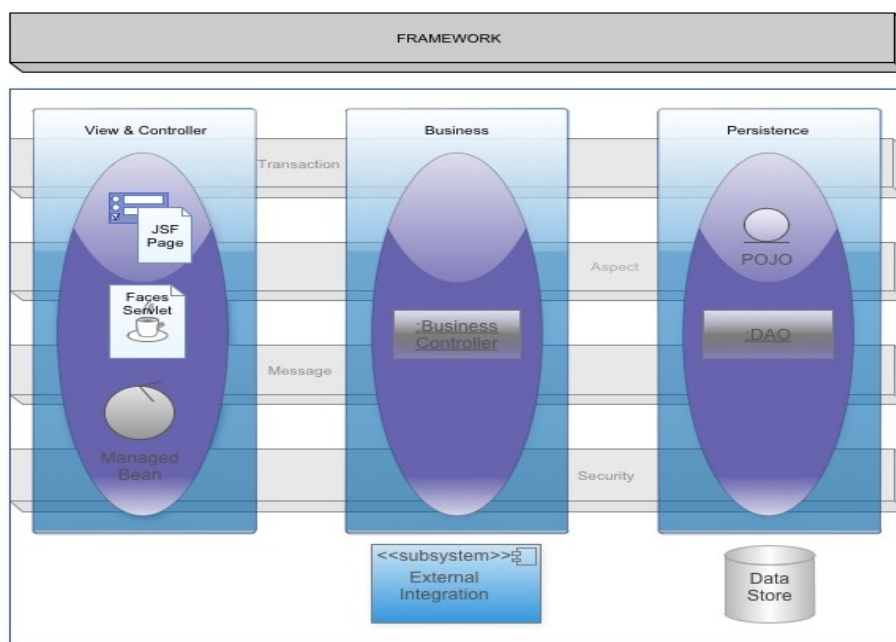


Figura 2. Camadas Verticais e Transversais do Demoiselle

3.3. Integração entre Camadas

Entre os aspectos a destacar sobre a arquitetura, o mais significativo é a utilização de padrões de projeto tais como Factory, Proxy, IoC e injeção de dependências para manter a integração de camadas em um nível de acoplamento baixo afim de garantir uma melhor manutenção e escrita/legibilidade das classes representantes de cada camada.

O mecanismo de integração entre camadas atua na camada de visão injetando objetos de negócio através de uma fábrica do próprio framework ou alguma fábrica definida pela aplicação. Esta fábrica poderá utilizar um proxy, do framework ou da aplicação, para a instanciação do objeto de negócio.

O mecanismo de integração entre camadas atua também na camada de regras de negócio injetando objetos de persistência através de uma fábrica do próprio framework. Esta fábrica poderá utilizar um proxy, do framework ou da aplicação, para a instanciação do objeto de persistência.

3.4. Desenvolvimento Baseado em Componentes

Os módulos citados no item 3.1 constituem o núcleo do framework. A ele podem ser acoplados componentes desenvolvidos por terceiros, desde que usem a interface definida pelo Demoiselle.

Essa arquitetura permite resolver um paradoxo do desenvolvimento de software. Padrões são necessários, para permitir a manutenção e promover o reuso. Padrões são coisas que não devem mudar (com facilidade). Por outro lado, a necessidade exige inovação, e a inovação geralmente provoca a quebra de padrões.

Dessa forma, o Demoiselle Framework faz uso de interfaces de classe em seu núcleo, as quais ditam o padrão de codificação para as aplicações. Por outro lado, a possibilidade de criar componentes e conectá-los ao Demoiselle dá ao desenvolvedor uma grande liberdade, pois ele pode criar qualquer funcionalidade para uma aplicação que use o framework, inclusive uma versão alternativa de uma funcionalidade já existente.

Os componentes tem ciclo de vida independente, não geram dependência obrigatória nas aplicações geradas e podem ser construídos colaborativamente. A facilidade de acoplar e desacoplar componentes, aliada com a padronização permite superar um problema do software desenvolvido em comunidade: o prazo de entrega. A componentização permite que um usuário crie e utilize uma nova funcionalidade para o framework sem ter de esperar que a mesma seja aprovada e incorporada pela comunidade. Mas uma vez que isso ocorra, ele pode trocar sua versão pessoal por aquela testada e melhorada pela comunidade.

4. Inovação e Ineditismo

Demoiselle é disponibilizado sob a licença LGPL 3 e qualquer componente utilizado ou desenvolvido para ele dever ser compatível com essa licença.

4.1. Comunidade

Apesar de ter sido feito para atender ao Governo Federal, o Demoiselle é um software livre, e o seu desenvolvimento é feito de forma colaborativa. Sendo assim, qualquer pessoa pode participar de seu desenvolvimento, bem como na evolução do

framework, quer seja reportando erros, efetuando melhorias de código ou ajudando na confecção da documentação.

4.2. Oportunidades de Negócio

À primeira vista parece ser mais um framework Java, o que pode levantar questões sobre o motivo de ter sido criado, em vez de se adotar um já existente como padrão. A intenção não é criar um produto para competir com outros frameworks, mas estabelecer uma plataforma que implemente o conceito de framework integrador. Ele realiza a integração entre vários frameworks especialistas e garante a evolução, manutenibilidade e compatibilidade entre cada um deles. Sua maior contribuição é dar um direcionamento ao uso das tecnologias.

O direcionamento tecnológico é importante para os prestadores de serviço, pois permite que eles se especializem nas tecnologias que foram definidas como o padrão a ser utilizado. Com a adoção de uma arquitetura de referência e uma plataforma integradora de tecnologias, um órgão do Governo Federal pode contratar uma empresa para desenvolver um sistema e depois ter a segurança de contratar outra para dar manutenção. Ou ainda, se tiver uma área de desenvolvimento de sistemas, pode ele mesmo realizar manutenção. Isso permite contratações mais flexíveis, assim como licitações de ampla concorrência.

Dessa forma, pequenas e médias empresas, ou até desenvolvedores ou consultores independentes que trabalhem como pessoa jurídica, podem participar de concorrências públicas junto com grandes empresas, pois todos terão acesso ao Demoiselle e às tecnologias relacionadas sem qualquer custo. O Governo Federal ganha, pois a ampla concorrência tende a diminuir os custos de manutenção dos sistemas de informação, além de que a própria plataforma é gratuita, o que o desonera em relação a licenças. A iniciativa privada ganha como um todo, pois a padronização dos sistemas democratiza a concorrência, evitando que um grupo de privilegiados tenha o mercado Governo reservado para si. Como diria o matemático John Nash, *“o melhor resultado virá quando todos do grupo fizerem o melhor para si mesmos e também para o grupo como um todo.”* (Nasar, 2002)

O Demoiselle extrapola as relações comerciais entre Governo e iniciativa privada. Qualquer empresa pode utilizar a plataforma para desenvolver sistemas para qualquer cliente. E esse cliente terá a segurança de que não dependerá do criador da aplicação para mantê-lo, ele terá liberdade de tomar conta de seus sistemas e de implementar funcionalidades sem ter que esperar por próximas versões. Tudo o que alguém precisa para fazer manutenção em uma aplicação desenvolvida pelo Demoiselle será conhecer os requisitos, o framework e a arquitetura de referência.

A qualidade é claramente afetada pelo estabelecimento de uma comunidade de usuários da plataforma, pois efetiva o reuso dos componentes criados. O Demoiselle oferece uma oportunidade de agregar a experiência de diversas comunidades de software relativas a cada um dos frameworks especialistas e criar uma grande rede colaborativa de desenvolvimento.

5. Ambiente de Desenvolvimento

O ambiente compatível para o desenvolvimento baseado no Demoiselle é composto de JVM, IDE e servidor de aplicação. A JDK requerida é a 1.5.x. Como IDE, recomendamos o Eclipse 3.3.2 JEE Developer (Europa) com o plugin AJDT

1.5.2.200804241330. Neste artigo o Eclipse é utilizado. Os servidores JEE com contêiner web 2.5 são compatíveis com o Demoiselle. Isso inclui Tomcat 6.x e JBoss 4.2.x.

6. O Nome

A nova plataforma é um sistema brasileiro, desenvolvido pelo SERPRO, empresa pública brasileira para uso nacional. A princípio deve causar estranheza que seja nomeado por uma palavra francesa. A explicação é a seguinte: Demoiselle é o nome de um modelo leve e pequeno de avião que o brasileiro Santos Dumont, conhecido como o Pai da Aviação, idealizou e pilotou em 1907, um ano depois do lendário vôo do 14 Bis. Diversas versões do Demoiselle foram testadas pelo avião de 1907 a 1909. Ao final das anotações e da documentação da construção da aeronave, Santos Dumont permitia a utilização, adaptação e cópia de seu trabalho. Em função dessa perfeita consonância com os conceitos atuais do software livre, Demoiselle foi o nome escolhido para batizar a plataforma de desenvolvimento de software.

Em dado momento de sua vida, Santos Dumont “dizia a todos que estava sem dinheiro. Ninguém acreditava em suas palavras, mas para animá-lo aconselharam-no a patentear o *Demoiselle*. Ele recusou. Era seu presente para a humanidade, disse que preferia terminar seus dias em um asilo de pobres que cobrar aos outros o privilégio de copiar sua invenção e de fazer experimentos aéreos.” (Hoffman, 2004)

7. Conclusão

Reusabilidade é palavra de ordem na utilização do Demoiselle Framework. Este artigo apresentou os objetivos, a proposta, a característica de ser software livre, a possibilidade do desenvolvimento colaborativo e, finalmente, o exemplo de criação de uma pequena aplicação web. Obviamente, uma aplicação simples, porém completa no tocante à utilização das camadas propostas pelo framework. Ela é suficiente para dar uma dimensão do que o Demoiselle é capaz, bem como instigar os desenvolvedores a completá-la, levando-os, é claro, a estudar sua documentação.

O Demoiselle apenas está em consonância com o ideal de um mundo de padrões. Padrões são bons, quando bem utilizados pois permitem que as pessoas se comuniquem. E, ainda que componentes sejam fabricados por pessoas diferentes e, mesmo assim, possam ser integrados a um único produto final. Assim, o desenvolvimento de um software que utiliza a mesma plataforma e processo tende a provê-lo de melhorias pela experiência acumulada com o tempo.

Referências

- Acesso a página do projeto no Sourceforge – <http://sourceforge.net/projects/demoiselle>
- Acesso ao portal do Demoiselle Framework – <http://www.frameworkdemoiselle.gov.br>
- BRASIL. MINISTÉRIO DA FAZENDA. SERPRO. (2009) “Sobre o Nome”. Em <http://www.frameworkdemoiselle.gov.br/menu/framework/sobre/>.
- Fowler, M. Patterns of enterprise applications architecture. Prentice Hall, 2002.
- Hoffman, Paul. (2004) Asas da Loucura: a Extraordinária Vida de Santos-Dumont. Objetiva.
- Johnson, R. E. How to develop frameworks. Disponível em

http://wiki.lassy.uni.lu/se2c-bib_download.php?id=453. Acesso em 20/03/2009.

Lisboa, F. G. S. Framework Java Serpro. TEMATEC, Ano X, nº XX, 2008.

Macias, A. Framework de Desenvolvimento – Visão Geral. Disponível em http://www.serpro.gov.br/clientes/serpro/serpro/imprensa/publicacoes/tematec/2008/tte_c92_a. Acesso em 25/03/2009.

Nasar, Sylvia. (2002) Uma Mente Brilhante. Record.