

# Reuso de Protótipos de Interface: uma Proposta para Utilização de Tecnologias Java para *Web*

Carlos Diego Quirino Lima  
 Luiz Carlos Rodrigues Chaves  
 Ygor de Oliveira de Carvalho  
 Herbert Anderson de Vasconcelos Dantas  
 Edemberg Silva da Rocha  
 Heremita Brasileiro Lira  
 Thiago Moura

Centro Federal de Educação Tecnológica da Paraíba  
 lucachaves@gmail.com

**RESUMO:** Reutilizar recursos é fundamental para minimizar esforços no processo de desenvolvimento de software. Em geral, não é fácil encontrar indicações, referências ou guias que tratem da aplicação do reuso de protótipos, sobretudo quando desenvolve-se sistemas Web. No entanto, o avanço substancial de estudos na área, aliado às novas tecnologias Java, tornou mais simples a adoção da prototipagem e, em paralelo, permite que se antecipe a construção da interface final do sistema Web, aproveitando-a completamente na fase de implementação. Diante disso, o Núcleo de Pesquisa e Desenvolvimento do CEFET-PB, do Projeto SIEP Gerencial resolveu, aliado à análise e projeto do SIPE (Sistema de Acompanhamento de Projetos Especiais), desenvolver um protótipo de interfaces, para demonstrar a possibilidade de reutilização.

**Palavras-chave:** prototipação de interfaces, reutilização de protótipos de interface, tecnologias Java Web e SIEP.

**ABSTRACT:** Reutilizar recursos é fundamental para minimizar esforços no processo de desenvolvimento de software. Em geral, não é fácil encontrar indicações, referências ou guias que tratem da aplicação do reuso de protótipos, sobretudo quando desenvolve-se sistemas Web. No entanto, o avanço substancial de estudos na área, aliado às novas tecnologias Java, tornou mais simples a adoção da prototipagem e, em paralelo, permite que se antecipe a construção da interface final do sistema Web, aproveitando-a completamente na fase de implementação. Diante disso, o Núcleo de Pesquisa e Desenvolvimento do CEFET-PB, do Projeto SIEP Gerencial resolveu, aliado à análise e projeto do SIPE (Sistema de Acompanhamento de Projetos Especiais), desenvolver um protótipo de interfaces, para demonstrar a possibilidade de reutilização.

**Keywords:** prototipação de interfaces, reutilização de protótipos de interface, tecnologias Java Web e SIEP.

## 1. Introdução

O reuso de artefatos de software trata-se de uma abordagem de desenvolvimento que prevê o aproveitamento do que já foi produzido, com a finalidade de acelerar o andamento das atividades de implementação dos requisitos, com a minimização de esforço. E nem tudo, a partir de um protótipo (seja ele de interface ou *software*), pode ser reutilizado no desenvolvimento, pois segundo Pura & Storey (2007) “*reuso refere-se ao projeto de novos sistemas a partir de artefatos reutilizáveis ou especificações de alto-nível.*”

Concordando com esta idéia, Pura & Storey (2007) enfatizam que as “*abordagens acerca de facilitar o reuso são necessárias uma vez que toda instância de reuso implica em custos. Se o custo percebido do reuso for maior do que o custo de se criar algo novo, é preferível este último.*”

Devemos considerar, ainda, a expressiva mudança dos mecanismos da *Web*, uma vez que ela se tornou uma verdadeira plataforma para aplicações distribuídas, ao contrário do que era anos atrás, quando funcionava como um repositório de informações em páginas estáticas (Gaedke & Reese, 2000).

Inserindo a prototipagem de interface dos ambientes WWW (*World Wide Web*) no contexto do reuso e da mudança freqüente das tecnologias *Web*, não há nada que prejudique a realização da reutilização de um protótipo no desenvolvimento de novas aplicações *Web*, principalmente devido o advento de arquiteturas baseadas em componentes e com os protótipos de alta-fidelidade.

Neste artigo, estaremos tratando da prototipagem do *Projeto Piloto* (um mini-sistema experimental para gerenciamento de projetos) do uso das tecnologias necessárias para sua produção e do uso das estruturas de composição em interfaces. Isso permitiu que o protótipo se tornasse reutilizável, ponto de partida para o desenvolvimento da interface final com o usuário.

Desta forma, iremos propor a aplicação da reutilização de protótipos para a elaboração de interfaces *Web* que venham a ser usadas efetivamente em projetos de desenvolvimento de *software*, baseados na plataforma JAVA, a partir das fundamentações do estudo de caso corrente.

## 2. Metodologia

No intuito de realizar a prototipação SIPE foi desenvolvido um projeto piloto no Centro Federal de Educação Tecnológica da Paraíba, no Núcleo de Pesquisa e Desenvolvimento, como forma de avaliar os resultados e benefícios de utilização de prototipação.

Esse projeto foi idealizado para a construção de uma pequena aplicação multicamadas, com o intuito de nivelar os bolsistas dotando-os dos conhecimentos técnicos necessários para as futuras atividades a serem desempenhadas no SIPE (Sistema de Acompanhamento de Projetos Especiais). O *Projeto SIEP* (Sistema Integrado da Educação Profissional), atualmente está ligado à *SETEC/MEC* (Secretaria de Educação Profissional e Tecnológica / Ministério da Educação).

Tomando por base os artefatos da fase de planejamento do Projeto Piloto - o levantamento de requisitos e o documento de visão - três fases foram planejadas.

Na primeira fase, os seguintes encaminhamentos foram realizados para a produção do protótipo do sistema:

- Decisão sobre que ferramentas de desenvolvimento seriam utilizadas para elaboração do protótipo.
- Decidir quais casos de uso seriam implementados na prototipagem, de forma que estivessem relacionados ao gerenciamento e a execução de atividades em projetos, ficando de fora as tarefas administrativas do sistema.
- Escolha das tecnologias para *Web* necessárias na construção do código fonte.

O resultado dos encaminhamentos acima levou em consideração a necessidade de aproveitar a interface resultante do processo de prototipagem, como base para o processo de implementação, resultando no seguinte ambiente de desenvolvimento:

- Utilização da ferramenta *Red Hat Development Studio Beta 2*, que integra o ambiente Eclipse 3.3 e seus plugins para *JavaServer Faces (JSF)*, *Facelets*, dentre outros para manipulação de código HTML, folhas de estilo, etc., com o servidor de aplicações *JBoss 4* embarcado.

- Foi escolhida a tecnologia *Facelets* (que usa XHTML – *eXtensible HTML*) para o *Framework JSF (JavaServer Faces)*, assim como a utilização de folha de estilos em cascata (*Cascading Style Sheets*).

A segunda fase foi a de implementação dos protótipos de interface do sistema. Uma vez definido todo o ambiente, iniciou-se a construção desses protótipos. Ao final, o protótipo da interface traria um modelo de interface para o sistema em desenvolvimento, assim como um produto para ser colocado em teste de validação dos requisitos do sistema junto ao cliente.

Nesta fase de prototipação da interface, foram desenvolvidos os seguintes artefatos:

- Folha de estilos em cascata (CSS).
- Página de Template com suporte a composição (*Facelets & JSF*).
- Páginas do protótipo para manipulação pelo usuário (JSF).
- Internacionalização (*resource bundle*).

A terceira e última fase foi levar estas páginas de prototipagem ao conhecimento dos de todos os desenvolvedores, com o intuito de validar os requisitos levantados, assim como angariar propostas de melhorias quanto à usabilidade, funções e necessidades para o sistema, e então prosseguir com o desenvolvimento do Projeto Piloto.

### 3. Fundamentação Teórica

#### 3.1. Desafios do reuso

De acordo com (Purao & Storey 2007), são dois os desafios para o reuso:

- fazer boas analogias, do protótipo inicial com o que deve ser o resultado final: para isso, é necessário conhecer bem padrões de análise de interfaces, ter o domínio do que deve ser desenvolvido, além de contar com a clareza no mapeamento dos requisitos.
- projetar sobre algo que já existe: é um desafio que exige superação de três dificuldades bem definidas: a recuperação (quanto às informações já existentes no protótipo), a adaptação (padronização do que já existe para o que deve ser feito) e a integração (na associação/junção do que

já existe com o que é exigido e será posto, no futuro, em produção).

Com intuito de aproveitar o esforço que já foi empregado, surge então a necessidade de enfrentar esses desafios, sobretudo pelos supostos ganhos de produtividade no quesito de construção de interfaces para ambiente *Web* – já que neste, é necessária a apresentação de um sistema com características visuais uniformes, conseguido apenas se tomarmos por base modelos (*templates*) de páginas, sobre os quais todas as demais unidades do sistemas serão construídas.

#### 3.2. Prototipagem

De acordo com (Sommerville 2007),

“um protótipo é uma versão inicial de um sistema de software usado para demonstrar conceitos, experimentar opções de projeto e, geralmente, conhecer mais sobre o problema e suas possíveis soluções.”

De acordo com a definição, existe o consenso entre os desenvolvedores de software de que o uso de protótipos para a concepção da interface com o usuário é um requisito indispensável, no contexto da Engenharia de Software. Serve para validar os requisitos levantados durante o planejamento, assim como agilizar o processo de documentação e treinamento dos usuários, sendo todo este aspecto vinculado ao processo de projeto do sistema.

Sobretudo quanto ao modelo evolutivo de prototipagem, (Aguiar 2007) discorre que

“os projetistas pré-supõem que o protótipo está em constante evolução e, portanto, pode ser modificado de acordo com o surgimento de novos requisitos ou alterações nos requisitos já existentes.”

Essa referência ao modelo evolutivo, onde o protótipo está em constante evolução, pode ser encarada como uma vertente para reutilização de protótipos. Diferentemente da abordagem *throw-away* – que considera o protótipo como um artefato de software que não virá ser utilizado em produção pelo cliente, muito menos mantido pelo desenvolvedor –, enxergar o protótipo como algo que cresce por demanda, é poder ampliar o seu es-

copo de utilização também para a fase de desenvolvimento.

Ainda de acordo com (Aguiar 2007),

“a classificação de protótipo pode se dar em termos de fidelidade, ou seja, o grau de similaridade entre o protótipo e a interface do produto final, incluindo características tais como métodos de interação, aparência visual, nível de detalhes, conteúdo, etc.. De acordo com a fidelidade, os protótipos são classificados, a princípio, em baixa-fidelidade e alta-fidelidade, sendo que, protótipos de alta-fidelidade são mais similares ao produto final do que protótipos de baixa-fidelidade.”

Sendo assim, temos nos protótipos de alta-fidelidade não apenas uma visualização estática das telas do sistema e seu conteúdo, mas também código construído com o uso de alguma linguagem de programação.

## 4. Estudo de Caso

### 4.1. Prototipagem do Projeto Piloto

De acordo com os níveis de classificação propostos por (Aguiar 2007), que divide protótipos de interface de software, a princípio, em dois níveis, respectivamente, baixa, e alta-fidelidade, podemos classificar o Protótipo de Interface do Projeto Piloto como um protótipo de alta-fidelidade.

O grande fator que impulsiona a utilização desse tipo de protótipo (de alta-fidelidade) é a necessidade de reutilização dos artefatos (telas, arquivos, etc.) produzidos nessa fase, de maneira que podem representar o ponto de partida para a construção efetiva da interface da aplicação.

Diferentemente do processo convencional de prototipagem - onde apenas procura-se demonstrar o funcionamento do *software*, aproximando a equipe responsável pelo desenvolvimento do sistema, com o cliente, que é responsável por validar algumas dessas funcionalidades -, existe a preocupação com a otimização do esforço empregado para a construção do protótipo.

Assim, podemos definir o protótipo da interface do Projeto Piloto como um protótipo reutilizável, ou seja, um protótipo de alta-fidelidade,

que se preocupa com a reutilização dos recursos empregados no seu desenvolvimento.

E no âmbito gerencial, podemos delimitar que o escopo das atividades se guiam na concretização do funcionamento das telas dos protótipos.

### 4.2. Utilizando tecnologias JAVA para Web na construção de protótipos reutilizáveis

Nesta seção serão apresentados os passos para elaboração do Protótipo da Interface do Projeto Piloto. Será sugerido um modelo que utilize as tecnologias Java para *Web* na elaboração de protótipos de alta-fidelidade, possibilitando a reutilização (reuso) de elementos da interface *Web* para a fase de implementação final da interface.

#### 4.2.1. Composição via *Templates* usando *Facelets*

*Facelets* é uma tecnologia para JSF, *open source*, que propõe corrigir as falhas decorrentes da integração entre Designers e Desenvolvedores de Software no que diz respeito ao desenvolvimento independente do visual e da implementação dos sistemas para *Web* (Geary & Horstmann, 2007).

Para isso, utiliza os conceitos de *views* (visões) e *subviews* (subvisões), que nada mais são que páginas, com código *markup* (linguagem de marcação) escrito e validado em XHTML (*eXtensible HTML*), o que padroniza o trabalho de Designers de Interface e Desenvolvedores no tratamento das visões da interface gráfica com o usuário.

É importante ressaltar que as visões são desacopladas de qualquer detalhe de implementação. Por isso, o Designer se sente livre para editar os modelos HTML, enquanto que o JSF é responsável por apresentar o visual do modelo de negócios.

No Protótipo de Interface do Projeto Piloto estas características foram bastante exploradas para garantir o reuso do que fosse produzido, sobretudo no que diz respeito à utilização de visões e subvisões.

Isso foi possível graças ao suporte à composição, nativo de *Facelets*. Com a composição, basta que seja criado um arquivo XHTML de modelo, delimitado por áreas com a *tag* `<ui:insert>`. Um exemplo de código fonte, que apenas define

áreas de uma página que compõe o modelo, pode ser visto na Fig. (1).

Em protótipos reutilizáveis, é uma grande vantagem empregar composição, uma vez que temos a padronização da interface do usuário, em todas as páginas, e os designers poderão gerar interfaces gráficas mais marcantes e com menor esforço, uma vez que apenas o modelo geral de páginas deverá ser alterado. O desenvolvedor preocupa-se apenas com o desenvolvimento do modelo de negócios, deixando a criação das páginas XHTML para o designer da interface.

Portanto, a página de modelo é chamada de visão. As partes que a compõem são as subvisões. Ao compor com um modelo, uma página XHTML deve declarar a tag `<ui:composition>` e informar o modelo (*template*) para o qual faz referências. Em seguida, basta definir as subvisões, que apenas definem (*tag <ui:define>*) o conteúdo da página.

#### 4.2.2. Uso de folha de estilo em cascata

Descrever como os documentos serão exibidos graficamente com a rederização de uma página *Web* não é a única utilidade das folhas de estilo em cascata. Elas também têm um papel fundamental nas questões referentes ao reuso de artefatos de interface construídos junto aos protótipos de alta-fidelidade para aplicações *Web*.

Note que, uma folha de estilos em cascata é previamente associada ao modelo.

Desta forma, basta que as alterações gráficas sejam realizadas neste único ponto (a folha de estilos) que toda a aplicação sofrerá os seus efeitos. Isso, além de garantir a uniformidade, ainda torna eficiente a manutenção (alterações) que, neste caso, tendem a ser constantes. A manutenção é uma atribuição do designer e, mais uma vez, melhor divide os papéis desse profissional e do desenvolvedor *Web*.

O reuso de folhas de estilo é garantido com a “tipagem” do corpo da página. Assim, uma vez criada para atender uma determinada necessidade, basta que novas classes de corpo sejam criadas para a folha de estilo e associadas ao *template*, para ter os resultados na tela, exibidos para os usuários dos sistemas *Web*.

Portanto, as folhas de estilos em cascata é artefato de interface que promove o reuso compondo um protótipo reutilizável.

#### 4.3. Internacionalização

Um dos pontos críticos da utilização de *Facelets* é o processamento de caracteres especiais. Experiências práticas com o desenvolvimento do protótipo do Projeto Piloto, durante a fase de implementação, mostraram que quando utilizamos composição via *templates*, estes deixam de ser exibidos (por exemplo, o caractere espaço em branco - “ ” - marcado como `&nbsp;`). Uma das saídas é a utilização de arquivos de recurso (ou de propriedades), comumente utilizados em I18N (internacionalização).

Quando consideramos que páginas *Web* são lidas nos mais diversos cantos do mundo, a internacionalização é importante para que o seu conteúdo possa ser exibido em diferentes línguas e o fato de se preocupar com este aspecto durante a elaboração do protótipo, em nada prejudica o seu desenvolvimento. Assim, a internacionalização pode nele ser associada, uma vez que, basicamente, alteramos o arquivo *faces-config.xml*, conforme a Fig. (2).

A alteração consiste em, nesse arquivo, ser criada em `<application>` as configurações de localidade (`<locale-config>`), assim como definido o recurso (`<resource-bundle>`).

Por fim, foram criados os arquivos *trI18N\_en*, *trI18N\_en\_US* e *trI18N\_pt\_BR*, com as respectivas traduções para o português e inglês, como, também apenas referenciar variáveis no *template* e as páginas que fazem composição com o mesmo, de acordo com o nome da variável especificada em `var` (neste caso, *keys*), nos recursos.

#### 4.4. Resultado

A Figura 3 mostra o resultado (telas) do protótipo que prioriza o reuso e serve como artefato a ser validado pelo cliente do sistema, analisado em conjunto com a equipe de desenvolvimento.

A construção de protótipos reutilizáveis de interface é possível concluir que é viável, desde que eles sejam protótipos de alta-fidelidade e utilizem uma tecnologia adequada para o favorecimento do reuso. No caso desse estudo, as tecnologias JAVA para *Web* (*Facelets* para *JSF*) atendem esse requisito.

Uma vez que o custo para a integração dos artefatos construídos não promove impacto significativo (devido o uso de composição), o ge-

renciamento é beneficiado pela boa divisão dos papéis na equipe de desenvolvimento e promove a arquitetura do software (padrão MVC). Podemos ter uma maior segurança em executar a prototipagem de interfaces com vistas no reuso, no processo de concepção e desenvolvimento de software.

## 5. Conclusões

Utilizando os recursos suportados por *Facelets*, para *JavaServer Faces*, foi possível elaborar o protótipo da interface do Projeto Piloto de maneira que o mesmo pudesse ser reaproveitado para a construção da aplicação final.

Graças a esse reaproveitamento foi possível construir a interface final fazendo pequenas adaptações do que já estava criado, incluindo, por exemplo, a possível adição de elementos na folha de estilos em cascata. Tudo foi aproveitado, havendo apenas a substituição dos conteúdos das subvisões *header2*, *sidebar1* e *mainContent* em cada página XHTML. Ao invés das subvisões terem elementos HTML, esses elementos foram substituídos por componentes JSF que possuem um melhor relacionamento entre a interface e o modelo de negócios do sistema.

Outro resultado positivo que pudemos perceber quanto ao reaproveitamento de protótipos de interface é que, em termo de papéis da gerência do desenvolvimento de software, é possível dividir as atividades que se referem à implementação do modelo de negócios e sua representação visual em páginas *Web* (atribuições do designer), promovendo uma arquitetura para projetos multicamadas.

O resultado (telas) mostram que o protótipo prioriza o reuso e serve como artefato a ser validado pelo cliente do sistema, analisado em conjunto com a equipe de desenvolvimento.

Com base nisso, é possível concluir que é viável a construção de protótipos reutilizáveis de interface, desde que eles sejam protótipos de alta-fidelidade e utilizem uma tecnologia adequada para o favorecimento do reuso. No caso desse estudo, as tecnologias JAVA para *Web* (*Facelets* para *JSF*) atendem esse requisito.

Uma vez que o custo para a integração dos artefatos construídos não promove impacto significativo (devido o uso de composição), o gerenciamento é beneficiado pela boa divisão dos papéis na equipe de desenvolvimento e promove a

arquitetura do software (padrão MVC). Podemos ter uma maior segurança em executar a prototipagem de interfaces com vistas no reuso, no processo de concepção e desenvolvimento de software.

## 6. Referências

AGUIAR, Y. P. C. **SMILE – uma ferramenta para geração automática, edição e simulação de protótipos de interface do usuário**. Campina Grande: UFCG, 2007. Dissertação de Mestrado.

GAEDKE, M.; REHSE, J. **Supporting Compositional Reuse in Component-Based Web Engineering**. Disponível em: [http://mwrwg.tm.unikarlsruhe.de/DownloadCenter/publications/dcpub/2000/2000SAC2000GaedkeRehse\\_pdf](http://mwrwg.tm.unikarlsruhe.de/DownloadCenter/publications/dcpub/2000/2000SAC2000GaedkeRehse_pdf). Acesso em: 29 de outubro de 2007.

GEARY, D.; HORSTMANN, C. **Core JavaServer Faces**. 2ª Ed. Rio de Janeiro: Alta Books, 2007.

PURAO, S.; STOREY, V. C. **Supporting Reuse during Conceptual Design of Information Systems: is it possible? Will They use it?** Disponível em: <http://puro.ist.psu.edu/working-papers/apsara-purao-storey.pdf>. Acesso em: 29 de outubro de 2007.

SOMMERVILLE, I. **Engenharia de Software**. 8ª Ed. Tradução: Selma Shin Shimuzu Melnikoff, Reginaldo Arakaki, Edilson de Andrade Barbosa. São Paulo: Pearson Addison-Wesley, 2007.