

Prototipagem de interfaces **na prática**

Otimizando a identificação e especificação de requisitos



Bruno Leite Alves

(brunoleitealves@gmail.com /Twitter @brunoleitealves): é bacharel em Ciência da Computação pela UNI-BH trabalha com análise e desenvolvimento de software desde 2006. Atualmente está cursando mestrado na UFMG. Possui certificação SCJP e nas horas vagas mantém um blog: <http://brunoleite.wordpress.com>. Atualmente trabalha na Spread Systems.



Rafael Saraiva Paoliello

(rafael.paoliello@gmail.com): é bacharel em Ciência da Computação pela PUC-MG trabalha com análise e desenvolvimento de software há 4 anos. Atualmente trabalha na Spread Systems.

A prototipagem ganha cada vez mais valor no mercado e já é utilizada em muitas empresas. Entretanto, ainda é vista muitas vezes apenas como uma forma de comunicação entre o requisito e o desenvolvimento, representando somente a disposição da tela. Na verdade, a prototipagem permite a validação e melhor entendimento dos requisitos levantados com o usuário, além de lhe proporcionar uma visão prévia do produto que será construído. As técnicas aqui apresentadas permitem otimizar a produtividade da equipe e minimizar o retrabalho.



F prototipagem é uma técnica que pode ser utilizada na construção de software. O protótipo pode ser construído de diversas formas, como, por exemplo, na construção civil onde os engenheiros e arquitetos constroem maquetes com o intuito de reproduzir conceitos visuais e funcionais do projeto. De forma análoga, temos o protótipo de software que pode ser construído usando esboços simples em papel ou ferramentas especializadas.

No desenvolvimento de software, a prototipagem ocorre durante o levantamento de requisitos e gera impactos em outras disciplinas. Uma de suas finalidades, talvez a principal, é fornecer ao cliente uma visibilidade sobre como será o sistema. Com a técnica de prototipagem podemos obter uma melhor definição das necessidades do cliente, já que é possível prever e corrigir inconsistências antes de um maior detalhamento.

A técnica de prototipagem é independente de processo ou metodologia de desenvolvimento de software, podendo ser usado em contextos mais formais, como CMMI, RUP e afins, ou em contextos ágeis, como Scrum ou XP, lembrando também que nada impede que tais contextos se mesquem. Em casos como o RUP, a prototipagem ocorre antes do detalhamento dos casos de uso, evitando assim retrabalhos desnecessários na documentação e possivelmente no desenho e codificação. Já em contextos ágeis como o Scrum, podemos ter a prototipagem ocorrendo na reunião de priorização do backlog e criação das histórias com o P.O. (product owner).

Este artigo está organizado da seguinte forma: inicialmente será abordada uma visão geral sobre o assunto e serão analisadas algumas ferramentas e técnicas utilizadas para prototipagem. Em seguida, é abordada a importância da criação da padronização de interfaces. Por fim, apresentamos um exemplo de utilização na prática.

Prototipagem

Durante a coleta de requisitos, temos várias barreiras que precisam ser quebradas, talvez a maior delas seja a dificuldade do especialista em entender o que o cliente deseja, ou até mesmo do cliente definir suas necessidades de forma clara e objetiva. Diante desse cenário, se faz necessário o uso de técnicas capazes de mitigar essas dificuldades, prevenindo possíveis deficiências no sistema e alto índice de retrabalho, em outras palavras, minimizando o custo do software e maximizando os resultados para o cliente.

Uma das formas de classificar um protótipo é baseando-se em sua fidelidade, podendo essa assumir dois tipos, baixa ou alta. O primeiro tipo indica que o protótipo não é visualmente nem funcionalmente fiel ao software, já o segundo tipo indica o contrário, podendo esse chegar ao nível final como o software real.

Ciclo de prototipagem

Como realizar a prototipagem? O que deve ser feito? Como todos sabem não existe de fato uma “bala de prata”, a solução pode variar de software para software, de cliente para cliente, enfim, cada cenário demanda a sua própria solução. Em softwares que seguem modelos ágeis, a técnica de prototipagem é bem favorecida, uma vez que o cliente está presente no dia-a-dia da equipe. Já em modelos nos quais o cliente é menos frequente, podemos usar a prototipagem para minimizar os efeitos de sua ausência.

A partir de uma conversa (formal ou informal) com o cliente, em que o especialista realiza a coleta inicial dos requisitos, é elaborado o primeiro protótipo. Este protótipo pode ser esboçado em papel (baixa fidelidade), feito utilizando ferramentas próprias para prototipagem ou o próprio código-fonte das telas. Este último sendo favorecido quando usado em conjunto com algum framework de template como Apache Tiles ou Facelets (alta fidelidade). Este protótipo então deve ser apresentado para o cliente para aprovação e auxílio em um melhor detalhamento dos requisitos iniciais, podendo gerar evoluções sobre o protótipo inicial.

A figura 1 mostra o fluxo de uma prototipagem, na qual o especialista inicia o processo com a coleta de requisitos com o cliente. Com os requisitos em mãos, a equipe de análise e prototipagem deve ser capaz de gerar algum tipo de protótipo, seja de baixa ou alta fidelidade. O especialista deve então apresentar os protótipos ao cliente, que por sua vez pode solicitar ajustes, definir melhor os requisitos podendo alterá-los ou até mesmo identificar novos. A grande vantagem desse ciclo é que mesmo em ambientes mais formais onde vários artefatos devem ser gerados, o custo é baixo, já que a manutenção está restrita somente ao protótipo. Caso contrário, as alterações em determinados tipos de especificações podem se tornar mais trabalhosas. A partir do protótipo aprovado, o especialista deve então iniciar a especificação baseada em requisitos com um maior grau de estabilidade e compreensão. A aprovação da especificação torna-se um trabalho menos desgastante para o cliente, uma vez que a documentação é apenas a consolidação do que já foi apresentado nos protótipos. Feito isso haverá menos desvios ao final da construção em relação aos desejos do cliente, ajudando a cumprir o planejamento de custo e prazo previamente acordados, evitando maiores desgastes.

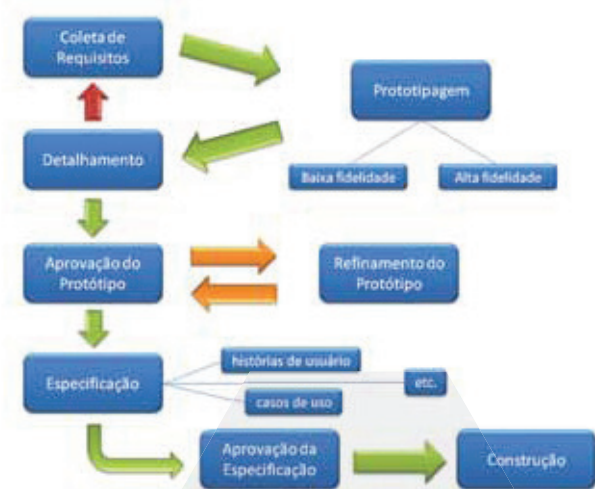


Figura 1. Evolução do requisito utilizando prototipagem.

Prototipagem na prática

A prototipagem pode ser um processo trabalhoso e complexo. Portanto, reunimos dez dicas para sua aplicação prática, que podem ser observadas a seguir.

- Deixe clara a interação entre os protótipos.
- Defina todos os componentes visuais que você irá precisar, monte sua biblioteca.
- Compartilhe e colabore – revisões em pares podem ajudar na troca de experiências.

- Envolver o cliente.
- Seja objetivo e atenha-se ao problema, evitando sair do escopo proposto.
- Seja criativo, mas cuidado para não extrapolar.
- Priorize a reusabilidade, a geração de código é interessante se o custo e o prazo permitirem.
- Dê a devida atenção à usabilidade.
- Envolver a área técnica para não inviabilizar a construção.
- Comece a prototipagem pelos requisitos que geraram mais dúvida.

Ferramentas e técnicas

No contexto atual temos várias opções de ferramentas e técnicas que podem variar das mais simples até as mais sofisticadas inclusive transformando simples protótipos em código reutilizável agregando maior valor ao negócio.

O protótipo esboçado em papel apresenta agilidade e baixo custo, tanto para criação como para treinamento do especialista na técnica. Entretanto, existem dificuldades de manutenção e armazenamento, além dos problemas evidentes na reutilização durante a construção do software. São gerados protótipos de baixa fidelidade, ou seja, que não refletem o software. Neste caso, deve-se deixar claro para o cliente que o software será construído com qualidade, evitando assim expectativas negativas. A figura 2 exemplifica este caso.

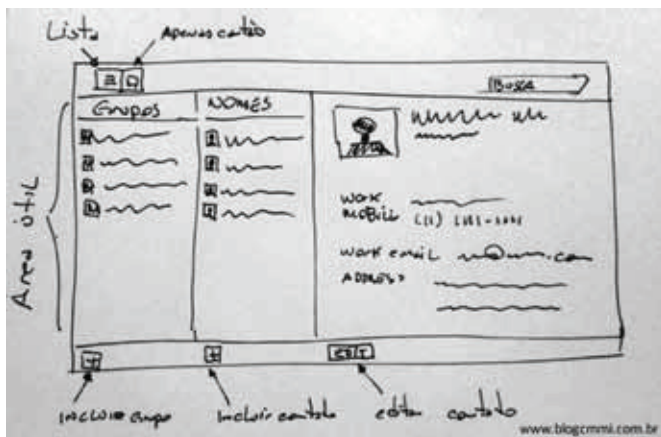


Figura 2. Esboço em papel.

Existem ainda ferramentas específicas para a criação de protótipos. A figura 3 exibe o exemplo da criação de uma tela simples de autenticação de usuário utilizando o Balsamiq (<http://www.balsamiq.com>). O Balsamiq proporciona uma maior interação entre o usuário e os protótipos, já que permite navegação e realização de ações simples através de links. Além disso, a ferramenta possui uma vasta biblioteca de componentes que podem variar desde campos do tipo texto até a exibição de vídeos ou ainda gráficos de vários tipos, como barra ou pizza.

A ferramenta apresenta um baixo custo para aquisição, US\$79, levando em conta seus benefícios, como aumento de produtividade e facilidade de uso. Para quem deseja conhecer a ferramenta, mas não está disposto a desembolsar o valor da licença, a ferramenta possui uma versão trial de 7 dias que pode ser obtida no site ou

também pode utilizar a versão web de demonstração que pode ser encontrada no endereço: <http://builds.balsamiq.com/b/mockups-web-demo/>.



Figura 3. Ferramenta Balsamiq.

A ferramenta Napkee (<http://www.napkee.com>) integra-se com arquivos do formato BMML (formato dos arquivos gerados pela ferramenta Balsamiq). Essa ferramenta foca em produtividade, pois converte os protótipos em códigos Web (HTML, CSS e JavaScript) como podemos observar na figura 4 ou Adobe® Flex 3 (MXML e ActionScript) como é apresentado na figura 5. Dessa forma, o código gerado pode ser usado para aumentar a produtividade (uma vez que estará pronta a camada de apresentação no caso de códigos Web e camadas de apresentação e controle no caso de códigos Flex) em arquiteturas que usam determinadas tecnologias, como PHP ou Adobe® Flex 3. Talvez o ganho seja menor com tecnologias como JSF (Java Server Faces) e Apache Wicket.

Assim como a Balsamiq, esta ferramenta possui um baixo custo, US\$49, também apresentando uma versão trial de 15 dias. Em casos onde se tem um ganho de produtividade, podemos considerar um ótimo custo x benefício, já que tempo é dinheiro.



Figura 4. Ferramenta Napkee – Formato web.

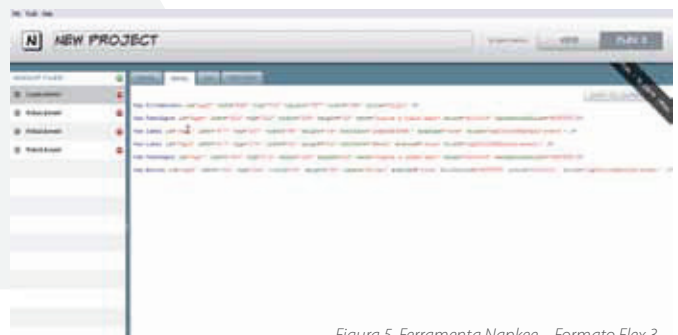


Figura 5. Ferramenta Napkee – Formato Flex 3.

As ferramentas Balsamiq e Napkee apresentadas anteriormente, ainda que justifiquem, possuem um custo de aquisição. A ferramenta Pencil (<http://pencil.evolus.vn>), ao contrário, não apresenta custos de aquisição, se apresentando como melhor alternativa para aqueles que não desejam pagar por uma ferramenta deste tipo. Há versões para instalação como plugin do Firefox ou standalone, figura 6. Ela é capaz de gerar códigos HTML, dentre outros formatos de imagem e documentos, a partir de um template. Existem alguns padrões disponíveis no endereço: <http://pencil.evolus.vn/en-US/Downloads/Templates.aspx>.

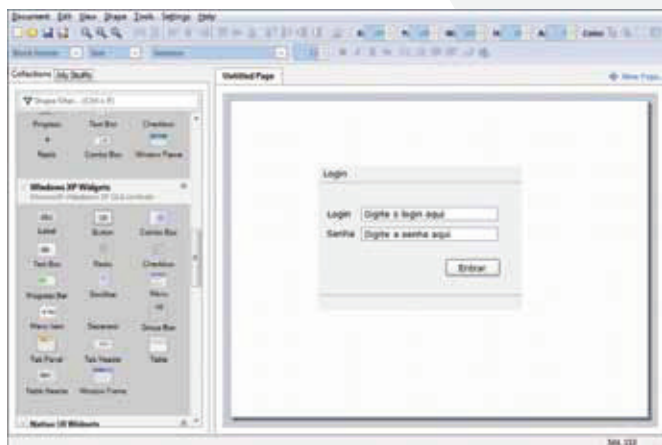


Figura 6. Ferramenta Pencil.

Outra boa opção de ferramenta que não possui custos de aquisição é a Tigr (<http://gotiggr.com>). Ela traz grande benefício principalmente quando estiver sendo utilizado RichFaces no sistema, pois são utilizados componentes desta biblioteca, figura 7. É uma ferramenta web que possibilita também navegação entre as telas criadas. Além de, como as demais, também exportar os arquivos para o formato HTML.



Figura 7. Ferramenta Tigr.

Quando a equipe de análise e levantamento de requisitos possui a habilidade de codificação é possível que os protótipos já sejam criados dentro da arquitetura e tecnologias do software. A grande vantagem do uso desta técnica é a alta taxa de reúso dos protótipos durante a implementação, além disso, com a utilização de algum framework ou técnica de template tem-se um aumento na produtividade e maior aderência aos padrões de interface. Neste caso, temos um protótipo de alta fidelidade, ou seja, com grande

proximidade do software que o usuário terá em funcionamento ao fim do projeto. Em contrapartida, é uma prototipagem mais demorada, uma vez que esta equipe envolve-se em parte da codificação.

A Listagem 1 representa um código JSF usando o framework de template Facelets de uma tela de autenticação de usuário semelhante à apresentada na figura 3. A primeira linha indica o arquivo de template utilizado, no qual o mesmo pode possuir componentes visuais que são repetidos em várias telas do sistema, como banner, menu, rodapé, dentre outros. A partir da tela gerada pela Listagem 1 a equipe de desenvolvimento irá codificar o controlador (além da camada de modelo, no modelo MVC – Model View Controller) e conectá-lo à tela, realizando pequenos ajustes.

Listagem 1. Código XHTML com Facelets da tela de login.

```
<ui:composition template="template.xhtml">
  <ui:define name="corpo">
    <h:form>
      <div>
        <fieldset>
          <legend>Login</legend>
          <div>
            <div>
              <h:outputLabel value="Login:" /><br/>
              <h:inputText size="40" maxlength="50" />
            </div>
            <div>
              <h:outputLabel value="Senha:" /><br/>
              <h:inputSecret size="50" maxlength="50" />
            </div>
            <div>
              <h:commandButton value="Entrar" />
            </div>
          </div>
        </fieldset>
      </div>
    </h:form>
  </ui:define>
</ui:composition>
```

A importância da padronização de interface

A padronização da interface normalmente é a definição de diretrizes do software relativas à interface com o usuário. Em metodologias ágeis esta definição pode estar acordada com o cliente e ter parte representada apenas nos arquivos de estilo do software, por exemplo, arquivos CSS (Cascade Style Sheet). Já em ambientes mais formais, onde são necessárias aprovações dos artefatos com assinatura do cliente, é interessante a criação de um documento com tais padrões especificados para que seja submetido à aprovação do cliente. É de suma importância que o cliente participe da criação destes padrões de forma a evitar desentendimentos.

O padrão de interface auxilia a equipe de prototipagem nas construções das telas, evitando dessa forma que novos componentes sejam usados a todo o momento, pois a variação visual pode gerar anomalias no sistema, além de dificultar o desenvolvimento, o que na prática representa maior esforço e custo. Como a prototipagem pode variar desde rabiscos no papel até o próprio código-fonte, duas abordagens podem ser feitas: 1) Protótipos de baixa fidelidade têm os padrões aplicados na implementação, ficando a cargo do desenvolvedor realizar as transformações, uma vez que o desenho gerado é apenas o esqueleto visual, por assim dizer; 2) Protótipos de alta fidelidade seguem os padrões à medida que são

construídos pela equipe de prototipagem, já que as telas geradas são de fato as telas que serão incorporadas ao sistema.

O fluxo da figura 8 representa a utilização do padrão de interface em vários pontos da evolução dos requisitos.

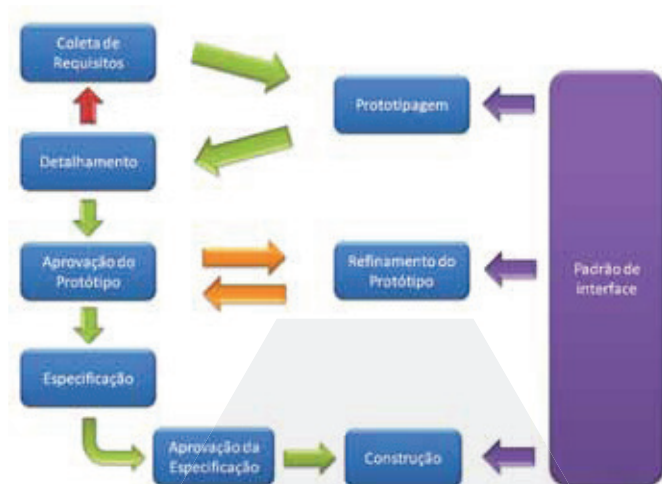


Figura 8. Evolução do requisito usando padrão de interface.

Como criar um guia de estilo

O guia de estilo é um documento que contém os padrões de interface que foram definidos junto ao cliente. Durante sua elaboração é fundamental também a observação de princípios de desenho, como:

- Visibilidade – Destaque visual das funcionalidades. O usuário deve ser capaz de identificar o próximo passo;
- Feedback – Usuário deve ter retorno de suas ações, por exemplo, mensagens ou barra de progresso;
- Restrições – Disponibilizar somente as opções necessárias, evitando erros;
- Correspondência – Os componentes visuais devem ser semelhantes ao mundo real;
- Consistência – Componentes similares para tarefas similares, facilitando o aprendizado;
- Affordance – Definir comportamentos semelhantes à familiaridade do usuário, por exemplo, um botão irá sugerir que o usuário o pressione.

Este documento deve ser o mais claro e objetivo possível, evitando ambiguidades. Para isso, podemos realizar detalhamentos textuais com exemplos nas próprias telas do sistema, como pode ser observado na figura 9.



Figura 9. Exemplo tela no guia de estilo.

A seguir, listamos alguns itens que julgamos importantes na criação de padrões de interface.

- Fontes e cores.
- Estrutura de telas e relatórios:
 - o Posição de banner, menu, rodapé e corpo.
- Características da logomarca utilizada.
- Janelas:
 - o Definição de quando usar cada tipo de janela, por exemplo, modal ou do browser.
- Formulários:
 - o Uso de agrupador de campos (fieldset), máscaras, disposição dos campos etc.
- Menus:
 - o Definições visuais e comportamentais, por exemplo, horizontais e verticais.
- Abas:
 - o Comportamento, cores e formas.
- Janelas de diálogo, por exemplo, janelas de confirmação de decisão tomada pelo usuário.
- Listagens:
 - o Zebrado, cabeçalho e alinhamento dos dados apresentados nas tabelas.
- Paginação:
 - o Estilo e forma, por exemplo, número de registros por página, forma de navegação entre as páginas e exibição da página atual.
- Botões e ícones de ação:
 - o Alinhamento de botões, definição de ícones etc.
- Mensagens:
 - o Posicionamento e cores.

Exemplo de utilização

Vamos exemplificar nesta seção a coleta de requisitos seguindo o fluxo detalhado neste artigo. Suponhamos que um dado cliente solicitou a uma empresa qualquer, denominada aqui de ACME, um sistema para gerência de seu estoque.

Em uma conversa inicial, os analistas identificaram as necessidades dos clientes. O cliente deixou claro que gostaria de um software que fosse capaz de realizar a autenticação dos usuários, mas não sabia como fazer isso, ele gostaria também que o software tivesse o controle de estoque de sua produção e que o mesmo fosse auditado, pois recentemente eles desconfiaram de fraudes na empresa. A empresa possuía outro sistema já implantado na empresa, um sistema de controle de pagamento, porém ele não possuía a especificação do mesmo.

Após a conversa inicial, a empresa optou por gerar um documento de visão fechando o escopo do projeto. Dessa forma, os requisitos do cliente foram documentados e formalizados para um posterior detalhamento. Com o documento de visão em mãos, a equipe de análise e prototipagem começou a trabalhar nos protótipos, mas antes que algo começasse a ser feito, alguém da equipe de prototipagem identificou um ponto importante levantado pelos analistas, a empresa já possuía um sistema implantado, seria então interessante para ambas as partes seguir o padrão de interface já existente neste sistema de folha de pagamento. Assim foi decidido que duas pessoas seguiriam em paralelo documentando seus padrões de interface. A equipe do comercial foi acionada para que a empresa pudesse cobrar por essa documentação.

Enquanto o guia de estilo era gerado, o restante da equipe começou a trabalhar nos protótipos tendo em mãos os requisitos levantados. Deve-se ressaltar a importância da participação da equipe de arquitetura neste ponto, uma vez que várias características do sistema impactam na parte técnica. Os primeiros protótipos gerados foram de baixa fidelidade, usando simples rabiscos no papel. Esta técnica foi escolhida para evitar custos desnecessários, uma vez que os requisitos nessa fase ainda se encontram imaturos. A figura 10 mostra como ficou a tela de controle de estoque após as primeiras conversas entre a equipe.

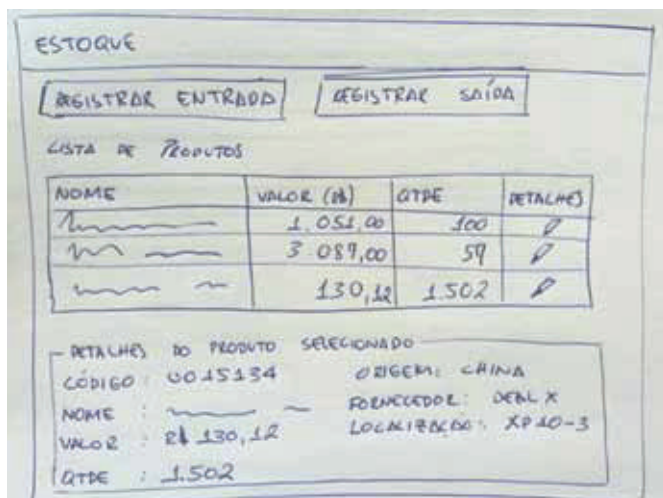


Figura 10. Esboço em papel da tela de controle de estoque.

Ao ver as telas de seu futuro sistema, o cliente acabou percebendo que gostaria de algumas melhorias e que havia se esquecido de informar que a impressão dos dados de estoque era extremamente importante para o funcionamento da empresa, pois tais informações eram colocadas no quadro principal da empresa para determinar metas. Com as novas informações em mãos, e com aprovação parcial do protótipo, os analistas se reuniram com a equipe de prototipagem para novas discussões. Enquanto isso, a equipe de documentação do guia de estilo finalizava seu trabalho, o que significava mais alguns ajustes a serem feitos no protótipo. Dessa vez, a equipe optou por usar a ferramenta Pencil, já que os requisitos estavam mais estáveis e o guia de estilo finalizado. A equipe técnica ressaltou também que a ferramenta escolhida poderia gerar o código para a construção do sistema, favorecendo assim a produtividade. Depois de vários ajustes, as novas telas ficaram prontas. A figura 11 mostra a nova tela de controle de estoque, após os pedidos do cliente e a aplicação do guia de estilo.

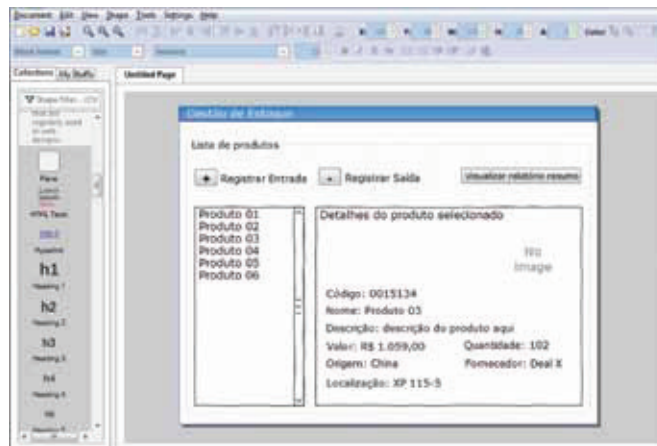


Figura 11. Protótipo da tela de controle de estoque.

Os protótipos foram submetidos à aprovação do cliente, que por sua vez, realizou aprovação com pequenas ressalvas. Com as telas aprovadas, a especificação dos casos de uso seguiu sem grandes alterações, o que minimizou os riscos do projeto. Após a geração dos documentos, que também foram submetidos à aprovação do cliente, a construção enfim foi iniciada, agora com os requisitos mais estáveis e com um padrão de interface disponível.

Considerações finais

O aumento no grau de compreensão das expectativas do cliente pela equipe de análise pode ser considerado como um dos principais benefícios da atividade de prototipagem na análise e levantamento de requisitos. Esta atividade também aproxima o cliente da equipe de análise, melhorando o entendimento dos requisitos que, normalmente, são apresentados somente em documentos, muitas vezes, de difícil compreensão. Desta forma, podem ser evitadas lacunas entre o software desenvolvido e o desejado pelo cliente, uma vez que a identificação de entendimentos errôneos das necessidades do cliente, que futuramente se tornariam possíveis alterações, ocorre o quanto antes. Estas alterações tendem a ser onerosas e geram retrabalho.

Usando mecanismos simples ou avançados os ganhos sempre serão consideráveis para todas as partes interessadas. Mesmo quando não se deseja despesar maiores recursos para a atividade de prototipagem utilizando mecanismos mais avançados, deve-se optar por uma prototipagem de baixa fidelidade, pois, desta forma, estaremos otimizando a identificação e especificação dos requisitos.

Referências

- <http://penta3.ufrgs.br/tutoriais/usabilidade/>
- <http://drdobbs.com/184414825>
- <http://www.requirementsdevelopment.com/>
- <http://java.dzone.com/articles/10-features-you-need-when>
- <http://www.napkee.com/>
- <http://www.balsamiq.com>
- <http://gotiggr.com>
- <http://pencil.evolus.vn>
- Figura 2 retirada do blog: <http://www.blogcmmi.com.br/>
- Figuras 3, 4 e 5 <http://rederia.net/2009/11/14/prototipagem-balsamiq-mockups-napkee/>

GUJ – Discussões sobre o tema do artigo e assuntos relacionados

Discuta este artigo com 100 mil outros desenvolvedores em www.guj.com.br/MundoJ