

# Framework FIB: um estudo experimental

Gislaine Camila Lapasini Leal<sup>1</sup>, Ana Paula Chaves<sup>1</sup>, César Alberto da Silva<sup>2</sup> e  
Elisa Hatsue Moriya Huzita<sup>1</sup>

<sup>1</sup>UEM - Universidade Estadual de Maringá  
Departamento de Informática. Maringá, PR, Brasil  
cammyleal@gmail.com, chavesana@hotmail.com, elisa@din.uem.br

<sup>2</sup>UTFPR - Universidade Tecnológica Federal do Paraná  
Coordenação de informática. Cornélio Procópio, PR, Brasil  
cesaralberto@utfpr.edu.br

**Abstract** Growing applications complexity, recent technological advances, time and cost restrictions for software development/delivery and the demand for product quality have been increased. So, tools that offer adequate support for the software development are currently needed. However, before creating them it is necessary to perform an evaluation their functionalities and also their usability and efficiency. Experimental Software Engineering practices have been adopted to perform experiments. This paper presents a case showing how the use of experimental studies can help evaluating the usability and efficiency of a framework developed for graphical user interface construction. The phases related to the experimental study execution are described from the planning until results discussion.

**Key words:** experimental study, experimental software engineering, framework.

## 1 Introdução

O cenário atual, marcado pelo avanço tecnológico e o aumento da complexidade dos aplicativos de software, demanda por ferramentas que auxiliem no processo de desenvolvimento. Uma das preocupações no desenvolvimento de software é a necessidade de agilizar a construção de aplicações desde a interface até o armazenamento dos dados, aumentando assim a reutilização no processo de software e, por conseguinte, reduzindo os custos e alcançando ganhos em qualidade. Diante desse cenário, *frameworks* têm sido desenvolvidos com o intuito de aumentar a produtividade, padronização e, por conseguinte, a qualidade.

Segundo [3], os estudos experimentais investigam as vantagens e desvantagens de métodos, técnicas e ferramentas na Engenharia de Software. Eles geram conhecimento experimental, que pode fazer parte do corpo geral de conhecimento da Engenharia de Software. Esse conhecimento é necessário para auxiliar organizações de Engenharia de Software a melhorar continuamente seus processos de desenvolvimento e manutenção para se tornarem competitivos. Além disso,

elas tem que melhorar a qualidade de seus produtos de software em um contexto de mudanças contínuas. O artigo apresenta um estudo experimental que avalia a usabilidade e eficiência de um *framework* para geração de interfaces gráficas. Esse estudo foi conduzido em turmas de duas instituições de ensino superior do Paraná. São discutidas as etapas que perfazem um estudo experimental e como estas foram conduzidas no estudo realizado.

O artigo encontra-se estruturado em quatro seções, além desta. A Seção 2 aborda o referencial teórico acerca de engenharia experimental. Na Seção 3 é apresentado o objeto de estudo do experimento, o *framework* FIB. O estudo experimental realizado é detalhado na Seção 4. E, por fim, são realizadas algumas considerações finais.

## 2 Engenharia Experimental

No contexto da engenharia de software, os estudos empíricos tem adquirido importância ao longo do tempo. Eles podem ser definidos como uma forma de se obter informações sobre um objeto de pesquisa. Desse modo auxiliam na construção de conhecimento com base em observações e evidências empíricas [1]. Parte da comunidade de pesquisa está voltando suas atenções para um conjunto de técnicas agregadas denominadas de Engenharia de Software Experimental [2].

A Engenharia de Software Experimental tem como objetivo apontar indícios sobre hipóteses levantadas a respeito de determinados artefatos gerados durante o desenvolvimento de software.

Um experimento é um tipo de estudo empírico realizado em um ambiente controlado, passível de observação e onde as hipóteses podem ser analisadas. Isto resulta em um estudo focado em variáveis específicas com um potencial estatístico significativo, constituindo-se na forma mais efetiva para provisão de evidência empírica e refinamento de conhecimento. Os resultados obtidos são analisados para identificar variáveis chaves e o relacionamento entre elas. Essa análise pode ser apoiada por habilidade humana, experiências, conhecimento de domínio entre outros.

A realização de experimentos tem por objetivo avaliar, caracterizar, prever, controlar e melhorar produtos, processos, recursos, modelos, teorias entre outros. Além disso, presume um relacionamento de causa e efeito [3]. Um experimento é desdobrado em diversas atividades, as quais possuem complexidade variável. Segundo [3] a condução de um experimento apresenta cinco fases:

- **Definição:** descreve os objetivos do experimento, objeto de estudo, foco da qualidade, ponto de vista e contexto;
- **Planejamento:** realiza a seleção do contexto, formulação das hipóteses, seleção das variáveis, seleção dos participantes e projeto do experimento;
- **Execução:** execução do experimento e coleta dos dados;
- **Análise e interpretação:** compara as hipóteses elaboradas com os dados alcançados com o uso de ferramentas estatísticas adequadas;
- **Apresentação e empacotamento:** documentação das informações obtidas a partir do experimento de modo estruturado e unificado.

No planejamento de experimentos é importante considerar a necessidade de replicação ou complementação do estudo a fim de ampliar a base de dados, a credibilidade e o índice de confiabilidade. A fase de apresentação e empacotamento oferece suporte para a replicação, oportunidades de variação e combinação de resultados [4].

Ao considerar as características de replicação e possibilidade de variação, surge a necessidade de incorporar técnicas e ferramentas para gerenciar bem como planejar, controlar e melhorar a qualidade do experimento. Segundo [6], após a realização do experimento, os dados coletados são caracterizados usando estatística descritiva, por meio de cálculos de centralidade, tendência e dispersão. Os dados anômalos ou falsos são excluídos, reduzindo os dados a um conjunto de dados válidos. Em seguida, os dados são analisados pelo teste das hipóteses e as hipóteses do estudo experimental são avaliadas estatisticamente.

A Estatística Descritiva é o nome dado ao conjunto de técnicas analíticas utilizadas para resumir o conjunto dos dados coletados em uma dada investigação a poucos números e gráficos. Com isto, objetiva-se entender a distribuição dos dados e sua natureza, além de identificar pontos anômalos. As medidas de tendência central (mediana, média e moda) são utilizadas como indicadores, que permitem que se tenha uma noção inicial de como os dados de um estudo experimental se comportam.

### 3 FIB (*Fast Interface Builder*)

O Grupo de Estudos em Engenharia de Software Distribuído (GEESD), que participa da construção de um ambiente distribuído de desenvolvimento de software conhecido como DiSEN (*Distributed Software Engineering Environment*), criaram o *framework* FIB, motivados pela ausência de apoio à criação da interface com o usuário, a estrutura e os métodos de acesso a banco de dados nos *frameworks* analisados (SwingBean e Genesis). Esse *framework* possui um conjunto de componentes para facilitar a construção de interface gráfica Swing, que possibilita a geração das classes java persistentes e os métodos de acesso ao banco de dados (CRUD) [1].

O FIB disponibiliza um conjunto de componentes, os quais facilitam a construção de formulários de maneira interativa, permitindo que o desenvolvedor apenas arraste os componentes para o formulário e aplique as configurações, de forma visual, para associar com a classe a ser persistida. Além disso, há a possibilidade de aplicar regras de validação aos componentes. Um conjunto de regras de validação está disponível para uso e, caso haja necessidade, o desenvolvedor pode criar suas próprias regras.

A concepção do FIB foi baseada nas necessidades e perspectivas do ambiente DiSEN e de seus participantes. Para sua consolidação como um *framework* de construção de interfaces gráficas independente de domínio, foram realizados estudos empíricos com o objetivo de averiguar se as funcionalidades oferecidas são suficientes para apoiar o desenvolvimento em diferentes contextos e capturar novos requisitos que possam contribuir com o aperfeiçoamento do mesmo. Maiores de-

talhes sobre o *framework* FIB pode ser obtido em [1]. A seção seguinte descreve como os experimentos foram conduzidos, métodos e ferramentas utilizadas.

## 4 Estudo Experimental

Ao introduzir processos, métodos e ferramentas é necessário analisar três aspectos no contexto em questão: viabilidade, usabilidade e eficácia [5]. Com base nessa necessidade foi planejado e executado um estudo empírico para identificar indícios desses aspectos no *framework* FIB. Esse estudo tem por objetivo oferecer informações que subsidiem o aprimoramento contínuo deste *framework*, enfatizando suas funcionalidades, utilidade e usabilidade. A seguir são descritos os experimentos realizados e é apresentada a análise e discussão dos resultados alcançados.

### 4.1 Características do Experimento

O objeto de estudo do experimento é o *framework* FIB e tem por objetivo identificar suas características de viabilidade, usabilidade e eficácia para desenvolvimento de interfaces gráficas independentes de domínio. O estudo foi desenvolvido sob a óptica do pesquisador, tendo como perspectivas, averiguar as funcionalidades oferecidas e detectar novos requisitos. O contexto do experimento consistiu em um cenário de desenvolvimento de software, em que os participantes deveriam desenvolver as interfaces da aplicação com base em um diagrama de classes. Foi realizado um projeto piloto, por dois integrantes do grupo GEESD, para estimar o tempo de desenvolvimento da atividade proposta e encontrar problemas no material planejado para o estudo.

No que se refere à instrumentação, foram elaborados: (i) um manual do *framework* descrevendo as funcionalidades, uma apresentação sobre seus objetivos e o cenário proposto e, (ii) dois questionários, sendo um para capturar o perfil do entrevistado e outro para avaliar a ferramenta. Com base no projeto piloto foram definidos alguns critérios, como o tempo máximo de 100 minutos para realização do experimento e que poderia haver comunicação entre os participantes.

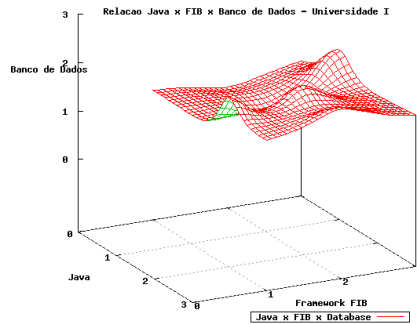
O experimento foi norteado pelas seguintes hipóteses: o FIB possibilita a criação ágil de interfaces gráficas para entidades persistentes; a utilização do *framework* requer uma curva de aprendizagem baixa; e apresenta um conjunto de funcionalidades suficientes para a construção das interfaces. A técnica de balanceamento foi utilizada para eliminação de casos extremos, como por exemplo, alunos que ofereceram resistência à realização da atividade proposta. A execução da tarefa consistiu na criação de duas interfaces de acordo com o diagrama de classes fornecido, preenchimento dos questionários e entrega por e-mail dos artefatos gerados na atividade. Cada questionário gerado, na fase anterior, possuía um número de identificação. Esse número foi utilizado para relacionar os questionários com os artefatos enviados pelos alunos, permitindo que sua identidade fosse preservada. Com o objetivo de verificar se as respostas fornecidas no questionário eram condizentes com o resultado da implementação, foram

selecionados aleatoriamente alguns questionários para que o artefato correspondente pudesse ser analisado. Após a tabulação dos dados, foram gerados gráficos utilizando a ferramenta GNUPlot 4.0. Para facilitar a construção dos gráficos, os valores do questionário foram graduados, indicando o grau de conhecimento ou dificuldade do aluno, da seguinte forma: 0 - nenhum, 1 - baixo/básico, 2 - intermediário, 3 - alto/avançado.

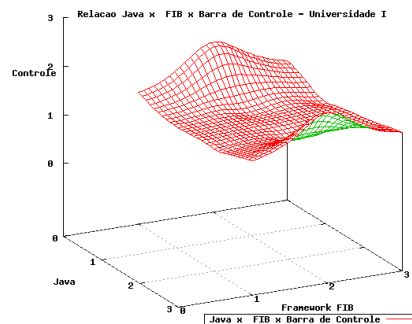
## 4.2 Experimento I

Os participantes do experimento I são 37 alunos do 3º ano do curso de Ciência da Computação, matriculados na disciplina Processos de Engenharia de Software II de uma universidade pública, que oferece cursos na área de informática há mais de 30 anos, incluindo um curso de pós-graduação em nível de mestrado. As razões que motivaram a escolha dos participantes foram: o GEESD e os participantes são da mesma universidade; o uso do *framework* pode ser integrado ao curso; a atividade pode ser utilizada como um exemplo prático; e, os participantes possuem um núcleo comum de conhecimentos que muitas vezes não é encontrado na indústria [5].

Os gráficos obtidos a partir dos dados coletados são apresentados nas Figuras 1, 2, 3 e 4. Esses gráficos ilustram a relação entre o conhecimento dos participantes na linguagem Java, a dificuldade em utilizar o *framework* FIB e, a terceira coordenada, representa o conhecimento dos participantes em Banco de Dados (Figura 1), dificuldade em utilizar os componentes Barra de Controle (Figura 2), Enum (Figura 3) e Grid (Figura 4).



**Figura 1.** Gráfico da relação Java x FIB x Banco de Dados.



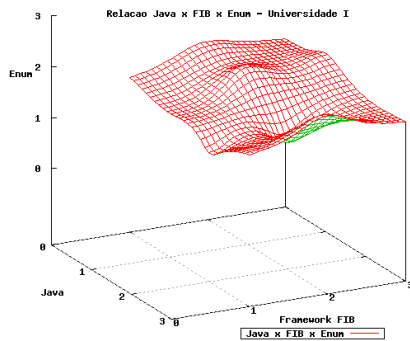
**Figura 2.** Gráfico da relação Java x FIB x Barra de Controle.

No gráfico da Figura 1, destacam-se três picos. Os participantes que possuíam conhecimento intermediário em banco de dados e básico em Java, tiveram um alto grau de dificuldade com o *framework* FIB. O segundo pico refere-se aos alunos com conhecimento intermediário em Java e banco de dados, que classificaram o FIB com dificuldade intermediária. No terceiro pico, os que possuíam

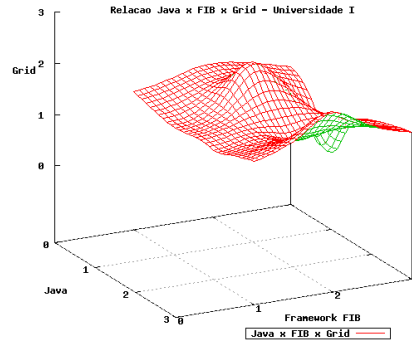
conhecimento intermediário em Java e básico em banco de dados, apresentaram dificuldade baixa na utilização do FIB.

Na Figura 2, é possível evidenciar que os alunos com nenhum conhecimento em Java tiveram dificuldade intermediária/alta para utilizar a Barra de Controle, embora tenham considerado o uso do *framework* entre os graus básico e intermediário. Os alunos com alto conhecimento em Java, classificaram a dificuldade em usar o *framework* e a Barra de Controle em intermediário e básico, respectivamente.

Analisando a Figura 3, pode-se observar que os participantes com nenhum ou pouco conhecimento em Java, tiveram maior dificuldade de utilizar o componente Enum e que classificaram, o nível de dificuldade em utilizar o *framework*, como intermediária. Nota-se também, que os alunos com conhecimento avançado em Java tiveram dificuldade intermediária em utilizar o *framework* e o componente Enum. Alguns alunos, com conhecimento em Java e dificuldade com o *framework* intermediário, disseram não encontrar dificuldades no uso do Enum.



**Figura 3.** Gráfico da relação Java x FIB x Enum.



**Figura 4.** Gráfico da relação Java x FIB x Grid.

Observando o gráfico da Figura 4, percebe-se que, de um modo geral, a dificuldade em utilizar o componente Grid foi considerada de intermediária a baixa.

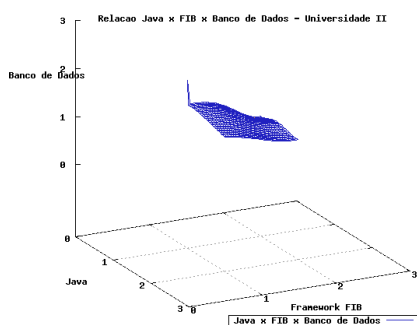
### 4.3 Experimento II

Os participantes do experimento II são 20 alunos do 3º semestre do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, matriculados na disciplina Projeto Integrador de Banco de Dados de uma universidade pública, que oferece cursos na área de informática há mais de 10 anos, incluindo um curso de pós-graduação em nível de especialização. As razões que motivaram a escolha dos participantes foram: interesse da parte dos professores da universidade na participação do experimento, o uso do *framework* pode ser integrado ao curso;

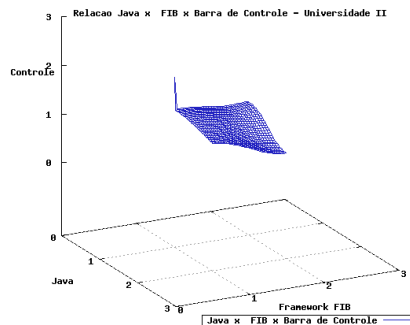
a atividade pode ser utilizada como um exemplo prático; e o núcleo comum de conhecimento dos participantes.

Os gráficos correspondentes aos dados coletados são apresentados nas Figuras 5, 6, 7 e 8. Assim como no experimento I, esses gráficos ilustram a relação entre o conhecimento dos participantes na linguagem Java, a dificuldade em utilizar o *framework* FIB e, a terceira coordenada, representa o conhecimento dos participantes em Banco de Dados (Figura 5), dificuldade em utilizar os componentes Barra de Controle (Figura 6), Enum (Figura 7) e Grid (Figura 8).

A Figura 5 mostra que os picos correspondem aos alunos com algum conhecimento em banco de dados e na linguagem de programação Java (básico/intermediário), tendo maior destaque, o conhecimento intermediário em banco de dados e básico em Java. Com relação ao *framework* FIB, a dificuldade dos participantes ficou entre básico e intermediário. Os indivíduos com maior conhecimento em Java e menor conhecimento em banco de dados apresentaram maior dificuldade.



**Figura 5.** Gráfico da relação Java x FIB x Banco de Dados.

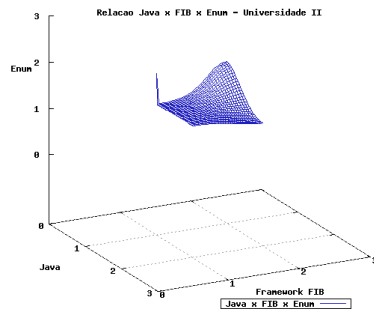


**Figura 6.** Gráfico da relação Java x FIB x Barra de Controle.

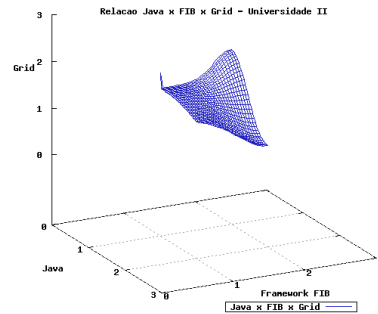
No gráfico da Figura 6, observa-se que os indivíduos que tiveram dificuldade intermediária para utilizar o *framework* FIB e, possuíam conhecimento intermediário em Java, tiveram dificuldade baixa na utilização da barra de controle. Por outro lado, os participantes com dificuldade baixa no uso do *framework* e conhecimento básico de Java, tiveram dificuldade intermediária na utilização da barra de controle.

A Figura 7 apresenta um pico, que corresponde aos participantes que, embora possuam baixo conhecimento em Java e tenham considerado a dificuldade em utilizar o *framework* entre intermediária e alta, tiveram dificuldade intermediária para utilizar o componente Enum. Grande parte dos participantes teve dificuldade baixa para utilizar esse componente.

Com relação à Figura 8, o pico destaca que os participantes com baixo ou nenhum conhecimento em Java e que consideraram intermediária a dificuldade em utilizar o *framework*, tiveram dificuldade entre intermediária e alta para



**Figura 7.** Gráfico da relação Java x FIB x Enum.



**Figura 8.** Gráfico da relação Java x FIB x Grid.

utilizar o componente Grid. O vale indica que os participantes com conhecimento intermediário em Java e que consideraram intermediária a dificuldade no uso do *framework* FIB, tiveram dificuldade baixa ou nenhuma para o componente Grid.

#### 4.4 Discussão dos resultados

Considerando os experimentos individualmente, observa-se que o experimento I apresentou uma dispersão maior dos dados, enquanto no experimento II os resultados se concentraram nos níveis básico/baixo e intermediário. Mesmo com características de dispersão diferentes, os resultados foram bastante similares. Em ambos os experimentos, o conhecimento em Java foi determinante para a dificuldade de utilização do *framework*.

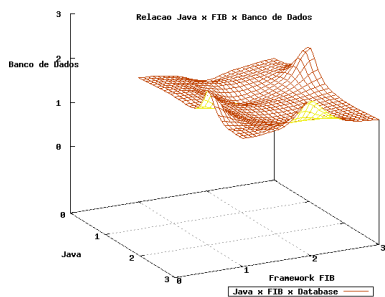
Com o objetivo de analisar a usabilidade e eficiência do *framework* FIB, os dados obtidos a partir dos experimentos realizados foram agrupados e novos gráficos foram gerados. Os resultados são reproduzidos nas Figuras 9, 10, 11 e 12.

O gráfico da Figura 9 mostra três picos, indicando os participantes que declararam maior conhecimento em banco de dados. Entretanto, observa-se que um maior conhecimento em banco de dados não foi determinante para o uso do *framework*.

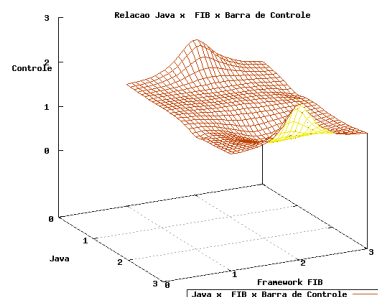
Na Figura 10, os participantes que consideraram o uso da barra de controle de dificuldade alta, consideraram o uso do *framework* FIB intermediário; o conhecimento em Java não foi determinante. A maioria dos participantes teve dificuldade entre nenhuma e baixa no uso desse componente.

O gráfico da Figura 11 mostra que grande parte dos participantes tiveram dificuldade entre baixa e intermediária para utilizar o componente Enum. O vale indica que os participantes com maior conhecimento em Java (intermediário), tiveram menor dificuldade. Com poucas exceções, os picos indicando maior dificuldade no uso do componente Enum, representam indivíduos com menor conhecimento em Java.

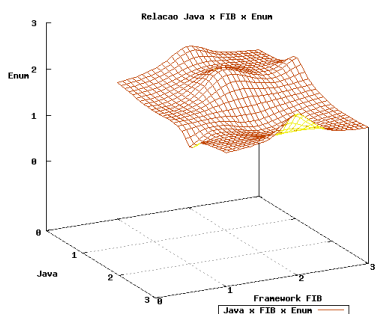




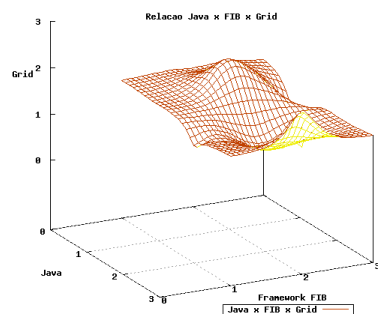
**Figura 9.** Gráfico de comparação entre os experimentos I e II da relação Java x FIB x Banco de Dados.



**Figura 10.** Gráfico de comparação entre os experimentos I e II da relação Java x FIB x Barra de Controle.



**Figura 11.** Gráfico de comparação entre os experimentos I e II da relação Java x FIB x Enum.



**Figura 12.** Gráfico de comparação entre os experimentos I e II da relação Java x FIB x Grid.

Por fim, a Figura 12 possui dois picos e um vale bastante salientes. Os picos indicam os participantes com maior dificuldade no uso do componente Grid e o vale indica menor dificuldade. Mais uma vez, com poucas exceções, o conhecimento da linguagem de programação foi determinante para a dificuldade dos indivíduos no uso do componente.

De maneira geral, o cálculo da média determina que, tanto o conhecimento dos participantes em Java e banco de dados, quanto à dificuldade em usar o *framework* e os componentes específicos estão entre básico/baixo e intermediário.

## 5 Considerações Finais

Este artigo tem como objetivo apresentar um estudo experimental para avaliar um *framework* de geração de interfaces gráficas, denominado FIB. O estudo experimental foi realizado em duas etapas. Primeiramente, os resultados foram

analisados isoladamente e depois comparados, a fim de averiguar a usabilidade e eficiência do FIB. Comparando os resultados dos experimentos com as hipóteses, concluiu-se que o conhecimento sobre a linguagem é determinante para a utilização do *framework*, já que facilita para o usuário a compreensão sobre as classes geradas automaticamente. Entretanto, participantes com pouco ou nenhum conhecimento em Java conseguiram construir as interfaces gráficas e realizar as operações de CRUD, embora tenham sentido mais dificuldade. A curva de aprendizagem para esses participantes é um pouco mais elevada, entretanto vantajosa, o que pode ser deduzido pela pequena dificuldade dos participantes em utilizar os componentes do FIB. Quanto às funcionalidades, para o domínio que o *framework* procura atender (construção de formulários para operações de CRUD), os componentes presentes no *framework* se mostraram satisfatórios. Contudo, o estudo ofereceu diretrizes para elicitação de novos requisitos para o *framework* FIB, tais como: geração de interfaces web; criação de *wizard* para geração de interfaces. Como trabalhos futuros, apontam-se: (i) a definição de hipóteses relacionadas a desempenho e custos para o uso do *framework* FIB; (ii) estender o estudo para outros ambientes, considerando participantes com perfis diferentes dos analisados (níveis de especialização, mestrado, profissionais de empresas); (iii) analisar o *framework* utilizando a técnica QFD (*Quality Function Deployment*) e FMEA (*Failure Mode and Effects Analysis*), para auxiliar a incorporação das necessidades dos usuários e identificar falhas potenciais e seus efeitos, respectivamente.

## Referências

1. C. A. da Silva, E. H. M. Huzita, G. Y. Sato, e W. C. Donegá. FIB: Um framework de apoio à construção de interface visando aumento de produtividade no desenvolvimento de software. In Simpósio Brasileiro de Engenharia de Software, XV Sessão de Ferramentas, Campinas, São Paulo, Brasil, 2008.
2. E. S. Doria. Replicação de estudos empíricos em engenharia de software. Master's thesis, Instituto de Ciências Matemáticas e de Computação - ICMC - USP, São Carlos, 2001.
3. D. S. Cruzes. Análise Secundária de Estudos Experimentais em Engenharia de Software. Tese de Doutorado - Faculdade de Engenharia Elétrica e Computação - Campinas, 2007.
4. R. dos Reis Alvarez. Métodos de identificação, análise e solução de problemas: uma análise comparativa. In XVII Encontro Nacional de Engenharia de Produção, Gramado - Porto Alegre, 1997. ABEPRO.
5. G. H. Travassos, D. D. Gurov, e E. A. G. Amaral. Introdução à engenharia de software experimental. Technical Report RT-ES-590/02, Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2002.
6. Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, Anders Wesslén. A. Experimentation in Software Engineering - An Introduction. Kluwer Academic Press, 2000.