

## **Dojo: Uma proposta de ensino e treinamento para o uso de Softwares Livres**

### Eixo 4 – Tecnologias da Informação e Comunicação para Transformação Social

Pedro H. da C. Braga<sup>1</sup>, Samantha B. de O. Cruz<sup>2</sup>, Lucimeri Ricas Dias<sup>3</sup>

<sup>1</sup>Núcleo de Solidariedade Técnica da Universidade Federal do Rio de Janeiro - SOLTEC/UFRJ, Rio de Janeiro - RJ - pedrohcb.ufrj@gmail.com

<sup>2</sup>Núcleo de Solidariedade Técnica da Universidade Federal do Rio de Janeiro - SOLTEC/UFRJ, Rio de Janeiro - RJ - samantha@poli.ufrj.br

<sup>3</sup>Núcleo de Solidariedade Técnica da Universidade Federal do Rio de Janeiro - SOLTEC/UFRJ, Rio de Janeiro - RJ - luricas@gmail.com

### **Resumo**

Os atuais bolsistas do projeto de extensão do Núcleo de Solidariedade Técnica – SOLTEC-UFRJ fazem parte de uma geração que desenvolveu com a tecnologia uma relação mais intensa que as gerações anteriores. Entretanto, a área de computação é marcada por contínuas modificações e conhecimentos específicos que não são acompanhados de igual modo pelos grupos que não tem formação em informática, gerando assim, uma demanda de aprendizagem e treinamento contínuo. Outro fator relevante para esse estudo é a baixa adesão e a falta de interesse em optar pelo uso de *softwares* livres, mesmo em ambientes que apoiam e defendem o seu uso. Por isso, os autores desse artigo, membros da equipe de TI do SOLTEC, propõem a adaptação do método de Dojo nas oficinas de treinamento para o uso de *softwares* livres para toda equipe do projeto de extensão em questão.

### **1 Introdução**

Dentro das propostas construtivistas, a aprendizagem resulta das interações que se estabelecem entre o sujeito e os agentes mediadores da cultura do grupo ao qual pertence, tendo como eixo norteador o trabalho cooperativo/colaborativo. Nesta perspectiva, é relevante buscar propostas metodológicas com a finalidade de instrumentalizar uma equipe para o uso adequado de *softwares* e programas, proporcionando a interação de forma recíproca, simultânea e com objetivos definidos em torno de uma tarefa comum. Nesse caso, oficinas de *softwares* livres, pois acreditamos para que os benefícios dos mesmos sejam amplamente aproveitados, é necessário divulgação e treinamento.

Através de questionários de pesquisa realizados com *Google Docs*<sup>1</sup>, é notório que muitos usuários, apesar de conhecerem algumas opções de *softwares* livres possuem certo preconceito com esse seguimento de programas por achá-los complicados e piores do que os *softwares* não livres. Outros membros do nosso projeto desconhecem totalmente o conceito de *software* livre e preferem utilizar em seus computadores pessoais, as versões piratas de programas da *Microsoft*, desconhecendo também, a possibilidade de acesso gratuito a essas assinaturas

---

<sup>1</sup>Modelo Disponível em: <http://goo.gl/k1Xvm>. Acesso em: 14 de junho de 2013.

para as instituições de ensino e pesquisa através do site da *Microsoft DreamSpark*<sup>2</sup> (DINIZ et al, 2012), conforme ilustrados no Gráficos 1 e 2:

Gráfico 1 – Comparação entre o uso de *Software* Livre e Proprietário

Fonte: elaborado pelos autores

Gráfico 2 – Número de usuários do *Microsoft Windows*

Fonte: elaborado pelos autores

Outro fator que acaba afastando a cultura do *Software* Livre dentro das universidades é a menor facilidade que seus programas apresentam na hora da utilização. Empresas como a *Microsoft* produzem programas automatizados, os quais não dependem de um grande conhecimento prévio para uso. Enquanto isso, sistemas como o Linux exigem técnicas avançadas para sua programação, o que dificulta seu uso e disseminação entre alunos de áreas diversas. Apesar da grande melhora adquirida através de colaboradores, tipos específicos de programas ainda dependem de usuários com alto nível de conhecimento (LIMA, 2012).

Diante desse contexto, escolhemos o método do Dojo, pois acreditamos que tal atividade tem um grande potencial para aprendizagem e treinamento com *softwares* livres. É compreensível que em grupos que utilizam computadores em seu ambiente de trabalho, algum membro da equipe não consiga dar os primeiros passos em algum programa, eventualmente o que falta é apenas um detalhe ou

---

<sup>2</sup>Disponível em: <https://www.dreamspark.com/>. Acesso em: 14 de junho de 2013.

uma simples orientação, num Dojo este membro pode observar outros da sua equipe executando desafios e ao perceber alguns detalhes, ele mesmo poderá praticar. Em caso de dúvida, contará com a ajuda dos companheiros da plateia (LUZ e NETO, 2012).

## 2 Conhecendo o Dojo

A palavra Dojo, em uma tradução literal do japonês, significa “lugar do caminho”, usada para designar o espaço de meditação dos monges budistas. Uma tradução livre significa “lugar onde se estuda a vida”, mas hoje, o termo é mais conhecido como o lugar onde se pratica artes marciais, uma sala dentro de uma academia com o chão coberto por tapetes. Atualmente, o termo tem sido usado para representar o lugar de encontro de um grupo de desenvolvedores que exercitam boas práticas de desenvolvimento, absorvendo a visão de desenvolvimento e aprendizagem compartilhada como uma arte. Dojo de programação, em inglês Coding Dojo, é uma atividade usada por programadores como um método de aprendizagem ágil, especialmente desenvolvimento dirigido para testes, programação pareada, e “passos de bebê” (Baby Step) (Beck e Andres, 2004). De um modo geral, o Dojo pode ser usado para denotar uma reunião de pessoas com o propósito de resolver um problema - Aprendizagem Baseada em Problemas (Problem Based Learning) (Wood, 2003) e ao Desenvolvimento Orientado a Testes (Test Driven Development) (Beck, 2003).

No Dojo, a visão central é a do organismo "ativo", em “fluxo”, não preocupado em responder a estímulos como no modelo behaviorista, mas engajado, buscando o sentido e o significado dos desafios propostos. O grupo pode explorar e fazer interpretações de suas próprias experiências e erros, elaborando novas saídas e testando as resoluções encontradas, para com isso, aprenderem com as informações dadas. Segundo Delgado et al. (2012) as principais vantagens do método Dojo são: “a velocidade de ensino ajustada à capacidade de absorção, teoria puxada pela prática, discussões feitas em torno de um código que compila e não sobre texto no quadro negro, onde o ambiente costuma ser amigável, e erros e inseguranças não são valorizados”. No entanto, para que possa ser aplicada com sucesso, a nossa proposta de Dojo como treinamento de *software* livre, precisa ser adaptada ao contexto, levando em conta os resultados dos questionários aplicados ao grupo. A oficina está sendo elaborada de acordo com um processo bem definido, discutido a seguir.

### 2.1 Esquema Dojo

Normalmente, uma sessão de Dojo tem duração de cerca de 2 (duas) horas, e requer uma sala para acolher os participantes, um computador e um projetor. A presença de um quadro para uso durante as discussões também é importante. O andamento do encontro varia, mas costuma ser organizado como no exemplo da Figura 1:

Figura 1 – Dinâmica de um Dojo

Fonte: Adaptado de Randori Coding Dojo (2011)

- Os 20 primeiros minutos são usados para a escolha do problema e os programas que serão utilizados;
- Os 10 minutos seguintes são usados para discutir a abordagem da solução;
- Dois voluntários iniciam a construção do primeiro teste, um como motorista e outro como navegador;
- A cada 5-7 minutos, o motorista volta para a plateia, o navegador assume como motorista, e um voluntário da plateia assume como navegador;
- Assim que o teste passa, a plateia pode opinar sobre refatorações e possíveis soluções alternativas para o problema;
- Quando não há mais refatorações a serem feitas, o próximo teste é planejado, e a dinâmica continua;
- Ao fim da sessão, são discutidos os pontos positivos e negativos da atividade.

### 3 Integração do Dojo ao treinamento e ensino de Software Livre

Em nível nacional é preciso que se destaque a opção pela utilização do *software* livre em estabelecimentos públicos federais e em escolas, expressa nas Diretrizes do Comitê Técnico de Implementação de *Software* Livre, dentre as quais merecem especial atenção as diretrizes que determinam a opção por “utilizar o *software* livre como base dos programas de inclusão digital”, “disseminar a cultura de *Software* Livre nas escolas e universidades” e finalmente, “popularizar o uso de *Software* Livre na sociedade” (CISL, 2009). Dentro da universidade, as principais razões pelas quais, nós da Equipe de TI do SOLTEC - UFRJ, defendemos o uso e adoção do *software* livre, podemos destacar (BORGES e EYER, 2003):

**Segurança:** o acesso ao código fonte permite que seja realizada uma auditoria. Um indicador da importância da segurança é o sistema operacional GNU/Linux na National Security Agency (NSA) – EUA;

**Custo menor:** a licença General Public License (GPL) da Free Software Foundation/GNU autoriza cópias ilimitadas;

**Maior qualidade:** o código aberto permite mais estudos e testes;

**Maior flexibilidade:** o acesso ao código fonte permite adaptações em tempo hábil;

**Mais vida útil:** em oposição ao conceito de obsolescência programada e a constante busca pela inovação tecnológica, os computadores não são descartados, podendo ser transferidos ou doados para aplicações que usam *Software Livre* (COUTO, 2012);

**Mais grupos de pesquisas e desenvolvimento:** nas universidades, empresas e demais centros, promovendo a cooperação internacional na resolução de complexos e de interesse comum;

**Mais cidadania:** o *Software Livre* estimula a solidariedade e a cooperação, refletindo positivamente na conscientização e participação política. A escolha de prioridades para o Movimento do *Software Livre* dentro das Universidades é um exercício saudável de política;

**Desenvolvimento Auto-Sustentável:** através do crescimento econômico com inclusão social.

O uso de *softwares* livres em ambientes de pesquisa permite o acesso a toda a capacidade tecnológica disponível a baixo custo e de maneira menos burocrática. Isso ainda contribui para o outro lado que é a participação da comunidade acadêmica na construção e melhoria desses *softwares*. Dessa forma é fechado um ciclo de contribuição entre as duas partes, favorecendo a disseminação da filosofia que há por trás do *software* livre (REIS, 2012).

### 3.1 Perfil da equipe SOLTEC-UFRJ

Dentre os membros do SOLTEC que responderam ao nosso formulário, podemos observar no Quadro 1 uma distribuição da faixa etária onde prevalecem nascidos entre 1985 e 1995, isto é, que se encontram hoje entre 18 e 28 anos de idade, cursando sua primeira graduação.

Quadro 1 – Perfil da equipe SOLTEC-UFRJ

Fonte: elaborado pelos autores

Também foi observado na análise dos resultados coletados que a maioria dos membros da equipe, independente da faixa etária, está aberta a participar de oficinas que orientem o uso de *softwares* livres.

Gráfico 3 – Nível de aceitação das oficinas de *Software Livre*

Fonte: elaborado pelos autores

Levando em consideração os dados coletados na pesquisa e o conhecimento das atividades do SOLTEC de seus quadros funcionais, determinou-se que os dois focos do treinamento-piloto que servirão de experimento prático desse estudo, descrito mais detalhadamente a seguir, serão o uso do sistema operacional GNU/Linux, em especial a distribuição Ubuntu, a suite de escritório LibreOffice, e ferramentas de design gráfico para uso específico pela equipe da coordenação de comunicação.

#### **4 Experimento Prático**

A partir da nossa própria capacitação em encontros com outros grupos que estudam e aplicam o método Dojo, buscamos estimular a criatividade, pensamento reflexivo e autônomo, juízo crítico, diferentes níveis de abstração e domínio de diferentes linguagens e fluência tecnológica. Acreditamos que para isso, o uso de Dojo no ensino pode representar uma mudança no paradigma de aprendizado: saímos de um cenário passivo, onde o usuário apenas absorve o conhecimento, para um cenário mais ativo, onde o usuário participa e aprende com os erros e acertos dos colegas (DELGADO, 2012). No entanto, não é possível aplicar o Dojo em sua forma tradicional a um ambiente de treinamento e aprendizado como o que estamos propondo. Algumas características do escopo do treinamento e tamanho da plateia limitam a flexibilidade oferecida pelos Dojos. Além disso, o foco original do Dojo é o aumento da habilidade em programação, enquanto que o objetivo estudado nesse artigo é a capacitação de usuários para o uso de ferramentas livres. Portanto, será uma proposta de aplicação inovadora dentro dos estudos pesquisados sobre Dojo.

A segunda fase planejada da pesquisa será o experimento prático com a equipe do SOLTEC. De acordo com as respostas obtidas em questionários, será realizado um treinamento os *softwares* livres mais usados pelo grupo. A equipe será dividida

em dois grupos, um grupo e a terá 2 (duas) horas-aula expositivas de teoria e prática intercaladas. O outro grupo participará de 20 horas de Dojo. O mesmo conteúdo será apresentado aos dois grupos, divididos aleatoriamente, pelo mesmo instrutor. Após essas 2 (duas) horas, ambos os grupos devem desenvolver um projeto semelhante. O código do projeto será submetido a um avaliador neutro de forma cega. Além dessa avaliação, o código deve ser analisado por ferramenta de análise de cobertura de testes, que examina quais partes do programa foram executadas pelo conjunto de testes.

#### **4.1 Resultados Esperados**

- capacitação de usuários para o uso de ferramentas livres;
- disseminação da cultura do *Software* Livre nas universidades;
- fluência tecnológica da equipe;
- aprendizagem colaborativa em um ambiente de treinamento e capacitação.

#### **4.2 Avaliação e Trabalhos Futuros**

A última etapa será a elaboração de questionários (que podem ser anônimos ou pessoais) onde teremos o feedback dos participantes da oficina. Através deles, faremos uma análise dos pontos positivos e críticos para elaboração de outras dinâmicas de treinamento que fazem parte da programação atual do projeto de extensão em questão.

A título de trabalhos futuros, apresentaremos os resultados das oficinas de Dojo para o treinamento em *Softwares* Livres com a nossa equipe e posteriormente, demonstrarmos a interação com outros trabalhos de pesquisa centrados no desenvolvimento de repositórios e recursos compartilhados da comunidade *Softwares* Livres.

### **5 Considerações Finais**

A atuação de coordenações de TI não deve ficar restrita apenas à instalação de programas e manutenção técnica, e sim, cumprir seu papel de disseminador do conhecimento dentro da comunidade acadêmica. Por isso, acreditamos que os *softwares* abertos sejam as ferramentas principais no ensino e produção de conteúdo, pois trazem consigo uma proposta de não serem vistos como mercadorias, acessível e disponível apenas de acordo com os interesses proprietários. Defendemos que os programas gratuitos e de código aberto deveriam, portanto, encontrar abrigo nas universidades públicas e em seus projetos de extensão, cuja natureza social também é inerente. Hoje, as interfaces dos programas gratuitos têm se tornado cada vez mais simples e semelhantes àsquelas dos conhecidos *softwares* proprietários, o que facilita o aprendizado da equipe.

### **6 Referências Bibliográficas**

BECK, K. (2003). Test-Driven Development: By Example. **The Addison-Wesley Signature Series**. Addison-Wesley.

BORGES, C.; GEYER, C. F. (2003). **Estratégias de governo para promover o desenvolvimento de software livre**. Revista Eletrônica de Sistemas de Informação ISSN 1677-3071 doi: 10.5329/RESI, 2(1).

CARMO, D.; BRAGANHOLO, V. (2012). **Um estudo sobre o uso didático de DOJOS de programação.** In: Workshop de Educação em Computação (WEI). Sociedade Brasileira de Computação.

DELGADO, C.; TOLEDO, R.; BRAGANHOLO, V. (2012). **Uso de Dojos no ensino superior de computação.**

FONTES, R. D.; GOLDSCHMIDT, J. P. (2009) **Diretrizes para Promover a Utilização eo Desenvolvimento do Software Livre no Governo Federal.**

LIMA, C. R. D. S. (2012). **Software Livre: para quê serve?** In: Anais do Congresso Nacional Universidade, EAD e Software Livre (Vol. 1, No. 1).

LUZ, R. B.; NETO, A. (2012). **Usando Dojos de Programação para o Ensino de Desenvolvimento Dirigido por Testes.** In: Anais do Simpósio Brasileiro de Informática na Educação (Vol. 23, No. 1).

REIS, J. P. S.; PENA, L. G. S.; OLIVEIRA, A. G. L. D.; BRAGA, A. F. (2012, November). **Software livre como subsídio em pesquisas na área de Ciência da Computação.** In: Anais do Congresso Nacional Universidade, EAD e *Software Livre* (Vol. 2, No. 2).

SATO, D. T.; CORBUCCI, H.; BRAVO, M. V. (2008). **Coding dojo: An environment for learning and sharing agile practices.** In: Proceedings of the Agile 2008, AGILE '08,, Washington, DC, USA. IEEE Computer Society.

TOLEDO, R. (2012). **O que é Dojo? GLSL Tutorial Sibgrapi.** Disponível em: <http://prezi.com/6z4xre7pt-vx/o-que-e-dojo/>. Acesso em: 20 de junho de 2013.