

Uma Linha de Produto de Software para Construção de Museus Virtuais para Aprendizagem

Luciana A. M. Zaina¹, Andréia D. de Leles², André Duarte², Guilherme Góis¹,
Eline F. de A. Welter¹

¹Departamento de Computação – UFSCar - Campus Sorocaba
Rod. João L. dos Santos, Km 110 – SP-264, CEP 18052-780 - Sorocaba – SP - Brasil

²Faculdade de Engenharia de Sorocaba - FACENS
Marginal Senador José Ermírio de Moraes, 1425, Sorocaba - SP - Brasil
lzaina@ufscar.br, adleles@yahoo.com.br, duarte.andre@hotmail.com,
guijocargo@gmail.com, eliwelter@gmail.com

Abstract. *This article presents a software product line for development of virtual museums for e-learning, which aims to enhance the development in this domain through software reuse and to promote the using of best practices of software engineering in e-learning area. A case study is reported to show the viability of the line and the advance in terms of software quality and productivity.*

Resumo. *Este artigo apresenta uma linha de produto de software para desenvolvimento de museus virtuais para aprendizagem, cujo objetivo é intensificar o desenvolvimento para este domínio através do reuso de software e promover o desenvolvimento de aplicações para a área de aprendizagem utilizando boas práticas da engenharia de software. Um estudo de caso é apresentado demonstrando a viabilidade da linha e os ganhos em termos de qualidade de software e produtividade.*

1. Introdução

A área de aprendizagem eletrônica (*e-learning*) vem atingindo um amadurecimento quanto ao desenvolvimento de aplicações para diferentes propósitos. Com isto, observa-se uma demanda pelo desenvolvimento sistematizado de aplicativos, já que estes tem se tornado cada vez mais complexos, como forma de obter produtos com maior qualidade sob o olhar das disciplinas da Engenharia de Software (ES). Dalmon e Brandão (2013) apontam os resultados de um levantamento realizado com grupos de pesquisa na área de aprendizagem eletrônica no Brasil, observando publicações realizadas no Simpósio Brasileiro de Informática na Educação (SBIE), onde relatam que há pouco rigor de sistematização nos processos de desenvolvimento de produtos nesta área, demandado por soluções que possam contribuir para a produção de aplicativos que contemplem manutenção, reuso e produtividade.

Neste sentido as Linhas de Produto de Software (LPS) podem ser uma solução, pois seu objetivo principal é sistematizar o processo de desenvolvimento de software. Como uma LPS deve atender uma família de produtos, observa-se que os museus virtuais possuem características semelhantes e outras variáveis, convergindo assim para

os fundamentos de uma LPS [Clements e Northrop 2001]. Além do aspecto tecnológico, de agilizar a produção dos museus virtuais, considera-se que a LPS irá intensificar a produção dos museus virtuais, contribuindo para o ensino-aprendizagem - formal, não formal e informal - de diversas áreas do conhecimento como História e Artes [Sartori et al. 2006]. Porém, há uma escassa produção de museus virtuais, sendo que o Brasil dispõe apenas de 22 [IBRAM 2009].

Partindo do pressuposto de intensificar a criação de museus virtuais de uma forma sistematizada sob o ponto de vista de ES e que estes museus possam ser utilizados no ensino-aprendizagem, o objetivo deste artigo é apresentar a LPS e-Museu - uma Linha de Produto de Software que permite a criação de museus virtuais para serem utilizados na *Web*. Não faz parte do escopo da LPS trabalhar com visualizações dos ambientes em 3D, pois os museus terão foco nas obras de arte e não nos seus contextos de imersão em ambientes históricos [Karoulis, Sylaiou e Branco 2006]. A LPS também não irá apresentar soluções de interfaces com o usuário elaboradas, já que estas devem ser desenvolvidas de acordo com públicos específicos. A LPS e-Museu possui um foco funcional e arquitetural, sendo estas as análises apresentadas no artigo.

Como contribuição da LPS pode-se destacar a produção de um conjunto de artefatos que intensifiquem o desenvolvimento de museus virtuais e conseqüentemente contribuindo diretamente com o ensino-aprendizagem independente de aspectos geográficos e a sistematização do desenvolvimento de aplicativos na área de aprendizagem eletrônica. O restante do artigo está organizado em: Seção 2 apresenta a fundamentação teórica; Seção 3 aponta os trabalhos relacionados; Seção 4 apresenta a proposta da LPS e-Museu (Engenharia de Domínio) e um estudo de caso (Engenharia de Aplicação); e por fim a Seção 5 realiza as considerações finais, apontando os trabalhos futuros.

2. Fundamentação teórica

O desenvolvimento de software educativo, que tem sido intensificado nos últimos anos, possui dois alicerces: (i) projeto instrucional – desenvolvido por uma equipe multidisciplinar técnica e pedagógica, (ii) projeto do software – desenvolvimento técnico do produto; sob este olhar há um esforço para trazer as boas práticas da ES para (ii) com o intuito de promover a qualidade e em alguns casos o reuso das aplicações desenvolvidas.

Uma LPS tem como função primordial gerar produtos específicos com base no reuso de uma infraestrutura que contempla: arquitetura de software, componentes, padrões de projeto e métodos de planejamento. Para a eficácia dessa reutilização, utilizam-se componentes de software, que permitem atuar nas similaridades (pontos comuns ou genéricos aos produtos da família) e variabilidades (que devem ser definidas e gerenciadas posteriormente) em uma família de produtos. As LPS são constituídas de: núcleo de artefatos (*core assets*); desenvolvimento sistemático (processo que deverá ser seguido para gerar os artefatos); e delimitadas por um segmento de mercado ou missão. O desenvolvimento de uma LPS é dividida em: Engenharia de Domínio (ED) - processo de identificação e organização do conhecimento acerca de uma classe de problemas (domínio) para apoiar sua descrição e solução de forma sistematizada; e Engenharia de Aplicação (EA) fase em que são construídas as aplicações dentro do domínio específico,

tendo como base o reuso de artefatos do núcleo base, previamente desenvolvidos na ED [Clements e Northrop, 2001].

Os Gerenciadores de Conteúdo (CMS – *Content Management System*) são aplicativos usados para criar, editar, gerenciar e publicar conteúdo, facilitando sua manutenção. Contudo, os CMS não são específicos para domínios e com isto há um esforço grande em adaptá-los a diferentes contextos [Trias 2012].

Considerando tal perspectiva, os autores deste artigo observaram que dentro do domínio de museus virtuais seria possível adotar uma LPS e estruturá-la para que pudesse não apenas ser reusável e extensível, mas que também contribuísse com a utilização dos museus para ensino-aprendizagem. Embora um museu virtual possa ser construído a partir de tecnologias *Web* como HMTL, a prática de seu reuso torna-se muito trabalhosa. Já os CMS, são genéricos demais para trabalhar com os elementos específicos do domínio de museus, dificultando inclusive a estensibilidade da aplicação.

3. Trabalhos relacionados

Os trabalhos relacionados a esta proposta refere-se ao estudo sobre museus virtuais e sobre o uso de LPS para desenvolvimento de ambientes de aprendizagem eletrônica. Observa-se que nos últimos anos tem crescido o número de trabalhos relacionados a adoção de LPS dentro do domínio de aprendizagem eletrônica.

Sob a perspectiva da criação de museus virtuais com fins educacionais, Macedo e Cozz (2005) realizaram um trabalho que mostra o desenvolvimento de sítios museológicos, categorizados como folhetos eletrônicos. Segundo os autores, museu interativo é a categoria que pode ser denominada de museu virtual, por criar um ambiente interativo virtual expositivo para visitação, com expositivas mais detalhadas [Berry, Sheard e Quartly 2011]. Vavoula et al (2009) aborda o potencial do uso de museus virtuais para aprendizagem em um experimento com vinte e três estudantes com idades entre 11-14 anos e professores - no projeto *MyArtSpace*.

Dentro do contexto de desenvolvimento de aplicações para aprendizagem eletrônica são encontrados trabalhos que apontam as vantagens da adoção de uma LPS: uso de desenvolvimento orientado a modelos (MDD – *Model Driven Development*) proposto por Cong et al. (2010) e de serviços web (*web services*) discutido por Zhou et al. (2008). Falvo et al. (2014) propõe uma LPS orientada a serviços (SOA – *Service Oriented Architecture*) - núcleo de artefatos e suas possíveis variabilidades - que dá suporte ao desenvolvimento de aplicativos para dispositivos móveis com fins de aprendizagem. Através da proposta de uma LPS para criação de aplicativos de atividades interativas (iMA), Dalmon e Brandão (2012) (2013) discutem a importância da adoção de uma LPS para melhorar não somente a produtividade e a qualidade do produto, mas também a satisfação dos desenvolvedores com os produtos gerados.

4. LPS e-Museu

Para o desenvolvimento da LPS e-Museu foi adotada a metodologia seguida no desenvolvimento de LPS, Engenharia de Domínio (ED) – estudo e desenvolvimento da LPS propriamente dita – e de Engenharia de Aplicação (EA) – uso da LPS dentro de seu domínio.

4.1. Engenharia de Domínio: desenvolvimento da LPS e-Museu

A ED seguiu as seguintes atividades: (i) estudo e análise do domínio; (ii) especificação da arquitetura; e (iii) desenvolvimento do núcleo de artefatos. Primeiro, para dar suporte ao estudo e análise do domínio (i) foram realizadas duas atividades: estudo comparativo (a) de seis museus virtuais e atividade exploratória (b) com os potenciais usuários para verificação e aceitação das possíveis funcionalidades do museu.

O estudo comparativo (a) teve como objetivo de levantar as principais funcionalidades encontradas nos museus virtuais – além da disponibilização das obras de arte – e verificar os aspectos relacionados a interação (canais de interação). Considerando as recomendações de Karoulis, Sylaiou e Branco (2006) e Marty e Twidale (2004) - que afirmam que os museus virtuais normalmente exibem uma carga excessiva de informação ou interfaces de usuário complexas, alterando a atenção dos usuários dos trabalhos de arte para a ferramenta - foram estudados seis museus virtuais a partir de uma lista disponível na *Web* (Universia Brasil 2014): Van Gogh, Louvre, Britânico, Egípcio, Salvador Dali e Inconfidência Mineira. O estudo permitiu identificar as características (*features*) comuns (presença em mais de um produto da família) e as consideradas opcionais (variabilidades) que se encontra presente em um ou em alguns produtos da família. O resultado do estudo foi importante para demonstrar a viabilidade de desenvolvimento de uma LPS para esse domínio.

Após o estudo comparativo, realizou-se uma atividade exploratória (b), para verificar a aceitação das funcionalidades e o potencial do uso do museu para o ensino-aprendizagem. Primeiro foi realizado uma chuva de ideias (*brainstorming*) com 2 museólogos e 4 professores com intuito de estudar os atores e suas respectivas atividades. Observou-se que para a LPS desejada, um museu com objetivos de ensino e aprendizagem, haveriam 3 atores distintos: museólogo, professor e aluno. O museólogo tem como atividades básicas a catalogação e disponibilização da obra de arte, dos cenários de visitação e da construção de um museu; o professor utiliza os cenários de visitação podendo criar atividades de pergunta/respostas com o intuito de promover a aprendizagem; e os alunos navegam pelos cenários e realizam as atividades de aprendizagem. Após a primeira interação, foi desenvolvido um protótipo funcional, para *Web* e *Android*, que permitiu validar os requisitos levantados no estudo comparativo e retificados no *brainstorming*. Com protótipo funcional foi realizada uma atividade com 174 alunos, 3 docentes de História e 1 museólogo do Museu Histórico Sorocabano, parceiro deste projeto, sendo os docentes e museólogos diferentes dos participantes do *brainstorming* [Welter et al., 2014]. Após a atividade foi realizada uma entrevista de retroalimentação do uso do protótipo com os professores e o museólogo. Tanto os professores quanto os museólogos apontaram a importância que as obras de um museu trazem para o aprendizado da relação entre o passado e o presente. Diante disto, esta característica foi adicionada a LPS como uma variabilidade. A atividade reforçou quais características a família de produtos teria em comum e suas variabilidades.

O resultado da análise do domínio foi a elaboração de um diagrama de características (*features*) (Figura 1) seguindo o método FODA - *Feature-Oriented Domain Analysis*, que apresenta as funcionalidades obrigatórias e suas variabilidades.

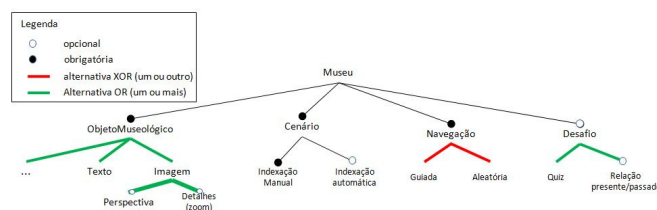


Figura 1. Diagrama de características da LPS.

Após a construção do modelo de características foi realizado o estudo e definição da arquitetura e das funcionalidades da LPS com o objetivo de ser flexível e extensível. Definiu-se que o padrão arquitetural base seria o MVC (*Model View Controller*) já que a LPS teria como plataforma de execução a *Web* e também por este modelo delimitar naturalmente as responsabilidades de cada elemento da arquitetura. Não foram consideradas na LPS aspectos relacionados a interface e interação do usuário, já que o objetivo principal é disponibilizar os elementos funcionais e permitir que as questões de interface-interação possam ser trabalhadas de acordo com o público-alvo. A partir do modelo de características, especificou-se um diagrama de classes conceitual que evidenciou que mesmo nos objetos de domínio (núcleo de artefatos) deveria ser possível a inserção de variabilidades. Optou-se por conduzir o desenvolvimento dos elementos de domínio com a abordagem de Desenvolvimento Orientado a Domínio (DDD - *Domain-Driven Design*) (Vernon 2013). As vantagens em usar o DDD é que ele permite um alinhamento do código com as regras de negócio, facilita o reuso, diminui o acoplamento, e permite o uso do produto independente de tecnologia, características estas fundamentais para uma LPS. O primeiro passo foi identificar os objetos que refletiam a regra de negócio, que no caso da SPL e-Museu são **museu**, **cenário** e **objeto museológico**, sendo **desafio** uma variabilidade.

Seguindo os princípios de um modelo baseado em DDD projetou-se a estrutura arquitetural em camadas independentes. A Figura 2 apresenta a arquitetura que cada componente do domínio deve seguir. A camada **Modelo** é composta pelos componentes entidade, o objeto de valor (DTO) e o padrão de objeto de acesso a dados (DAO - *Data Access Object*) que juntos viabilizam os requisitos de persistência. Já a camada de Serviços é responsável por prover o serviço adequado à camada de aplicação sendo que o componente **Serviço** (padrão Fachada - *Facade*) pode ser referenciado por um **Controle** de uma aplicação, que representa o produto desenvolvido a partir da LPS. Nesta camada ainda existe um serviço **Construtor** (padrão *Builder*) que trata da criação complexa dos principais objetos da LPS - *objeto museológico*, cenário e museu - e é utilizado pela camada serviço. O objetivo do padrão **Construtor** é facilitar a criação de objetos complexos, por meio da invocação de um único método de criação. Foi adicionada à arquitetura um elemento de **Comportamento** cuja função é tratar de componentes que dão comportamento ao museu como a **navegação guiada** - domínio - ou **aleatória** - variabilidade, ou mesmo acrescentar variabilidade como **compartilhamentos** a redes sociais ou outros repositórios. A arquitetura definida permite que a LPS seja independente e flexível para trabalhar com arcabouços na camada de aplicação como *Spring*, *Java Server Faces* entre outros.

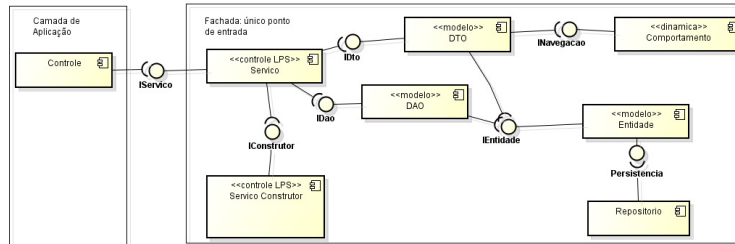


Figura 2. Arquitetura em camadas dos componentes da LPS.

A partir da estrutura arquitetural base de cada elemento do domínio, foram criados os elementos pertencentes ao núcleo de artefatos da LPS, prevendo os respectivos pontos de variabilidade. A Figura 3 apresenta o modelo de componentes indicando o que é mandatório e variável. Os elementos em azul são aqueles que fazem parte do núcleo de artefatos e foram implementados, em laranja são variabilidades disponíveis na LPS e em amarelo são as variabilidades preparadas para serem implementadas por extensão. Observa-se que os principais componentes apresentam funcionalidades mandatórias e de variabilidades, isto porque podem haver extensões nestes pontos. É possível observar internamente em cada componente os conceitos (classes) que podem ser estendidas. Procurou-se implementar as regras de orientação a objetos que induz o reuso (programar para supertipo e abstração). Neste sentido, o padrão de projeto **Estratégia** (*Strategy*), que visa a mudança por extensão foi adotado permitindo que novos requisitos possam ser adicionados como pontos de variabilidade. Deve-se salientar que além dos componentes, faz parte do núcleo de artefatos da LPS e-Museu a documentação (modelos e documentos) e os testes unitários automatizados que foram desenvolvidos para os componentes mandatórios da LPS (objeto museológico, cenário e museu). Para implementação da LPS foram usadas as seguintes tecnologias: Java, *Eclipse Link*, *Maven* e *JPA*.

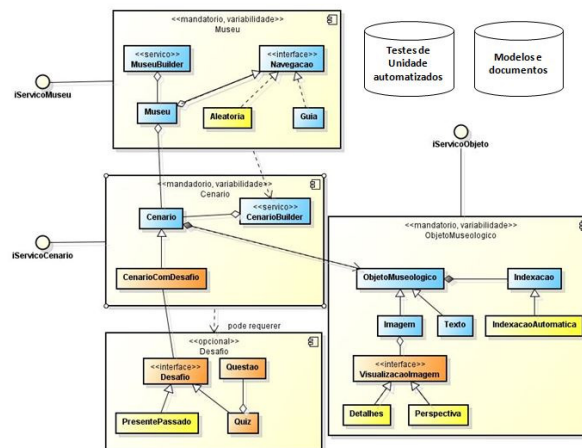


Figura 3. Estrutura da LPS - componentes, testes e modelos.

4.2. Implementação a partir da LPS - Engenharia de Aplicação

Na EA os componentes de software da ED são reutilizados para o desenvolvimento das aplicações do domínio. Neste trabalho adotou-se o mesmo método de verificação da

LPS empregado por Dalmon e Brandão (2013), comparando as visões dos mesmos desenvolvedores e os resultados produzidos utilizando e não utilizando a LPS para desenvolver um produto.

Para avaliar o uso da LPS proposta foi conduzido um estudo de caso com o objetivo de *comparar o desenvolvimento de um museu virtual sob o aspecto funcional com e sem utilizar a LPS*. Um ponto primordial neste estudo é que o escopo era a *verificação funcional e arquitetural*. Um grupo de quatro estudantes do último de graduação em Engenharia da Computação da FACENS desenvolveu um museu virtual sem utilizar a LPS e depois utilizando a LPS. Para que a futura análise pudesse ser realizada, foram realizadas (i) *entrevistas de acompanhamento* durante o uso e não uso da LPS e ao final foram levantados (ii) *indicadores* sobre os aspectos arquiteturais dos dois produtos gerados. Antes do início do estudo de caso realizou-se uma entrevista com o grupo para caracterizar o perfil dos participantes, onde dos quatro participantes: (a) dois já estagiavam em empresas de desenvolvimento utilizando a linguagem Java para *Web*; os outros dois tinham experiência em desenvolvimento apenas na academia; (b) apenas um conhecia as tecnologias Eclipse Link e JPA; (c) nenhum deles havia utilizado o Maven ou uma LPS previamente; (d) todos conheciam padrões de projeto, mas nunca haviam empregado em projetos reais. Dada a caracterização foi realizado um aquecimento de nivelamento de conhecimento com os participantes (tecnologias usadas para construção da LPS), que durou três semanas, porém sem apresentar a LPS para que não houvesse influência durante o não uso da LPS.

Após o aquecimento foi conduzida a etapa (A) de *não uso da LPS*. Foi apresentado ao grupo o domínio do problema e solicitado que realizassem a implementação de um museu considerando: o uso das tecnologias então conhecidas e uma arquitetura que fosse facilmente extensível, com baixo acoplamento e alta coesão. Durante um mês e meio, o grupo seguiu as fases de ES, realizando anotações sobre tempo gasto em cada fase, e com entrega de artefatos: (i) Análise – documento com os requisitos funcionais, não-funcionais e um diagrama conceitual, (ii) Projeto – diagrama de classes parcial visando a arquitetura, (iii) Implementação – código fonte e (iv) Testes – apenas de validação, não foram desenvolvidos testes unitários automatizados. Durante e ao final do desenvolvimento, foram conduzidas entrevistas buscando coletar informações sobre os (a) pontos positivos e (b) negativos, incluindo as dificuldades encontradas. Os integrantes relataram que: (a) “foi uma experiência interessante e difícil, pois não estamos acostumados a desenvolver pensando na arquitetura; buscamos normalmente apenas desenvolver o código funcional”; (b) “o desenvolvimento de um software pensando nos requisitos arquiteturais demanda muito tempo de análise, principalmente porque os requisitos funcionais deverão ser encaixados na arquitetura. Deixamos de realizar testes no final porque já queríamos terminar o produto e entregá-lo”.

Num segundo momento, foi realizada a etapa (B) *uso da LPS*. Inicialmente, foi apresentado ao grupo os modelos arquiteturais (Figura 2 e 3) e todos os artefatos da LPS. Não foi realizado um novo aquecimento, pois os grupos já conheciam os conceitos e tecnologias, e também porque a LPS provê modelos que dão suporte ao seu uso. Isto também seria uma maneira de verificar a necessidade ou não de outros artefatos que auxiliassem o uso da LPS. Da mesma forma, durante vinte dias, seguiu-se as mesmas fases de ES da etapa (A), sendo que na etapa (B) foram entregues artefatos: (i) um

modelo de classes do museu que seria desenvolvido; (ii) um modelo refinado do anterior com a arquitetura da LPS proposta; (iii) museu desenvolvido; (iv) casos de testes automatizados executados. Como na etapa (A), também foram realizadas *entrevistas* de acompanhamento e de finalização com o grupo, em que eles já apontavam um comparativo sobre o uso e não uso da LPS: (a) “sentimos um aumento de produtividade grande e a arquitetura foi nos guiando durante o desenvolvimento; os testes unitários nos ajudaram nas primeiras verificações nos dando confiança sobre o caminho que estávamos seguindo no desenvolvimento; o código ficou mais legível, separado por responsabilidades e com isto, ficou mais fácil de dar manutenção; a extensão da LPS é demais!”; (b) “no começo tivemos dificuldades em encaixar os requisitos na arquitetura, mas depois conseguimos; a preparação do uso do ambiente para usar a LPS foi demorado na primeira vez”. A Figura 4 apresenta a interface para preparação do cenário e a visão de navegação no museu.

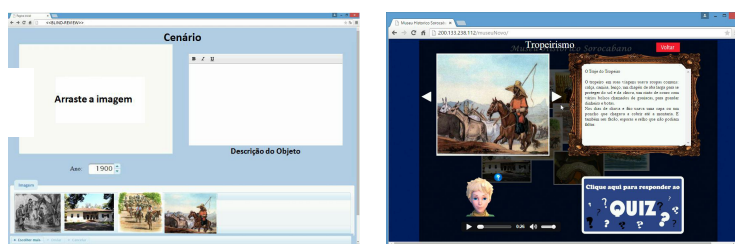


Figura 4 - Preparação do cenário e navegação no museu.

A partir do acompanhamento das atividades dos alunos, observou-se que, embora a LPS demande um tempo de aprendizado inicial, ela propicia benefícios para que os produtos sejam aderentes a uma arquitetura, estimulando a divisão natural de responsabilidades do código e fomentando uma melhor legibilidade e manutenção. Outro ponto importante, foi que a documentação provida pela LPS auxiliou no aprendizado de seu uso e de suas potencialidades. Os testes unitários automatizados melhoraram a produtividade e qualidade, permitindo que estes fossem executados durante o desenvolvimento. Na etapa (A) sem a LPS o grupo havia deixado de executar os testes.

Além das *entrevistas de acompanhamento* (i) também foi realizado um comparativo entre os códigos fontes gerados nas etapas (A) e (B) através de *indicadores* (ii), selecionados de Li e Henry (1993). Os indicadores buscam verificar a falta de coesão, o alto acoplamento e os métodos ponderados por classes, que são métricas que indicam o quanto um produto atende requisitos de qualidade como reuso, manutenção e complexidade. Além disto, foi verificada a produtividade - linha de códigos geradas (LOC) - manual e por reuso/automaticamente - por unidade de tempo, e a divisão do código pelas camadas da arquitetura. Para geração das métricas foi utilizado o *plugin* do NetBeans - Source Code Metrics. A partir da Tabela 1(a) pode-se observar que o produto desenvolvido *sem a linha* apontam deficiências em atributos relevantes. A partir das entrevistas e da análise pontual do código produzido, foi possível observar que a arquitetura proposta pela LPS influenciou fortemente no resultado. Considerando, os resultados da Tabela 1(b) verifica-se que houve um esforço menor para geração de código *com o uso da LPS* (maior valor para código reusado). Algum código foi gerado automaticamente *sem o uso da LPS* já que os participantes usaram uma IDE para o desenvolvimento. Outro indicador interessante é a distribuição do código pelas camadas.

Na etapa (B) o maior peso está concentrado na camada **Modelo**, que pelas boas práticas arquiteturais é quem concentra a lógica do negócio. O uso do padrão Fachada contribuiu para uma boa separação das responsabilidades na arquitetura. Em contrapartida, na Etapa (A) houve uma alta concentração de código na camada de **Controle**, o que representa uma distribuição das responsabilidades não adequadas.

Tabela 1. Métricas do código fonte das etapas A e B.

Métricas	Etapa A - sem LPS		Etapa B - com LPS	
	Médi a	Máximo	Médi a	Máxim o
Métodos ponderados por classe	11,73	32	6,26	10
Acoplamento entre classes	1,76	5	0,31	1
Falta de Coesão entre métodos	4,87	15	2,66	7

(a)

Métricas	Etapa A - sem LPS	Etapa B - com LPS
LOC - manual	702	333
LOC - reusado/gerado	77	1218
Tempo gasto (h)	33	14
Média (LOC/h)	23,6	110,79
Camada com maior LOC	52% - Controle	40% - Modelo

(b)

5. Conclusões e Futuros Trabalhos

Este trabalho apresentou uma LPS com o objetivo de desenvolver museus virtuais com foco na aprendizagem. Acredita-se que este trabalho trouxe contribuições que demonstram a viabilidade de utilizar as técnicas de reuso dentro do contexto de desenvolvimento de software que apoiem a aprendizagem introduzindo boas práticas de desenvolvimento nesta área de pesquisa. Também apresenta contribuições importante para a sociedade, pois a LPS proposta permite intensificar o desenvolvimento de museus virtuais que poderão ser utilizados por alunos de diferentes níveis escolares. São trabalhos futuros: testes com outras funcionalidades da LPS; estudo sobre a flexibilização da arquitetura para ser utilizada por diferentes tipos de interface do usuário; e desenvolvido um museu virtual completo, a partir da parceria com a secretaria da cultura do município de Sorocaba, que será disponibilizado para uso da aprendizagem de História e Artes.

6. Agradecimentos

Agradecemos o apoio financeiro da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) ao projeto, processo nº 2013/25572-7.

Referências Bibliográficas

- Berry, G.; Sheard, J.; Quartly, M. (2011) “A Virtual Museum of Computing History: an educational resource bringing the relationship between people and computers to life”. In: 13th Australasian Computer Education Conference.
- Clements, P.; Northrop, L. (2001). Software Product Lines: Practices and Patterns. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Cong, X.; Zhang, H.; Zhou, D.; Lu, P.; Qin, L. (2010) “A Model-Driven Architecture Approach for Developing E-Learning Platform”, Edutainment, p. 111-122.
- Dalmon, D. L., Brandão, L. O. (2012) “Uma Linha de Produto de Software para Módulos de Aprendizagem Interativa”. In: Simpósio Brasileiro de Informática na Educação (SBIE 2012), p. 1-10.

- Dalmon, D. L.; Brandão, L. O. (2013) “Sobre o Desenvolvimento de Software Educacional: proposta de uma Linha de Produto de Software para Módulos de Aprendizagem Interativa”. *Revista Brasileira de Informática na Educação*, Vol. 21, p. 113-130.
- Falvo, V.; Duarte Filho, N.F. ; Oliveira, E.; Barbosa, E.F. (2014) “A contribution to the adoption of software product lines in the development of mobile learning applications”. In: *Frontiers in Education Conference, IEEE*, p. 1-8.
- IBRAM (2009) “Portal de Comunicação: 94% dos brasileiros nunca visitaram um museu”. Disponível em: <http://www.clicrbs.com.br/especial/rs/portal-social/19,0,3054406,94-dos-brasileiros-nunca-visitaram-um-museu.html>. Acesso em: 02/05/2013.
- Karoulis, A.; Sylaiou, S.; White, M. (2006) “Usability Evaluation of a Virtual Museum Interface”. *INFORMATICA*, Vol. 17, N. 3, p. 363–380.
- Li, W.; Henry, S. (1993) “Object-oriented metrics that predict maintainability”. In: *Journal of Systems and Software*, Vol 23, N. 2, p. 111-122.
- Macedo, D.P.S; Cozz, S.S.M (2005) “A realidade virtual na museologia: uma análise das vantagens e desvantagens para o turismo cultural”. *Revista Científica Ciência e Administração*, Vol. 11, N. 2, p. 229-240.
- Marty, P.; Twidale, M. (2004) “Lost in gallery space: a conceptual framework for analyzing the usability flaws of museum web sites”. *First Monday*, Vol. 9, N. 9.
- Sartori, A.S; Maximiano, F.; Porto, S. C. D.; dos Santos, S. M. (2006) “Museus Virtuais: Memória e Educação em Tempos de Ciberespaço”. *Intercom Junior - Jornada de Iniciação Científica em Comunicação*. Disponível em: <http://www.intercom.org.br/papers/nacionais/2006/resumos/R1490-1.pdf>.
- Trias, F. (2012) “Building cms-based web applications using a model-driven approach”. In: *Sixth International Conference on Research Challenges in Information Science (RCIS)*, p. 1-6.
- Universia Brasil (2012) “46 museus virtuais para você visitar de graça”, Disponível <http://noticias.universia.com.br/destaque/noticia/2012/02/16/912114/46-museus-virtuais-voce-visitar-graca.html>.
- Vavoula, G.; Sharples, M; Rudman, P.; Meek, J. ; Lonsdale, P. (2009) "Myartspace: Design and evaluation of support for learning with multimedia phones between classrooms and museums". In: *Computers & Education*, Vol. 53, No. 2, p. 286–299.
- Vernon, V. (2013) *Implementing Domain-Driven Design*. Pearson.
- Welter, E. F. A.; Zaina, L. A. M.; Leles, A. D.; Ribeiro, M. S. (2014) “Aprendendo História através de Museus Virtuais: uma parceria entre Professores e Museólogos”, In: *20º Workshop de Informática na Escola (WIE 2014)*, 3º Congresso Brasileiro de Informática na Educação (CBIE 2014), Dourados, p. 76-85.
- Zhou, D., Zhuo, Z., Zhong, S., Pan X. (2008) "The Design of Software Architecture for E-Learning Platforms". In: *Edutainment*, p. 32-40.