

Provendo um Serviço Web para Interação e Coleta de Dados de Aplicativos Educacionais

Walter Gomes, Pedro Castro, Emanuella Cardoso, Marcelo Malheiro,
Rafael Castaneda, Gustavo Paiva Guedes, Renato Mauro, Eduardo Ogasawara

¹Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ

{rcastaneda, gguedes, rmauro}@cefet-rj.br

eogasawara@ieee.org

Abstract. *Due to the large amount of mobile devices, educators and researchers are faced with many opportunities to explore and experiment different m-learning applications. In this scenario, there is a demand for tools that both interact and collect data from these applications. However, it is a common practice to develop these tools in an ad-hoc way, which makes them very strongly coupled to the m-learning applications. This inhibits the possibility to reuse software and also makes it difficult to analyze all applications in a holistic way. To address these issues, this paper presents SWEI, a web service that enables both interaction and data collection with different types of m-learning applications. Additionally, SWEI integrates application data in a data warehouse. SWEI was evaluated through a set of pilot m-learning applications and seems to be a promising approach.*

Resumo. *Devido a grande quantidade de dispositivos móveis, educadores e pesquisadores se deparam com oportunidades para explorar e experimentar diferentes aplicativos de m-learning. Nesse cenário existe uma demanda por ferramentas que possibilitam tanto a interação quanto a coleta de dados a partir desses aplicativos. Entretanto, é uma prática comum desenvolver essas ferramentas de modo ad-hoc, o que as tornam muito acopladas aos aplicativos de m-learning. Isso inibe o reuso de software e dificulta análise dos diferentes aplicativos por meio de uma visão holística. Para abordar esses problemas, este artigo apresenta o SWEI: um serviço web que possibilita tanto a interação quanto a coleta de dados a partir de diferentes aplicativos de m-learning. Adicionalmente, o SWEI integra os dados coletados pelos aplicativos em um data warehouse. O SWEI foi avaliado por meio de um conjunto de aplicativos piloto e mostrou-se uma abordagem promissora.*

1. Introdução

No cenário atual da área da educação, a adoção de recursos tecnológicos têm sido amplamente difundida. Essa tendência resultou na adoção do conceito do ensino eletrônico (do inglês, *e-learning*). Universidades, escolas e demais instituições de ensino utilizam o ambiente web para facilitar o compartilhamento de conteúdo e comunicação com os alunos [Hershkovitz and Nachmias, 2011]. Nessa mesma linha, há um número crescente de jogos e aplicações no âmbito educacional que abordam uma determinada área de conhecimento e possibilitam que o aluno pratique determinados conceitos de forma intuitiva e interativa.

Mais recentemente, os conceitos já utilizados via ensino eletrônico passaram a serem incorporados no ambiente *mobile* [Georgiev et al., 2004]. Essa nova plataforma permite que os recursos tecnológicos possam ser utilizados em qualquer lugar e a qualquer momento.

Os aplicativos baseados em ensino eletrônico proporcionam o acesso de seu conteúdo a um número grande de usuários. Conseqüentemente, o tráfego e quantidade de dados produzida é considerável. A coleta de dados em prol de uma análise posterior para melhorar métodos e abordagens de ensino tem sido foco de estudo [Romero and Ventura, 2007]. No contexto das aplicações educacionais, a quantidade de acessos a um determinado conteúdo, o padrão de respostas dos alunos sobre um questionário, ou, em um jogo, a qualidade do algoritmo usado pela aplicação oponente são exemplos de questões que interessam tanto a desenvolvedores quanto a educadores. A partir dessas informações é possível aprimorar as técnicas utilizadas, identificar dificuldades apresentadas pelos alunos, avaliar a eficiência desses métodos de ensino e traçar novas estratégias. Um outro grupo de aplicativos educacionais compreende aqueles que necessitam, durante a execução, acessar um repositório para, por exemplo, contextualizar as informações de sua interface dependendo de algum estado ou localização [Brassai et al., 2014]. Finalmente, existem aplicações que necessitam da combinação destas duas propriedades [Sibaldo et al., 2006].

Nesse diapasão, é de suma importância a existência de mecanismos para gerenciar essas interações. É comum à maioria das aplicações existentes, ter a sua estrutura de interação baseada em serviço web (do inglês, *web service*). A característica geral dos serviços implementados é ser fortemente acoplado à aplicação, inibindo a sua modularização e seu reuso. Isso impede, por exemplo, realizar uma análise consolidada dos dados provenientes de duas ou mais aplicações distintas pela falta de uniformidade nos dados que, eventualmente, é potencializada quando se tratam de plataformas diferentes (por exemplo, *mobile* versus *desktop*).

Sendo assim, esse artigo busca explorar o SWEI, o Serviço Web Educacional Integrado, que tem como meta maior modularização e versatilidade da interação entre o aplicativo *m-learning* e o servidor educacional por meio de um serviço web integrado e expansível. Sua arquitetura permite que o serviço seja incorporado de modo fácil a um grande número de aplicações. Além disto, o SWEI consolida os dados em forma de *data warehouse*, abrigando dados de diversas aplicações ao mesmo tempo, mesmo quando oriundas de plataformas distintas.

Além desta introdução, este trabalho está organizando em mais cinco seções. A seção 2 desse artigo revisa conceitos dos serviços web, suas características e abordagens para armazenamento e transmissão de dados. A seção 3 destaca os trabalhos relacionados. A seção 4 apresenta a metodologia: com a arquitetura e o funcionamento do SWEI, enquanto a seção 5 relata os resultados e as avaliações experimentais. Finalmente, a seção 6 apresenta as considerações finais.

2. Serviços Web

A utilização de serviço web tem sido adotada como forma de prover comunicação entre aplicações de diferentes plataformas. O serviço web é uma aplicação com finalidade bem definida que pode ser integrada a outra aplicação mais complexa, utilizando um modelo de dados e de comunicação padronizados sob protocolos gerais da internet [Sheng et al.,

2014]. No que se diz à implementação de um serviço web, existem três pontos principais: (i) a escolha do protocolo de comunicação a ser usado; (ii) o modelo de dados a padronizar a comunicação; e (iii) a forma de interação a ser exercida pelo serviço.

2.1. Protocolos de Comunicação

Os protocolos de comunicação são conjuntos de regras que são usados para transmitir dados de um computador a outro. Os primeiros serviços web adotavam o *SOAP* (do inglês, *Simple Object Access Protocol*), que é um protocolo para troca de informações estruturadas em ambientes descentralizados e distribuídos [Silberschatz et al., 2010]. Ele é versátil e extensível, podendo utilizar diversos protocolos de transporte. A maior desvantagem do *SOAP* é a sua verbosidade.

Recentemente, começou-se a adotar o padrão *RESTful* (do inglês, *Representational State Transfer*). O *RESTful* encapsula o seu conjunto de regras e restrições nas primitivas gerais de sistema web tradicional [Silberschatz et al., 2010]. Esse padrão vem sendo comumente adotado por sua estrutura mais simples e mais fácil de atualização do que o *SOAP*.

2.2. Modelo de Dados para Comunicação

Os modelos de dados para comunicação padronizam as relações entre elementos de dados que trafegam entre a aplicação cliente e o serviço web. Destacam-se dois padrões: XML e JSON. O formato XML (do inglês, *Extensible Markup Language*) se tornou uma linguagem padrão para representação de dados. Ele pode ser utilizado independentemente da plataforma, o que representa uma grande vantagem em ambientes heterogêneos como a internet [Silberschatz et al., 2010]. A desvantagem do XML é a sobrecarga de desempenho tanto na transmissão quanto no processamento análise (*parse*) sobre o documento [Rodrigues et al., 2011].

O formato JSON (do inglês, *Java Script Object Notation*) por outro lado é considerado mais leve e intuitivo para desenvolvedor, podendo ser facilmente gerado e analisado pelas aplicações [Silberschatz et al., 2010]. Esse formato tem sido comumente adotado atualmente devido as suas vantagens: menor verbosidade, menor tamanho e facilidade de *parse*.

2.3. Formas de Interação

Um serviço web opera em modo cliente-servidor, mas a intensidade pela qual os dados trafegam entre cliente e servidor estabelece a forma de interação, caracterizando cenários de coleta de dados, acesso a dados ou uso combinado. Na coleta de dados, uma aplicação cliente somente submete dados de interesse para o serviço web, que posteriormente faz o armazenamento em um repositório. No sentido oposto da coleta, a aplicação cliente pode apenas solicitar dados do serviço web, fornecendo um conteúdo contextualizado com a demanda da aplicação cliente. Finalmente, quando a finalidade é de uma interação completa, a aplicação cliente tanto envia quanto solicita dados para o serviço web. De acordo com o tipo de interação, aumenta-se ou diminui-se o acoplamento entre a aplicação cliente e o serviço web.

3. Trabalhos Relacionados

A análise de dados provenientes de ambientes educacionais tem sido o foco de alguns trabalhos na área da computação. As ferramentas educacionais são cada vez mais numerosas e diversas, envolvendo a utilização de jogos, aplicativos educacionais e LMS (do inglês, *Learning Management Systems*). Os dados produzidos por essas ferramentas podem ser analisados com o objetivo de acompanhar o desenvolvimento dos estudantes, identificando as dificuldades observadas, aprimorar as técnicas de ensino e fornecer informações para elaboração de políticas educacionais. Abordagens como a de Romero et al. [2008] utilizam a base do Moodle para analisar o comportamento dos usuários. Esse tipo de abordagem é complementar ao SWEI, uma vez que ambos podem compartilhar a mesma base de dados e produzir uma visão holística das aplicações.

O conceito de um modelo para coleta de dados advindos de uma ou múltiplas fontes de dados é explorado por Hansen and Pinto [2003] e Ruchika and Goel [2012], respectivamente. A arquitetura empregada é a *SOAP* com *XML*. A utilização dessa arquitetura resulta em uma necessidade maior de desenvolvimento tanto por parte de quem fornece os dados, quanto para quem os recebe e os armazena.

Diversos trabalhos empregam os serviços web como ferramenta para a comunicação com aplicações ricas (como *m-learning*), com objetivo de interação, acesso ou coleta de dados de um repositório. No GeoQuesting [Brassai et al., 2014], adota-se a arquitetura *RESTful* com *JSON* de modo a obter informações contextualizadas pela localização fornecida pelo GPS. A cada localização previamente cadastrada, formulários específicos sobre o local/atração são apresentados. Finalmente, existem situações nas quais há a necessidade de acessar e atualizar um repositório educacional [Sibaldo et al., 2006].

A busca por trabalhos relacionados ao tema mostrou a carência de serviços que possam ser empregados de modo mais generalizado, tendo um mesmo serviço disponível para amplo espectro de aplicações. Essa carência aumenta quando se procuram serviços baseados em arquitetura *RESTful* com *JSON*. Sendo assim, a proposta do SWEI, ao fornecer um serviço integrado para envio, recebimento e interação entre aplicações, apresenta-se como relevante.

4. SWEI – Serviço Web Educacional Integrado

Esta seção descreve o SWEI e encontra-se organizada em quatro subseções. A seção 4.1 apresenta a sua arquitetura. A seção 4.2 apresenta o *data warehouse* proposto. Finalmente, as seções 4.3 e 4.4 apresentam dois benefícios gerais do SWEI, a saber: visão integrada das aplicações e reutilização.

4.1. Arquitetura

O SWEI é um serviço web modularizado para ser utilizado no contexto de aplicações educacionais. Adota como principal conceito a versatilidade, posicionando-se como um serviço que faz coleta de dados e, ao mesmo tempo, viabiliza a implementação de pacotes [Birsan, 2005] para o desenvolvimento de aplicações interativas. No que se refere à camada servidora, a Figura 1 apresenta os componentes. O SWEI adota a arquitetura *RESTful* com *JSON* e banco de dados PostgreSQL para *data warehouse*.

Pelo lado do cliente, a Figura 1 apresenta também a indicação da instanciação de quatro aplicações piloto. O ControlHarvest [Alves et al., 2014] é um jogo para *desktop* onde são abordados conceitos sobre controle biológico. O LabVetor [Honorato et al., 2015] e o JOE [Paschoal et al., 2014] são aplicações *m-learning*: a primeira aborda o ensino de vetores, enquanto a segunda explora os conhecimentos sobre ortografia da língua portuguesa. As três aplicações citadas utilizam o SWEI exclusivamente para enviar os dados gerados durante as atividades dos usuários. As setas indicam que o fluxo de dados é apenas da aplicação para o SWEI, que internamente faz armazenamento dos dados coletados no SGDB. A quarta aplicação apresentada é o Amê [Silva et al., 2015] que faz uso do modo interativo do SWEI, demandando um pacote próprio.

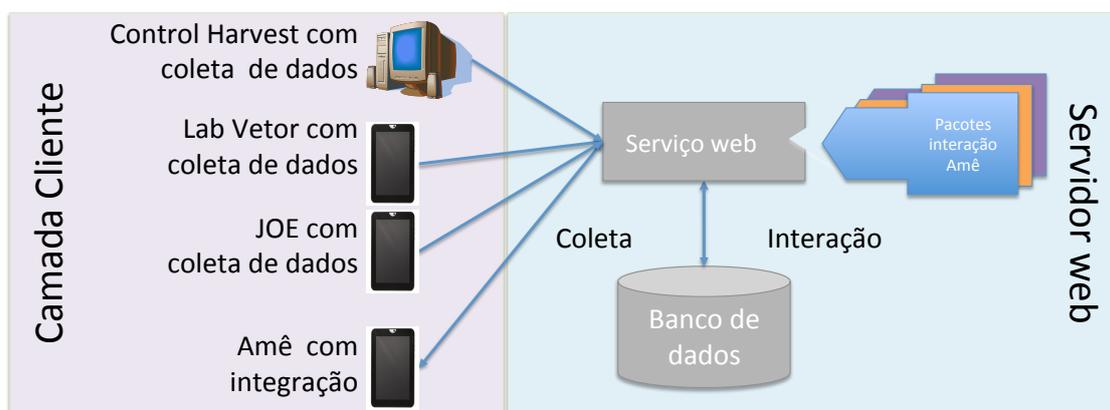


Figura 1. Arquitetura do SWEI

No que tange ao mecanismo de coleta de dados, o objetivo do SWEI é ser bastante genérico, *i.e.*, pode ter a sua interface cliente instanciada em um número bem distinto de aplicações. Um bom exemplo é o LabVetor que foi desenvolvido para a plataforma mobile usando AppInventor. Nela, basta montar uma URL com o conteúdo codificado em JSON que o SWEI o armazena.

Finalmente, no cenário interativo, os pacotes de interação são módulos específicos desenvolvidos para uma determinada aplicação. Nesse contexto existe uma maior dependência entre SWEI e a aplicação cliente. Para que o serviço funcione corretamente é preciso implementar o pacote no lado do servidor. A aplicação cliente submete, ao longo da sua execução, pedidos de dados ao SWEI, que são tratados pelos pacotes customizados. A grande vantagem é que o SWEI abstrai toda a parte da comunicação, deixando que o desenvolvedor foque apenas em desenvolver o protocolo de mensagens trocadas.

4.2. Data warehouse

Os dados coletados pelo SWEI são automaticamente consolidados em um *data warehouse* instanciado em um banco de dados PostgreSQL. O modelo de dados é apresentado pelo diagrama de classes (notação UML) da Figura 2. Ela corresponde a uma constelação de fatos [Silberschatz et al., 2010]. No diagrama, *Usó* e *Item* (indicados em amarelo) são fatos e as demais classes (indicadas em azul) correspondem às dimensões. O *Usó* é caracterizado quando o usuário inicia uma atividade na aplicação, e as informações relacionadas a tempo e localização são armazenadas. Usando, como exemplo, o Amê, ao iniciar o jogo as informações sobre data/hora e localização são coletadas. Ao término do

jogo, existe uma coleção de itens (representados pela classe *Item*) com os valores associados ao jogo. Os valores, nesse caso, são a pontuação do jogador, do algoritmo desafiado e o seu tipo. A cada diferente aplicação, existe uma série de itens que apresentam diferentes domínios e tipos de dados, podendo ser textos, números reais, inteiros, data e dados complexos, que no diagrama são abstraídos por um objeto.

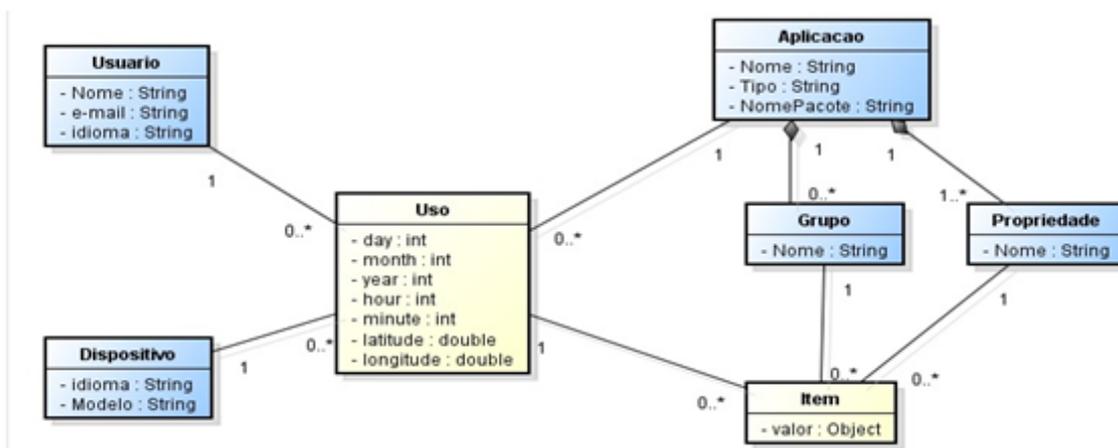


Figura 2. Representação do data warehouse em UML

As classes de dimensões são responsáveis por adicionar e manter informações importantes, principalmente na hora de realizar consultas e integração dos resultados. *Usuário* representa a pessoa que está utilizando uma determinada aplicação, a partir de um *Dispositivo*, que pode ser um *tablet*, *smartphone* ou um *desktop*. A *Aplicação* pode ser de diferentes tipos (*mobile*, *desktop*, etc). Em alguns casos, a aplicação pode ter a especificação do pacote para a interação. Uma aplicação determina as propriedades (representada por *Propriedade*) que os valores de um item estão associados. No caso do *LabVetor*, a direção do vetor *a*, ou o comprimento do vetor *b*, por exemplo. Cada aplicação possui um conjunto de propriedades. Assim, como propriedades, cada aplicação define um *Grupo*, responsável por agrupar os itens. *Grupo* e *Propriedade* são partes de *Aplicação*.

4.3. Visão integrada das aplicações

A presença de um repositório centralizado contendo dados de diversas aplicações pode ser muito útil na hora de analisar os resultados por meio de uma perspectiva holística. As pessoas interessadas em realizar análise de dados ou aplicar técnicas de mineração de dados tem, à disposição, uma fonte abundante de material em um único repositório. Nesse modelo, torna-se possível comparar aplicações e abordagens com propósitos similares de forma facilitada. Principalmente, ao avaliar os resultados produzidos por diferentes aplicativos, pode-se obter um retorno importante para aprimorar as estratégias educacionais usadas pelas aplicações.

4.4. Reutilização

Em muitas aplicações já desenvolvidas não foi levada em consideração uma estrutura sistêmica para coleta de dados. Cada aplicação acaba concebendo uma solução própria,

gerando muito retrabalho e aumentando a quantidade de erros de programação. No desenvolvimento de uma nova aplicação, as funcionalidades fornecidas pelo SWEI não precisam ser desenvolvidas novamente, bastando incorporá-las ao projeto, reduzindo o trabalho e número de defeitos. A utilização do SWEI permite aos desenvolvedores focar no objetivo da aplicação, abstraindo a dificuldade de modelagem de serviços de coleta e integração de dados.

5. Avaliação Experimental

O SWEI foi concebido para apoiar a análise dos dados produzidos por diferentes aplicativos educacionais. Foram selecionadas algumas aplicações para incorporar o módulo de cliente do SWEI e permitir a interação com o serviço web. Essas aplicações originalmente não possuíam um sistema de coleta e análise de dados. Dessa forma, o SWEI foi integrado às seguintes aplicações: LabVetor [Honorato et al., 2015], ControlHarvest [Alves et al., 2014], Amê [Silva et al., 2015] e JOE [Paschoal et al., 2014]. Nessas aplicações, houve um contato com os responsáveis pelo desenvolvimento dos projetos para que os mesmos incluíssem no código de suas aplicações, as primitivas para coleta de dados usando o SWEI.

Para apresentar o potencial do SWEI, apresentamos o seu uso em duas aplicações: LabVetor e Amê. Elas representam, respectivamente, um aplicativo codificado no *AppInventor* e outro no *Android Studio*. Ambos os aplicativos podem ser encontrados no *Google Play Store*. O LabVetor foi utilizado predominantemente por alunos do ensino médio que puderam testar seus conhecimentos de vetores. Já o Amê é um jogo para a plataforma *Android* voltado para o aprendizado de mecanismos de buscas em inteligência artificial. No Amê, o cliente de serviço web foi implementado para fins de interação, tanto para armazenar resultados dos jogos, como pontuação do jogador e do algoritmo do oponente, quanto para receber dados, como, por exemplo, ranking de desempenho dos algoritmos de busca.

A partir do preenchimento das tabelas de fatos pelo SWEI, pode-se realizar as consultas de análise. Como a base de dados tem topologia de constelação, para analisarmos os dados das aplicações é necessário realizar consultas baseadas em tabulação cruzada (do inglês, *cross-tab*) [Silberschatz et al., 2010]. A Figura 3.a apresenta a visão *v_hanafuda* que integra as tabelas de fatos com as dimensões por meio de uma consulta de tabulação cruzada. Para tanto, a visão usa uma função de agrupamento, que possibilita visualizar, a cada jogo, o algoritmo utilizado pela aplicação e as respectivas pontuações. No resultado da consulta, os valores *computer points* e *player points*, originalmente presentes na tabela de fatos, passam a ser vistos como atributos na tabulação cruzada. A utilização desse recurso é importante no processo de tomada de decisão [Gray et al., 1997] e possibilita análises mais sofisticadas usando métodos estatísticos e mineração de dados.

O resultado das consultas realizadas a partir das visões foram importados para o ambiente R [R Development Core Team, 2008] para fins de manipulação dos dados e criação de gráficos que ilustram os padrões e comportamentos das atividades exercidas pelos usuários. Os gráficos gerados correspondem a diferentes perspectivas fornecidas pelo *data warehouse*. Neste artigo, o objetivo não é realizar análises individuais de cada aplicativo, mas sim apresentar o potencial das consultas e análises que o SWEI é capaz de oferecer. É importante ressaltar que a qualidade das análises depende da qualidade dos

dados que os aplicativos enviam para o SWEI, *i.e.*, são ditados pelo desenvolvedores dos aplicativos.

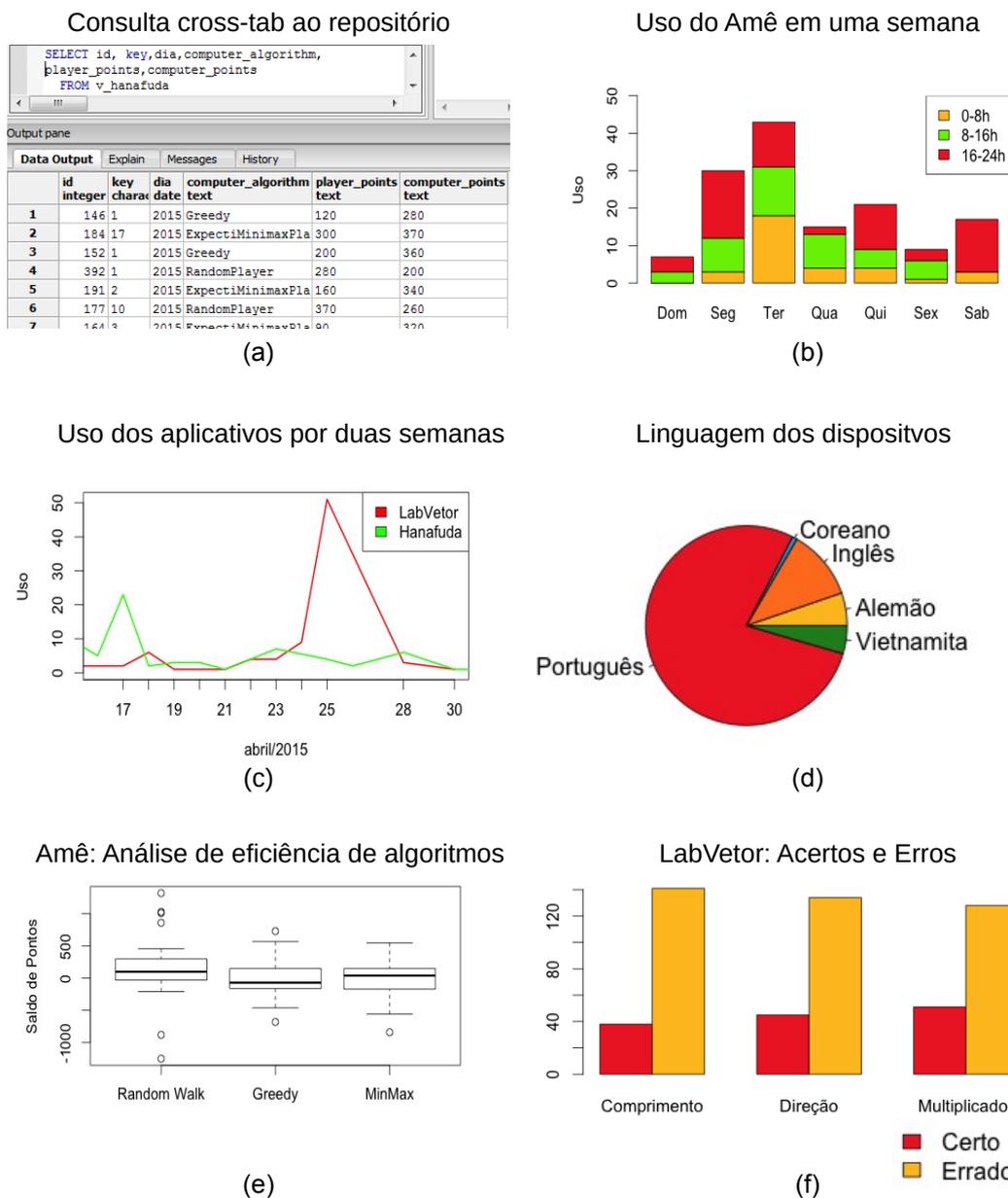


Figura 3. Análise do uso do SWEI

No que se refere ao monitoramento de uso, a Figura 3.b apresenta o horário do uso do aplicativo Amê. Nota-se que, por ser um jogo, o seu maior uso foi nos horários do final do dia. Por ser uma base integrada, é possível avaliar o uso dos diferentes aplicativos ao mesmo tempo. A Figura 3.c apresenta o uso dos aplicativos ao longo dos dias. Note que o aplicativo LabVetor foi muito utilizado no dia 25, que foi o dia no qual alunos do ensino médio realizaram um experimento controlado com o aplicativo. Outra consulta global pode ser vista na Figura 3.d, na qual é apresentada a linguagem utilizada nos dispositivos móveis. Como os aplicativos estão disponíveis no *Google Play Store*, apesar da

predominância do português, temos uso de dispositivos em inglês, alemão, vietnamita e coreano.

Ao mesmo tempo que o SWEI permite análise de uso dos aplicativos, ele possibilita análises relacionadas ao conteúdo propriamente abordado. A Figura 3.e apresenta a eficiência dos algoritmos de busca nas suas partidas medidas pelo saldo entre o oponente e o algoritmo adversarial. Quanto menor for o saldo, mais eficiente é o algoritmo. O *box-plot* apresentado possibilita essa análise, onde é possível ver que o algoritmo *Greedy* foi o único a apresentar mediana negativa, ou seja, venceu mais vezes o oponente.

Um outro exemplo deste tipo de análise pode ser visto na Figura 3.f, onde são apresentados os erros e acertos dos usuários ao utilizar o LabVetor considerando-se os diferentes assuntos abordados. O aplicativo trabalha com o conceito de vetores, onde o aluno deve acertar o módulo (comprimento), direção e escalar (multiplicador) dos vetores. Observa-se que há mais erros do que acertos em todas as operações. Isso pode ser usado como um indicador para rever a usabilidade do aplicativo ou da metodologia usada durante o ensino e aprendizado do conteúdo.

6. Conclusões e perspectivas

O trabalho apresenta o SWEI, um serviço web integrado para interação e coleta de dados de aplicativos educacionais. Um dos pontos fortes do SWEI é a sua versatilidade. A introdução de chamadas clientes em diferentes aplicativos de diferentes ambientes foi feita de modo a facilitar a vida do desenvolvedor. A integração foi avaliada com aplicativos desktops e móveis, tanto via ambiente de programação, como o *Android Studio*, quanto para ambientes baseados em programação visual, como o *AppInventor*. Foi observado um interesse tanto pelos desenvolvedores quanto pelos idealizadores dos aplicativos nos recursos oferecidos pelo serviço web de modo a facilitar tanto a interação quanto o armazenamento dos resultados coletados.

A disponibilização dos dados coletados pelo SWEI em um *Data Warehouse* potencializa análises individuais das aplicações, desde a exploração de dados, até testes estatísticos e métodos de mineração de dados. Ademais, a visão integrada dos diferentes aplicativos em uma base integrada é bastante positiva ao permitir análises correlacionadas e uma visão holística do processo de ensino e aprendizado.

Agradecimentos

Os autores agradecem à FAPERJ e ao CNPq pelo financiamento parcial do trabalho.

Referências

- Alves, G., Warley, P., Quadros, J., Lignani, L., and Ogasawara, E. (2014). ControlHarvest: Ensino de Ecologia por Meio de Gamificação do Controle Biológico. *Anais do Simpósio Brasileiro de Informática na Educação*, 25(1):342–351.
- Birsan, D. (2005). On Plug-ins and Extensible Architectures. *Queue*, 3(2):40–46.
- Brassai, B., Varga, B., Simon, K., and Torok-Vistai, T. (2014). GeoQuesting: Mobile adventure game and web-based game editor. In *2014 IEEE 12th International Symposium on Intelligent Systems and Informatics (SISY)*, pages 99–103.

- Georgiev, T., Georgieva, E., and Smrikarov, A. (2004). M-learning: A New Stage of e-Learning. In *Proceedings of the 5th International Conference on Computer Systems and Technologies*, CompSysTech '04, pages 1–5, New York, NY, USA. ACM.
- Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., and Pirahesh, H. (1997). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery*, 1(1):29–53.
- Hansen, R. P. and Pinto, S. C. S. C. (2003). Construindo Ambientes de Educação Baseada na Web Através de Web Services Educacionais. *Anais do Simpósio Brasileiro de Informática na Educação*, 1(1):61–70.
- Hershkovitz, A. and Nachmias, R. (2011). Online persistence in higher education web-supported courses. *The Internet and Higher Education*, 14(2):98–106.
- Honorato, E., Schocair, C., Mauro, R., Castaneda, R., Duarte, S., and Ogasawara, E. (2015). LabVetor: A aplicação M-Learning para ensino de vetores na Física. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 26.
- Paschoal, L., Bento, T., Velasco, T., Schocair, C. O., Castaneda, R., Oliveira, T., and Ogasawara, E. (2014). JOE: Jogo Ortográfico Educacional. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 25, pages 652–656.
- R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Rodrigues, C., Afonso, J., and Tomé, P. (2011). Mobile Application Webservice Performance Analysis: Restful Services with JSON and XML. In Cruz-Cunha, M. M., Varajão, J., Powell, P., and Martinho, R., editors, *ENTERprise Information Systems*, number 220 in Communications in Computer and Information Science, pages 162–169. Springer Berlin Heidelberg.
- Romero, C. and Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1):135–146.
- Romero, C., Ventura, S., and García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1):368–384.
- Ruchika and Goel, A. (2012). EwayDataCollect: Web Service-Based Data Collection Tool for Education System. *Procedia Technology*, 5:397–406.
- Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., and Xu, X. (2014). Web services composition: A decade's overview. *Information Sciences*, 280:218–238.
- Sibaldo, M. A. A., Loureiro, E., Bittencourt, I. I., and Costa, E. d. B. (2006). Infra-estrutura para Acesso a Comunidades Virtuais na Web Através de Dispositivos Móveis. *Anais do Simpósio Brasileiro de Informática na Educação*, 1(1):58–60.
- Silberschatz, A., Korth, H., and Sudarshan, S. (2010). *Database System Concepts*. McGraw-Hill Science/Engineering/Math, New York, 6 edition edition.
- Silva, A. B. C., Serique, S. P., Preuss, L. S., Ogasawara, A., Quadros, J., Bezerra, E., Souza, U., and Ogasawara, E. (2015). Amê: An Environment to Learn and Analyze Adversarial Search Algorithms Using Stochastic Card Games. In *ACM Symposium on Applied Computing*.