REVISTA DO CCEI
ISSN Impresso: 1415-2061
ISSN Online: 2356-6635
Centro de Ciências
Economia e Informática
Revista Técnico-Científica
URCAMP
EDIURCAMP

LIMESURVEYGENERATOR: UM *FRAMEWORK* PARA GERAÇÃO DE *SURVEYS* EXPLORANDO TECNOLOGIAS DE *SOFTWARE* LIVRE

LIMESURVEYGENERATOR: A FRAMEWORK FOR GENERATING SURVEYS EXPLOITING OPEN SOURCE TECHNOLOGIES

Bruno Moura Paz de Moura[1], *João Roberto Ricalde Gervasio*[2], *Luciano de Pelegrini Lopes*[3]

RESUMO: Este trabalho apresenta a proposta de um *framework* para geração, envio de *surveys* e relatórios na avaliação acadêmica explorando os artefatos das tecnologias de *software* livre em conjunto com os padrões de projeto de arquitetura de *software* no seu desenvolvimento e fazendo uso de especificações Java (JSE), conjuntamente com o Sistema de Gerenciamento de Banco de Dados (SGBD) MySQL, e a ferramenta *open source LimeSurvey*. O objetivo geral é apresentar o *framework* desenvolvido, os padrões e tecnologias usadas na implementação do *LimeSurveyGenerator*. O resultado obtido foi uma aplicação para geração dos *surveys*, *tokens* e envio de *emails* na avaliação acadêmica mostrando a integração destas tecnologias, e a geração dos relatórios das avaliações.

Palavras-chave: LimeSurveyGenerator, Java Standard Edition, Survey

***ABSTRACT**: This work presents the proposal of a framework for generation, sending surveys and reports in the academic evaluation exploring the artifacts of free software technologies in conjunction with the software architecture design patterns in its development and making use of Java (JSE) together with the Database Management System (DBMS) MySQL and the open source tool LimeSurvey. The general objective is to present the framework developed, the standards and technologies used in the implementation of LimeSurveyGenerator. The result was an application for generation of surveys, tokens and emails in the academic evaluation showing the integration of these technologies, and the generation of evaluation reports.*

*Keywords**:** LimeSurveyGenerator, Java Standard Edition, Survey*

1 INTRODUÇÃO

Com a necessidade de obter um processo de pesquisa para avaliação de desempenho docente pelo discente nas instituições de ensino superior, pode ser considerada de fundamental importância a utilização de uma ferramenta para aplicação e gerenciamento de s*urveys* na elaboração das pesquisas para avaliação do desempenho do docente. O método *survey* (MORAL, p. 2013), é um método de coleta de informações diretamente de pessoas a respeito de suas idéias, sentimentos, saúde, planos, crenças e de fundo social, educacional e financeiro. Outras especialidades como clima organizacional, opinião pública, relacionamento, consumo, dentre outras, podem ser exploradas com o seu emprego.

[1] Mestrando em Computação.
{brunomoura@unipampa.edu.br}

[2] Mestrando em Tecnologias Educacionais em Rede.
{joaogervasio@unipampa.edu.br}

[3] Aluno Especial do Mestrado em Computação Aplicada.
{lucianolopes@unipampa.edu.br}

A coleta de informações é realizada através de questionários e aplicados ao público alvo selecionado para realização da pesquisa. O método utiliza um instrumento predefinido para obter descrições quantitativas de uma população (FINK, p. 2003a; FINK, p. 2003b). Para (MORAL, p. 2013), o *survey* deve ser administrado pelo pesquisador que pode enviá-lo aos entrevistados por meio impresso ou eletrônico.

O emprego de *surveys* na avaliação acadêmica estimula para que os desafios dos docentes sejam alcançados em busca da eficiência no seu dia a dia de trabalho, e dentre eles destacam-se: (i) as instituições de ensino superior possuem métodos para avaliação de seus docentes? (ii) a avaliação vai ao encontro dos aspectos curriculares visando interação à teoria, e prática aos aspectos da realidade? (iii) os docentes estimulam seus alunos a produção de pesquisa e extensão?

Este trabalho tem como objetivo central a proposição de um *framework* para geração de *surveys* em larga escala empregando da ferramenta *LimeSurvey*.

O artigo esta estruturado em nove seções. A primeira seção trata dos fundamentos contextuais do trabalho, na segunda seção são apresentadas as tecnologias utilizadas, na terceira seção os padrões de projeto de *software* empregados são expostos. Na seção quatro é abordada sobre a metodologia empregada na pesquisa, a seção cinco expõem o projeto do módulo *LimeSurveyGenerator*, logo na seção seis o modelo de *survey* colocado em prática é exposto, na continuidade a seção sete descreve o relatório elaborado, e por fim na seção oito são apresentas as considerações finais.

2 PADRÕES E TECNOLOGIAS UTILIZADAS

2.1 LIMESURVEY

Em (SCHMITZ, p. 2012) é proposta uma ferramenta intitulada de *LimeSurvey*[4], um *software* livre fornecido sobre os termos da *General Public Licence v2* (GNU) para aplicação de questionários *online* escrito em *Hypertext Preprocessor* (PHP), podendo ser utilizada junto a bancos de dados como *MySQL*, *PostgreSQL* ou *Microsoft SQL Server* para persistência de dados. Esta ferramenta de *software* fornece funcionalidades para usuários sem conhecimento sobre desenvolvimento de *software* possam publicar e coletar respostas de questionários.

No projeto do LimeSurvey (SCHMITZ, p. 2012), dentre os diversos recursos desse sistema estão o número ilimitado de questionários ao mesmo tempo; número ilimitado de participantes; uma variedade de tipos de questões com a possibilidade de condições para as perguntas de acordo com respostas anteriores; integração com imagens; pesquisas anônimas e identificadas; envio de convites e lembretes por *e-mail*;

possibilidade de salvar questionários parcialmente respondidos e recomeçar posteriormente; questionários com datas de início e expiração automatizadas; exportação dos dados em diversos formatos; análise através de sínteses estatísticas e apoio de gráfico, dentre outras funcionalidades.

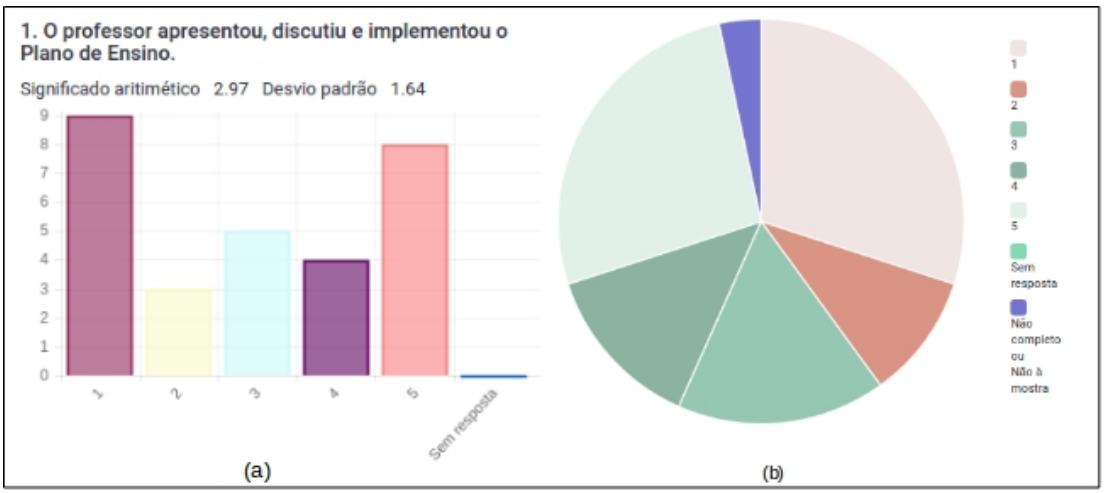
A Tabela 1 apresenta um resumo estatístico extraído de relatório oferecido pelo LimeSurvey, referindo-se a primeira pergunta empregada no *survey*, e destacando-se como recurso para análise estatística das respostas dos *surveys*. Posteriormente a Figura 1 (a) refere-se ao gráfico de barras, e a Figura 1 (b) ao gráfico de pizza, que sumarizam informações úteis obtidas através dos *survey*.

Tabela 1: Resumo estatístico da pergunta 1 da avaliação no *LimeSurvey*.

Sumário dos campos para pergunta 1 do *survey*			
1. O professor apresentou, discutiu e implementou o Plano de Ensino?			
Resposta	**Contagem**	**Percentagem**	**Soma**
1(1)	9	30%	40,00%
2(2)	3	10%	
3(3)	5	16,67%	16,67%
4(4)	4	13,33%	
5(5)	8	26,67%	40,00%
Soma(Resposta)	**29**	**100%**	**100%**
Sem resposta	0	0%	
Não completo ou Não à mostra	1	3,33%	
Significado aritmético		0	
Desvio padrão		0	

Fonte: (SCHMITZ, p. 2012).

Figura 1: (a) Gráficos (a) de Barras, (b) de Pizza Pergunta 1 da avaliação no *LimeSurvey*.



Fonte: (SCHMITZ, p. 2012).

[4] https://www.limesurvey.org/

2.2 MySQL

Para (MYSQL, p. 2017), o *MySQL* é um popular Sistema de Gerenciamento de Banco de Dados (SGBD) relacional SQL *Open Source* desenvolvido, distribuído e apoiado pela *Oracle Corporation*.

Um banco de dados relacional armazena dados em tabelas separadas em vez de colocar todos os dados em um só local. As estruturas de banco de dados são organizados em arquivos físicos otimizados, visando à obtenção de acesso rápido e consistente.

O modelo lógico do banco de dados é constituído de objetos como tabelas, visualizações, tuplas e colunas, ao mesmo tempo oferecendo um ambiente de programação flexível para emprego na etapa da elaboração das regras que ditam as relações entre os diferentes objetos, agregando assim consistência e integridade ao modelo relacional.

De acordo com (LUCKOW, DE MELO, 2010, p.63), o *MySQL* é o banco de dados que atualmente tem uma ampla utilização nos mais diversificados seguimentos, tendo mais de 70 milhões de instalações no mundo todo. É utilizado por empresas como Amazon.com, Google, Motorola, MP3.com, NASA, Silicon *Graphics*, Texas *Instruments* e Yahoo *Finance*.

O SGBD MySQL foi empregado para este projeto considerando a ampla utilização por empresas que atuam nos mais diversificados seguimentos do mercado, também considerando ofertar uma base de ferramentas sólidas para empregar na concepção de projetos livres a toda comunidades, podendo assim serem ampliados e distribuídos por outros pesquisadores.

2.3 JAVA

Conforme (LUCKOW, DE MELO, 2010, p.25-26,), a linguagem Java surgiu em 1991 na *Sun Microsystems*. Inicialmente era parte de outro projeto chamado *Green Project*, que ainda tinha como objetivo possibilitar a convergência entre computador, equipamento eletrônico e eletrodoméstico. O principal destaque do Java refere-se na execução do código de suas aplicações sobre uma *Java Virutal Machine* (JVM), ou seja, qualquer *hardware* ou equipamento eletrônico que suporte uma JVM poderá executar o Java, justificando o slogan "*write once, run anywhere*", traduzindo: "escreva uma vez, rode em qualquer lugar".

REVISTA DO CCEI
ISSN Impresso: 1415-2061
ISSN Online: 2356-6635
Centro de Ciências
Economia e Informática
Revista Técnico-Científica
URCAMP
EDIURCAMP

2.3.1 OpenSwing

O *OpenSwing* [5] é um *framework open-source* que pode ser utilizado para desenvolver aplicações Java baseadas em *Swing* (EIJE, 2010, p.7-8; OPENSWING, p. 2017). Através de seu emprego é possível desenvolver tipos de aplicações Java conforme a categorização apresentada no Quadro 1.

Quadro 1: Tipos de Aplicações com *OpenSwing*.

Tipo de Aplicação	**Descrição**
***stand-alone*:**	sem acesso a banco de dados, como um bloco de notas, por exemplo;
cliente-servidor:	apresentação + regras de negócio na máquina cliente e banco de dados na máquina servidor;
***web* em três camadas:**	*Swing* na máquina do cliente que acessa o servidor via *Hypertext Transfer Protocol* (HTTP), e no servidor residem a camada de regras de negócio (*servlet* Java) e o banco de dados; Fisicamente essa camada de regra de negócio, que são os *servlets*, poderia estar em uma máquina servidora única, a qual seria um servidor de aplicações. O banco de dados pode residir em outro servidor.
aplicações distribuídas:	*Swing* + aplicações do lado do servidor remotamente acessadas via *Remote Method Invocation* (RMI) ou por outro protocolo mais banco de dados.

Fonte: (EIJE, p. 2010)

Todos os componentes gráficos seguem as especificações Java Beans, desta forma, o emprego de *Integrated Development Environment* (IDE) Java disponíveis como *JDeveloper*[6], *NetBeans*[7] e *Eclipe*[8] podem ser exploradas. O desenvolvimento é visual, arrastando e soltando, do mesmo modo feito no *Deplhi*[9] ou mesmo no *NetBeans* com os componentes *Swing* atualmente disponíveis (EIJE, 2010, p.9-10).

Segundo (OPENSWING, p. 2017), o *OpenSwing* permite criar aplicações baseadas no paradigma *Single Document Interface* (SDI), onde cada janela é independente da outra, ou no paradigma *Multiple Document Interface (*MDI) onde existe uma janela principal da aplicação e as demais são filhas. Este é o modelo aplicado para o referido projeto.

Para a concepção do *framework* proposto, foi utilizado uma estrutura que oferece mecanismo entre os componentes e modelo de dados, com base no padrão de arquitetura de *software Model, View, Controller* (MVC), enquadrado-se na classificação de aplicação em duas camadas segundo a taxonomia de aplicações do *OpenSwing*.

Todos os componentes do *OpenSwing* são fortemente desacoplados uns dos outros, por meio de uma rigorosa concepção orientada a objetos baseada na adoção de *interfaces* que facilitam a implementação de novos comportamentos dos componentes do *framework* (EIJE, p. 2010).

[5] http://oswing.sourceforge.net/
[6] http://www.oracle.com/technetwork/developer-tools/jdev/overview/index.html
[7] https://netbeans.org/
[8] https://eclipse.org/

2.3.2 JavaMail

A plataforma Java possui funcionalidades adicionais nas quais abrangem computação distribuída com RMI e *Common Object Request Broker Architecture* (CORBA)[10] e uma arquitetura de componentes *JavaBeans*. A *Application Programming Interface* (API) JavaMail (HAROLD, p. 2013; SUN, p. 1999; LUCKOW, DE MELO, 2010, p.465), fornece um conjunto de classes abstratas definindo objetos que compõem um sistema de correio eletrônico. A API define classes como Mensagem, Armazenamento, Transporte e pode ser estendida para fornecer novos protocolos com objetivo de adicionar funcionalidade quando necessário. Além disso, a API fornece subclasses concretas das classes abstratas, nas quais incluem *MimeMessage e MimeBodyPart*, implementado amplamente a utilização de protocolos de correio eletrônico em conformidade com as especificações RFC822[11] e RFC2045[12].

3 PADRÕES DE PROJETO

3.1 TRANSFER OBJECT

Segundo (NETO, 2011, p.734), o objetivo desse padrão é reduzir a quantidade de requisições necessárias para recuperar um objeto. O *Value Object* permite encapsular em um objeto um subconjunto de dados utilizáveis pelo cliente e utilizar apenas uma requisição para transferi-lo.

Para (FOWLER, p. 2002) nesse padrão, os dados que vêm do modelo são expostos à camada de visão pelos *Data Transfer Objects (DTOs)* que pode ser representado através do *Value Object*. Um *Value Object* é um objeto Java arbitrário que encapsula os valores de retorno dos componentes de negócio, como o retorno de métodos de um *Data Access Object* (DAO).

Geralmente, seus atributos são definidos como públicos e o construtor recebe cada um de seus valores. O nome das classes Java que representam *Value Object* podem receber um “VO” ou “DTO” em sua nomenclatura para identificar participação deste padrão (VAN NIEUWPOORT et al. 2005).

3.2 DATA ACCESS OBJECT

Conforme (NETO, 2011, p. 732), o principal objetivo do padrão *Data Access Object* (DAO) é separar recursos de dados de seus mecanismos de acesso, permitindo que sejam modificados independentemente do código que os utiliza. Um DAO implementa o mecanismo de acesso para se trabalhar com uma fonte de dados específica.

[9] https://www.embarcadero.com/br/products/delphi
[10] http://www.corba.org/
[11] https://www.ietf.org/rfc/rfc822.txt
[12] https://www.ietf.org/rfc/rfc2045.txt

REVISTA DO CCEI
ISSN Impresso: 1415-2061
ISSN Online: 2356-6635
Centro de Ciências
Economia e Informática
Revista Técnico-Científica
URCAMP
EDIURCAMP

Um componente de negócio fica exposto apenas à *interface* do DAO, que esconde toda complexidade relativa à interação com a fonte de dados que está sendo utilizada. Como a *interface* de um DAO não altera quando sua implementação precisa ser modificada, esse padrão permite alterar a fonte de dados que está sendo utilizada numa aplicação sem afetar os componentes de negócios que fazem uso dele (DEEPAK et al., 2001; GAMMA et al., 1994).

Quando é utilizado o padrão DAO é comum também ser empregado um *GenericDAO,* ou seja, um DAO genérico somente tem o conhecimento do tipo de entidade que trabalhará no momento em que é instanciando (MEDINA et al. 2017; NETO, p. 2011).

3.3 MVC

É um padrão de arquitetura muito utilizado no desenvolvimento de projetos web, como aplicado no trabalho em (MOURA, p. 2013), e também pode ser aplicado a qualquer outro tipo de projeto que exija interação com usuário.

Para (LUCKOW, DE MELO, 2010, p.180-181), padrão MVC determina que um sistema seja separado em três camadas, denominadas de *Model*, *View* e *Controller*.

Segundo (NETO, 2011, p. 738), o principal objetivo do MVC, entretanto, não é definir o modo de separar em camadas, mas sim, definir como as camadas devem interagir. Em outras palavras, a separação de responsabilidade é um conceito benéfico na implementação do MVC, também prega que a camada *Model* não pode conhecer a *View*, e vice-versa. A camada responsável pela comunicação entre elas é a *Controller*.

Este padrão é utilizado neste trabalho com objetivo de prover portabilidade, modularização e uma organização arquitetural para o projeto do *framework*, e também tendo em vista facilitar a reusabilidade de código. No Quadro 2, é exposto o detalhamento das responsabilidades de cada camada deste padrão.

Quadro 2: Responsabilidades das camadas do MVC.

Camada	Responsabilidade
***Model*:**	gerencia um ou mais elementos de dados, responde a perguntas sobre o seu estado e responde a instruções para mudar de estado. O modelo sabe o que o aplicativo quer fazer e é a principal estrutura computacional da arquitetura, pois é ele quem modela o problema que está se tentando resolver;
***View*:**	representa tudo que compõe a *interface* de um sistema. É na *View* que as informações geradas pela *Model* serão exibidas, e é por meio da camada *View* que o usuário poderá inserir os dados para utilização do sistema. Com o emprego do *OpenSwing* no projeto, esta camada é representada pelas classes do pacote *view*, e são todas estendidas de *javax.swing*;
***Controller*:**	conecta *View* a *Model*. Responsável por buscar as informações da camada *Model* para gerar a *View* e por receber as informações da *View* e enviar para o *Model*.

Fonte: (NETO, p. 2011)

4 METODOLOGIA

O desenvolvimento do projeto foi realizado com base em métodos de modelagem e implementação no paradigma da orientação a objetos, utilizando os *frameworks* da especificação *Java Standard Edition* (JSE), e a *API OpenSwing*, empregando os padrões de projeto citados nas seções anteriores. Esta abordagem contribui para a organização da modelagem e implementação do *framework* proposto, obtendo assim a possibilidade de obter se múltiplas visões de um mesmo contexto e desacoplando a *Interface* do sistema da lógica da aplicação.

A tecnologia de banco de dados aplicada foi o SGBD *MySQL*, na qual tem grande utilização por empresas dos segmentos públicos e privados. O modelo lógico de dados empregado foi originado pela instalação padrão da ferramenta *LimeSurvey*. Os ambientes de desenvolvimento utilizados foram: Netbeans IDE em sua versão 8.0.2, juntamente com as especificações Java JSE para a implementação das rotinas de geração dos *surveys*, *tokens*, e envio de e-mail ao público-alvo da pesquisa. Tanto na etapa de geração dos *surveys* quanto nas etapas da geração e exportação de relatórios, foram aplicados os artefatos do SQL, tais quais *Functions*, *Storage Procedures* e SQL Dinâmicas, para isso foi colocada em prática a ferramenta *MySQL Workbench* na versão 6.0. Na próxima seção é apresentado o detalhamento das funcionalidades do módulo *LimeSurveyGenerator*.

5 LIMESURVEYGENERATOR

Esta seção apresenta as funcionalidades e características do *framework LimeSurveyGenerator*[13], tendo enfoque uma sequência de passos pré estabelecidos, iniciando na (i) parametrização de conexão com o banco de dados; (ii) parametrização das configurações de *Simple Mail Transfer Protocol* (SMTP); (iii) configuração do padrão de modelo do e-mail utilizado para envio dos convites dos questionários; (iv) informação dos parâmetros para a geração dos questionários e oferecendo uma (v) visualização dos relatórios de estatísticas dos *surveys*.

5.1 PARÂMETROS DE CONEXÃO COM BANCO DE DADOS

Na tela de configuração de parâmetros para conexão com banco de dados, é permitido realizar o gerenciamento do registro utilizado para a conexão, as informações de URL, Driver, Usuário e Senha são aplicadas.

[13] https://sourceforge.net/projects/limesurveygenerator/

Seguindo a sequência de passos da geração dos *surveys*, a funcionalidade ilustrada na Figura 6 exposta na seção de apêndices, apresenta as configurações necessárias para o carregamento do arquivo do tipo *Comma-separated values* (CSV), que deverá ser composto por uma estrutura constituída da seguinte forma: (i) Nome do Avaliado, (ii) Nome do Avaliador, (iii) Email do Avaliador e (iv) empregando como delimitador de campos o caracter ponto e vírgula ";".

As configurações como (i) Proprietário da pesquisa, (ii) E-mail, (iii) Título do survey, (iv) Data e Hora Inicial para abertura, (v) Data e Hora Final para fechamento, (vi) URL do Domínio Principal utilizado na instalação do *LimeSurvey*, assim como também o (vii) Diretório do Apache[14] onde o mesmo está instalado devem ser informados, e ainda, caso necessária a definição da geração para os *tokens* e envio dos mesmos por e-mails, basta habilitá-las estas opções na etapa da geração dos questionários.

A partir desses dados o sistema tem condições de gerar os questionários, *tokens* e disparar os *e-mails* aos seus investigados. Esta funcionalidade vem a contribuir na administração, agilizando assim o processo para geração de *surveys*, relatórios e envio em larga escala, fornecendo desta maneira uma fonte de informações. O Algoritmo 1, mostra uma simplificação dos passos executados na geração dos *surveys*, dando uma visão geral de como é implementada esta rotina.

Algoritmo 1: Algoritmo para Geração das Pesquisas no LimeSurvey

```
Input: CSVFile, strName, strTitle, strEmail, strSubject, dtInitial, dtFinal
initialization;
while (End-Of-File CSV) do
    sid=getNextSurveyID;
    insert into lime_surveys;
    insert into lime_surveys_languagesettings;
    insert into_lime_groups;
    insert into lime_questions;
    create table lime_survey_sid;
    create table lime_tokes_sid;
    insert into table_tokens_sid;
end
```

Os comandos de *Data Manipulation Language* (DML) *insert, delete, update*, e os de *Data Definition Language (DDL)*, como por exemplo, *create table* são gerados dinamicamente através de *Storatge Procedure com* o uso de SQL dinâmica. No Algoritmo 1, a variável "*sid*", representa o ID obtido na criação da estrutura de cada *survey*.

6 AVALIAÇÕES

O modelo de questionário aplicado é anônimo e baseado na (RESOLUÇÃO 80, p. 2014). O registro das respostas não contém nenhuma informação de identificação sobre o respondente. A utilização de um

[14] https://www.apache.org/

código/*token* de identificação a pesquisa é utilizado, o código não é armazenado junto às respostas. Este é armazenado em uma tabela específica para esta finalidade no banco de dados e será atualizado apenas para indicar se a pesquisa está completa ou incompleta, sendo assim, não há nenhuma maneira de relacionar os códigos de identificação com as respostas.

O instrumento avaliativo é utilizado a partir da escala de 1 a 5, onde o conceito 1 revela conceito mínimo de desempenho didático no item avaliado e 5, conceito máximo. As perguntas apresentadas no questionário utilizado nas avaliações são descritas no Quadro 3.

Quadro 3: Perguntas realizadas na avaliação.

1. O professor apresentou, discutiu e implementou o Plano de Ensino?
2. O professor na condução do componente curricular estabelece interação entre a teoria, a prática e/ou os aspectos da realidade?
3. O professor promove a articulação entre o ensino e/ou a pesquisa e/ou a extensão, seja através da vinculação da disciplina com esta atividade ou através da utilização de ferramentas afetas a esta atividade finalística?
4. O professor dispensa aos alunos tratamento cordial em um clima ético e de respeito pessoal, aceita críticas, opiniões e sugestões?
5. O professor mostra-se receptivo às necessidades dos alunos e cooperativo na solução de suas dificuldades com o componente curricular. É acessível/disponível para orientação extraclasse?
6. O professor elabora avaliações compatíveis (coerentes) com o conteúdo desenvolvido, discute e analisa os resultados com os alunos?
7. O professor utiliza linguagem clara e compreensível na condução do processo de ensino aprendizagem?
8. O professor é assíduo e pontual?

Fonte: (RESOLUÇÃO 80, p. 2014)

As primeiras avaliações processadas com o emprego do *framework LimeSurveyGenerator* foram através de consultas aplicadas ao publico alvo constituído de alunos dos cursos de graduação, pós-graduação da Universidade Federal do Pampa, e que compreendeu os períodos de 2015/01, 2015/02, 2016/01 e 2016/02. De forma sumarizada a Tabela 2, apresenta os totalizadores de *surveys* gerados e enviados aos seus avaliadores nos referidos períodos, mostrando a escalabilidade que o *framework* proporciona para geração e envio de *surveys* em massa.

Tabela 2: Total de *Surveys* Gerados com *LimeSurveyGenerator*

Período	**Total de *Surveys* Gerados**
2015/01	730
2015/02	736
2016/01	760
2016/02	805
Total	**3031**

Fonte: (Autores)

7 RELATÓRIO E EXPORTAÇÃO

Após o *survey* ter sido ativado, a partir da barra de administração do *LimeSurvey* é possível gerar as estatísticas. Para analisar os resultados pode se carregá-los numa planilha no formato para Microsoft *Excel*, *Microsoft Word*, *CSV, PDF, HTML*. Opcionalmente, também é possível exportar os dados e usar a estatística do sistema para obter informações úteis a partir dela.

Complementarmente aos recursos do *LimeSurvey*, no trabalho proposto foi elaborado um modelo de relatório que levou em consideração alguns critérios da (RESOLUÇÃO 80, p. 2014) estabelecidos para fins de desempenho didático que são eles:

- A avaliação dar-se-á a partir da média de desempenho de cada questão;
- A soma total das médias será gerada a parir da média das questões;
- O conceito atribuído ao avaliado é dado com base na Tabela 3.

Tabela 3: Atribuição do Conceito

Conceito obtido na avaliação	**Pontuação**
Entre 0,0 e 1,0	0,0
Entre 1,01 e 2,0	0,25
Entre 2,01 e 3,0	0,5
Entre 3,01 e 4,0	0,75
Entre 4,01 e 5,0	1,00

Fonte: (RESOLUÇÃO 80, p. 2014).

8 CONSIDERAÇÕES FINAIS

A utilização desta técnica para coleta de informações através de pesquisas proporciona múltiplos recursos, na qual fornece suporte a tomada de decisões com base no seu público-alvo. Através deste método quantitativo é possível obter dados precisos servindo deste modo de base para a elaboração dos relatórios e gráficos. Surgem também como alternativa para análise simplificada pelos proprietários da pesquisa e aos próprios avaliados. De posse destas informações é possível propor alternativas no desenvolvimento das metodologias aplicadas ao meio acadêmico.

Sugere-se para trabalhos futuros a elaboração de outros tipos de questionários e métodos para avaliação dos resultados, buscando agilizar o processo de gerenciamento das pesquisas e nas decisões do meio acadêmico. Este trabalho apresentou um *framework* para a elaboração de *surveys* em grande escala na avaliação acadêmica.

REFERÊNCIAS

CHANTAL, Morley. UML para a análise de um sistema de informação. 2017.

DEEPAK, Alur; CRUPI, John; MALKS, Dan. Core J2EE Patterns: Best Practices and Design Strategies. Sun Microsystems. Palo Alto, 2001.

EIJE, Albert Eije Barreto. Dominando o OpenSwing. São Paulo: Ciência Moderna, 2010.

FINK, Arlene. The survey handbook, volume 1. Sage, 2003a.

FINK, Arlene.. How to design survey studies, volume 6. Sage, 2003b.

FOWLER, Martin. Patterns of enterprise application architecture. Addison-Wesley Longman Publishing Co., Inc., 2002.

FUNDAÇÃO UNIVERSIDADE FEDERAL DO PAMPA – UNIPAMPA. Resolução 80, de 28 de Agosto de 2014, Bagé, 2014.

GAMMA, Erich et al. Design patterns: elements of. 1994, 1994

HAROLD, Elliotte Rusty. JavaMail API, volume 1. O Reilly Media; 1 Edition, 2013.

LUCKOW, Décio Heinzelmann; DE MELO, Alexandre Altair. Programação Java para a WEB. Novatec Editora, 2010.

MEDINA-SANTIAGO, A. et al. Web Application Development by Applying the MVC and Table Data Gateway in the Annual Program Budget Management System. development, v. 8, n. 2, 2017.

MOURA. Bruno Moura Paz, Abner. GUEDES. Aplicação dos Principais Padrões de Projeto utilizando o J2EE, Bagé/RS, v. 17, n.32, p-33-51, 2013.

MORAL, Assédio et al. Método de Pesquisa Survey, 2013 Disponível em: <http://www.partes.com.br/2013/12/09/metodo-de-pesquisa-survey/> Acessado em 20 de Abril de 2017.

MYSQL. MySQL 5.7 reference manual. Oracle Corporation, 2017.

NETO, Antônio Gonçalves dos Santos. Java na Web. Ciência Moderna Ltda, 2011.

OPENSWING. Disponível em: <http://oswing.sourceforge.net>, Acessado em: 18 de Abril de 2017.

SCHMITZ, Carsten et al. LimeSurvey: An open source survey tool. LimeSurvey Project Hamburg, Germany. URL http://www. limesurvey. org, 2012.

SUN , Inc Sun Microsystems. JavaMail Guide for Service Providers. Sun Microsystems, 1999. Disponível em: <https://javamail.java.net/docs/Providers.pdf> Acesso em: 20 de Abril de 2017.

VAN NIEUWPOORT, Rob V. et al. Ibis: a flexible and efficient Java-based Grid programming environment. Concurrency and Computation: Practice and Experience, v. 17, n. 7-8, p. 1079-1107, 2005.

REVISTA DO CCEI
ISSN Impresso: 1415-2061
ISSN Online: 2356-6635
Centro de Ciências
Economia e Informática
Revista Técnico-Científica
URCAMP
EDIURCAMP

APÊNDICES

Figura 2: Caso de Uso.

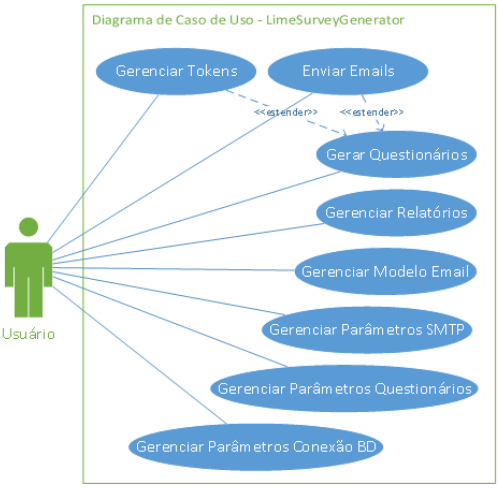


Figura 3: Diagrama de Classe.

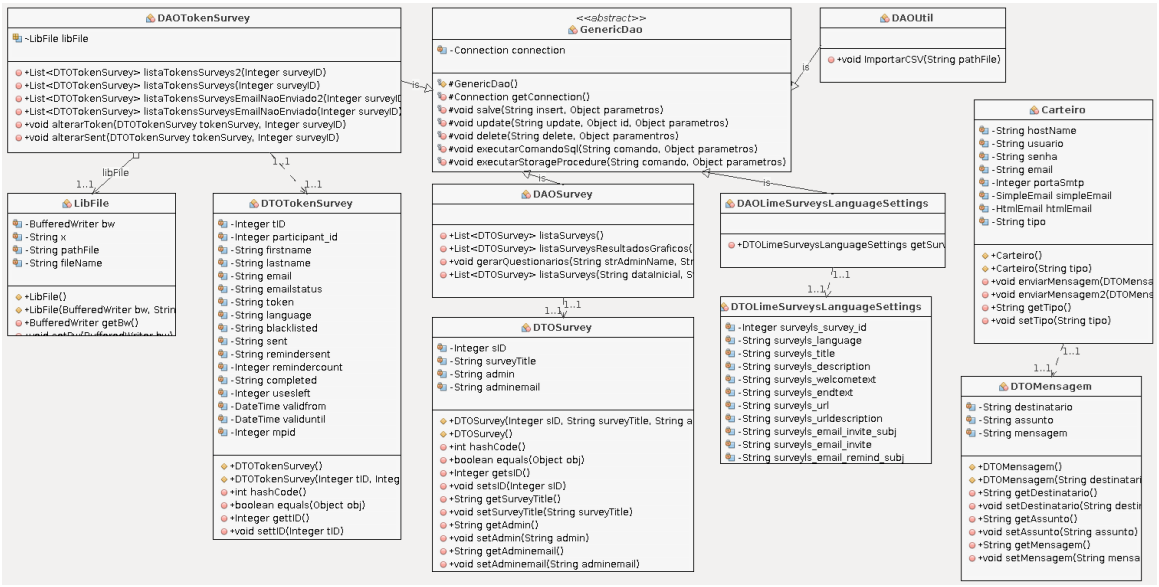


Figura 4: Menu Principal do Projeto *LimeSurveyGenerator*

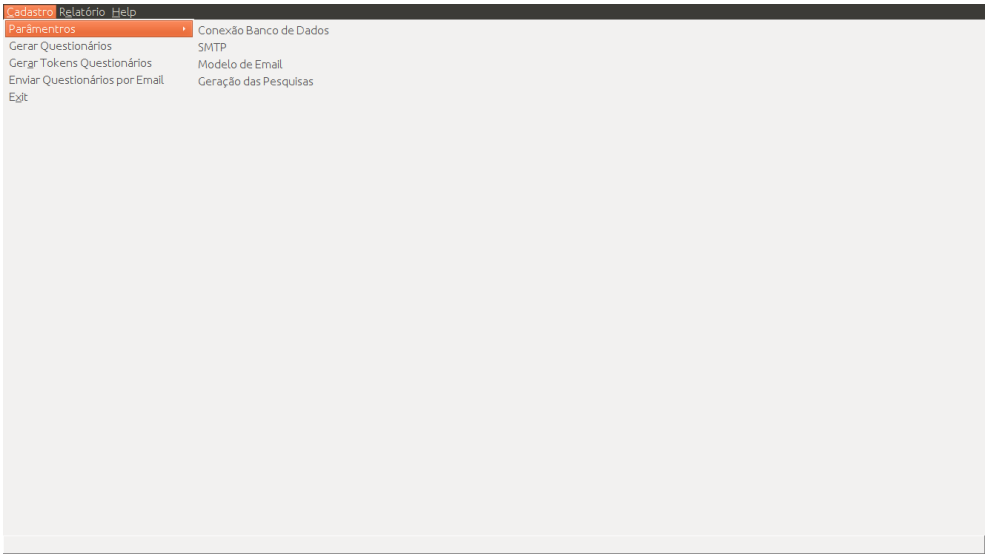


Figura 5: Tela de Parametrização da Conexão com Banco de Dados

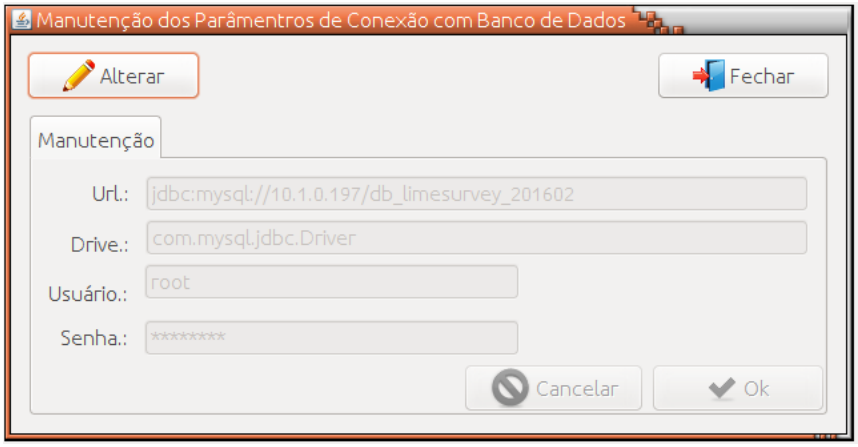


Figura 6: Tela de Geração dos *Surveys*.