

FIDA: Ferramenta para Importação de Dados no formato planilha eletrônica para um sistema de Avaliação Institucional

Kawany Iracema Fernandes¹, Michele Cristiani Barion²

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) – *Campus Hortolândia* – São Paulo – SP – Brasil

kakawmelf@gmail.com¹, michele_barion@hotmail.com²

Abstract. *Currently, regarding the technology area, the use of data becomes inevitable, but commonly the sources are varied, such as spreadsheets, text files, or even PDF files, thus making it difficult to generate results. Focusing on data provided via electronic spreadsheet and with the application of the SCRUM methodology, this work presents the development of functionality created on the web platform that make it possible to import the records of an electronic spreadsheet directly into the database, aiming at managing reports in a system for an institutional evaluation in the educational segment.*

Resumo. *Atualmente quando se trata da área de tecnologia, a utilização de dados se torna algo inevitável, porém muitas vezes as fontes são variadas, tais como planilhas, arquivos de texto ou até mesmo arquivo PDF, dificultando, assim, a geração de resultados. Tendo como foco dados passados via planilha eletrônica e com a aplicação da metodologia SCRUM, esse trabalho apresenta o desenvolvimento de uma funcionalidade criada na plataforma web que possibilita a importação dos registros de uma planilha eletrônica diretamente para o banco de dados, objetivando o gerenciamento de relatórios em um sistema para avaliação institucional no segmento de ensino.*

1. Introdução

No decorrer dos anos, a tecnologia da informação se tornou cada vez mais presente em nosso cotidiano, tanto pessoal quanto profissional. No meio empresarial, podemos notar que ela está presente na maioria dos processos, sendo eles nas áreas de Tecnologia da Informação (TI), Recursos Humanos (RH), atendimento ao cliente, entre outras. Com a quantidade crescente de áreas usando sistemas paralelos para exercerem seu trabalho, na qual estes muitas vezes acabam gerando processos operacionais repetitivos, profissionais de desenvolvimento de *software* buscam propor ferramentas computacionais que diminuam os trabalhos corriqueiros e que facilitem a geração de resultados mais eficazes e eficientes para todos os setores que integram informações.

Segundo os autores [Laudon e Laudon 2014], sistemas de informação são fundamentais nas organizações, principalmente para atingir seus objetivos organizacionais, contudo se não houverem pessoas capacitadas para desenvolvê-los, mantê-los e utilizá-los, não poderão levar ao alcance de resultados satisfatórios. Diante disso, a fim de que os objetivos em prol da organização sejam alcançados, é necessário que a cultura da organização mude como um todo, aderindo assim os sistemas de informação para facilitar o cotidiano das pessoas e com isso

melhorando a qualidade do serviço entregue, afinal como apresentam os autores [Laudon e Laudon 2014], sistemas de informação são formados pelas dimensões organizacional, humana e tecnológica.

Há diversos exemplos de sistemas que foram criados para automatizar serviços, desde atividades mais complexas como gerenciamento de estoque, pagamento de contas via aplicativo, como simples soluções para criação de questionários, tais como [*SurveyMonkey*] e o [*Google Forms*], usados por muitas empresas para a realização de pesquisas de satisfação.

Tendo como foco pesquisas de satisfação em instituições de ensino superior, muitas destas instituições possuem seus métodos de avaliação institucional, sendo estes desenvolvidos de acordo com suas necessidades específicas. Também é importante salientar que o Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), define a autoavaliação como uma das modalidades obrigatórias para gerar futuros relatórios que apresentem informações gerais da instituição, tais como a qualidade das aulas, a infraestrutura e serviços associados aos diversos setores da Instituição de Ensino Superior (IES).

[...] acrescido de indicadores específicos, projeto pedagógico, institucional, cadastro e censo. O relatório da autoavaliação deve conter todas as informações e demais elementos constantes no roteiro comum de base nacional, análises qualitativas e ações de caráter administrativo, político, pedagógico e técnico-científico [INEP 2015].

Em um processo de avaliação há a geração de um grande volume de dados, tais como informações sobre a instituição e questões relacionadas aos laboratórios e qualidade do curso, sendo que seus resultados manipulados manualmente geram um acúmulo de trabalho, tais como filtrar e classificar os dados para que sejam registrados de forma correta, armazenar as informações em suas respectivas bases de dados, gerar relatórios e possíveis gráficos para apresentação de resultados. Levando em consideração a grande movimentação de dados que é feita, os resultados gerados nem sempre são entregues de forma a facilitar o tratamento e o carregamento dos mesmos, como por exemplo, arquivos em formato de texto.

Considerando como exemplo uma rede particular de ensino, a avaliação institucional é feita em etapas. A primeira etapa trata da coleta e tratamento de dados, visto que é comum nas empresas os dados estarem associados à diversas fontes, na qual a coleta é realizada através de bancos de dados do sistema acadêmico, porém muitas vezes ele não possui todas as informações. Estas são complementadas através de dados passados por planilhas eletrônicas, tornando necessária a intervenção do desenvolvedor, pois todo processo se torna muito manual.

A Figura 1 apresenta um diagrama com a execução dos processos. Observa-se na Figura que as ações “processar dados” e “gerar planilha eletrônica” são realizadas pela área de negócio por meio das seguintes ações: a cada nova análise e processamento dos dados pela área de negócio, são geradas planilhas eletrônicas com dados “brutos” que então são encaminhadas à equipe de desenvolvimento para filtragem dos dados e preparação do documento para ser exportado. Todavia, toda vez que há alguma mudança nos dados da planilha “bruta”, todo o processo se repete.

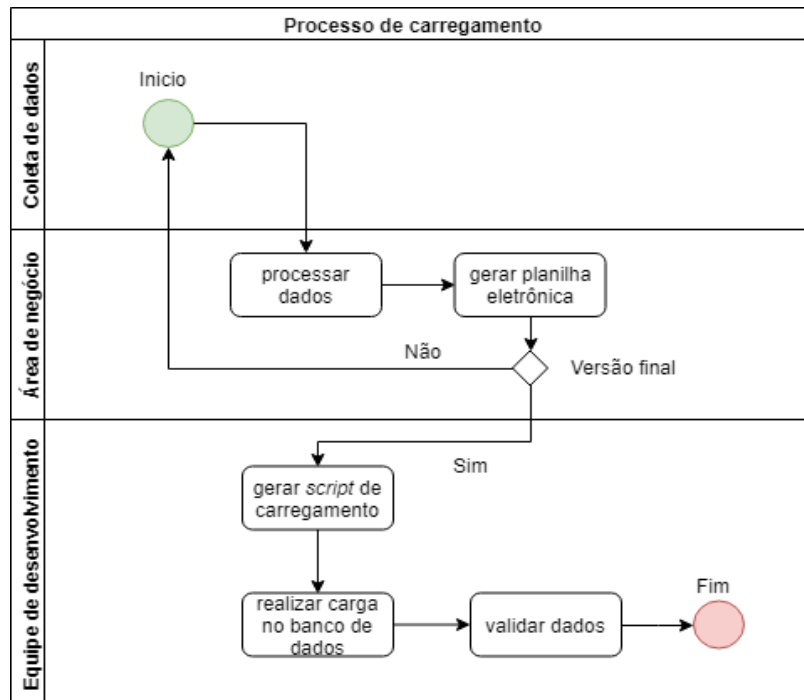


Figura 1. Processo para leitura dos dados da planilha para carregamento no banco de dados

Após os dados serem carregados no banco, uma validação para verificar a coerência dos mesmos é realizada e nela são verificados se os dados mostrados pelo sistema coincidem com os que estão contidos na planilha.

Diante deste contexto, o objetivo da proposta é o desenvolvimento de uma funcionalidade para plataforma *web* que faz a importação de dados editados em massa via planilha eletrônica para uma ferramenta já existente chamada Painel de Conferência para manutenção dos dados acadêmicos. A proposta será implementada pela equipe de desenvolvedores, já que esta é responsável pelo armazenamento das informações em um banco de dados para que as unidades de ensino de uma Instituição particular possam realizar a avaliação institucional. Seguindo as diretrizes de trabalho da equipe, os processos de desenvolvimento seguirão a metodologia ágil.

Para abordagem do objetivo exposto, além desta introdução, o presente trabalho está organizado em seis seções ordenadas em: fundamentação teórica, metodologia, desenvolvimento, conclusão, trabalhos futuros e relação das referências utilizadas tanto na pesquisa sobre as metodologias adotadas quanto das ferramentas que foram fundamentais para o desenvolvimento.

2. Referencial teórico

Esta seção aborda a pesquisa bibliográfica realizada para gerar o embasamento teórico que fundamenta o uso de cada escolha para realização do presente trabalho.

2.1. Sistema de Informação

Segundo o autor [Silva 2007], as informações podem ser criadas de diferentes maneiras, estando elas não ligadas somente a computadores. Um sistema de informação tem como objetivo coletar dados de diversas fontes, armazená-los e por fim processá-los de forma que os dados possam ser distribuídos de modo a alcançar alguma finalidade prática.

Segundo o autor [Rezende 2016], sistemas de informações podem ser separados em três níveis, sendo eles operacional, tático e estratégico.

a. Sistemas de nível operacional são aqueles associados ao fluxo de dados muito grande, ou seja, voltados às operações do dia-a-dia, na qual dão suporte aos negócios da empresa. Eles são a base de fornecimento de informação para os demais sistemas e sua interrupção quanto ao funcionamento poderá prejudicar os outros níveis de sistemas de informação. Considerando a proposta do presente trabalho, a aplicação se enquadra neste tipo de sistema.

b. Sistemas de nível tático são usados para controle e gerenciamento quanto às transações operacionais realizadas no nível operacional. Permitem aos gestores o acompanhamento quanto às informações gerais, comparando o seu desempenho por meio de padrões.

c. Sistemas de nível estratégico são usados por gestores visando, por meio de informações gerados no nível tático, mudanças em seu ambiente organizacional, além de futuros planejamentos e decisões. Os autores [Laudon e Laudon 2014] classificam este tipo como sistemas de apoio aos executivos.

2.2. Avaliação Institucional

Em 14 de abril de 2004 foi constituída a Lei nº 10.861 para criação do Sistema Nacional de Educação Superior (SINAES) que tem como foco principal a avaliação de quatro elementos, sendo eles: uma autoavaliação, avaliação externa, censo da educação superior e cadastro de cursos e instituições [INEP 2015]. A autoavaliação é realizada pelos discentes e se resume em avaliar o Projeto Pedagógico do Curso (PPC), a instituição de ensino e o censo. A avaliação externa é realizada com pessoas que pertencem ao corpo acadêmico e científico, com base nas áreas de especialidades que cada um possui. No Censo da Educação Superior são contidas as informações que são disponibilizadas para a área acadêmica e para a população, e por esse motivo a avaliação das instituições deve estar contida nesta entidade, para que assim se torne possível um entendimento melhor sobre as instituições. Para cadastros dos cursos e instituições são avaliados os dados levantados pela autoavaliação e com a análise é possível dizer se é viável ou não a abertura de um curso/instituição nova.

Segundo o SINAES, estas avaliações foram criadas para que fosse possível melhorar a qualidade das IES através do reconhecimento de seus valores, meios de gestão, formação, cursos e pesquisas realizadas [CONGRESSO NACIONAL 2004].

2.3. Ferramentas de desenvolvimento

Para esta subseção será apresentado o embasamento teórico estudado e aplicado nos processos e em todo o desenvolvimento, considerando a escolha da metodologia, das linguagens e tecnologias.

2.3.1. Scrum

Para os autores [Schwaber e Sutherland 2017] o *Scrum* é um *framework* para gerenciamento de projetos, desde a organização até o desenvolvimento de produtos complexos e adaptativos.

É utilizado um período de tempo chamado *Sprint* para o desenvolvimento do produto. Um projeto pode ter quantos *Sprints* forem necessários para concluir o produto final, sendo que

em cada um é feita uma implementação extra no código como, por exemplo, uma nova opção no cadastro ou um botão extra na tela.

2.3.2. HTML

[Mazza 2014] aborda o *HTML (Hypertext Markup Language)* como uma linguagem de hipertexto utilizada para criar páginas *web* através de apontadores denominados hiperligações, no qual conectam sua página a informações na *Internet*.

2.3.3. Javascript

Segundo [Flanagan 2011] *Javascript* é uma linguagem dinâmica, interpretada e sem tipagem. Pode ser utilizada junto ao paradigma de programação orientada a objeto. Tem como base a linguagem Java em relação à criação de classes.

2.3.4. CSS

Para o autor [Zemel 2015] *CSS (Cascading Style Sheets)* é uma linguagem de estilização de páginas *web* feitas com linguagens de marcação (*HTML* e *XHTML*) e com ela é possível trazer cor, diferentes fontes, formatos e personalizações na página, para que assim ela não mostre apenas texto quando o usuário utilizá-la.

2.3.5. PHP

Segundo os autores [Lerdorf, MacIntyre e Tatroe 2013], *PHP (Personal Home Page Tools)* é uma linguagem de programação que pode ser usada como um *script* do lado do servidor e pode gerar conteúdo *web* dinâmico. Possibilita a interação com o usuário por meio de formulários, parâmetros *URL* e *links*. Uma das suas características é o suporte a um grande número de bancos de dados, sendo uma tarefa simples a construção de uma página com esta integração.

Com o tempo a linguagem foi se adaptando e com isso foram surgindo novas bibliotecas, sendo uma delas a biblioteca *PhpSpreadsheet*¹ usada para a criação e edição de planilhas eletrônicas com o auxílio da linguagem *PHP*.

2.3.6. Bootstrap

Segundo o autor [Schmitz 2014], *Bootstrap* é um *framework* para programação *front-end* onde possui um pacote de elementos prontos para serem usados com o *HTML* e *CSS*. Ele não afeta outras linguagens quando se trata de desenvolvimento. É usado para criação de *sites* responsivos através de bibliotecas de *plugins* que estão disponibilizadas para uso.

2.3.7. MS SQL Server

Segundo [Miyagusku 2008], *MS SQL Server* é um SGBD (Sistema Gerenciador de Banco de Dados) relacional desenvolvido pela *Microsoft* e possui uma melhor interação quando se trata de linguagens pertencentes ao *framework .Net*, como *Visual Basic*, *C#*, *F#*, entre outros.

Permite a criação de tabelas relacionadas, fornecendo integridade referencial e outras restrições para manter a precisão dos dados. Suas linguagens de consulta primárias são *T-SQL*

¹ Abrindo e editando planilhas com o *PHPSpreadsheet*. Disponível através do link <https://imasters.com.br/desenvolvimento/abrindo-e-editando-planilhas-com-o-phpspreadsheet>.

(*Transact-SQL*) e *ANSI* (American National Standards Institute) *SQL* (*Structured Query Language*).

2.3.8. Diagrama Entidade-Relacionamento

Considerando banco de dados relacional, aplica-se o desenvolvimento do modelo conceitual, que é um esquema que representa a realidade das informações existentes, assim como as estruturas de dados que representam estas informações. Usa-se, então, a técnica de modelagem conceitual Entidade-Relacionamento.

Nesta técnica, um modelo conceitual é usualmente representado através de um diagrama, chamado Diagrama Entidade-Relacionamento (DER) [HEUSER 2009].

3. Metodologia

A proposta deste trabalho seguiu o seguinte escopo na metodologia e desenvolvimento:

- Foi realizado um estudo de caso em uma escola particular de ensino universitário. Diante deste cenário, foi realizada uma pesquisa de campo com o intuito de investigar mais a fundo a expectativa da área de negócio a respeito da nova funcionalidade e, conseqüentemente, definir os requisitos deste trabalho.
- O desenvolvimento se deu com o auxílio do modelo iterativo e incremental *Scrum*.
- Foi desenvolvido um estudo bibliográfico acerca de bibliotecas em *PHP* para possibilitar a leitura de planilhas eletrônicas. Considerou-se o *PHP* por ser compatível com a linguagem já utilizada pela área de negócio para visualização após o carregamento dos dados e, assim, tornando-se possível a integração entre a proposta deste trabalho e a rotina já existente para geração e impressão de resultados.
- Foi feito o estudo das colunas oriundas da planilha que é exportada através de outro sistema e, após, a definição da estrutura das tabelas físicas para armazenamento dos dados.
- Para criação da proposta, definiu-se o uso das linguagens *HTML*, *CSS*, *Bootstrap*, *Javascript* e *PHP* para criação da rotina que importa os dados. O *MS SQL Server* foi definido como o SGBD para armazenamento dos dados.

4. Desenvolvimento

Nesta seção serão abordados os *Sprints* do projeto, considerando o tempo específico para cada tarefa definida, desde a análise até a entrega final da proposta.

4.1. Etapas da análise e implementação da proposta

Para abordagem da proposta foram definidas duas informações que serão recebidas por arquivos externos, sendo registros dos cursos e das unidades de ensino. Diante deste contexto e para o desenvolvimento de cada funcionalidade associada a cada informação a ser recebida, foi determinado o período de um mês com 2 *Sprints* e em um intervalo de duas semanas para cada um.

Na Figura 2 são representadas, a partir de casos de uso, as funcionalidades da aplicação proposta. Observa-se que o usuário terá gerenciamento e controle sobre todos os processos.

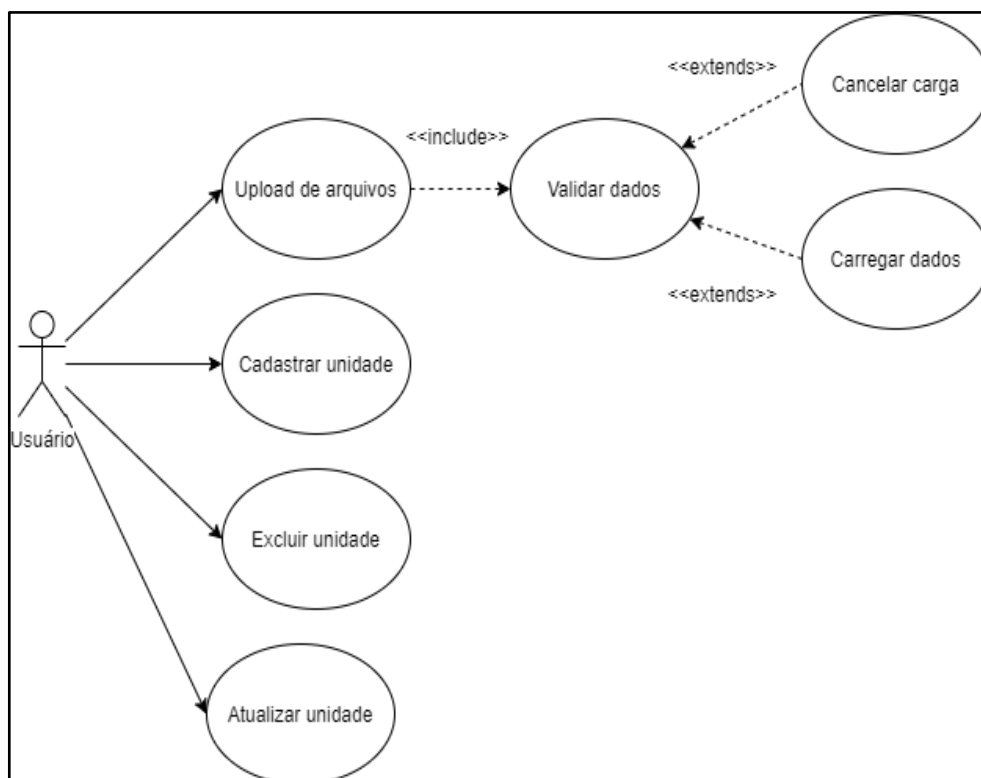


Figura 2. Diagrama de Caso de uso

Logo após o mapeamento do processo foi usado o banco de dados *MS SQL Server* para a criação das tabelas temporárias, pois caso ocorresse a carga das planilhas com dados errados isso não afetaria as tabelas finais que alimentam a ferramenta que está em produção, e com isso a aplicação já existente continuaria funcionando normalmente. A Figura 3 apresenta o diagrama referente às tabelas temporárias que guardarão os dados oriundos das planilhas eletrônicas.

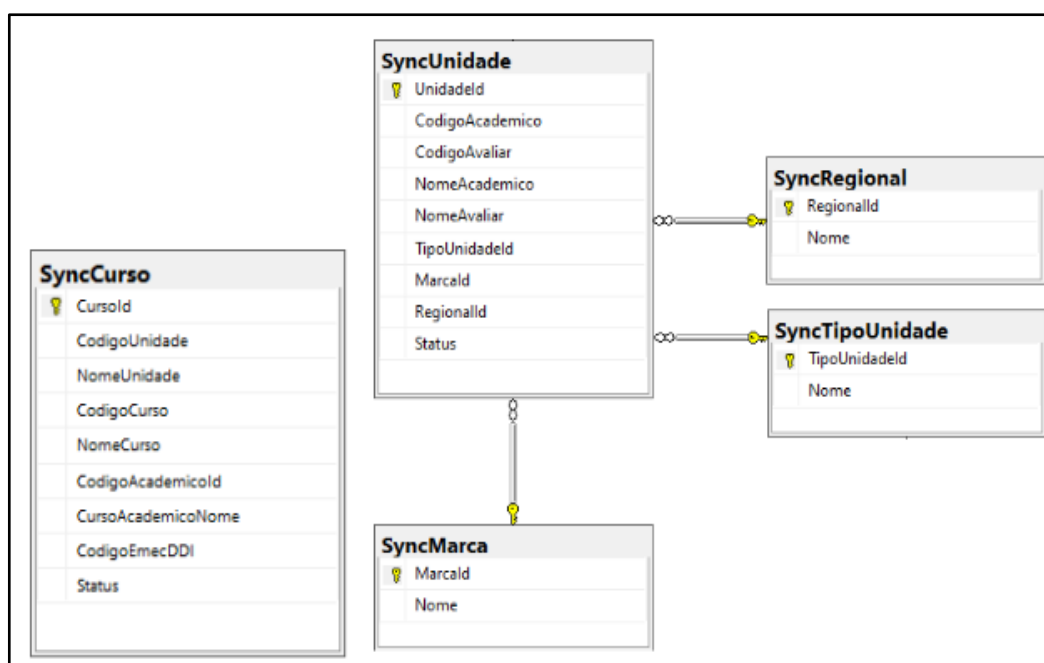


Figura 3. DER tabelas temporárias

A Figura 4 é o DER que representa as tabelas que guardarão os dados após a confirmação do usuário quanto ao processo de importação.

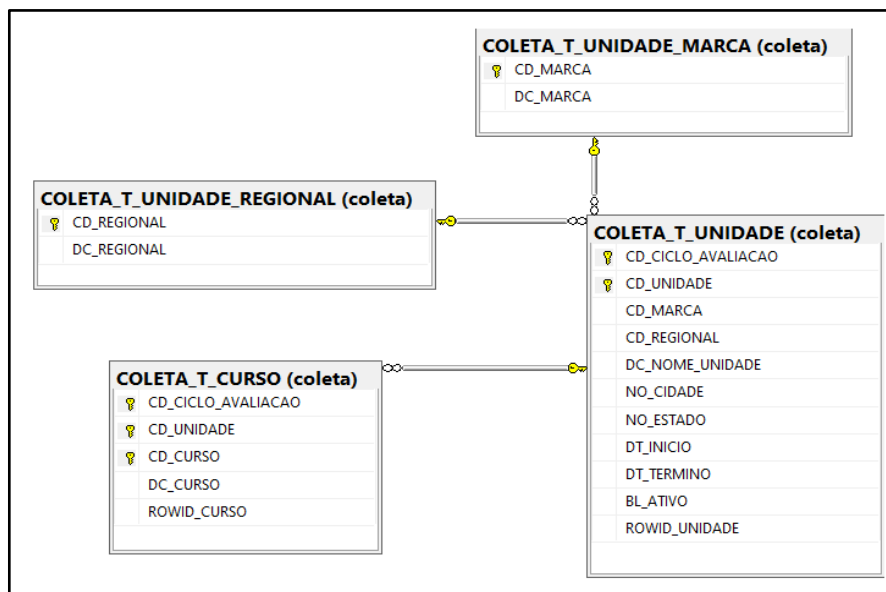


Figura 4. DER tabelas destino

Conforme apresentado na metodologia, a proposta foi desenvolvida para plataforma *web*, usando para *front-end HTML, CSS, Bootstrap e Javascript*. Quanto ao *back-end* foi usada a linguagem *PHP*, com armazenamento dos dados no SGBD relacional *MS SQL Server*. A sua implantação/armazenamento foi em um servidor local localizado no ambiente de estudo, tendo o acesso apenas localmente, já que todo *upload* de arquivos é feito pela equipe de desenvolvedores.

Para implementação da rotina responsável pelo processo de importação dos dados através de planilhas eletrônicas foi usado o auxílio da biblioteca *PhpSpreadsheet*, sendo extraída dentro da classe *FileHandling*, por ser mais fácil de ser utilizada no *controller* para *upload* do arquivo.

Como pode ser visto na linha 36 até a 39 da Figura 5, o arquivo é armazenado em uma pasta de *cache* do projeto, pois desta forma todos os arquivos são guardados como um histórico, para que seja possível ter um controle do processo de importação. Na linha 42 é criada uma variável capaz de ler arquivos com extensão *xlsx*, e a partir dela é lido o arquivo que foi salvo na pasta de *cache* e, então, após a leitura é mostrada uma mensagem de sucesso.

```

32 //status 1, segue com leitura do arquivo
33 if ($out["status"]) {
34     //mover para pasta cache do projeto
35     if (!move_uploaded_file($_FILES["file"]["tmp_name"], $target_file)) {
36         $out["msg"] = "notMovedToDir";
37         $out["status"] = false;
38     }
39
40
41     //ler arquivo com a lib office php
42     $reader = IOFactory::createReader('Xlsx');
43     $spreadsheet = $reader->load($target_file);
44     $sheetData = $spreadsheet->getActiveSheet()->toArray(null, false, false, false);
45     $out["sheetData"] = $sheetData;
46     $out["msg"] = "uploadAndReadSucessfull";
47
48 }

```

Figura 5. Biblioteca *PhpSpreadsheet*

Na função `importStore` dentro da classe `controller`, observa-se na linha 102 da Figura 6, que a classe `FileHandling` é instanciada para a variável de mesmo nome e, a partir dela, é chamada a função `uploadAndReadXlsFile`, onde é passado o arquivo e o tamanho máximo em *bytes* que o mesmo pode ter. Já na linha 111 verifica-se que a variável `avSyncUnidade` está chamando a função `importNew` que está dentro da classe `AvSyncUnidade`.

```

102     $fileHandling = new FileHandling();
103     $uploadAndRead = $fileHandling->uploadAndReadXlsFile($_FILES["file"], 'xlsx', 2 * 1024 * 1024);
104
105     //msg retorno
106     $this->dados["import-msg"] = $uploadAndRead["msg"];
107     $this->dados["import-status"] = $uploadAndRead["status"];
108     if ($uploadAndRead["status"]) {
109         $avSyncUnidade = new AvSyncUnidade($uploadAndRead["sheetData"]);
110         if ($post["import-option"] == "new") {
111             $this->dados["import-log"] = $avSyncUnidade->importNew();
112         }
113     }

```

Figura 6. Função de importação

Na classe `AvSyncUnidade`, as colunas passadas na planilha eletrônica são lidas no construtor da mesma. A ordem do armazenamento dos dados equivale a cada campo da tabela que receberá o número da coluna da planilha. Esta sequência é representada na Figura 7.

```

16     public function __construct($unidadesDataSheet)
17     {
18         $this->unidadesDataSheet = $unidadesDataSheet;
19         $this->columnsDataSheet["Código_Acadêmico"] = 0;
20         $this->columnsDataSheet["CodigoAvaliar"] = 1;
21         $this->columnsDataSheet["NomeAcademico"] = 2;
22         $this->columnsDataSheet["NomeAvaliar"] = 3;
23         $this->columnsDataSheet["TipoUnidade"] = 4;
24         $this->columnsDataSheet["Marca"] = 5;
25         $this->columnsDataSheet["Regional"] = 6;
26         $this->columnsDataSheet["Status"] = 7;
27     }

```

	CodigoAcademico	CodigoAvaliar	NomeAcademico	NomeAvaliar	TipoUnidade	Marca	Regional	Status
1								
2								
3								
4								

Figura 7. Colunas da tabela

A função `importNew` pertence à classe `AvSyncUnidade`, na qual nas linhas 54, 57 e 60 são atribuídos os valores passados pela planilha às suas respectivas colunas pertencentes ao banco de dados, como é mostrado nas Figura 8.

Já na linha 63 foi instanciado um novo banco de dados, pois como observado no DER da Figura 3, para que fosse possível a carga de unidades, é necessário que outras tabelas já estejam carregadas, já que estão relacionadas pelas respectivas chaves.

Na linha 65 é atribuído à variável `daoSyncMarca` o valor recebido pelo arquivo lido e, após, é feita uma busca na tabela física para verificar se o dado já existe. É importante enfatizar que antes do armazenamento das unidades, é necessário ter o cadastro das marcas², já que há uma relação entre elas.

```
53 //set codigo avaliar
54 $daoSyncUnidade->setCodigoAvaliar($unidade[$this->columnsDataSheet["CodigoAvaliar"]]);
55
56 //set nome avaliar
57 $daoSyncUnidade->setNomeAvaliar($unidade[$this->columnsDataSheet["NomeAvaliar"]]);
58
59 //set status
60 $daoSyncUnidade->setStatus($unidade[$this->columnsDataSheet["Status"]]);
61
62 //recuperar codigo da marca a partir do nome e set de marcaid
63 $daoSyncMarca = new DaoSyncMarca();
64 $nomeMarca = StringTransform::stToUpper($unidade[$this->columnsDataSheet["Marca"]]);
65 $daoSyncMarca->setNome($nomeMarca);
66 $marcaId = $daoSyncMarca->getMarcaIdByNome($daoSyncMarca);
67 if ($marcaId) {
68     $daoSyncUnidade->setMarcaId($marcaId);
69 } else {
70     $importStatus[$index]["status"] = false;
71     $importStatus[$index]["msg"][] = "notExistsSyncMarca";
72 }
```

Figura 8. Função `importNew` da classe `AvSyncUnidade`

A Figura 9 aborda um exemplo de tela para todas as ações de *upload*. Para que a ferramenta tivesse maior aceitação pelo usuário final, foi realizado o desenvolvimento de uma tela simples e intuitiva, conforme apresenta a Figura 9, sendo composta por um botão para a escolha do arquivo a ser carregado e dois outros botões sendo um para confirmar a leitura do documento e outro para o cancelamento da mesma.



Figura 9. Tela para *upload*

Com a confirmação da leitura do arquivo, este passará por algumas validações, tais como se o arquivo carregado é do tipo `xlsx` ou se o nome da marca registrado em uma das tuplas do arquivo já existe na tabela real. A Figura 10 aborda dois exemplos de *upload* com erros ocorridos, sendo o primeiro ocasionado pelo tipo de arquivo e o segundo pela importação de dado inconsistente.

² Para explicar marca e unidade, pode ser exemplificado como o Instituto Federal de São Paulo - IFSP, sendo a marca e o Câmpus Hortolândia como a unidade.

Unidades Importar

Dados não importados! O arquivo deve ser do tipo .xlsx.

Voltar

Unidades Importar

STATUS	MENSAGEM	DADOS
OPS!	MARCA NÃO CADASTRADA.	EST0066
	REGIONAL NÃO CADASTRADA.	HORTOLANDIA
	TIPO DE UNIDADE NÃO CADASTRADO.	AMANDA 00000000000
OPS!	MARCA NÃO CADASTRADA.	EST0066
	REGIONAL NÃO CADASTRADA.	HORTOLANDIA
	TIPO DE UNIDADE NÃO CADASTRADO.	CARLA 11111111111

Figura 10. Telas de erro no processo de importação

Já a Figura 11 aborda a leitura do arquivo realizada com sucesso, apresentando uma listagem dos dados recebidos.

Unidades Importar

STATUS	MENSAGEM	DADOS		
SUCESSO!		-1000 EST0066 BOULEVARD SHOPPING LONDRINA - BOULEVARD UNIDADE PRESENCIAL UNOPAR SAO PAULO - SUL 1		
	SUCESSO!		-1002 EST0024 PITÁGORAS - CONTAGEM CONTAGEM UNIDADE PRESENCIAL PITAGORAS LESTE 1	
		SUCESSO!		-1004 EST0089 PITÁGORAS - SÃO LUÍS SÃO LUÍS UNIDADE PRESENCIAL PITAGORAS CENTRO - NORTE 1

Figura 11. Telas que confirmam o processo de importação

Após a carga completa dos dados, os mesmos são listados e apresentados em uma tela para que o usuário possa realizar a conferência dos registros que foram carregados, podendo

exportá-los para arquivos do tipo texto e xlsx e gerenciar os mesmos, conforme apresenta a tela da Figura 12.

Unidades						
PESQUISAR <input type="text"/>						
CÓD. ACADÊMICO	CÓD. AVALIAR	NOME ACADÊMICO	NOME AVALIAR	TIPO UNIDADE	MARCA	REGIONAL
-1000	EST0055	LONDRINA	LONDRINA	UNIDADE PRESENCIAL	MARCA	SAO PAULO - SUL
-1002	EST0024	CONTAGEM	CONTAGEM	UNIDADE PRESENCIAL	MARCA	LESTE
-1004	EST0089	SÃO LUÍS	SÃO LUÍS	UNIDADE PRESENCIAL	MARCA	CENTRO - NORTE
-1006	EST0034	GOV.VALADARES	GOV. VALADARES	UNIDADE PRESENCIAL	MARCA	LESTE
-1007	EST0094	SORRISO	SORRISO	UNIDADE PRESENCIAL	MARCA	CENTRO - NORTE

Figura 12. Listagem dos dados inseridos no processo de upload

Além da possibilidade de inserir dados pelo processo de importação, houve a implementação das funcionalidades adicionar (Figura 13), remover (Figura 14) e atualizar unidade (Figura 15), permitindo que o usuário gerencie os registros individualmente. É importante destacar que unidade é apenas um exemplo das opções implementadas para esta proposta, já que as demais seguem o mesmo padrão no desenvolvimento.

Unidades Adicionar

Cód. Acadêmico *

Cód. Avaliar *

Nome Acadêmico *

Nome Avaliar *

Tipo Unidade *

Marca *

Regional *

* Preenchimento obrigatório.

Figura 13. Tela de cadastro unidade

Unidades Atualizar

Cód. Acadêmico *
-1002

Cód. Avaliar *
EST0024

Nome Acadêmico *
Contagem

Nome Avaliar *
CONTAGEM

Tipo Unidade *
UNIDADE PRESENCIAL

Marca *

Regional *
LESTE

Cancelar Atualizar

Figura 14. Tela de atualização unidade

Unidades Remover

Cód. Avaliar: EST0015
Cód. Acadêmico: FAC
Nome Acadêmico: Campinas
Nome Avaliar: CAMPINAS
Tipo Unidade: Unidade presencial
Marca: Faculdade
Regional: LESTE

Confirma a remoção da unidade acima? Não Sim

Figura 15. Tela de remoção unidade

A Figura 16 apresenta um diagrama com a execução dos processos depois da ferramenta proposta e implantada. Em comparação à Figura 1, observa-se que houve uma otimização do fluxo dos processos e, conseqüentemente, agilidade no carregamento dos arquivos à ferramenta para avaliação institucional e com isso a equipe de desenvolvimento não está envolvida durante o processo.

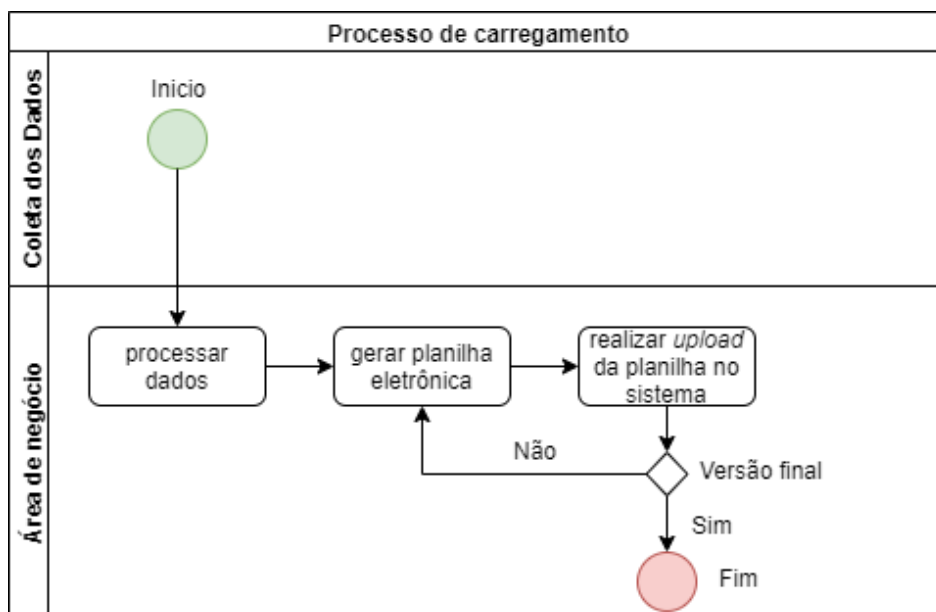


Figura 16. Processo para *upload* de arquivos após a criação da ferramenta

5. Conclusão

A proposta foi implantada em uma empresa para auxiliar no *upload* e tratamento dos dados que serão utilizados para realizar a avaliação institucional de diversas instituições de ensino e com isso tornando o processo mais simples e eficaz.

Para desenvolvimento deste trabalho e até mesmo destacando as competências e habilidades adquiridas no decorrer do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, houve aplicação de conceitos associados às disciplinas de Desenvolvimento *Web*, Banco de Dados, Engenharia de *Software*, Metodologia de Pesquisa e Projeto de Sistemas para planejamento e desenvolvimento teórico-técnico.

6. Trabalhos futuros

Como trabalhos futuros sugere-se a implementação do *upload* de registros associados aos dados de disciplinas e alunos. Um dos requisitos a ser estudado para importação de dados é a inclusão apenas de alunos que estão aptos no processo de avaliação institucional, já que atualmente este processo é feito manualmente pela equipe de desenvolvimento com a exclusão desses registros.

Referências

- Congresso Nacional. (2004). “Lei nº 10.861, de 14 de abril de 2004. Institui o Sistema Nacional de Avaliação da Educação Superior – SINAES e dá outras providências, Brasília, DF”. http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2004/lei/110.861.htm. [Online: acessado em 23 de outubro de 2019].
- Flanagan, D. (2011). “JavaScript: The Definitive Guide”. Editora O’Reilly, 6ª edição.
- Google Forms. <https://www.google.com/forms/about/>. [Online: acessado em 21 de agosto de 2019].
- Heuser, C. A. (2009). “Projeto de banco de dados”. Editora Bookman, 6ª edição.

- INEP. <http://inep.gov.br/processo-de-avaliacao>. [Online: acessado em 18 de setembro de 2019].
- Laudon, K. C., Laudon, J. P. (2014). “Sistema de informação”. Pearson Education do Brasil, 11ª edição.
- Lerdorf, R., Macintyre, P., Tatro, K. (2013). “Programming PHP”. Editora O’Reilly, 3ª edição.
- Mazza, L. (2014). “HTML5 e CSS3: Domine a web do futuro”. Editora Casa do Código.
- Miyagusku, R. (2008). “Curso Prático de SQL - Guia de referência completo para usar a linguagem SQL nos bancos de dados: MS SQL Server, Oracle, PostgreSQL, MySQL”. Editora Digerati Books.
- Rezende, D. A. (2016). “Planejamento de sistemas de informação e informática”. Editora Atlas, 5ª edição.
- Schmitz, D. (2014). “Bootstrap: framework front-end para desenvolvimento web e mobile”. https://kupdf.net/download/livro-bootstrappdf_5d2c9520e2b6f5b11da54a5b_pdf. [Online: acessado em 20 de outubro de 2019].
- Schwaber, K., Sutherland, J. (2017). “The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game”. <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>. [Online: acessado em 20 de outubro de 2019].
- Silva, N. (2007). “Análise e Estruturas de Sistemas de Informação”. Editora Érica.
- SurveyMonkey. <https://pt.surveymonkey.com>. [Online: acessado em 20 de agosto de 2019].
- Zemel, T. (2015). “CSS Eficiente: Técnicas e ferramentas que fazem a diferença nos seus estilos”. Editora Casa do Código.

Documento Digitalizado Restrito

Versão Final TCC

Assunto: Versão Final TCC
Assinado por: Michele Barion
Tipo do Documento: Projeto
Situação: Finalizado
Nível de Acesso: Restrito
Tipo do Conferência: Documento Original e Cópia

Documento assinado eletronicamente por:

■ **Michele Cristiani Barion, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 20/04/2020 13:47:53.

Este documento foi armazenado no SUAP em 20/04/2020. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 396171

Código de Autenticação: 22bc76e350

