



ISSN: 2230-9926

Available online at <http://www.journalijdr.com>

# IJDR

*International Journal of Development Research*

Vol. 10, Issue, 09, pp. 40942-40952, September, 2020

<https://doi.org/10.37118/ijdr.20099.09.2020>



RESEARCH ARTICLE

OPEN ACCESS

## AUDITMODEL TOOL: UMA FERRAMENTA PARA AUDITORIA BASEADA NA ISO 19011: DIRETRIZES PARA AUDITORIAS EM SISTEMAS DE GESTÃO

<sup>1</sup>Allison Ramon Araújo de Santana, <sup>1</sup>Paulo Caetano da Silva and <sup>2</sup>Marcio A. Pereira da Silva

<sup>1</sup>PPGCOMP – UNIFACS; <sup>2</sup>Universidade Federal de Pernambuco – UFPE

### ARTICLE INFO

#### Article History:

Received 19<sup>th</sup> June 2020

Received in revised form

14<sup>th</sup> July 2020

Accepted 22<sup>nd</sup> August 2020

Published online 30<sup>th</sup> September 2020

#### Key Words:

Isolation, Marginalization, Displacement, Patriarchal, Diasporic sensibility, Supremacy, suppression and dominance.

\*Corresponding author: Kalpana Pandey,

### ABSTRACT

Auditorias tradicionais, *in-loco*, são realizadas todos os anos em várias organizações, aumentando o custo (hospedagem, passagem, alimentação) envolvidos neste processo. Quando as auditorias são realizadas através do uso de ferramentas, que costumam ter custos elevados, a produtividade da equipe de auditor é reduzida em virtude do excesso de configurações durante as auditorias. A adoção de modelos pode diminuir os custos de desenvolvimento e viabilizar a criação de ferramentas com menos scripts configuráveis. Neste trabalho é proposto um modelo de representação de processos de auditoria baseado na ISO 19011, denominado AuditModel. São especificados 14 elementos, classificados em 04 domínios, que juntos compõem o modelo. Para validação do AuditModel são identificadas características presentes em ferramentas de auditoria e implementada uma ferramenta, denominada AuditModelTool. Um processo de negócio foi mapeado e auditado de forma a ilustrar quais os passos necessários para realização de uma auditoria. Espera-se que com o modelo e ferramenta propostos seja possível mapear e executar qualquer processo de auditoria. Disponível em um repositório público para desenvolvedores a ferramenta proposta fomenta o desenvolvimento de *software* de maneira padronizada, baseados na ISO 19011 e em um modelo genérico de representação de processos de auditoria.

Copyright © 2020, Kalpana Pandey and Dr. Vinod Kumar Singh. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Citation: Kalpana Pandey and Dr. Vinod Kumar Singh, 2020. "The treatment of women and family in anjana appachana's "Bahu"", *International Journal of Development Research*, 10, (09), 40942-40952.

### INTRODUCTION

Organizações têm buscado informatizar seus ambientes operacionais de forma a possibilitar que as auditorias internas e externas possam ser realizadas de forma automatizada, possibilitando uma menor utilização de papel. Segundo a ISO (*International Organization for Standardization*) em 2018 foram realizadas 1.307.622 (um milhão trezentos e sete mil e seiscentos e vinte e duas) auditorias. Para Vasarhelyi et al em algumas empresas as auditorias ainda são realizadas de maneira tradicional, *in loco*, o que acarreta elevação dos custos com auditoria em virtude de custos com alimentação, locomoção e hospedagem dos auditores. Lyra entende que a produtividade da equipe de auditoria é menor quando são necessárias configurações em ferramentas de auditoria. De acordo com Teruel, algumas ferramentas de auditoria têm alto custo, bem como as atualizações de ferramentas diferentes instaladas no mesmo ambiente computacional são mais propícias a erros em virtude da diversidade de características de cada ferramenta, da necessidade de configuração de *scripts*, que em alguns casos são sobrescritos gerando incompatibilidade com a versão anterior.

A definição coerente dos processos e a padronização diminuem o retrabalho e garantem a redução de erros, que podem causar impactos negativos no ambiente organizacional. A padronização de dados para auditoria auxilia na redução de problemas: falta de acessibilidade, transparência dos dados disponibilizados e/ou coletados, alto custo no ato da coleta de dados bem como esforços superiores aos necessários em várias fases da auditoria. Segundo Flowerday et al a variedade de formato de dados e registros gerados pelos sistemas em uso nas organizações atrapalha o desenvolvimento de soluções computacionais focadas em auditoria. Existem algumas iniciativas para a padronização de dados de auditoria, um dos exemplos mais significativos é o *Audit Data Standards*, iniciativa da AICPA (*American Institute of Certified Public Accountants*) cujo objetivo é desenvolver um modelo de dados padronizado que facilite o uso de análises aprimoradas. Segundo Huanzhuo, quando se tem a necessidade de garantir requisitos, nem só de produtividade, mas também de adaptabilidade, flexibilidade e interoperabilidade, a padronização é muito discutida, tanto nas esferas organizacionais públicas, quanto nas privadas.

Entretanto, com exceção da AICPA, são poucos ou quase inexistentes grupos de trabalho que buscam criar padrões, normas e modelos focados em diretrizes que possibilitem a automatização da auditoria. Na perspectiva do entendimento de quais passos devem ser seguidos e quais diretrizes devem ser consideradas como requisito para o desenvolvimento de projetos de *softwares* tendo a auditoria como foco, os estudos são ainda mais escassos. Para os procedimentos tradicionais de auditoria (inspeção de documentação, entrevistas) consomem muito tempo dos auditores e aumentam a chance de erros. A adoção de um modelo de representação de processos de auditoria no desenvolvimento de sistemas poderá diminuir o custo de desenvolvimento apontado por Teruel, através do fomento e compartilhamento de informações na academia. O compartilhamento de informações possibilita desenvolvimento de ferramentas *open-source* que podem ser distribuídas através de repositórios públicos, incentivando a criação de grupos de pesquisa que busquem soluções e aprimorem o modelo contribuindo para o aumento de ferramentas gratuitas de qualidade. Com boas ferramentas é possível forçar empresas privadas a diminuir o custo de suas aplicações em virtude do aumento da concorrência e descentralização de informações.

De acordo com a ISO atualmente são 23 mil 117 padrões normativos, desenvolvidos por ela, que cobrem quase todos os aspectos de tecnologia e fabricação no mundo, muitos deles são auditados utilizando as diretrizes da ISO 19011, que é genérica e independente do sistema de gestão definido como requisito de auditoria (e.g. qualidade, meio ambiente, segurança do trabalho, segurança da informação). Assim como a ISO, o modelo apresentado neste trabalho propõe uma independência dos sistemas de informação em relação ao processo específico auditado e aos requisitos a serem utilizados na auditoria. A representação do processo de auditoria através do modelo proposto fornecerá uma estrutura padrão que possibilitará, uma vez que os elementos propostos já são utilizados em auditorias de certificação, uma maior abrangência dos *softwares* quanto ao atendimento em diversos seguimentos de mercado (e.g. indústria, agronegócio, logística). A padronização da representação do processo de auditoria nos sistemas de informação (SI) tornaria sua execução mais ágil. A adoção de padrões e modelos reduz erros, evita problema de comunicação e melhora a segurança, além de reduzir custos financeiros e operacionais (MÍKVA, PRAJOVÁ, YAKIMOVICH, KORSHUNOV, 2016).

A fim de identificar um modelo ou padrão que represente o processo de auditoria automatizada foi feita uma revisão da literatura sobre a padronização de processos de auditoria que identificou 97 normas, sendo que destas, noventa e dois padrões (94,84 %) são relacionados à definição de conceitos aplicados na auditoria e apenas cinco (5,15%) relacionados à realização de auditoria contínua através da padronização de dados. Entre as propostas dos modelos de representação de dados existe o *Audit Data Standards*, uma proposta de padronização de dados de auditoria baseada em XBRL GL do Instituto Americano de Contadores Públicos Certificados (*American Institute of Certified Public Accountants*). Através do estudo observa-se a necessidade da criação de um modelo genérico para representação do processo de auditoria, uma vez que a AICPA busca definir padrões de dados e as normas evidenciadas não descrevem formas de automatizar o processo de auditoria. É uma contribuição positiva exemplificar o processo de forma a possibilitar o entendimento por parte de

desenvolvedores quanto aos requisitos necessários para realização de uma auditoria. Para proporcionar uma visão geral sobre ferramentas de auditoria, com o objetivo de identificar, classificar e compará-las, foi realizada outra revisão da literatura sobre Ferramentas de Auditoria Auxiliadas por Computador – TAAC's. Foram identificadas 8 ferramentas, das quais 62,5% são multiplataforma (i.e. compatíveis com sistemas operacionais distintos: *Windows, Android, IOS e Linux*), porém, apenas 25% são ferramentas classificadas como *open-source*. A parcela de *softwares* responsivos corresponde a 37,5% do total e 50% possibilitam a realização de auditorias remotas. Apenas 25% das ferramentas encontradas não emitem *Reports* de auditoria e 50% não emitem planos de ação. Observou-se com essa revisão da literatura que algumas ferramentas possuem um grande acoplamento do *software* ao processo de auditoria. O ideal é que a ferramenta possa ser adaptativa em relação ao processo de auditoria que seja aplicado, pois torna a sua utilização mais abrangente.

Destarte, neste trabalho é proposto o AuditModel, um modelo para representação de processos de auditoria. O AuditModel é baseado na ISO 19011: Diretrizes para Auditorias em Sistemas de Gestão e propõe a representar processos de auditoria de forma que permita a construção de sistemas de informação de auditoria desacoplados do processo a ser auditado e aos requisitos da auditoria. Para os autores o AuditModel possibilita o desenvolvimento de ferramentas de fácil entendimento por parte das equipes de desenvolvimento de *software*, suprimindo a deficiência apresentada por Lyra referentes às configurações de algumas ferramentas de auditoria, por meio de *scripts*, antes de serem utilizadas em processos específicos.

Diante do exposto, para validação do modelo proposto, apresenta-se neste trabalho uma ferramenta baseada no AuditModel e denominada AuditModelTool com o propósito de possibilitar a realização de auditorias em processos de negócios diferentes. Neste artigo, na segunda seção, são apresentados trabalhos correlatos sobre Ferramentas de Auditoria Auxiliadas por Computador – TAAC's. Na terceira seção encontra-se o AuditModel (modelo para representação de processos de auditoria), e as subseções 2 (arquitetura da ferramenta *Audit Model Tool*), 3 (o processo de implementação da ferramenta *Audit Model Tool*) e 4 (validação da ferramenta *Audit Model Tool* através a realização de uma auditoria em um processo de negócio mapeado). Na conclusão, são apresentadas as considerações finais deste trabalho.

**Ferramentas de Auditoria Auxiliadas por Computador – TAAC's** A partir da aplicação do protocolo de pesquisas nas bases de dados selecionadas (*ACM Digital Library, IEEE, Sielo, Science Direct e no Google Academy*) foram identificados 179 artigos, que tiveram seus títulos e resumos lidos de forma a identificar uma possível correlação com ferramentas de auditoria auxiliadas por computador. Porém, 92,07% (165) dos artigos identificados não tinham correlação direta com os critérios e objetivos deste trabalho, considerando a extensão e aplicabilidade dos temas relacionados a auditoria.

Após a leitura dos títulos e resumos, 8,48% (14) foram selecionados e lidos completamente, destes, 64,28% (09) discutiam sobre ferramentas de auditoria e foram selecionados para análise. Os 35,71% (05) artigos não selecionados foram

trabalho que apresentaram: um protótipo para fraudes visuais, inovações em análises de dados em banco, determinantes para aceitação em auditorias, perspectivas de sistemas de informações contábeis e fundamentos de auditoria contínua. Não sendo possível obter dados sobre ferramentas de auditoria propriamente ditas. As ferramentas de auditoria possibilitam a atuação do auditor em diversas informações simultâneas provocando uma redução considerável do tempo para conclusão de uma auditoria (HUANG e HUANG, 2017). A abordagem e consolidação das ferramentas TAAC's começou em meados de 1980 e para Alves são utilizadas para automatizar e simplificar a análise de dados de forma eficiente. Segundo Alves, sistemas informatizados para auditoria aprimoram o planejamento e gerenciamento das auditorias, geram análises mais robustas, promovem a melhoria contínua e reduz o custo com análises manuais, além de aprimorar os procedimentos de auditoria das organizações.

Para estas ferramentas podem ser classificadas como generalistas ou especializadas, sendo que as generalistas costumam realizar simulações, análises geral de amostras, sumarizações e verificações de duplicidades enquanto as especializadas são desenvolvidas para realização de tarefas específicas e objetivas, normalmente são parametrizadas pelos próprios auditores. Segundo as ferramentas generalistas são pré-configuradas e podem ser utilizadas por profissionais que não tem conhecimento técnico em auditoria, porém, para Lyra a realização de configurações em ferramentas de auditoria, por parte de auditores experientes diminui a produtividade. Lyra afirma que as ferramentas especialistas proporcionam resoluções para problemas específicos, normalmente focados em seguimentos de mercado especializado e comumente são desenvolvidas com foco no crédito imobiliário, leasing, cartão de crédito, entre outras especialidades. Para Teruel as especificidades das ferramentas especialistas aumentam seu custo e dificultam a atualização dos ambientes computacionais. Ademais, ele discute vantagens e desvantagens de algumas ferramentas de auditoria, e.g. IDEA, Galileo, Pentana e Suíte Trauma Zer0. Um protótipo de uma ferramenta para uso na nuvem é apresentado por S. Huang et al., o projeto foi denominado de J-CAAT's (HUANG e HUANG, 2017).

Os softwares Audit Command Language (ACL) e Interactive Data Extraction&Analysis (IDEA) são utilizados como referência e comparação nesse trabalho. Apesar dos autores afirmarem que a ferramenta J-CAAT's é open-source, não foi possível evidenciar sua disponibilização na internet em repositórios públicos. Em seu trabalho, Shing-Han apresenta a ferramenta Real-Time Log AuditMechanism (RTLAM) que foi desenvolvida para identificação de fraudes. Seu foco é analisar grandes quantidades de logs, além de não operar em rede, ela é baseada no sistema operacional Windows 7. Seu processo de execução é composto por quatro etapas: importação de dados, configuração, compactação e auditoria. A ferramenta é apresentada como open-source, mas, assim como a ferramenta J-CAAT's, não foi encontrada na internet. M. R. Lyra e J. O. Imoniana, em seus respectivos livros apresentam diversos conceitos utilizados na fundamentação deste trabalho, bem como nos resultados apresentados neste artigo.

A partir desta revisão da literatura, identificamos os seguintes aspectos para serem considerados neste trabalho: A necessidade de suprir a deficiência apresentada por uma vez

que são citadas as principais técnicas de auditoria, entretanto, não apresenta um modelo para representação de processos de auditoria e nem soluções para o desenvolvimento de sistemas de informações de auditoria. Com os trabalhos de (HUANG e HUANG, 2017) e entendemos a importância de mantermos os arquivos do projeto disponível em repositórios públicos para que sejam utilizados em propostas posteriores. Além disso (HUANG e HUANG, 2017) pondera a importância da responsividade em ferramentas web, por isso utilizamos um layout responsivo na *AuditModel Tool*. A ferramenta apresentada por J-CAAT's não pode ser executada em rede, considerando este aspecto, inserimos a *AuditModelTool* na internet para que seja possível o acesso em qualquer ambiente, desde que exista um navegador web disponível.

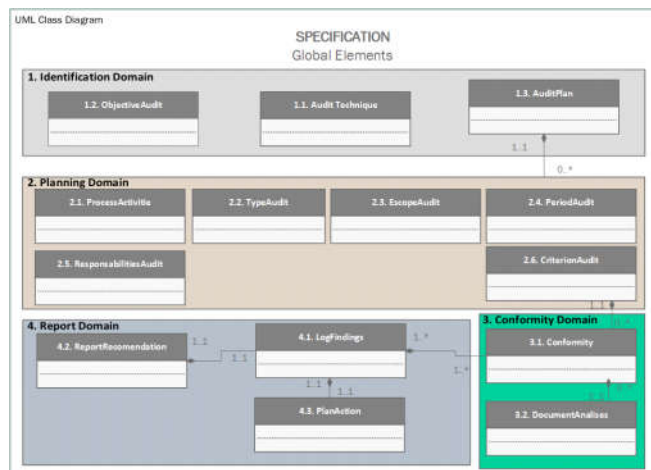
## Audit Model

Nesta seção será apresentada a solução proposta para os problemas discutidos na Introdução deste artigo. Para a questão de padronização da representação de processos de auditoria, um modelo de representação, o *AuditModel*, é apresentado, nele é possível representar diferentes processos de auditoria, que se materializam na ferramenta proposta, *AuditModelTool*. Na ferramenta é possível representar e executar diferentes processos de auditoria, no estágio atual de desenvolvimento, apenas o processo de auditoria da ISO 19011 foi implementado. A partir dessa flexibilidade é possível mapear e executar diferentes processos de negócio para serem auditados. Outras características também foram implementadas na ferramenta para solucionar os problemas como: falta de compatibilidade com diferentes sistemas operacionais, adaptabilidade a dispositivos móveis, baixa disponibilização de ferramentas *open-source*, além da diminuição do acoplamento do software ao processo de auditoria.

**Audit Model, um modelo para representação de processos de auditoria:** O *AuditModel* é apresentado com o propósito de suprir as deficiências apresentadas por diversos autores e discutidas na introdução, i.e. erros e baixa produtividade de auditores, configuração de ferramentas de auditoria, baixa padronização de processos e ferramentas de auditoria, interdependência entre ferramenta e processos. O *AuditModel* é composto por 14 elementos que foram definidos em virtude da sua importância para realização de auditorias. Esses elementos compõem os blocos de domínios que fazem alusão às fases a serem executadas durante um processo macro de auditoria: Identificação (interna ou externa), Planejamento (processo, responsável, local), Conformidade (requisitos, evidências) e Relatório (conclusão, ações futuras).

Os elementos do modelo representam as informações que devem ser fornecidas no processo de auditoria e foram especificados considerando nomenclaturas que já são utilizados por auditores. Exemplo destas nomenclaturas são: escopo de auditoria (definição de onde as auditorias são realizadas), registros (documentos a ser auditado), critérios (requisitos a serem verificados), relatórios de auditorias (contém as não conformidades evidenciadas), plano de ação (descreve as atividades a serem realizadas para sanar a não conformidades).

A Figura 1 apresenta os quatro modelos do domínio e seus respectivos elementos.



Fonte: Próprio Autor.

**Figura 1. Domínios e seus elementos**

A finalidade dos domínios e a explicação de cada elemento é apresentado a seguir:

**Identification:** Identificar a auditoria. Neste domínio é possível evidenciar qual tipo de auditoria será realizada e associar a um plano de auditoria específico. Nele ocorre a definição do tipo de auditoria, se é de rotina, análise de conformidade ou reanálise para validação de uma auditoria anterior. É composto por três elementos: *ObjectiveAudit* (verificação, conformidade ou reanálise), *AuditTechnique* (e.g 19011, referência à norma que a auditoria é baseada) e *AuditPlan* (identifica o plano de auditoria a ser preenchido nos elementos do domínio de *Planning*);

**Planning:** Se especifica a estrutura necessária para o planejamento das atividades a serem realizadas. Seus elementos visam possibilitar a criação de um plano (direcionado e estratégico) que é utilizado como base para realização da auditoria. Através deste domínio é possível definir o período, qual é o processo que será auditado, o tipo de auditoria a ser realizada, quem são os responsáveis pela auditoria e quais serão os critérios a serem utilizados. Sua estrutura é formada por seis elementos: *Process Activitie* (processo a ser auditado), *Scope Audit* (nível hierárquico da auditoria e confidencialidade), *PeriodAudit* (data para realização da auditoria), *TypeAudit* (interna ou externa), *Responsabilities Audit* (auditor responsável) e *Criterion Audit* (critérios que serão utilizados como parâmetros para análise de conformidade);

**Conformity:** Tem a função de atestar, através da evidência, a conformidade de uma amostra quando comparado aos requisitos estabelecidos no plano de auditoria. Seus elementos são: *Conformity* (confronta os critérios definidos no elemento *CriterionAudit*, do *Planning*, com os documentos a serem analisados), *Document Analyses* (documentos a serem analisados);

**Report:** É o domínio no qual se apresenta os resultados da auditoria, sendo ele o responsável por comunicar aos envolvidos as evidências, bem como as respectivas não conformidades evidenciadas no processo de auditoria. Seus três elementos são: *Report Recomendation* (contém informações sobre as não conformidades e os requisitos), *Log Findings* (registro das não conformidades encontradas) e *Plan*

(plano com prazo para resolução das não conformidades). As relações existentes na Figura 01 são: de composição, entre os domínios *Planning* e *Identification*, isto porque na identificação da auditoria é necessário definir um plano de auditoria, o qual será composto pelos elementos do domínio de *Planning*. Estes elementos possuem detalhes que possibilitam a programação formal do processo de auditoria a ser realizado; outra relação de composição é necessária entre os domínios *Planning* e *Conformity*, pois, para realização da análise da conformidade de um dado será necessário utilizar os critérios definidos durante a definição do planejamento, no plano de auditoria e, com base nestes critérios compará-los com o dado a ser auditado de modo atestar a conformidade; uma agregação existe entre os domínios *Conformity* e *Report*, pois, os *relatórios* agregam informações oriundas das não conformidades e critérios utilizados no processo de auditoria pelo domínio *Conformity*.

Quanto as multiplicidades representada na Figura 01 pelos números 1, 0 e pelo símbolo \*, mostram que o domínio *Identification* pode estar relacionado a vários ou nenhum domínio *Planning*, relação do tipo associação (0..\*), mas um domínio *Planning* obrigatoriamente deve ter uma identificação (1..1). O domínio *Planning* pode ter várias ou nenhuma análise de conformidade através do domínio *Conformity* (0..\*), mas a conformidade só ocorre se houver o plano (1..1). O domínio *Conformity* pode não precisar gerar um *Report* (0..\*), porém, o domínio *Report* deve ter no mínimo uma análise de conformidade (1..\*).

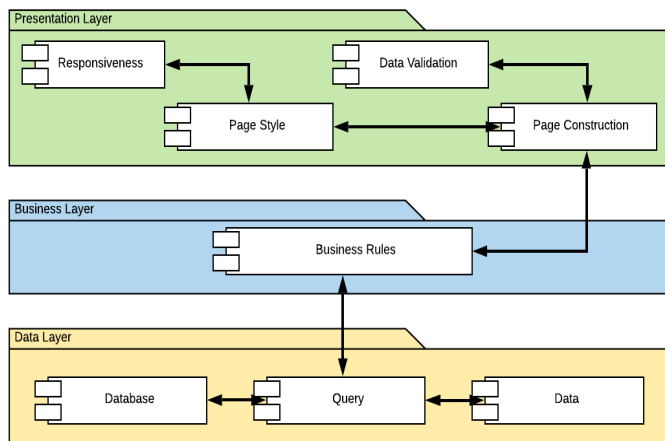
Os domínios podem ser utilizados de forma independentes, por exemplo: Um processo organizacional a ser auditado pode ser definido no domínio *Identification* e registrado um plano de ação no domínio *Planning*, porém, o domínio *Conformity* pode não ser acionado e, conseqüentemente, o *Report* também não será, já que a auditoria programada não aconteceu. Os elementos propostos neste modelo foram definidos e conceituados tendo como base as seções 5 (estabelecendo objetivos do programa de auditoria) e 6 (Condução de auditorias) da norma ISO 19011. Entretanto, o *AuditModel* é um modelo extensível, permitindo que novas definições sejam adicionadas, possibilitando assim a modelagem de outros processos de auditoria diversos da ISO 19011. Os 14 elementos propostos podem conter informações de qualquer processo, para isso é necessário mapeá-lo, obtendo as informações necessárias para que sejam inseridos no sistema todos os dados necessários de maneira coerente, possibilitando a realização da auditoria com o auxílio da ferramenta. A seguir é apresentada a arquitetura da ferramenta *AuditModelTool* que incorpora os conceitos aqui descritos, esta mesma arquitetura pode ser implementada utilizando tecnologias diferentes daquelas usadas no seu desenvolvimento.

**AuditModel Tool, Arquitetura de Software:** A arquitetura de *software* compreende as propriedades, relações e estruturas entre os elementos de um *software*. Ainda segundo esse autor os principais motivos para o mapeamento de uma arquitetura para um sistema são: a comunicação que a arquitetura oferece entre as diferentes partes interessadas envolvidas no projeto, a possibilidade de definir, logo em primeiro momento, as restrições de implementação, desempenho, escalabilidade, atributos de qualidade e, por último, a abstração, que possibilita reutilização do modelo especificado em outros projetos de *softwares*. *Softwares* são complexos e,



consequentemente, requerem um maior esforço da equipe de desenvolvimento, além de uma postura disciplinada no desenvolvimento das aplicações (SUNAGA ET AL., 2017).

Considerando os motivos apresentados por para mapeamento de arquitetura dos sistemas e a complexidade no desenvolvimento de ferramentas de *softwares* é apresentada neste trabalho a arquitetura da *Audit Model Tool*, vide Figura 2. A arquitetura proposta considera a existência de três camadas, sendo: *Presentation*, *Business* e *Persistence*.



Fonte: Próprio Autor.

**Figura 2. Audit Model Tool System Architecture**

A *Presentation Layer* é composta por quatro componentes que juntos possibilitam uma melhor interação do usuário com a ferramenta. É através desta camada que todos os dados são validados, configurados e adaptados para apresentação, formando a interface da aplicação.

Os quatro componentes da *Presentation Layer* são apresentados a seguir:

- **Page Construction:** Componente responsável por viabilizar a interpretação, em diferentes navegadores web, de dados obtidos através das requisições realizadas pela *Business Layer* a *Persistence Layer*. Este componente apresenta, através dos navegadores, parágrafos, listas, títulos, etc;
- **Data Validation:** É utilizado para analisar criticamente as entradas de dados na ferramenta quando estes foram inseridos pelo usuário. Alguns campos de formulários têm seus preenchimentos obrigatórios e essa validação é importante para o funcionamento da ferramenta e para realização da auditoria. Os dados validados são direcionados para o componente *Page Construction* de forma possibilitar uma interação com usuário;
- **Page Style:** Se propõe a organizar graficamente a página web com os dados vindos do componente *Page Construction*. Este componente posiciona os dados recebidos em várias locais do navegador, bem como formata e apresenta imagens e dados de maneira intuitiva;
- **Responsiveness:** Responsável por identificar o dispositivo ao qual o usuário está fazendo acesso a ferramenta, de forma a adaptar o layout para este dispositivo proporcionando uma boa experiência de usabilidade por parte do usuário.

A *Business Layer* é responsável por materializar, através do componente *Business Rules*, os elementos e a lógica do modelo *Audit Model*. Esta camada detém as definições e regras lógicas que viabilizam a realização da auditoria pela ferramenta. A *Business Layer* é a única camada que realiza e recebe requisições das demais camadas, uma vez que as informações são consultadas na fonte de dados (camada de dados), tratadas através do componente *Business Rules* e apresentadas através da *Presentation Layer*.

A última camada da arquitetura da ferramenta *Audit Model Tool*, denominada *Persistence Layer*, é responsável por manter os dados a serem utilizados e tratados no sistema. Esta camada é composta por três componentes (*Data*, *Query* e *Database*) apresentados a seguir:

- **Database:** Modelado com base nos elementos do *Audit Model* este componente se comunica e troca informações diretamente com a camada *Query*. É responsável por armazenar os dados recebidos de forma viabilizar sua utilização para operacionalização da ferramenta;
- **Query:** Este componente tem o objetivo de extrair e configurar dados do componente *Data* e armazenar estas informações no componente *Database*. É através dele que são realizados acessos às informações que serão repassadas para o componente *Business Rules* (da *Business Layer*);
- **Data:** Componente que representa os dados que serão extraídos para realização da auditoria. Podem ser enviados para realização que será auditada em diversos formatos e.g relacional, CSV, XML, XBRL (XBRL, 2017), *Audit Data Standards - XBRL GL* (AICPA, 2020), ASCII, Json.

Espera-se que com essa arquitetura sistemas de informação de auditoria possam ser construídos com de forma a executar diferentes processos de auditoria, baseados na ISO 19011 e em outros padrões de processos de auditoria.

**Audit Model Tool: Implementação:** Nesta seção serão apresentados características, decisões e tecnologias usadas para a implementação da ferramenta *Audit Model Tool*, são mostrados os componentes de cada camada da arquitetura e seus elementos.

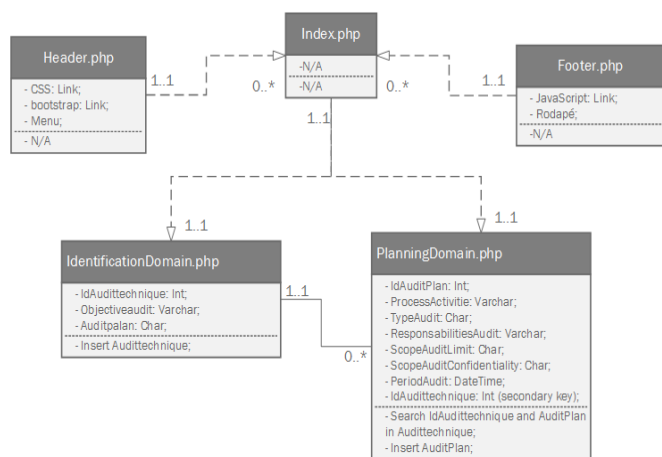
Primeiro serão apresentadas as tecnologias utilizadas para implementação dos componentes apresentados na arquitetura (*Presentation*, *Business* e *Persistence*):

i) **Presentation Layer:** O componente *Page Construction* foi implementado por meio da linguagem de marcação HTML5 (*HyperText Markup Language*) uma vez que esta linguagem permite exibir parágrafos, listas, cabeçalhos. A formatação do conteúdo HTML é realizada pelo componente *Page Style*, através do uso da tecnologia CSS (*Cascading Style Sheets*). Com o CSS é possível configurar os aspectos visuais dos documentos web, e.g. cores, tamanhos e imagens. Para que fosse possível implementar recursos de validação de formulários necessários no componente *Data Validation*, utilizou-se a linguagem de programação *Javascript*. Completando a lista de componentes da *Presentation Layer* o componente *Responsiveness* foi materializado através do uso do *framework Bootstrap* para permitir a responsividade da ferramenta;

ii) **Business Layer:** O componente *Business Rules* foi implementado através do uso da linguagem de programação PHP (*Hypertext Preprocessor*) que foi utilizada para codificação e representação, através de classes, das regras de negócio da aplicação que foram elaboradas através do mapeamento dos elementos do AuditModel;

iii) **Persistence Layer:** Para implementação do componente *Database* foi realizada uma análise de requisitos afim de identificar quais registros deveriam conter no banco de dados relacional (MARIADB, 2020) utilizado durante o desenvolvimento da ferramenta. Com base neste estudo foi realizada uma modelagem através da ferramenta *WorkBeanch* e criado o banco de dados da ferramenta. O componente *Query* é representado através das classes *Document Analises Import.php* e *Criterion AuditImport.php*. Através destas classes é possível obter dados do componente *Data* e importá-los para o banco de dados através de métodos construídos em PHP para este fim. Estes códigos recebem os arquivos de dados importados, estabelecem a comunicação com o banco de dados, separam os dados recebidos e os inserem nas tabelas do banco de dados. Nesta primeira versão da ferramenta o componente *Data* é representado através da tecnologia .CSV. A importações de outros tipos de dados, e.g. XML, XBRL, Json, serão implementadas em versões posteriores desta ferramenta.

As saídas de dados do Audit Model Tool ocorrem com a exportação de arquivos em formato PDF (*Portabledocumentformat*). Que são gerados pelas classes *Log Findings.php*, *ReportRecommendation.php* e *PlanAction.php*. A seguir são apresentadas as classes implementadas no projeto. Segundo as aplicações em PHP costumam ter excelentes performances, é uma linguagem interpretada por navegadores web e pode ser considerada de fácil aprendizado. Por se tratar de um sistema web com mecanismos de responsividade que adapta o *layout* para qualquer dispositivo, a PHP foi a linguagem de programação selecionada para a implementação do projeto. Todas as estruturas de composição de cada arquivo PHP que constam na ferramenta foram representadas nos diagramas de classes mostrados nas Figura 3, cujas classes são: *Home* (representa a página inicial da ferramenta), *Identification Domain* e *Planning Domain*, Figura 4, classes que compõem o domínio *Conformity Domain* e Figura 5, classes do domínio *Report Domain*.

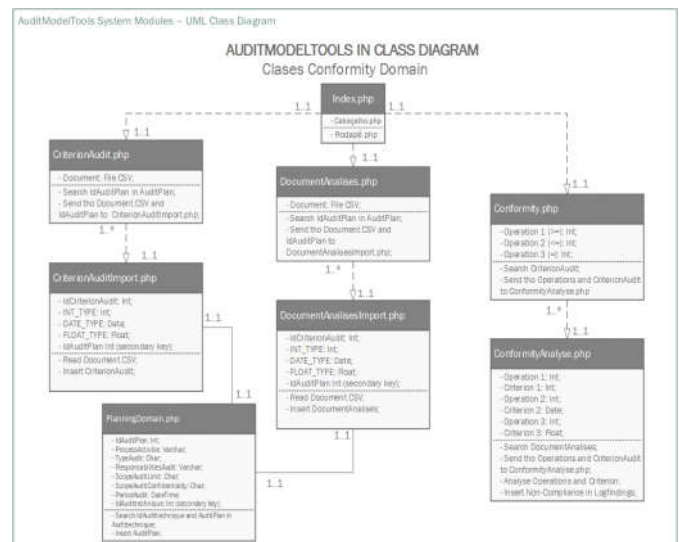


Fonte: Próprio Autor.

**Figura 3. Classes PHP que compõem o componente Business Rules (Business Layer): Home, Identification Domain (Pacote do Audit Model) e Planning Domain (Pacote do AuditModel)**

Na Figura 3 são representadas as classes *Index.php*, *IdentificationDomain.php* e *PlanningDomain.php*. A classe *index.php* possui duas subclasses na sua estrutura hierárquica, *Header.php* e *Footer.php*. Tanto os elementos do *header* como do *footer* são incluídos pelo sistema nas demais classes sempre que elas forem executadas. Utilizando esta metodologia não é necessário reescrever os códigos já escritos para o menu. O arquivo *index.php* deve, obrigatoriamente conter um *header* e um *footer* (1..1) porém o *footere* o *header* não precisam da *index* e podem ser inseridos em outros *scripts* PHP (0..\*), por isso toda vez que o *index.php* for apresentado, a sua relação com o *script* será obrigatória (1..1), já que o *script* para apresentação do layout deve conter o *footer* e o *header*. A classe *IdentificationDomain.php* é composta pelos elementos do domínio *Identification* existentes no AuditModel.

A classe *Planning Domain.php* define todos os elementos do domínio *Planning* existente no AuditModel. Com essa classe é possível associar um plano a somente uma *Identification*, numa relação (1..1). A classe *Identification Domain.php* permite realizar inserções de dados e consulta no banco de dados, porém, antes da realização da inserção a classe busca a *Identification* do plano ao qual as informações serão inseridas.



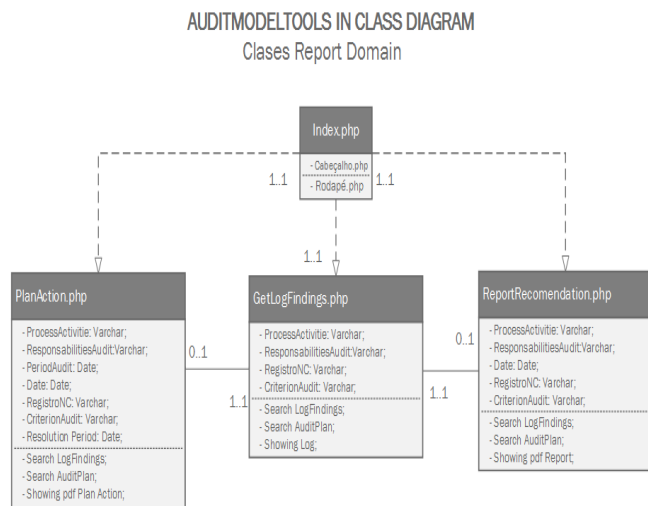
Fonte: Próprio Autor.

**Figura 4: Classes PHP que compõem o componente Business Rules (Business Layer) do pacote Conformity Domain do Audit Model.**

Na Figura 4 pode-se identificar as classes *CriterionAudit.php*, *Document Analises.php* e *Conformity.php*. Na classe *CriterionAudit.php* é realizada a pesquisa ao banco de dados a fim de obter o identificado plano de auditoria para associar o arquivo CSV que será importado e servirá como critério da auditoria a ser realizada. Após obtenção do arquivo a classe envia-o para *CriterionAuditImport.php*, onde será realizado o processo de leitura dos dados do arquivo e inserção destas informações no banco de dados relacional. A classe *CriterionAuditImport.php* deve receber uma ou várias informações do *CriterionAudit.php* (1..\*), porém, *Criterion Audit.php* deve enviar dados para uma única classe *CriterionAuditImport.php* (1..1). A obtenção das informações do processo a ser auditado são obtidas pela classe *DocumentAnalises.php* que realiza o mesmo procedimento da classe *CriterionAudit.php* e envia o documento para o arquivo *DocumentAnalisesImport.php* as relações são as mesmas definidas entre a classe *Criterion Audit.php* e a

*CriterionAuditImport.php*. Os dados a serem importados dos arquivos .csv devem obrigatoriamente seguir a ordem e tipos de dados definidos na tabela do banco relacional para que possam ser gravados com sucesso. Neste caso o tipo de dados definidos foram: *Int*, *Date* e *Float*.

A classe *Conformity.php* realiza uma consulta no banco de dados para obtenção dos critérios definidos através da extração dos dados pela classe *Document Analises Import.php* e dispõe de operadores lógicos que devem ser configurados para que sejam enviados, junto com os critérios definidos, para classe *Conformity Analyse.php*. Na classe *Conformity Analyse.php* é realizada uma busca na tabela de banco de dados que contempla as informações importadas pela classe *Document Analises Import.php*. A *Conformity Analyse.php* analisa os critérios e informações obtidas no banco de dados para posteriormente compará-los de forma atestar a conformidade. Neste teste de conformidade todas as inconsistências evidenciadas serão gravadas no banco de dados e utilizada como log das auditorias. O log de auditoria é representado pela classe *GetLog Findings.php* explicada a partir da Figura 5.



Fonte: Próprio Autor.

**Figura 5. Classes PHP que compõem o componente Business Rules (Business Layer): do pacote ReportDomain do AuditModel**

Na Figura 5 é possível identificar as classes *GetLogFindings.php*, *ReportRecommendation.php* e *PlanAction.php*. A classe *GetLogFindings.php* é composta pelos elementos do modelo *AuditModel* e mantém o registro das não conformidades, dos critérios, do processo auditado e do responsável pela auditoria. Para dispor dos logs de auditoria a classe *GetLogFindings.php* realiza consulta ao banco de dados e registra as não conformidades de auditoria em um log que será exportado em arquivo .pdf para o usuário.

A classe *PlanAction.php* e a classe *ReportRecommendation.php* não têm tabelas definidas no banco de dados, uma vez que as não conformidades utilizados por estas duas classes estão gravados no arquivo de log gerado pela classe *GetLogFindings.php*. A existência destas duas classes estão condicionadas ao log (*GetLogFindings.php*) da auditoria, uma vez que o plano de ação (*PlanAction.php*), que tem a função de controlar os prazos para resolução das não conformidades, e o relatório final (*ReportRecommendation.php*), que contém o resultado da auditoria, serão gerados a partir deste log.

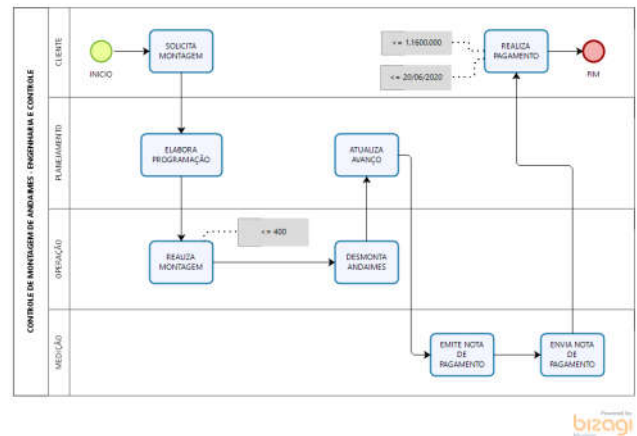
A classe *ReportRecommendation.php* obtém o log de auditoria e gera um arquivo do tipo .pdf contendo as informações definidas no *AuditModel* como processo auditado, responsável pela auditoria, data da realização, registro da não conformidade e o critério.

A classe *PlanAction.php*, apresenta as informações definidas no modelo como: não conformidades, data para fechamento das não conformidade evidenciadas, nome do auditor, gerando um arquivo .pdf de forma caracterizar um plano de ação.

Desta forma o *AuditModel* foi implementado com base na arquitetura proposta. Na Seção 4 será mostrado um exemplo de auditoria realizado a partir da solução proposta neste trabalho

**Validação da ferramenta AuditModel Tool:** Afim de realizar uma auditoria e validar a funcionalidade do *Audit Model*, foi mapeado, usando a *BPMN (Business Process Model and Notation)*, um processo de negócio de uma organização. O nome da empresa a qual este processo pertence não consta no mapeamento por questões de sigilo. Foi mapeado o processo de controle de montagem de andaimes na Figura 6. Os processos e os resultados de auditoria são apresentados a seguir.

**Auditoria - Processo de Controle de Montagem de Andaimes:** O processo corporativo apresentado foi mapeado com a finalidade de apresentar o fluxo de controle de um contrato para montagem de andaimes. Nesta suposição é definida uma verba para realização das atividades com recurso financeiro e prazo limitado. A Figura 6 apresenta o processo mapeado.



Fonte: Próprio Autor.

**Figura 6. Mapeamento do processo de controle de montagem de andaime**

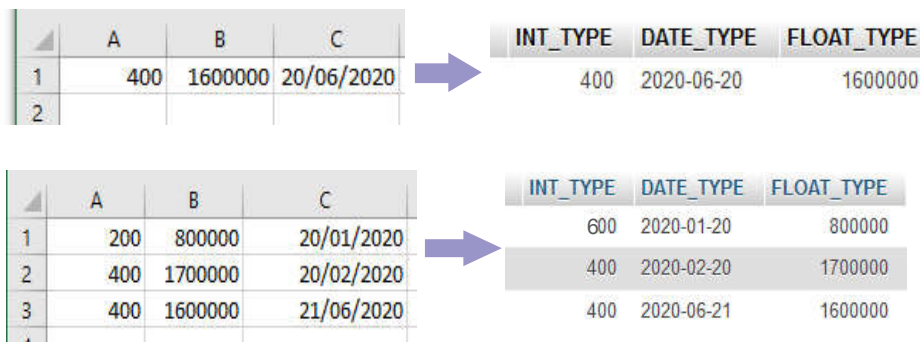
Através do diagrama é possível identificar o nome do processo (Controle de montagem de andaimes) e o setor a qual ele pertence (Engenharia e Controle). O processo macro é dividido em quatro subprocessos internos, setores ou atores que participam e interagem entre si, no exemplo são eles *Cliente*, *Planning*, *Operação* e *Medição*. O fluxo representado pelas linhas em amarelo leva às ações esperadas após cada interação da tarefa anterior. A fim de entender os requisitos existentes no contrato, assumiu-se que o custo mensal com andaimes montados não deve ultrapassar 1.600.000,00 (um milhão e seiscentos mil reais) o que significa aproximadamente 400 andaimes faturados. O contrato em questão tem vigência até 20/06/2020 e suas notas fiscais devem ser emitidas até esta



data. Os critérios do processo em questão, considerando os valores assumidos no mapeamento do processo são: Nota mensal com valor inferior ou igual a 1.600.000,00 (um milhão e seiscentos mil reais) relacionado ao pagamento de no máximo 400 andaimes montados.

Para a organização e para o cliente é fundamental a relação de confiança, bem como, o atendimento dos requisitos definidos entre as partes uma vez que as não conformidades corriqueiras podem gerar desvios e, conseqüente, rescisão contratual. Para realização de uma auditoria através da ferramenta AuditModel Tool se faz necessário o preenchimento do objetivo da auditoria (*AuditObjective*), a definição de um nome que identifique o plano de auditoria (*AuditPlan*) e seja preenchido o plano de auditoria (*AuditPlan*). Depois de preenchidos os dados do plano, deve-se importar, via arquivo .csv, os critérios e as informações oriundas da execução do processo a fim de possibilitar a análise de conformidade.

Através da análise de conformidade é possível identificar e existência de 3 não conformidades, ou seja, três registros do processo não atendem aos requisitos definidos na ferramenta. Os registros de 600 andaimes montados apresentados em 20/01/2020 não conferem com o requisito de limite de andaimes no mês (<400). O valor de 1.700.000,00 (um milhão e setecentos mil reais) apresentado no dia 20/02/2020 é superior que ao valor definido em contrato (1.600.000,00, um milhão e seiscentos mil reais) e por fim, os dados fornecidos no registro do dia 21/06/2020 foram apresentados fora da data limite que encerra o contrato e, conseqüentemente, o faturamento da nota (20/06/2020). Os logs de auditoria (*Log Findings*), bem como os respectivos relatórios e plano de ações podem ser consultados através do menu da ferramenta. A seguir é apresentado, na Figura 9, o log de auditoria realizada. Por definição, o modelo sugere informações para criação de relatórios de auditoria (*ReportRecommendation*) em formato .pdf.



Fonte: Próprio Autor.

**Figura 7. Importação dos critérios e informações do processo para o banco de dados. Componentes *Data* e *Database*, da Camada de *Persistence***

Registro no DocumentAnalises	Critério	Situação
600	Deve ser Menor ou Igual a 400	NÃO CONFORMIDADE
2020-01-20	Deve ser Menor ou Igual a 2020-06-20	Conforme
800000	Deve ser Menor ou Igual a 1600000	Conforme
Registro no DocumentAnalises	Critério	Situação
400	Deve ser Menor ou Igual a 400	Conforme
2020-02-20	Deve ser Menor ou Igual a 2020-06-20	Conforme
1700000	Deve ser Menor ou Igual a 1600000	NÃO CONFORMIDADE
Registro no DocumentAnalises	Critério	Situação
400	Deve ser Menor ou Igual a 400	Conforme
2020-06-21	Deve ser Menor ou Igual a 2020-06-20	NÃO CONFORMIDADE
1600000	Deve ser Menor ou Igual a 1600000	Conforme

Fonte: Próprio Autor.

**Figura 8. Análise de Conformidade**

Na Figura 7 foram realizadas duas inserções no banco de dados através da importação dos arquivos .CSV, sendo a primeira dos critérios (*Criterion Audit*) do processo e a segunda dos dados fornecidos pelos usuários do processo (*Document Analises*). Após a inserção dos critérios (*Criterion Audit*) e os dados do processo (*Document Analises*) foi realizada uma análise de conformidade (*Conformity*), conforme apresentado na Figura 8.

Visando realizar o controle das não conformidade e possibilitar um acompanhamento das correções dos problemas evidenciados, a ferramenta Audit Model Tool gera um plano de ação (*PlanAction*) em formato .pdf, Figura 11, com os prazos para correção das não conformidade. O relatório da auditoria em questão é disponibilizado na Figura 10.



LOG GERADO COM SUCESSO.

Processo Auditado	Auditor Responsável	Log Gerado em:
Scaffolding Assembly Control	Allison Santana	05/07/2020
Registro no DocumentAnalises	Critério	Situação
600	Deve ser Menor ou Igual a 400	NÃO CONFORMIDADE
1700000	Deve ser Menor ou Igual a 1600000	NÃO CONFORMIDADE
2020-06-21	Deve ser Menor ou Igual a 2020-06-20	NÃO CONFORMIDADE

Fonte: Próprio Autor.

Figura 9. Log da auditoria realizada

Processo Auditado	Auditor Responsável	Data
Scaffolding Assembly Control	Allison Santana	05/07/2020
Registro no DocumentAnalises	Critério	Situação
600	Deve ser Menor ou Igual a 400	NÃO CONFORMIDADE
1700000	Deve ser Menor ou Igual a 1600000	NÃO CONFORMIDADE
2020-06-21	Deve ser Menor ou Igual a 2020-06-20	NÃO CONFORMIDADE

Fonte: Próprio Autor

Figura 10. Relatório da auditoria

Processo Auditado	Auditor Responsável	Data da Auditoria	Data Atual
Scaffolding Assembly Control	Allison Santana	2020-07-05 00:00:00	05/07/2020
Registro no DocumentAnalises	Critério	Situação	Prazo para Resolução da Não Conformidade
600	Deve ser Menor ou Igual a 400	NÃO CONFORMIDADE	04/08/2020
1700000	Deve ser Menor ou Igual a 1600000	NÃO CONFORMIDADE	04/08/2020
2020-06-21	Deve ser Menor ou Igual a 2020-06-20	NÃO CONFORMIDADE	04/08/2020

Fonte: Próprio Autor

Figura 11. Plano de ação de auditoria

## Considerações Finais

Foi apresentado neste trabalho, a *AuditModelTool*, ferramenta, implementada com base no modelo para representação de processos de auditoria, *AuditModel*. A validação do modelo e da ferramenta se deu por meio da execução de um processo de auditoria realizado em um ambiente real. A ferramenta mostrou eficiência no processo de realização de auditoria e simplicidade na sua utilização. Não foram necessárias configurações de *scripts*. O modelo se mostrou eficiente para representar o processo de auditoria no processo de negócio da organização.

A utilização da notação gráfica BPMN foi de grande valia para o entendimento dos processos de auditoria e de negócio. São apresentados no fluxo mostrado na figura que representam o processo de negócio, Figura 6, os principais componentes, suas operações e relacionamentos de forma destacar os principais aspectos a serem observados na realização da auditoria. Observa-se que a aplicação da solução proposta, com o *AuditModel*, *AuditModel Tool* e a representação BPMN, foi possível se obter uma visão simplificada da realidade, permitindo eliminar detalhes irrelevantes para a realização da auditoria e concentrando em aspectos específicos. Destarte, o processo de realização ficou mais

simples, prático e automatizado. Este trabalho oferece conceitos e insumos para futuros pesquisadores desenvolverem outras ferramentas ou melhorarem a ferramenta disponibilizada no repositório público *GitHub* de endereço <https://github.com/allisonrasantana/AuditModelTool>. demais, serão disponibilizados adocumntação a especificação do modelo proposto, eo processo mapeado utilizado neste trabalho.

Embora só tenha sido apresentado neste artigo um exemplo de utilização da proposta, outros exemplos foram efetuados em processos de negócios diferentes, mas com base no processo de auditoria definido pela ISO 19011. Com isso, foi possível inferir que o *AuditModel* consegue representas diferentes processos de negócios a serem auditados e que sua implementação na ferramenta auxilia auditores especificarem seus processos de auditoria com base na ISO 19011. Portanto, a ferramenta atendeu a todas as perspectivas de fatores relacionados a aspectos técnicos e de auditoria definidos neste trabalho. Foi evidenciado que a ferramenta é compatível com quatro sistemas operacionais, possibilita importação de dados em arquivos .csv e de banco de dados, é adaptada para uso em diversos dispositivos e está disponível em um repositório *open-source*. Quando aos aspectos de auditoria a *AuditModelTool* pode ser utilizada remotamente e emite *Reports* e planos de ação conforme definido no modelo.

Os exemplos apresentados na validação da ferramenta ilustram como utilizar a ferramenta *AuditModelTool* para implementar os elementos do *AuditModel* e reforça a característica do uso do modelo independente do processo de negócio a ser auditado.

Os 14 elementos propostos podem conter informações de qualquer processo, basta mapeá-lo, obter as informações necessárias e adaptar os critérios e os documentos a serem analisados a tecnologia utilizada no desenvolvimento de *softwares*. Embora tenha sido tomado como base o padrão de auditoria da ISO 19011, que é o mais utilizado pelas organizações no mundo, percebe-se que a adaptação para um outro modelo é possível. Como trabalhos futuros pretende-se estender a ferramenta e modelo para outros processos de auditoria, estender a ferramenta para extrair dados de diferentes fontes, e.g. XML, XBRL, Json, diferentes bancos relacionais. Pretende-se estender e utilizar o trabalho de de Bragança et al. (BRAGANCA, LADISLAU, SILVA e SILVA, 2019) para incorporar a extração de dados XBRL. Um Benchmark de auditorias com a *AuditModel Tool* também é idealizado.

## REFERÊNCIAS

- ABNT. ABNT ISO 9001:2015 - Sistema de Gestão da Qualidade / Requisitos, 30 set. 2015. Disponível em: <<https://www.abntcatalogo.com.br/norma.aspx?ID=345041>>. Acesso em: 26 maio 2020.
- ABNT. NBR ISO 19011:2018 - Diretrizes para auditorias de sistema de gestão da qualidade e/ou ambiental, 20 dez. 2018. Disponível em: <<https://www.abntcatalogo.com.br/norma.aspx?ID=415425>>. Acesso em: 30 mar. 2020.
- AICPA. Audit Data Standards. AICPA ORG, 2020. Disponível em: <<https://www.aicpa.org/interestareas/frc/assuranceadvisoryservices/auditdatastandards.html>>. Acesso em: 03 jan. 2020.
- ALVES, P. M. D. A. Ferramentas Informatizadas Utilizadas na Auditoria. Revista Brasileira de Contabilidade, n. 225, Volta Redonda, 2017.
- BOOTSTRAP. The most popular HTML, CSS, and JS library in the world., 2020. Disponível em: <**Error! Hyperlink reference not valid.**>. Acesso em: 22 jun. 2020.
- BRAGANCA, H. A. ; LADISLAU, S. P. ; SILVA, M. A. P. ; SILVA, PAULO CAETANO DA . Motor XBRL-ETL: Uma ferramenta para transformação de dados baseada na taxonomia XBRL-SICONFI. In: 16° Contecsi - International Conference on Information Systems and Technology Management, 2019, São Paulo. International Conference on Information Systems and Technology Management. São Paulo: FEA, 2019, DOI: 10.5748/16CONTECSI/AIS-6233.
- COSTA, D.; SILVA, P. C. A framework for auditing xbrl documents based on the gri sustainability guidelines. : 14th International Conference on Information Technology: New Generations ITNG 2017, VIII. Las Vegas: [s.n.]. 2017.
- DAVID.Y. CHAN, CHIU V, VASARHELYI, M.A., "New Perspective: Data Analytics as a Precursor to Audit Automation". Continuous Auditing (Rutgers Studies in Accounting Analytics), Emerald Publishing Limited, pp. 315-322. <https://doi.org/10.1108/978-1-78743-413-420181016>, 2018.
- ELMASRI, R.; NAVATHE, B. Sistema de Banco de Dados. 6. ed. PEARSON BRASIL - ISBN: 8579360854, 9788579360855, 2011. p. 788.
- HUANG, S.; HUANG, S.-M. J-CAATs: A Cloud Data Analytic Platform for Auditor's. The 2017 International Conference on Computer Auditing, June 19-20, 2017, London, UK.
- IETF. Common Format and MIME Type for Comma-Separated Values (CSV) Files, 2005. Disponível em: <<https://tools.ietf.org/html/rfc4180>>. Acesso em: 05 jun. 2020.
- IMONIANA, J. O. Auditoria de sistemas de informação, Ed 3. São Paulo: Atlas, 2016.
- ISO. Certification & Conformity - The ISO Survey. International Organization for Standardization, 2018. Disponível em: <<https://www.iso.org/the-iso-survey.html>>. Acesso em: 25 maio 2020.
- ISO. ISO - International Organization for Standardization, 2020. Disponível em: <<https://www.iso.org/home.html>>. Acesso em: 10 jun. 2020.
- ISO. Portable document format - ISO 32000-2:2017, 2017. Disponível em: <<https://www.iso.org/standard/63534.html>>. Acesso em: 28 jun. 2020.
- LAUDON, C. ; LAUDON, J. P. Sistemas de Informações. Rio de Janeiro: p. 35, v. 4, 1999.
- LYNCH, The Hidden Role of Software in Educational Research: Policy to Practice (Routledge Research in Education) - ISBN 978-1138084933. Routledge, 2017. p. 230.
- LYRA, M. R. Segurança e auditoria em sistemas de Informação. Rio de Janeiro: Ed. Ciências Moderna, 2008. p.85.
- MACHADO, N. R. Banco de Dados - Projeto e Implementação - ISBN 9788536500195. Editora Érica, v. 1, 2009.
- MARIADB. Maria DB Enterprise Open Source Database, 2020. Disponível em: <<https://mariadb.com/>>. Acesso em: 13 maio 2020.

- MÍKVA, MIROSLAVA., PRAJOVÁ, VANESSA., YAKIMOVICH, BORIS., KORSHUNOV, ALEXANDER.; TYURIN, I. Standardization – one of the tools of continuous improvement. International Conference on Manufacturing Engineering and Materials ICMEM 2016. Procedia Engineering. Vol. 149, p 329-332. 2016.
- MIRANDA, R. C.. Arquitetura de referência para construção, validação e implantação de serviços de integração entre sistemas, Uberlândia, 2018. p.25.
- MORAES, Jacleciane,. MORAES, Eliezer,. LOBATO, Antônio,. MELO, Alexandre. Um estudo comparativo de portais colaborativos e uma proposta do portal Websocial. Revista de Tecnologia da Informação e Comunicação da Faculdade Estácio do Pará, Belém, v. 2, p. 58 - 77, Agosto 2019.
- OMG. Business Process Model And Notation, 2020. Disponível em: <<https://www.omg.org/spec/BPMN/2.0/About-BPMN/>>. Acesso em: 27 jun. 2020.
- OMG. Unified Modeling Language - Specification, 2020. Disponível em: <<https://www.omg.org/spec/UML/About-UML/>>. Acesso em: 27 jun. 2020.
- PHP GROUP. O que é o PHP?, 2020. Disponível em: <[https://www.php.net/manual/pt\\_BR/intro-what-is.php](https://www.php.net/manual/pt_BR/intro-what-is.php)>. Acesso em: 26 jun. 2020.
- SANTANA, A. R. A.; SILVA, P. C. Padronização de Processos de Auditoria Contínua, Campinas. Revista de Auditoria, Governança e Contabilidade, v. 8, n. 35, 2020. ISSN 2317-0484.
- SHING-HAN LI, D. C. Y. Y.-P. C. A Real-Time Audit Mechanism Based on the Compression Technique. ACM Transactions on Management Information System, v. Vol. 7, n. 2, p. 7, mar. 2016.
- SUNAGA, et al. Relation Between Combinations of Personal Characteristic Types and Educational Effectiveness for a Controlled Project-Based Learning Course. IEEE Transactions on Emerging Topics in Computing, mar. 2017. 69-79.
- TERUEL, E. C. Principais ferramentas utilizadas na auditoria de sistemas e suas características, São Paulo, 2017. p. 96.
- VASARHELYI, M.; ALLES, ; WILLIAMS, K. Continuous assurance for the now. Australia: Institute of Chartered Accountants in Australia, Sydney, 2010.
- W3C. HTML 5 Recommendation, 2017. Disponível em: <<https://www.w3.org/TR/html52/>>. Acesso em: 20 jun. 2020.
- W3C. HTTP - Hypertext Transfer Protocol, 2014. Disponível em: <<https://www.w3.org/Protocols/>>. Acesso em: 29 jun. 2020.
- W3C. JavaScript Tutorial, 2020. Disponível em: <<https://www.w3schools.com/js/default.asp>>. Acesso em: 20 jun. 2020.
- WERNER,. Process Model Representation Layers for Financial Audits. 49th Hawaii International Conference on System Sciences. New Zealand: [s.n.]. 2016.

\*\*\*\*\*