

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE MINAS,
METALÚRGICA E DE MATERIAIS (PPGE3M)

APLICATIVO WEB RESPONSIVO PARA GESTÃO DE AUDITORIAS
INTERNAS/EXTERNAS E NÃO CONFORMIDADES

Roberval Silva Bett

Porto Alegre

2019

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE MINAS,
METALÚRGICA E DE MATERIAIS (PPGE3M)

APLICATIVO WEB RESPONSIVO PARA GESTÃO DE AUDITORIAS
INTERNAS/EXTERNAS E NÃO CONFORMIDADES

Roberval Silva Bett

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais da Universidade Federal do Rio Grande do Sul, como requisito parcial à obtenção do título de Mestre em Engenharia modalidade Acadêmica.

Área de concentração: Tecnologia Mineral, Ambiental e Metalurgia Extrativa

Porto Alegre

2019

Esta dissertação foi julgada adequada para obtenção do título de mestre em ENGENHARIA, área de concentração: Tecnologia Mineral, Ambiental e Metalurgia Extrativa e aprovada em sua forma final pela orientadora e pela banca examinadora.

Profa. Dra. Rejane Maria Candiota Tubino
Orientadora
Escola de Engenharia
Universidade Federal do Rio Grande do Sul

Dr. Afonso Reguly
Coordenador PPGE3M

Banca Examinadora

Prof. Dr. Afonso Reguly- PPGE3M/UFRGS
Dr. Pelo PPGE3M-UFRGS

Profa. Dra. Thaiana Pereira dos Anjos- IFSC/Campus Lages
Doutora pela UFSC

Prof. Dr. Wilson Castello Branco Neto- IFSC/Campus Lages
Doutor pela UFSC

Dedico este trabalho a minha Família e a Deus.

AGRADECIMENTOS

Aos meus pais, irmão e especialmente a minha esposa pelo incentivo e pressão;

A empresa Edutec Satc pelo incentivo;

A minha orientadora, professora Rejane Tubino, pela amizade, contribuição, paciência e pela inestimável contribuição técnica e pessoal;

Aos colegas que de alguma forma contribuíram para a realização deste estudo;

Aos colaboradores da controladoria SATC e do LAMEF UFRGS;

A colaboradora do LAMEF UFRGS Etiene Benini Mendes;

Ao amigo Lucas Boeing Scarduelli e aos alunos do curso de Informática Pronatec – SATC, pelo esforço colaborativo no desenvolvimento do protótipo.

SUMÁRIO

1.	INTRODUÇÃO.....	13
1.1.	OBJETIVOS	14
1.1.1	OBJETIVO GERAL.....	14
1.1.2	OBJETIVOS ESPECÍFICOS	14
1.2.	Delimitação do trabalho	14
1.3.	Estrutura do trabalho.....	15
2.	REFERENCIAL TEÓRICO	16
2.1.	Gestão da Qualidade	16
2.2.	Certificação	17
2.3.	NBR ISO 19011:2018	19
2.4.	Auditorias	20
2.5.	Processos de Software	21
2.6.	Desenvolvimento Web	23
2.7.	Responsive Web Design (RWD)	24
2.8.	Mobile First.....	25
2.9.	Single Page Application (SPA).....	26
2.10.	Banco de Dados.....	27
2.11.	Requisitos de Software	28
3.	PROCEDIMENTOS METODOLÓGICOS	29
3.1	Descrição da empresa, do setor/processo foco do trabalho.....	29
3.2	Método de Pesquisa.....	30
4	APLICAÇÃO PRÁTICA DO TRABALHO	32
4.1.	Documento de requisitos	32
4.2.	Diagrama de Classes.....	35
4.3.	Node.Js	36
4.4.	MongoDB	37

4.5.	Json	37
4.6.	Webservices.....	38
4.7.	AngularJS.....	40
4.8.	Servidor Web	41
5.	APLICAÇÃO E DISCUSSÃO DOS RESULTADOS.....	43
6.	CONSIDERAÇÕES FINAIS.....	47
6.1.	Conclusões	47
6.2.	Sugestões de trabalhos futuros.....	48
	Referências.....	49
	Anexos.....	53

LISTA DE FIGURAS

Figura 1: Modelo Iterativo Incremental.....	22
Figura 2 - Web Design Responsivo	24
Figura3 - Mobile first	26
Figura 4 - Single Page Application.....	26
Figura 5- Diagrama de Classes DbAudit.....	36
Figura 6 - Arquitetura Back-end x Front-End	39
Figura 7 - Webservice Rest.....	40
Figura 8 - Diagrama Aplicação AngularJS – MVC	41
Figura 9 - Tela de Clientes.....	44
Figura 10 - Tela de Auditores	45
Figura 11 - Tela de Auditoria	45
Figura 12 - Tela de CheckList	46

LISTA DE TABELAS

Tabela 1 - Nº de Organizações Certificadas pela ISO.....	19
--	----

LISTA DE ABREVIATURAS E SÍMBOLOS

ABNT – Associação Brasileira de Normas Técnicas

CHECKPOINTS – Pontos de checagem

DbAudit – Sigla que dá nome ao protótipo

DER – Diagrama entidade e relacionamentos

DFD – Diagrama de fluxo de dados

FRAMEWORK - Códigos comuns entre vários projetos de software

HTML - Linguagem de Marcação de Hipertexto

HTTPS - Protocolo de transferência de hipertexto seguro

INMETRO – Instituto Nacional de Metrologia, Qualidade e Tecnologia

ISO – International Organization for Standardization

JSON – Formato para transferência de dados entre camadas da aplicação

LAMEF – Laboratório de Metalúrgica Física

LAMETRO – Laboratório de Metrologia

NBR – Norma Brasileira

PMBOK – Project management body of knowledge

PMI – Project management Institute

REST – Estilo de arquitetura de software

RWD - Responsive web design

SATC – Escola Técnica da Sociedade de Assistência aos Trabalhadores do Carvão

SGBD – Sistema gerenciador de banco de dados

SOAP – Protocolo para troca de informações

SPA – Single page application

STAKEHOLDERS – Partes interessadas

UFRGS – Universidade Federal do Rio Grande do Sul

UML – Linguagem de modelagem unificada

WEBSERVICES - Componentes que permitem às aplicações enviar e receber dados

XML –Extensible markup language.

RESUMO

Com o aumento da busca por produtos e serviços com melhor padrão de qualidade, os métodos de apuração de falhas e desvios nos processos de fabricação são cada vez mais exigidos e aplicados. Em uma era em que a informação por si só não é mais tão significativa, é necessário agregar valor a esta informação gerando conhecimento e auxiliando os processos de tomada de decisão. A tecnologia necessita ser inserida neste processo como ferramenta de auxílio, agilização da documentação e correção dos desvios visando à padronização e melhoria da qualidade, objetivando a redução de custos e da utilização dos recursos naturais. As auditorias surgem neste contexto como ferramenta de gestão visando melhoria de processos, adequações a legislações vigentes e/ou aumento da competitividade das organizações. Um protótipo de aplicativo web responsivo para realização de auditorias internas/externas e controle da sua documentação foi desenvolvido. Analisou-se as vantagens e desvantagens da aplicação da ferramenta na realização e emissão do relatório final de auditoria. O protótipo se mostrou eficiente no que se refere à redução do tempo de execução da auditoria e aumento da rastreabilidade do processo.

Palavras Chaves: Processo de auditoria, não conformidades, padrão de qualidade.

ABSTRACT

Due to the increasing request for products and services with better quality standards, the calculating failures and deviations methods in manufacturing processes are increasingly required and applied. In a time when information by itself is no longer important, it is needed to give value to this information by generating knowledge and helping decision-making processes. Technology needs to be inserted in this process as a helping tool, speeding up of the documentation and deviation correcting, aiming the standardization and the quality improvement, searching for costs reduction and the use of natural resources. In this context the audits emerge as management tools, aiming to improve processes, adapt to current laws and/or improve the organizations competitiveness. A responsive web application prototype was designed to support the conduct and documenting internal/external audits. The evaluation of advantages and disadvantages of the tool usage on the audits and on the final report generation was performed. The prototype was efficient as audit reduction time and to improve rastreability of process.

Key words: Audit process, non-compliances, quality standards.

1. INTRODUÇÃO

Segundo a definição utilizada pelo *Project Management Institute* – PMI “qualidade é o grau até o qual um conjunto de características satisfaz as necessidades” (PMBOOK, 2017).

Segundo Fernandes (2015), com o aumento das dimensões da economia mundial, passou-se a exigir cada vez mais que os produtos e/ou serviços atendam a padrões normatizados de qualidade, sustentabilidade social e proteção à integridade física e à saúde de seus funcionários. Neste contexto a implantação de sistemas de gestão da qualidade tem se tornado um diferencial para a sobrevivência e competitividade das empresas, o que torna estas ferramentas importantes aliadas no sentido de melhorar a eficiência da gestão de processos e gerando uma realidade de desenvolvimento sustentável.

Entenda-se desenvolvimento sustentável como sendo a necessidade de se harmonizar o desenvolvimento econômico com a proteção ambiental no seu sentido mais amplo, ou seja, caracteriza-se por ser um modelo de desenvolvimento que prioriza os aspectos econômicos, ambientais e sociais. Segundo Tronco (2005).

“[...] um sistema de qualidade define-se como a estrutura operacional de trabalho de toda a empresa documentada em procedimentos técnicos integrados, em que cada processo é definido para o direcionamento das ações a serem desenvolvidas para obter melhores práticas e meios de assegurar qualidade e menores custos para a produção”(TRONCO, 2005).

Para assegurar-se que um produto e/ou serviço possa apresentar um padrão de qualidade e um melhoramento contínuo é preciso auditar os processos visando comprovar a execução dos procedimentos ou o levantamento, acompanhamento e resolução de não conformidades nos processos.

As linhas gerais da pesquisa estão pautadas nas orientações da norma NBR ISO 19011(ABNT, 2011), que orientam para a definição dos objetivos, escopo e critérios individuais de cada auditoria. Vale salientar que a norma não indica os requisitos da auditoria e sim fornece orientações sobre o planejamento e execução de

um processo de auditoria, bem como sobre as competências dos auditores e líderes de equipes de auditoria.

A aplicação de recursos tecnológicos, tanto nos produtos quanto nos processos, deixou de ser um problema estritamente tecnológico para agir como agente importante na capacidade de gestão dos recursos e processos que compõe o sistema produtivo.

A partir destes pressupostos foram levantadas informações sobre os métodos de aplicação das rotinas de auditorias, visando uma melhoria nos processos com o objetivo de criar uma aplicação que sirva como ferramenta especialista de auxílio à equipe de auditoria, seja ela interna ou externa.

1.1. OBJETIVOS

1.1.1 OBJETIVO GERAL

O projeto focaliza o desenvolvimento de um protótipo de aplicação que permita realizar e documentar uma auditoria (interna/externa) e fazer o acompanhamento das não conformidades levantadas durante estes procedimentos.

1.1.2 OBJETIVOS ESPECÍFICOS

- Acompanhar as não-conformidades e ocorrências levantadas, reduzindo substancialmente as responsabilidades de controle dos tempos e status destas incidências por parte dos auditores;
- Emitir o relatório final das auditorias realizadas no aplicativo;
- Documentar uma auditoria no aplicativo;
- Analisar as vantagens e desvantagens da utilização do aplicativo na realização de auditorias.

1.2. Delimitação do trabalho

Durante o processo de análise de campo junto aos auditores da Escola Técnica SATC (Sociedade de Assistência aos Trabalhadores do Carvão), e de algumas

certificadoras que prestavam consultoria nesta instituição de ensino, apurou-se através de observações das atividades e entrevistas, que o protótipo de aplicação deveria ser desenvolvido levando-se em consideração a portabilidade de hardware e a alta disponibilidade dos recursos. A partir destas características o protótipo foi desenvolvido como uma aplicação *Web*.

O protótipo segue as especificações para implementação de um processo de auditoria, seja ela interna ou externa, permitindo-se criar *checklists*, controlar as não conformidades/ocorrências, aumentando a rastreabilidade e documentando toda a auditoria, sendo toda a atividade embasada na norma NBR ISO 19011 (ABNT, 2018).

Visando comprovar a aplicabilidade do protótipo foram lançadas auditorias no aplicativo, sendo que a auditoria fornecida pelo Laboratório de Metalurgia Física (LAMEF) UFRGS serviu como ponto final da bateria de testes e avaliação do aplicativo. O acesso a informações de auditorias reais para avaliação do aplicativo se mostrou como o maior desafio em virtude da confidencialidade das informações.

1.3. Estrutura do trabalho

A presente pesquisa foi dividida em seis capítulos. O primeiro traz a contextualização e apresentação do tema, o problema, os objetivos e a delimitação.

O segundo capítulo apresenta o embasamento teórico, tratando sobre controle de qualidade, certificações e não conformidades.

O terceiro capítulo apresenta a caracterização da empresa/área de atuação, e a metodologia empregada na pesquisa e desenvolvimento do projeto.

O quarto capítulo trata da aplicação prática dos métodos no processo de desenvolvimento do protótipo. Suas etapas, características e particularidades tecnológicas.

O quinto capítulo aborda a aplicação do protótipo e a discussão dos resultados.

O sexto capítulo apresenta as considerações finais e as sugestões propostas para trabalhos futuros.

O capítulo Anexos apresenta a auditoria realizada com dados do LAMEF.

2. REFERENCIAL TEÓRICO

Este capítulo contém uma breve revisão dos conceitos que são relevantes para a boa compreensão do desenvolvimento deste trabalho.

2.1. *Gestão da Qualidade*

Abrantes (2009), cita em seu livro, conceitos relevantes ligados ao termo qualidade como: “Qualidade é a conformidade às especificações” e “Prevenir não conformidades é mais barato que corrigir ou refazer o trabalho”.

Segundo Abrantes (2009), uma organização pautada na gestão pela qualidade total preconiza que todos são responsáveis pela qualidade, desde os níveis mais baixos (nível operacional) até os níveis hierarquicamente mais altos (nível estratégico). Para garantir o controle de qualidade de produtos, serviços e/ou processos os funcionários devem conhecer os procedimentos e devem ser treinados periodicamente para agir em casos de não-conformidades e/ou alterações de qualidade.

Buscar uma melhoria na integração entre os setores da organização é apenas uma das características importantes para que se possa melhor dimensionar o uso de recursos existentes, minimizando custos e adequando o processo produtivo de tal forma que as necessidades do cliente e a minimização dos impactos ambientais sejam alcançados.

Segundo Fernandes (2015), o tema da qualidade recebeu mais atenção das empresas nos últimos anos devido ao crescimento do mercado e aumento da concorrência, objetivando garantir qualidade e eficiência de seus produtos, aumentando desta forma o seu potencial de competitividade.

Desenvolver e documentar procedimentos não são consideradas tarefas fáceis, porém são de suma importância para que seja possível localizar problemas, analisar os processos, padronizar e estabelecer controles, de tal forma que os mesmos problemas não voltem a ocorrer e que se minimize a probabilidade de novas não conformidades aparecerem.

Avaliar a conformidade de produtos, serviços e/ou processos, de acordo com diretrizes e normas pré-estabelecidas, é uma oportunidade de atestar a qualidade e aumentar a competitividade de mercado.

Na prática o controle de qualidade tem como objetivos principais planejar, manter e se possível melhorar a qualidade dos produtos e/ou serviços.

O termo qualidade não está mais associado ao simples mérito de conformidade com as especificações estabelecidas. A evolução deste conceito posicionou a satisfação do cliente como foco principal do tema e esta característica trouxe um conjunto de outros requisitos e objetivos atendendo a amplitude de *stakeholders* da organização e a busca da excelência organizacional da empresa.

Segundo esta ótica deve-se atentar para a implantação de sistemas de gestão da qualidade e conseqüentemente procurem implantar em seus produtos, serviços e processos normas de certificação que venham a agir como ferramentas para o destaque da organização não somente no cenário nacional como internacional.

2.2. Certificação

As certificações de produtos e/ou processos podem ser compulsórias ou voluntárias. Essas classificações surgem no momento em que um produto ou serviço possui um grau de periculosidade maior na sua construção e o Estado considera que uma certificação se faz necessária para garantir a legalidade, sendo esta a certificação compulsória. Pode ocorrer de forma voluntária pela organização, quando objetiva-se garantir conformidade com os requisitos das normas vigentes (TICONA, 2003).

Segundo Abrantes (2009), “as certificações de conformidade podem ser: de sistemas de gestão, de produtos, de serviços e de pessoas”. As certificações, independentemente do tipo de aplicação, possuem a sua fundamentação a partir de auditorias certificadoras (externas), realizadas por organizações independentes acreditadas para executar essa modalidade de avaliação de conformidade, e internas desenvolvidas por profissionais da própria organização objetivando avaliar e melhorar a eficácia dos processos.

Relatos históricos comprovam que civilizações antigas, por volta de 3500AC, buscavam simplificar produtos e unidades de medida. A própria linguagem escrita e falada caracteriza-se como sendo o princípio da padronização na sociedade humana (GARBI, 2009). Como exemplo pode-se citar a sociedade moderna, na França por volta de 1790, quando se padronizou o metro como unidade de medida de comprimento.

Por volta de 1940 é criada no Brasil a ABNT (Associação Brasileira de Normas técnicas) e em 1947 a ISO (*International Organization for Standardization*), que se tornaria no futuro a maior organização mundial de desenvolvimento de normas e padrões. A ISO é uma organização não-governamental, que reúne organismos Nacionais de normalização de aproximadamente 163 Países e desde a sua fundação já publicou mais de 19.000 normas Internacionais como cita Abrantes (2009).

“A sigla ISO significa *International Organization for Standardization*, ou seja, Organização Internacional para Normalização, cuja sede fica em Genebra na Suíça. O objetivo básico da ISO é desenvolver e promover normas e padrões mundiais, que possam facilitar o comércio internacional. [...] ISO é importante, pois permite as empresas competir no mercado mundial e os países compradores querem garantia de qualidade. No Brasil, segundo a Associação Brasileira de Normas Técnicas – ABNT, a série ISO recebe a denominação NBR ISO”(Abrantes, 2009).

Dentre as mais diversas normas ISO existentes, pode-se citar a norma ISO 19011, que pauta sobre a auditoria dos sistemas de gestão e embasa esta pesquisa. Esta norma surgiu como alternativa de unificação das diversas normas que tratavam do tema “realização de auditorias” nas mais diversas áreas de atuação.

Pode-se então concluir que uma organização certificada pela ISO, foi aprovada em um processo independente de auditoria, realizado por uma certificadora, quanto ao grau de qualidade em determinada área, produto e/ou serviço da auditada.

A implantação de rotinas de governança dentro das organizações, a necessidade de comprovar o grau de segurança de investimentos, além das exigências da legislação Brasileira e mundial já fazem parte da rotina de um número considerável de organizações e impulsionam as empresas para a busca de certificações Internacionais.

Segundo pesquisa realizada e publicada pela ISO anualmente (ISO, 2018), pode-se analisar os números comparativos de certificações anuais, e a Tabela 1 apresenta os números comparativos dos anos de 2016 e 2017. Nota-se uma redução no quantitativo de certificações de um ano para o outro e cabe discutir se esta redução tem uma correlação com a crise financeira estabelecida mundialmente e/ou é gerada por um conjunto de outros fatores a serem levantados e analisados.

Tabela 1 - Nº de Organizações Certificadas pela ISO.

Norma	Nº de Certificações em 2016	Nº de Certificações em 2017	Alterações	Alterações %
ISO 9001	1.105.937	1.058.504	-47.433	-4%
ISO 14001	346.147	362.610	16.463	5%
ISO 50001	20.216	22.870	2654	13%
ISO 27001	33.290	39.501	6211	19%
ISO 22000	32.139	32.722	583	2%
ISO 13485	29.585	31.520	1935	7%
ISO 22301	3.853	4.281	428	11%
ISO 20000-1	4.537	5.005	468	10%
ISO 28000	356	494	138	39%
ISO 39001	478	620	142	30%
Total	1.576.538	1.558.127	-18.411	-1

Fonte: ISO Survey (2018).

No Brasil, a ABNT é a organização responsável por representar o país perante a ISO, já o INMETRO (Instituto Nacional de Metrologia, Qualidade e Tecnologia) é o organismo de certificação para o Brasil e ligado à ABNT.

A certificação, é a conquista de um certificado de aprovação para um sistema de gestão baseado em normas internacionais. Para manter esta certificação as organizações investem em inovações e aperfeiçoamentos de seus produtos, serviços e/ou processos, paralelamente com a qualificação de seus colaboradores.

2.3. NBR ISO 19011:2018

O processo de auditoria em si é o foco desta pesquisa, embasada nas resoluções da norma NBR ISO 19011:2018 que discorre sobre a realização dos processos de auditoria ABNT (2018).

A NBR ISO 19011:2018 é a terceira edição da norma que foi editada primeiramente em 2002. Esta terceira edição traz pontos importantes que foram adicionados e devem ser observados e implantados durante o processo de auditoria como alguns dos citados pelo comitê de estudo de tecnologia de suporte do comitê Brasileiro de qualidade:

- ✓ Confidencialidade;
- ✓ Gestão de riscos;

É importante salientar que esta norma não tem por objetivo estabelecer os requisitos que uma auditoria deve conter, mas fornece diretrizes sobre a gestão de um programa de auditorias, seja ela interna ou externa.

2.4. Auditorias

Para garantir o padrão de qualidade nos processos, produtos e/ou serviços se faz necessário executar medições e validações a partir dos padrões estabelecidos e documentados. De acordo com Falconi (2004):

“a implantação do controle de qualidade em uma empresa precisa ser monitorada não só para verificar seus pontos fortes e fracos, mas também para orientar as pessoas e demonstrar o interesse contínuo da empresa pela qualidade” Falconi (2004).

Neste sentido a auditoria surge como ferramenta de gestão para apurar se a política de qualidade e o sistema de gestão da qualidade estão compreendidos, se todos os processos seguem os padrões estabelecidos, se todos os colaboradores estão devidamente adequados e treinados, se os produtos e/ou serviços estão de acordo com a legislação vigente e uma série de outros fatores estão sendo observados na execução dos procedimentos.

Durante a auditoria para certificação de uma organização é possível certificar a organização como um todo ou somente alguns setores, processos e/ou serviços unitariamente.

A auditoria deve ser conduzida por auditores que tenham experiência em controle de qualidade e preferencialmente que conheçam a área de atuação auditada para que a ela sirva como uma ferramenta de gestão e para que as pessoas auditadas possam enxergá-la como uma oportunidade e não como uma punição como cita Kaoru Ishikawa (apud Falconi, 2004).

“o auditor pode assumir uma pilha de fórmulas e checklist, mas sem o conhecimento baseado em experiência ele não

conduzirá bem a sua função... Auditorias devem ser usadas para promover a qualidade, e não para inspecionar”.

É através das observações, sugestões e recomendações dos auditores, que a organização vai identificar suas falhas e desenvolver as melhorias contínuas e adaptações nos processos.

O auditor líder tem como responsabilidades durante o planejamento da auditoria levantar as áreas, processos e/ou serviços que serão auditados, criar uma lista de pontos de checagem e a montagem da equipe de auditores.

A NBR ISO 19011 (ABNT, 2018) orienta sobre a formação e conduta da equipe de auditoria e de seu líder, estabelecendo que o auditor líder não tem a obrigação de ser um profundo conhecedor da área a ser auditada e neste caso a equipe deve contar com a figura de um profissional que conheça com propriedade a área (auditor especialista) e que o auditor líder tem a responsabilidade de programar a auditoria e orientar/coordenar a equipe.

A norma NBR ISO 19011(ABNT, 2018) teve sua primeira versão lançada em 2002 e passou por uma profunda mudança quando do lançamento de sua nova versão em 2011 em virtude do surgimento neste período de uma gama considerável de novas normas e da inclusão de referências importantes para a execução de auditorias internas bastante abrangentes.

No ano de 2018 a norma foi atualizada para NBR ISO 19011(ABNT, 2018), versão esta que aprofundou vários requisitos como por exemplo as questões de confidencialidade e gestão de riscos.

Durante o desenvolvimento da pesquisa um protótipo de software foi criado e este passou por um processo tecnicamente conhecido como “processo de software”, aonde se observou os requisitos da NBR ISO 19011:2018.

2.5. Processos de Software

Segundo Sommerville (2011), “um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software...”, e traduz de maneira muito resumida toda a complexidade que estas atividades remetem. Quando se fala em modelos de processo de Software pode-se citar a modelo cascata,

que possuía uma abordagem sequencial, até os modelos mais atuais como os interativos incrementais, como demonstrado na Figura 1.



Figura 1: Modelo Iterativo Incremental. Fonte: Pontes, Aleixo e Minora (2006).

O desenvolvimento de aplicações passa por inúmeras evoluções e transformações que precisam ser revistas periodicamente, dando origem aos modelos de processos de software. Mesmo desenvolvendo os processos de software a partir de modelos mais modernos ou mais tradicionais, todas as técnicas estão embasadas sob 4 alicerces como cita Sommerville (2011).

- Especificação de software:

Esta é a primeira etapa do processo de software e equivale ao momento em que os analistas definem em conjunto com o cliente quais funcionalidades o software deve atender e quais restrições devem ser implementadas no mesmo. É como um contrato de escopo definidos entre a equipe de desenvolvimento e o cliente.

- Projeto e implementação de software:

É nesta etapa que a equipe de desenvolvedores (Analistas de sistemas e negócios, programadores, designers) devem construir as especificações a partir das necessidades e restrições levantadas na etapa anterior.

- Avaliação de software:

A etapa de avaliação é responsável por uma das fases mais importantes do processo de desenvolvimento, pois é nesse momento que o aplicativo passa por análises das mais diversas, como:

- Suporte a carga de dados extrema;
- Suporte a volatilidades do ambiente externo à aplicação;
- Ambientes diversos;
- Avaliação de requisitos e restrições;

- Evolução de software:

Já há algum tempo os estudiosos da área de engenharia de software no mundo corporativo da TI, vem discutindo alterações na etapa de construção dos requisitos devido as características que o mercado vem apresentando e a velocidade que as necessidades dos clientes mudam na sociedade atual. A organização dos processos deve estar preparada para lidar então com as possíveis mudanças de requisitos e/ou projeto do software. A etapa de evolução do software é responsável por analisar e aplicar possíveis modificações e/ou evoluções nos requisitos dos clientes e/ou mudanças nas tecnologias aplicadas na construção da aplicação, principalmente no que tange ao modelo arquitetural que será implementado.

2.6. Desenvolvimento Web

Viu-se nos últimos anos o desenvolvimento de aplicativos baseados na arquitetura da internet ditarem tendência na área de desenvolvimento e esta característica está diretamente ligada à necessidade de executar aplicações da mesma forma, independente da arquitetura de *hardware* e/ou *software* (portabilidade).

Este modelo de desenvolvimento prevê a divisão da aplicação em duas partes completamente distintas, a aplicação que roda no servidor e que é chamada de *back-end* e a que roda diretamente no dispositivo do cliente chamado de *front-end*.

Embora o *front-end* e o *back-end* sejam certamente distintos um do outro, eles também são como dois lados da mesma moeda. A funcionalidade de um

site/aplicação depende de cada lado se comunicar e operar efetivamente com o outro, como uma única unidade (FERGUSON, 2018).

2.7. Responsive Web Design (RWD)

Em uma sociedade em que a velocidade e a disponibilidade da informação são fatores determinantes, os aplicativos têm papel fundamental nesta tangente como ferramentas de auxílio à tomada de decisão, e as aplicações *Web* tem ganho muito espaço neste mercado justamente por vir de encontro a estas necessidades. A alta disponibilidade e portabilidade de *hardware* compõem as vertentes mais fortes desta realidade altamente tecnológica.

Com o aumento do uso de dispositivos móveis aparece a necessidade de adaptar os *websites* aos diversos tamanhos existentes de telas, como consta na Figura 2. São duas as vertentes mais utilizadas atualmente: *Adaptive Web Design* e *Responsive Web Design*.

“[...] dispositivos possuem dimensões, resolução de tela, capacidade de armazenamento e processamento diferentes. Uns mais e outros menos, dependendo da marca e modelo. Essas características diferentes dos dispositivos tornaram-se um verdadeiro desafio para web designers construírem aplicações web que se adaptem a todos eles de maneira a oferecer um nível de usabilidade admissível aos usuários que estão cada vez mais exigentes”(KOCH e GASPARINI, 2015).



Figura 2 - Web Design Responsivo – Fonte: Altermann (2012).

Os *Web-Designers* têm procurado desenvolver *layouts* de aplicações que se ajustem automaticamente ao tamanho e resolução dos dispositivos (ALTERMANN, 2012).

Este termo foi utilizado pela primeira vez por Ethan Marcotte em seu livro *Responsive Web Design* (Marcotte, 2014). Ethan introduz uma série de técnicas que, através de sua combinação, resultavam em um desenvolvimento responsivo, ou seja, adaptável a qualquer tela de qualquer dispositivo, seja ele móvel ou *desktop*.

2.8. *Mobile First*

O *mobile first* é uma metodologia de desenvolvimento *Web* em que se planeja primeiro o aplicativo para ser carregado em dispositivos móveis, e somente depois deve-se projetá-lo para dispositivos com dimensões de *displays* maiores (Figura 2).

Os dispositivos móveis possuem características limitantes como displays menores, menor capacidade de processamento, *touch* com sensibilidade inferior aos mouses etc. Tem como foco a construção de uma base sólida de código portátil (executado independente de arquitetura de *hardware* e/ou *software*), simples e funcional.

Contudo, o *mobile first* (Figura 3), não foca somente nas restrições que estes dispositivos possuem, mas também nas suas características únicas, como estar inseridos em outros contextos que não os de casa ou escritório, como por exemplo, tirar proveito da localização do usuário (SAUER, 2017).

Esta característica é importante na construção do protótipo visto que o auditor líder consegue visualizar em tempo real como está se desenvolvendo a auditoria, quais os *checkpoints* já foram executados, quais estão pendentes e quais as ocorrências e não conformidades que por ventura estejam sendo apuradas pelos auditores diretamente em campo foram documentadas.

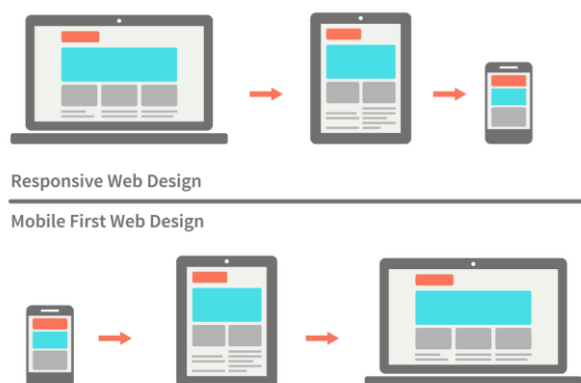


Figura 3 - Mobile first – Fonte: www.organicadigital.com

2.9. Single Page Application (SPA)

O desenvolvimento de aplicações *Web* vem sofrendo muitas mudanças e atualizações, como por exemplo os desafios de poupar tráfego de dados e melhorar performance e apresentação nos *displays*. Páginas desenvolvidas com *SPA* tendem a ser carregadas mais rapidamente devido ao isolamento de camadas e uma fluidez muito semelhante à de uma aplicação *desktop* (aplicação que tem a sua execução independente de *Internet/Intranet*). Nesta metodologia de desenvolvimento a renderização é feita diretamente no *front-end* exigindo do *back-end* somente o pacote de dados a ser apresentado e atualizando desta forma somente os *containers* necessários para a apresentação.

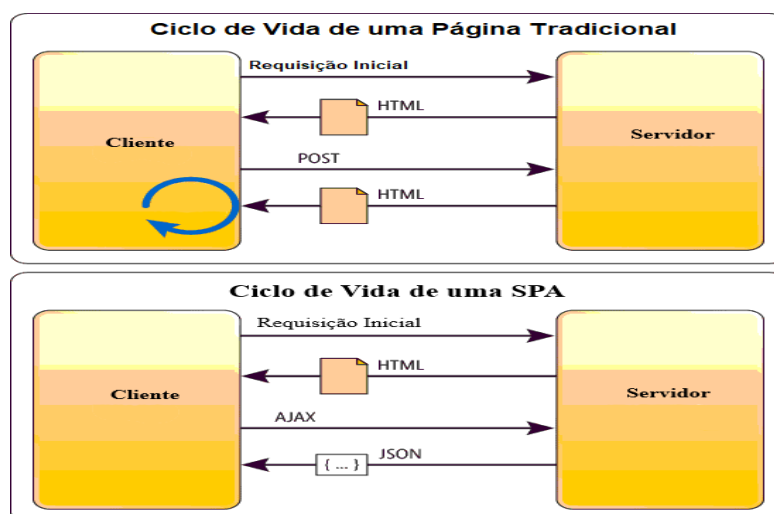


Figura 4 - Single Page Application– Fonte: www.moz.com

Analisando-se a Figura 4 fica evidente que no formato de apresentação da SPA a requisição é passada ao servidor e este retorna uma estrutura de dados, como por exemplo no formato *Json* ou até mesmo *XML*.

Quando uma aplicação faz uma requisição a um servidor tem-se a interação *front-end* x *back-end* acontecendo e os dados que por ventura este *back-end* retorna para ser renderizado no dispositivo do cliente que está rodando a parte *front-end* da aplicação podem ser empacotados em objetos de duas formas, *XML* ou *Json* (RIDLEY, 2017).

A partir destes objetos o container específico dos mesmos é renderizado, evitando-se neste caso um aumento de fluxo de dados de formatação na rede e minimizando os impactos no servidor devido ao repasse de funções de formatação para o *front-end*.

Esta característica então evita que dados de formatação da apresentação dos dados fiquem trafegando na rede e somente a informação faça este trajeto, sendo formatada somente no momento da apresentação, o que além de ser mais simples no foco do desenvolvedor, minimiza os impactos sobre o desempenho da aplicação em virtude do tipo de conexão que o cliente possui.

2.10. Banco de Dados

No processo de software é etapa comum e relevante a escolha das ferramentas que serão utilizadas para escrever a aplicação bem como o banco de dados que será responsável pelo armazenamento e gerenciamento das informações para futuras interações via aplicação.

Tradicionalmente os sistemas gerenciadores de bancos de dados, que são um conjunto de dados e de ferramentas para gerenciar e manipular esses dados, tem seguido um modelo denominado relacional.

Na construção deste protótipo foram levadas em consideração vertentes mais atuais de sistemas gerenciadores de bancos de dados e optou-se por aplicar um modelo denominado *NoSql*. Este modelo permite uma interação simples no formato *key-value* e para grandes massas de dados tem se mostrado uma solução interessante do ponto de vista do desempenho, de acordo com Li e Sathiamoorthy (2015).

“Sistemas de banco de dados tradicionais foram baseados no modelo relacional. Estes são amplamente conhecidos como bancos de dados SQL. Nos últimos anos, no entanto, os bancos de dados não relacionais aumentaram drasticamente em popularidade. Esses bancos de dados são comumente conhecidos como bancos de dados NoSQL, marcando-os claramente de forma diferente dos bancos de dados SQL tradicionais. A maioria deles se baseia no armazenamento de pares de valores-chave simples, com base na premissa de que a simplicidade leva à velocidade” (LI e SATHIAMOORTHY, 2015).

2.11. *Requisitos de Software*

Segundo Sommerville (2011) “os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços oferecidos e as restrições a seu funcionamento”. O levantamento de requisitos é uma das etapas do processo de software mais importantes devido ao grau de comprometimento das etapas subsequentes e da habilidade dos analistas em abstrair as ideias e proposições dos clientes transformando-as em itens claros e mensuráveis pela equipe de desenvolvimento. São muitas as abordagens existentes também nesta etapa, desde os tradicionais modelos de dados e funções (*DER* e *DFD*) até os diagramas que compõe a *UML*.

Segundo Sommerville (2011), os requisitos podem ser divididos ainda segundo diversas ópticas, mas comumente são classificados como funcionais e não funcionais:

- Requisitos Funcionais: Os requisitos funcionais são uma descrição das funções que devem ser desempenhadas pelo aplicativo.
- Requisitos Não Funcionais: Os requisitos não funcionais estão diretamente ligados ao uso da aplicação, ou seja, estão relacionados a desempenho, usabilidade, confiabilidade e segurança entre outros.

3. PROCEDIMENTOS METODOLÓGICOS

Neste capítulo são abordadas as metodologias utilizadas para formatação e condução da pesquisa. Trata-se de uma pesquisa exploratória, pois durante sua elaboração foram abordados os processos desenvolvidos dentro do LAMETRO/SATC, consultas a auditores externos da certificadora que atua na instituição, bibliografias que denotam os conteúdos relativos a controle de qualidade.

A abordagem foi embasada em pesquisa-ação não estruturada, modelo no qual a entrevista é desenvolvida de uma forma mais flexível e a sensibilidade do entrevistador e o desenrolar da entrevista direcionam o fluxo.

A princípio não se formulou uma pauta, partindo-se das observações primárias e relatos do funcionário do laboratório LAMETRO/SATC. Em um segundo momento, já de posse de algumas informações primárias pode-se direcionar as próximas entrevistas para os pontos mais importantes e ambíguos dos levantamentos primários.

3.1 Descrição da empresa, do setor/processo foco do trabalho

As entrevistas, observações e auditorias utilizadas no desenvolvimento desta pesquisa foram realizadas e fornecidas pelo LAMETRO SATC e LAMEF UFRGS.

Em entrevistas com os colaboradores do LAMETRO e da controladoria da SATC notou-se que as auditorias internas e externas realizadas demandavam de um tempo para captação, formatação e disponibilização das informações para a montagem do relatório final da auditoria. A partir destas observações, detectou-se a necessidade de se desenvolver uma ferramenta para auxiliar na realização, documentação e controle de auditorias internas nos laboratórios da instituição. Optou-se por utilizar o LAMETRO (Laboratório de Metrologia) como foco da pesquisa, principalmente por ser o alvo da próxima certificação que a instituição buscava.

Para buscar mais informações que pudessem trazer novas proposições e indagações a respeito da pesquisa, foram feitas entrevistas com alguns auditores que prestam serviços à instituição e nestas entrevistas detectou-se que as certificadoras não se utilizam de ferramentas tecnológicas que auxiliassem e automatizassem o processo de auditorias à época.

Nos primeiros meses da pesquisa, no início de 2013, não foram identificados aplicativos que viessem de encontro às necessidades e que pudessem ser aplicados

na instituição. No final do ano de 2015 surgiram algumas ferramentas no mercado que se assemelham em alguns pontos do que foi proposto para o protótipo, mas com custo relativamente alto. Não foi possível realizar uma análise mais profunda destas ferramentas visto que as empresas não possuem versões de testes e não forneceram cópias para avaliação.

Para avaliar o protótipo de aplicação foram lançadas algumas auditorias de teste, sendo que o principal teste desenvolvido foi a partir de uma auditoria feita no LAMEF UFRGS.

3.2 Método de Pesquisa

Em uma primeira etapa o pesquisador adquiriu informações a respeito da execução de auditorias, partindo da aquisição e estudo da NBR ISO 19011:2011 que pauta sobre a realização de processos de auditoria e que nesta edição de 2011 traz alterações importantes. Uma destas alterações foi a que a norma não se restringe mais as auditorias ambientais e de qualidade podendo ser utilizada por variados sistemas de gestão.

A partir deste procedimento foram realizadas reuniões com os funcionários da controladoria da SATC e posteriormente com auditores externos ligados à instituição para levantar características de execução das auditorias a partir da norma NBR ISO 19011, que tinham potencial de se transformar em requisitos do protótipo.

De posse destas informações iniciou-se a primeira etapa do processo de software que tem por finalidade transformar as características e necessidades levantadas em um documento que representa as funções que o software deve permitir executar (levantamento de requisitos).

Com o documento de requisitos em mãos pôde-se partir para a etapa de modelagem de dados. É nesta etapa que se transformam os requisitos levantados e acordados com o cliente, em estruturas digitais que permitam armazenar as informações para posterior interação com a aplicação (banco de dados).

Como já citado anteriormente, uma aplicação *web* está dividida em duas partes distintas no que tange às suas responsabilidades, mas que em conjunto representam o protótipo da aplicação propriamente dito.

No lado do *back-end* está instalado um aplicativo que permite levantar serviços (*webservices*) para que possam ser consumidos por aplicações a partir da estrutura

de internet/intranet e optou-se por utilizar a ferramenta *Node.js*. Neste servidor os serviços foram disponibilizados através de *webservices* no padrão *rest*, por ser o modelo mais simples, eficiente e utilizado do mercado.

Em relação ao *front-end*, é a camada da aplicação que roda no equipamento do cliente desenvolvida com o *framework AngularJS*. Este *framework* é o padrão de grandes empresas como por exemplo a *Google*, para acessar *webservices*, e que aumenta a produtividade e a escalabilidade, além de reduzir sensivelmente o trabalho do desenvolvedor visto que o mesmo reduz o número de linhas de código.

Após a criação do protótipo iniciou-se uma fase de testes por parte da equipe de desenvolvimento para levantamento de erros e não conformidades com os requisitos levantados. Algumas observações com relação a funcionalidades, desempenho e layout foram apuradas e algumas alterações foram realizadas. Após os testes citados e de posse de uma versão estável do protótipo, partiu-se para os testes de avaliação com auditorias reais. O protótipo está implantado em um serviço/servidor *web* denominado *Heroku*, que se apresentou como uma ferramenta interessante levando-se em consideração que é gratuita e apresenta bom desempenho.

Neste momento as auditorias do LAMETRO SATC e do LAMEF UFRGS foram as ferramentas utilizadas para a avaliação do aplicativo.

A simulação partiu do cadastramento dos dados da auditada, da certificadora e da equipe de auditores. Na sequência o auditor líder levantou uma lista de pontos de checagem para incluir como *checklist* da auditoria e foram inseridos no protótipo. Após a realização dos procedimentos foi emitido o relatório final da auditoria.

4 APLICAÇÃO PRÁTICA DO TRABALHO

O capítulo aborda as características de cada etapa do desenvolvimento do protótipo, iniciando pelas proposições das funcionalidades do documento de requisitos, levantamento e consumo dos serviços e armazenamento das informações.

4.1. Documento de requisitos

Ao final do levantamento dos requisitos, observando-se as funcionalidades, restrições, características e ferramentas tecnológicas disponíveis e as questões relativas ao desempenho e qualidade do aplicativo, deve-se gerar um documento contendo todas essas informações e este documento deve ser apresentado ao cliente firmando um contrato que estabelece todas as atividades propostas e aprovadas por ambas as partes. Este documento é denominado “documento de requisitos”, como afirma (SOMMERVILLE, 2011).

“Isso mostra que os requisitos não são independentes e que muitas vezes geram ou restringem outros requisitos. Portanto, os requisitos de sistema não apenas especificam os serviços ou as características necessárias ao sistema, mas também a funcionalidade necessária para garantir que esses serviços/características sejam entregues corretamente”(SOMMERVILLE, 2011).

Por uma questão de especificação e para que se possa analisar as funcionalidades implementadas, é apresentada a seguir uma breve lista com os requisitos funcionais do protótipo:

- **RF01 – Controle de Colaboradores**

O requisito deve permitir a inclusão, consulta e alteração dos dados pessoais dos colaboradores. Optou-se por solicitar apenas informações que remetam ao contato direto com o colaborador (nome, fone e e-mail) preservando, desta forma, outras informações do colaborador que não necessitam estar disponíveis na aplicação.

- **RF02 – Controle de auditores**

O requisito deve permitir inclusão, consulta e alteração de auditores que serão responsáveis por efetuar um processo de auditoria, seja ela interna ou externa. Deve-se configurar no momento da inclusão se o auditor é interno ou externo. Decidiu-se por solicitar informações como documentos, e-mail e fone do auditor, pois se faz necessário inserir no documento final de auditoria.

- **RF03 – Controle de Certificadora**

O requisito deve permitir a inclusão, consulta e alteração da empresa certificadora no caso de uma auditoria externa e da própria empresa a ser auditada no caso de se tratar de uma auditoria interna. Este requisito exige alto nível de detalhamento dos dados cadastrais devido às exigências do relatório final de auditoria.

- **RF04 – Controle de Clientes**

O requisito deve permitir a inclusão, consulta e alteração da empresa a ser auditada. São solicitadas informações como Razão social, endereço, responsável pela auditoria, e-mail e fone pois são exigências da norma e devem constar no documento final de auditoria.

- **RF05 – Controle de Auditoria Interna e Externa**

O requisito deve permitir a inclusão, consulta e alteração de uma auditoria, definir no momento da inclusão se a mesma é interna ou externa. Deve-se poder escolher, no momento da inclusão ou posteriormente, quais são os auditores e quem é o auditor líder.

- **RF06 – CheckList**

O requisito deve permitir a inclusão, consulta e alteração de pontos a serem auditados dentro da instituição. Estes requisitos devem aparecer na forma de uma estrutura de árvore. Deve-se definir em cada ponto de controle qual é o documento que descreve o processo, sendo que este documento é composto por um valor de controle alfanumérico de aproximadamente 10 dígitos. Deve-se desenvolver também uma rotina de checklist para efetuar a verificação diretamente no ponto de controle, preferencialmente executada em um dispositivo móvel, escolhendo-se o documento, o funcionário, o auditor, as observações e as ocorrências (caso ocorram).

- **RF07 – Ocorrências**

O requisito deve permitir cadastrar, consultar e alterar ocorrências levantadas a partir de um ponto de controle do checklist. Durante uma auditoria pode-se documentar mais de uma ocorrência por ponto de controle, podendo esta ocorrência ser uma não conformidade que é basicamente o não atendimento de uma necessidade ou expectativa que esteja implícita ou obrigatória, ou uma sugestão que é um apontamento de acompanhamento e melhoria de processos.

A partir do momento que uma não conformidade é levantada deve-se identificar o problema, a forma de conduta e o prazo de correção.

- **RF08 – Emissão do relatório final da Auditoria**

O requisito deve permitir a geração do relatório final de auditoria, a partir de um modelo previamente apresentado, contendo os dados do cliente, da certificadora, dos auditores, checklist, observações, ocorrências, etc.

- **RF09 – Controle de Acesso**

O requisito deve permitir a checagem do usuário que está se logando no aplicativo, visto que o mesmo estará alocado em um servidor na nuvem. Deve-se também permitir a inclusão, consulta e alteração de usuários do sistema, inclusive com níveis de acesso. Isso se faz necessário devido à possibilidade de alguns usuários somente

poderem visualizar o documento final de auditoria ou alguma outra funcionalidade específica, mas sem poder de alteração na aplicação.

4.2. Diagrama de Classes

A linguagem de modelagem unificada (UML) teve origem na compilação das melhores práticas de Engenharia de Software, emergindo como padronização para os documentos produzidos durante o processo de software.

O diagrama de classes figura como um dos elementos centrais da UML e permite documentar a estrutura das classes, sua hierarquia e relacionamentos (MCNEISH, 2002).

A Figura 5 demonstra a abstração das classes do protótipo, a hierarquia e o relacionamento existente entre elas a partir dos elementos levantados no documento de requisitos.

Analisando o diagrama nota-se uma segmentação com o intuito de separar as classes de tal forma que facilite o entendimento, mesmo sendo analisado por leitores que não sejam da área de TI.

Na parte superior da Figura 5 pode-se notar um conjunto de classes que representam tipos de pessoas que interagem com a aplicação, com suas devidas particularidades, hierarquias e relacionamentos.

Na parte inferior da Figura 5 nota-se um conjunto de classes que organiza as atividades de realização da auditoria, a lista de pontos de checagens com suas respectivas ocorrências e não conformidades.

Pode-se identificar na Figura 5, que cada classe possui um conjunto muito particular de atributos e estes foram apurados durante o levantamento de requisitos.

Estes atributos representam as informações a respeito de cada objeto da aplicação, que necessitam ser armazenados e processados.

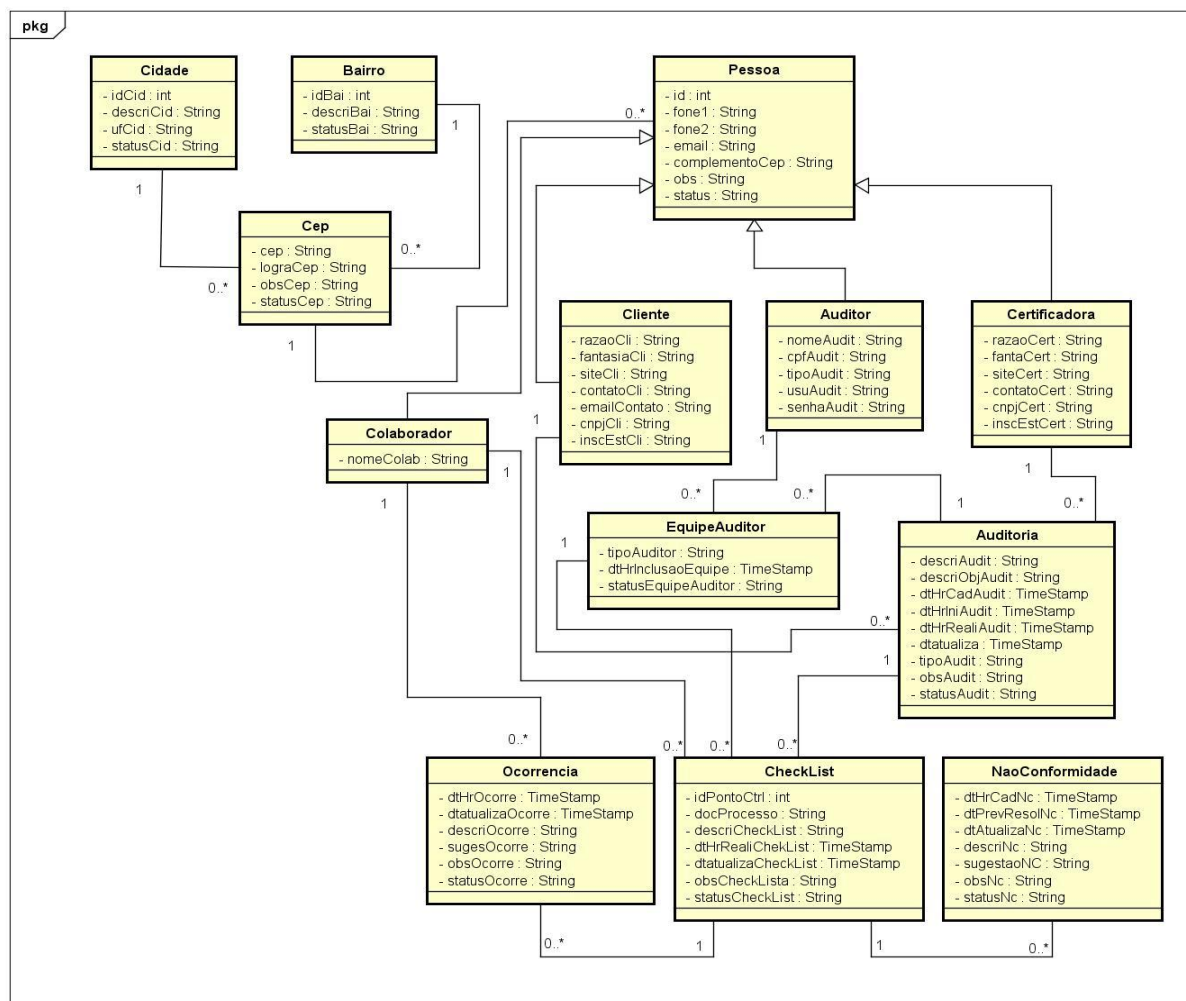


Figura 5- Diagrama de Classes DbAudit – Fonte: O Autor.

4.3. Node.Js

O Node.Js é uma plataforma *server-side*, ou seja, fica localizada no servidor em que a aplicação está implantada. Esta plataforma disponibiliza uma série de funções que reduzem a carga de processamento na máquina do servidor, além de permitir um maior controle por parte do desenvolvedor (CARRENHO, 2016).

Optou-se por codificar a camada da aplicação localizada no servidor em *Node.Js*, também em virtude de ser uma plataforma *JavaScript*, linguagem em que o *framework AngularJS* foi escrito e que foi utilizada na camada de aplicação do protótipo.

Vários testes foram efetuados com outros servidores e linguagens de programação para disponibilizar os *webservices*, mas o *Node.Js* foi o que apresentou melhor escalabilidade e desempenho o que justifica a sua aplicação.

4.4. MongoDB

Desde o advento da criação dos primeiros computadores nota-se que algumas ferramentas ganham força e evoluem paralelamente à medida que novas tecnologias são desenvolvidas. Neste contexto pode-se enquadrar os sistemas gerenciadores de bancos de dados como sendo uma das ferramentas mais importantes do mercado, sem a qual seria impossível manter-se o nível de disponibilidade das informações que são a base da sociedade moderna.

O *MongoDb* é considerado hoje um dos principais bancos de dados *NoSql* do mercado. Esse novo modelo de banco de dados difere da estrutura tabela, registros e colunas do modelo relacional e se caracteriza por ser um modelo de banco de dados orientado a documentos. Toda a informação fica armazenada em coleções disponibilizadas no formato *Json*, ou seja, dentro das coleções tem-se pares (chave-valor) para localizar cada informação específica (Rodrigues, 2019).

Vale salientar que não se pode afirmar que um banco de dados *NoSql* é melhor que um banco de dados relacional. Cada modelo de banco de dados atende melhor um determinado cenário e considerando os prós e contras decidiu-se por utilizar o *MongoDB*.

4.5. Json

Uma das metodologias de desenvolvimento mais difundidas atualmente são os sistemas *web*, e muitos destes sistemas compartilham uma tecnologia denominada *WebService*. Esses serviços podem ser consumidos pelas mais diversas tecnologias e se baseiam na troca de informações entre os componentes da aplicação.

Historicamente o formato de troca de dados mais comum e utilizado era o *XML-Extensible Markup Language*. O *Json* surgiu nos últimos anos e já se transformou no formato mais utilizado para intercâmbio de informações entre as camadas das aplicações na internet, principalmente devido à facilidade que apresenta para o desenvolvedor montar e transferir seus dados através de objetos de simples formatação como afirma Zemel (2016).

“Já não é nenhuma novidade no mundo de desenvolvimento web o uso e abuso da tecnologia JSON. Entretanto, nem todos conhecem este padrão de intercâmbio facilitado de dados entre aplicações. Não é obrigatório, mas saber JSON pode ser bastante útil em projetos um pouco mais sofisticados e/ou que necessitam de uma demanda/política de transferência de dados mais simples, leve e eficiente”(ZEMEL, 2016).

“Usar JSON proporciona inúmeras vantagens ao desenvolvedor web. Desde a simplicidade em tratar os dados à maneira fácil como é possível integrar tecnologias *client-side* com *server-side*, o JSON certamente é bastante recomendado”(ZEMEL, 2016).

Json é um acrônimo para “*JavaScript Object Notation*” e se trata de um subconjunto de notações *JavaScript*, mas seu uso não está condicionado ao uso do *JavaScript* como linguagem de desenvolvimento, visto que o *Json* atualmente é utilizado por mais de 15 linguagens de programação.

Correa (2016) afirma que:

“JSON (JavaScript Object Notation) é um modelo para armazenamento e transmissão de informações no formato texto. Apesar de muito simples, tem sido bastante utilizado por aplicações Web devido a sua capacidade de estruturar informações de uma forma bem mais compacta do que a conseguida pelo modelo XML, tornando mais rápida a conversão dessas informações para uma estrutura de objetos. Isto explica o fato de o JSON ter sido adotado por empresas como Google e Yahoo, cujas aplicações precisam transmitir grandes volumes de dados.”(CORREIA, 2016).

Durante esta etapa constatou-se um pequeno gargalo no fluxo de desenvolvimento devido ao pouco conhecimento da tecnologia por parte da equipe também em função de ser uma tecnologia nova e extensa. O emprego de *webservices REST* em detrimento de outras tecnologias que também estavam disponíveis se deu em função da curva de aprendizado que é muito menor.

4.6. Webservices

Durante a fase de análise e levantamento de dados a respeito do processo de auditoria propriamente dito, ficou evidenciado que o protótipo parte de uma necessidade básica e elementar que é a alta disponibilidade.

Para que se possa alcançar um alto grau de disponibilidade em uma aplicação, não se pode deixar de considerar um parque tecnológico com alta diversidade de

ambientes e arquiteturas, destacando-se as embarcadas. Como pode-se verificar na Figura 6, qualquer aplicação *Web* hoje compartilha de uma premissa de estruturação que parte da divisão em duas fases distintas e independentes, mas que possuem um alto grau de comunicação chamado de *Server side / ClientSide* ou *Back End / Front End* (PEDCHENKO, 2019).

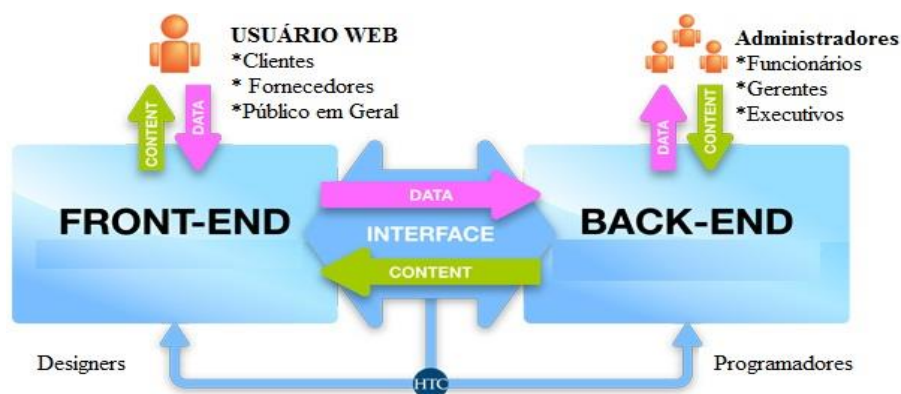


Figura6 - Arquitetura Back-end x Front-End – Fonte: Pedchenko (2019).

Alcançar alta disponibilidade aliada à performance e escalabilidade exige uma tecnologia que permita dividir as responsabilidades dentro do processo de desenvolvimento e os *webservices* vem de encontro a esta necessidade.

Tradicionalmente um *webservice* utiliza SOAP (*Simple Object Access Protocol*) para efetuar a invocação de métodos remotos (*webservices*). O SOAP sofre algumas críticas devido ao alto grau de burocracia que o processo exige e neste sentido o *webservice Rest* é muito mais simples e ágil, o que foi determinante para a utilização do modelo. De acordo com Poutsma (2016), “Nos últimos anos, REST surgiu como uma alternativa interessante as arquiteturas distribuídas como SOAP / WSDL / WS”.

Segundo Williamson (2015), os serviços *Rest* (*Representational State Transfer*) permitem que se faça uma separação entre as responsabilidades (Figura 7).

Ficam isoladas dos serviços *Rest*, as responsabilidades de desenvolver e controlar interface e serviços de usuários. Os serviços focam no armazenamento de dados e na lógica de negócios.

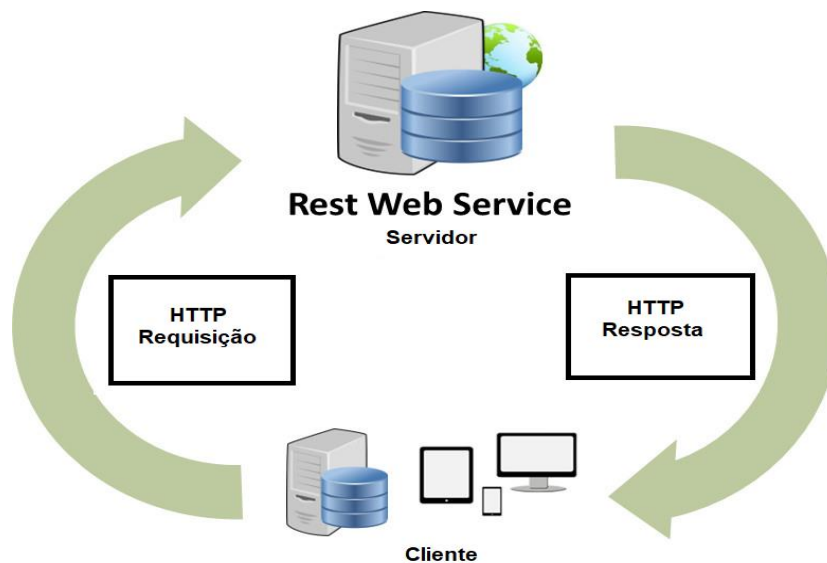


Figura7 - WebService Rest – Fonte: Adaptado de <http://www.stsoftware.com.au>

4.7. AngularJS

O desenvolvimento de aplicações corporativas vem passando por constantes modificações e um conjunto de metodologias e ferramentas tem surgido com o objetivo de facilitar, documentar e planejar o desenvolvimento de projetos, principalmente os de grande porte. Grandes corporações no mundo têm desenvolvido *Frameworks* próprios que atendam às suas necessidades, e em alguns casos essas ferramentas se transformam em padrões mundiais para desenvolvimento de diversos tipos de aplicações. O *AngularJS* surgiu como um *framework* de código aberto baseado em *JavaScript* com o objetivo de fornecer um conjunto de ferramentas bem estruturadas, de fácil compreensão, de alta performance e produtividade, e rapidamente se transformou no *framework JavaScript* do lado do cliente mais conhecido e amplamente utilizado. A Figura 8 demonstra a capacidade de dividir a aplicação em módulos, com funções específicas e independentes entre si, que podem ser consumidas pelas mais diversas plataformas e ferramentas.

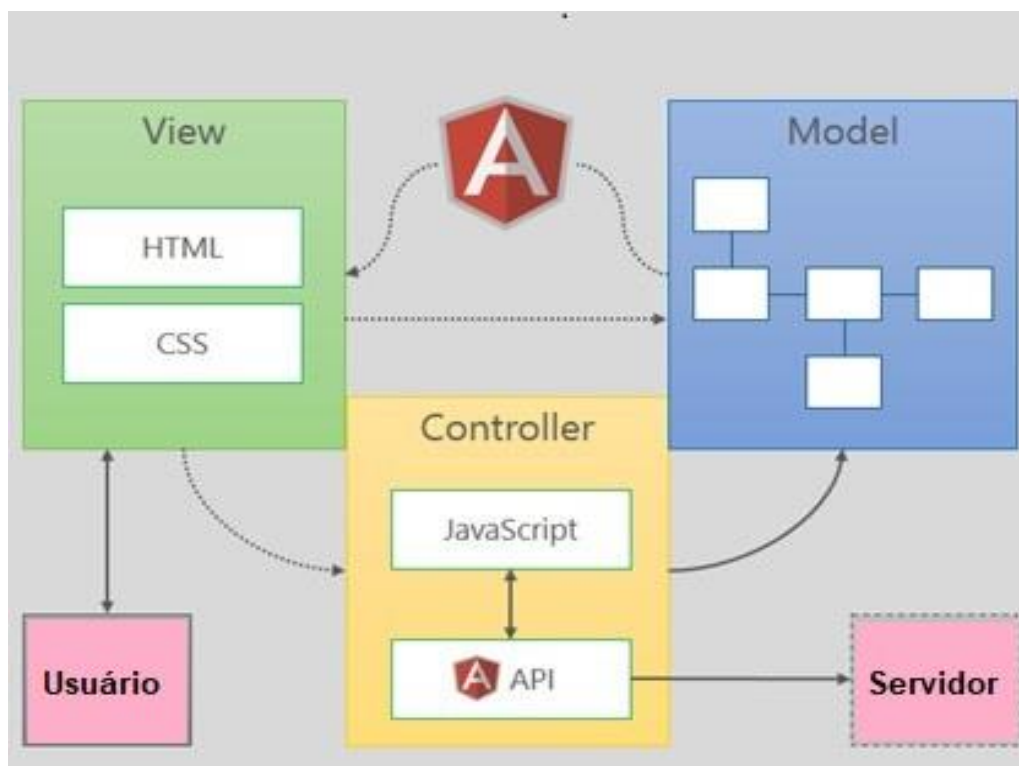


Figura 8 - Diagrama Aplicação AngularJS – MVC – Fonte: Adaptado de: Stropek (2013)

Esse *framework* ganhou notoriedade e ultrapassou os limites da *Google*, ganhando fama e tornando-se uma das ferramentas mais comentadas e utilizadas já em um curto espaço de tempo, desde o seu lançamento, segundo Williamson (2015).

“O AngularJS do Google é um framework Java script MVC (Model-View-Controller, ou Modelo-Visão-Controlador) completo, que facilita bastante a criação rápida de aplicações que executem adequadamente em qualquer plataforma desktop ou móvel” (WILLIAMSON, 2015).

A escolha do *AngularJS* se deu em função então da necessidade latente de criar uma aplicação enxuta em que se pudesse aproveitar toda a capacidade e características das aplicações embarcadas ou *desktop* de uma forma única e eficiente.

4.8. Servidor Web

Da Silva (2016), do site portal da educação, conceitua os servidores *Web* como sendo responsáveis por armazenar e trocar informações com outras máquinas. O serviço exige pelo menos dois participantes envolvidos em cada troca de informações: um cliente, que solicita informações, e um servidor, que atende a esses pedidos.

O funcionamento da ferramenta, observando-se do lado cliente é muito simples, bastando para tal que o usuário disponha de um browser para disparar as suas requisições baseadas no protocolo *https* (*web*) e receber de volta a informação solicitada juntamente com sua formatação geralmente baseadas no formato de *html* (linguagem de formatação de dados *web*).

Já do lado do servidor a complexidade das atividades é bem maior e um conjunto complexo e interdependente de ferramentas e serviços precisam ser configurados e disponibilizados para atender a estas requisições disparadas pelos clientes.

Muitos foram os testes feitos durante a fase de desenvolvimento do protótipo com diversos servidores *web*, analisando a qualidade dos serviços, tempo de resposta, facilidades de configuração e de levantamento dos serviços entre outras características importantes até a escolha do servidor *Node.js*, visto que foi o que apresentou a melhor relação dos requisitos apresentados.

Após abordadas as etapas para a construção do protótipo, é no servidor *web* que está implantada a aplicação, que pode ser acessada pelo link "<https://appdbauditheroku.herokuapp.com/app/dbauditoria/index1.html#/>". Foi implantado um usuário padrão (usuário: admin, senha: admin), para permitir o acesso inicial ao aplicativo.

5. APLICAÇÃO E DISCUSSÃO DOS RESULTADOS

Durante a documentação desta pesquisa argumentou-se sobre o porquê da utilização de cada ferramenta do processo de software de tal maneira que mesmo os leigos no assunto consigam entender a sua aplicação no processo e possam utilizá-las de forma simples e eficiente.

O procedimento de teste levantou algumas ocorrências e não-conformidades que foram inseridas na aplicação e posteriormente incluídas no relatório geral da auditoria gerado pelo protótipo e que se encontra no anexo da dissertação. Algumas versões diferentes do relatório final de auditoria foram implementadas, mas ficou evidenciado que esta é uma funcionalidade que necessita um estudo mais aprofundado para que se possa encontrar um modelo que seja mais enxuto e representativo.

Após a realização dos testes foram levantadas algumas situações que podem ser aprimoradas, principalmente no que tange à funcionalidade da aplicação.

Levando-se em consideração que o protótipo roda em ambientes que exigem uma portabilidade, disponibilidade e funcionalidades extremas, o layout é algo importante e em alguns momentos apresentou baixa de produtividade por parte do auditor devido à responsividade gerada pela aplicação. Cabe salientar que estas baixas de produtividade estão assim descritas objetivando-se um retorno máximo da aplicação, porém já representam um ganho significativo em relação ao processo manual realizado atualmente.

Cabe salientar que os dados das auditorias, auditores e clientes utilizados para a avaliação do protótipo são informações de domínio público e estão disponíveis para acesso via sites de busca na internet.

Para melhorar a compreensão do funcionamento do protótipo, optou-se por apresentar uma linha de execução do aplicativo, durante um processo de auditoria.

Essa sequência pode ser comprovada observando-se as Figuras 9 até a 12.

A Figura 09 ilustra a tela de cadastro de clientes, contendo as informações necessárias para posterior geração do documento de auditoria. Esta funcionalidade, assim como todas as outras telas, está contida no *front-end* da aplicação e foram desenvolvidas com foco na usabilidade e funcionalidade. Pode-se notar na tela o cadastro efetuado do cliente LAMEF UFRGS.

Figura 9 - Tela de Clientes - Fonte: O Autor

Pode-se notar nesta tela a solicitação das informações que devem constar no momento da emissão do relatório final da auditoria, além das informações do responsável pela auditoria na organização.

A Figura 10 remete à funcionalidade do protótipo, que permite cadastrar os auditores vinculados à certificadora. Note-se que durante a inclusão do auditor no protótipo não se define se o mesmo é auditor especialista ou auditor líder, funcionalidade que só será preenchida no momento em que for inserido um auditor na equipe de uma auditoria. O que deve ser definido é se o auditor pode executar auditorias internas ou externas.

Figura 10 - Tela de Auditores - Fonte: O Autor

A Figura 11 representa a funcionalidade mais importante do processo de auditoria. É nesta tela que é configurada a auditoria, se é interna ou externa, qual o cliente, a empresa certificadora, as datas previstas e de realização da auditoria e qual norma será aplicada.

Figura 11 - Tela de Auditoria - Fonte: O Autor

A partir deste momento o auditor líder deve levantar os pontos de controle (checklist), lançá-los no protótipo, informar qual a equipe de auditores e gerenciar a realização da auditoria nos pontos de controle.

Durante o processo de auditoria os auditores vão preenchendo os dados dos pontos de controle configurados pelo auditor líder. Se surgirem ocorrências e/ou não conformidades, estas devem ser preenchidas de acordo com as opções das telas das funções citadas.

A Figura 12 demonstra a tela que permite aos auditores informar qual a situação encontrada no ponto de controle, qual o colaborador estava no ambiente e como o processo estava sendo desenvolvido. Existindo alguma ocorrência ou não conformidade o auditor deve lançá-la e indicar quais as ações corretivas devem ser tomadas no tempo delimitado.

CheckPoint	Descrição	Colaborador	Equipe	Documento	Data/Hora		Status	NC	Ocorrencia	Excl
					Cadastro	Realizacao				
5c37f0515cd78d0004c008e0	Equipe de Trabalho; Confidencialidade das informações e imparcialidade; Controle de Registros e Dados(Rev 03)	<input type="text"/>	<input type="text"/>	Manual Qualidade Rev 04 Item 4.1	05/05/2017	11:00:00	Ativo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5c37f0515cd78d0004c008df	Cronograma de Análise Crítica do sistema de gestão da qualidade no ano de 2016.	<input type="text"/>	<input type="text"/>	Manual Qualidade Rev 04 Item 4.2	05/05/2017	11:10:00	Ativo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5c387b97dc2a5f0004581649	Elaboração, Alteração e Controle de Documentos.	<input type="text"/>	<input type="text"/>	Manual Qualidade Rev04 Item 4.3	05/05/2017	11:15:00	Ativo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5c387cb1dc2a5f000458164b	Análise Crítica de Pedidos, Propostas e Contratos.	<input type="text"/>	<input type="text"/>	Manual Qualidade Rev 04 Item	05/05/2017	11:20:00	Ativo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 12 - Tela de CheckList – Fonte: O Autor

Após a averiguação de todos os pontos do *checklist*, o auditor líder deve gerar o relatório final de auditoria e fazer os ajustes que achar conveniente.

É importante salientar que todas as telas possuem as mesmas funcionalidades e layout com o intuito de reutilizar os procedimentos desenvolvidos, facilitar os processos de manutenção e treinamento dos usuários do protótipo.

6. CONSIDERAÇÕES FINAIS

6.1. Conclusões

Vivencia-se no Brasil uma era de adequação dos processos produtivos e da prestação de serviços, aumentando o poder de competitividade com organizações Internacionais e melhorando o controle de qualidade de seus processos.

Durante o desenvolvimento desta pesquisa analisou-se os procedimentos realizados durante um processo de auditoria, seja ela interna ou externa, embasada na norma NBR ISO 19011:2018, a fim de estabelecer parâmetros que possibilitassem desenvolver uma ferramenta para auxiliar os auditores no planejamento, execução e acompanhamento de auditorias e as possíveis não conformidades levantadas.

Os auditores planejavam a auditoria recolhendo as informações necessárias para a montagem da lista de pontos de checagem (*checklist*), e posteriormente, iam a campo realizar as atividades fazendo as suas devidas anotações. De posse destas anotações então era feita a digitação das informações para posterior montagem e impressão do relatório final de auditoria.

Como resultado da pesquisa desenvolveu-se um protótipo de aplicação *web* que possibilitou executar a auditoria e inserir as informações diretamente no ponto de checagem. A própria aplicação se encarrega de montar o relatório final de auditoria, buscando as informações diretamente na base de dados alimentada pelos auditores e permitindo que se faça as devidas correções antes da impressão do relatório, além de permitir-se efetuar um controle mais adequado das ocorrências e não conformidades.

Ficou evidenciado o ganho de desempenho com a aplicação do protótipo nos testes efetuados nos laboratórios LAMEF E LAMETRO, visto que as informações encontradas nos pontos de controle, suas ocorrências e não conformidades são inseridas imediatamente e estão disponíveis para todos os envolvidos no processo além de permitirem a rastreabilidade das ações.

Durante os meses de desenvolvimento da pesquisa e protótipo constatou-se que algumas ferramentas semelhantes foram criadas no mercado e estão disponíveis, sobretudo possuem um valor de aquisição e aplicação altos. Uma das características mais importantes do protótipo e já discutidas anteriormente é a alta disponibilidade. Esta característica permite sugerir que todas as funcionalidades aqui desenvolvidas e

evidenciadas podem ser transformadas em um serviço público e gratuito para execução de auditorias, que poderiam ser utilizadas por organizações nas suas auditorias internas e por certificadoras nos processos de certificação.

Para avaliar a utilização do aplicativo foi inserida no aplicativo uma auditoria interna realizada e disponibilizada pelo laboratório LAMEF UFRGS, o que permitiu verificar as funcionalidades, desempenho e aplicação do protótipo. Essa atividade possibilitou comprovar que o protótipo é uma ferramenta útil e que pode ser implantada em qualquer organização que deseja melhorar seus processos e/ou produtos, servindo como uma ferramenta de gestão.

6.2. Sugestões de trabalhos futuros

O *DbAudit* apresenta características favoráveis para emprego como ferramenta nos processos de auditoria interna e/ou externa e se portou muito bem em um ambiente distribuído e de alta volatilidade. Em conjunto com a equipe de controladoria da SATC notou-se que algumas ferramentas desenvolvidas pela equipe de TI da instituição representam um ganho significativo no controle de qualidade e logística da instituição e poderiam trabalhar em conjunto com o *DbAudit*, como por exemplo o “Controle de documentos”.

Pode-se elencar algumas sugestões vislumbradas ao final desta pesquisa como:

- A integração com um aplicativo que normatize o controle de documentos referenciado no protótipo.
- A transformação do *DbAudit* em um serviço público e gratuito servindo como ferramenta de gestão para as organizações e/ou certificadoras.

Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT. **NBR ISO 9000**. Fundamentos para sistemas de gestão da qualidade. Rio de Janeiro, Rio de Janeiro, 2005.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT. **NBR ISO 19011**: Diretrizes para auditoria de sistemas de gestão. Rio de Janeiro, 2011.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT. **NBR ISO 19011**: Diretrizes para auditoria de sistemas de gestão. Rio de Janeiro, 2018.

ABRANTES, J. **Gestão da Qualidade**. Rio de Janeiro: Interciência, 2009.

ALTERMANN, D. Midiatismo. **Midiatismo cultura, comunicação e marketing digital**, Disponível em: <<https://www.midiatismo.com.br/design-responsivo-entenda-o-que-e-a-tecnica-e-como-ela-funciona>>. Acesso em: 20 dez. 2012.

BEZERRA, F. www.portal-administracao.com. **Portal da Administração**. Disponível em: <<http://www.portal-administracao.com/2014/08/ciclo-pdca-conceito-e-aplicacao.html>>. Acesso em: 20 mai 2016.

CARRENHO, R. M. Criando serviços com Node.Js. **DevMedia**. Disponível em: <<https://www.devmedia.com.br/criando-servico-de-microblog-com-node-js/31036>>. Acesso em: 16 jan. 2016.

CORREA, E. Sobre o Site DevMidia. **Site DevMidia**. Disponível em: <<http://www.devmedia.com.br/introducao-ao-formato-json/25275>>. Acesso em: 20 maio 2016.

DA SILVA, J. A. F. Sobre o Portal da Educação. **Site Portal da Educação**. Disponível em: <<http://www.portaleducacao.com.br/informatica/artigos/17165/como-funciona-um-servidor-web>>. Acesso em : 23 maio 2016.

DEITEL, PAUL; DEITEL, HARVEY. **Java: Como Programar**, 8ª Edição. São Paulo: Pearson Prentice Hall, 2010.

FALCONI, V. C. **TQC - Controle da Qualidade Total (no estilo Japonês)**. Nova Lima: Editora INDG TecS, 2004.

FELLIPE, ALEIXO; PEREIRA, BRUNO PONTES; ATAÍDE, LEONARDO MINORA. Processo Acadêmico simplificado: Uma proposta de Processo para o CEFET-RN. **Holos**, p. 74-85, 2006.

FERGUSON, N. CareerFoundry. **CareerFoundry**. Disponível em: <<https://careerfoundry.com/en/blog/web-development/whats-the-difference-between-frontend-and-backend/>>. Acesso em: 17 jul. 2018.

FERNANDES, J. L. S. Etapas Necessárias para a Implantação de um sistema de Gestão Integrado. **Rev. Adm. UFSM, Santa Maria**, v. 8, número 1., p. 1-12, 2015.
GARBI, G. G. **A rainha das ciências: um passeio histórico pelo maravilhoso**. Lavras: Livraria da Física, 2009.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO. **The ISO Survey of certifications 2018**. Genebra. 2018.

KOCH, M.; GASPARINI, I. **Abordagem Web Design Responsivo para Desenvolver Aplicações Web que se Adaptam a qualquer Dispositivo**. Blumenau: [s.n.], 2015.

LI, Y.; SATHIAMOORTHY, M. A performance comparision of SQL and NoSQL databases. **Site da Abook apart**, Auckland, 21 nov. 2015. Disponível em: <<http://abookapart.com/products/reponsive-web-design>>. Acesso em: 21 nov. 2015.

MCNEISH, K. UML Class Diagrams. **Code Magazine**, p. 10, 2002.

OLIVEIRA, L. M. R. Repositório da Universidade de Lisboa. Disponível em: <<http://repositorio.ul.pt/handle/10451/20405>>. Acesso em: 30 jul 2017.

PEDCHENKO, A. Front-end x back-end web development. **Aog It**. Disponível em: <<https://www.aog.jobs/blog/frontend-vs-backend-web-development-whom-do-you-need-for-your-project/>>. Acesso em: 16 jan. 2019.

PROJECT MANAGEMENT INSTITUTE. PMI. **Guide PMBOOK**. Pennsylvania - USA, 2017.

POUTSMA, A. Sobre Spring MVC. **Site da Spring MVC**. Disponível em: <<http://spring.io/blog/2009/03/08/rest-in-spring-3-mvc/>>. Acesso em: 22 maio 2016.

RIBEIRO, L. DevMedia. **DevMedia**. Disponível em: <<http://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>. Acesso em: 20 Fevereiro 2016.

RIDLEY, J. Optimizing AngularJS Single-Page Applications for Googlebot Crawlers. **The Moz Story**. Disponível em: <<https://moz.com/blog/optimizing-angularjs-single-page-applications-googlebot-crawlers>>. Acesso em: 31 maio 2017.

RODRIGUES, J. NoSql e MongoDB. **DevMedia**. Disponível em: <<https://www.devmedia.com.br/guia/nosql-e-mongodb/34482>>. Acesso em: 10 jan. 2019.

RODRIGUES, N. D. S. Um Estudo corporativo Utilizando Diferentes Abordagens de Bancos de Dados. **XI Brazilian Symposium on Information System**, 26 a 29 Maio 2015. 548.

SAUER, G. Organica digital Seeds. **Organica Digital Innovation**. Disponível em: <<https://www.organicadigital.com/seeds/por-que-utilizar-mobile-first-no-web-design/>>. Acesso em: 20 abr. 2017.

SOMMERVILLE, I. **Engenharia de Software**. São Paulo: Pearson Education do Brasil, 2011.

STERN, E. H. Sobre o Site DevMidia. **Site da DevMidia**. Disponível em: <<http://www.devmedia.com.br/artigo-sql-magazine-6-postgresql-introducao-e-conceitos/7185>>. Acesso em: 20 maio 2016.

STROPEK, Rainer. AngularJS with TypeScript and Windows Azure Mobile Services, 2013. Disponível em: <<http://goo.gl/O2NENn>>. Acesso em 01 ago. 2019.

STSOFTWARE. Disponível em: <<https://www.stsoftware.com.au/site/ST/blog/article/how-to-access-st-data-from-microsoft-excel-using-the-rest-web-service/>>. acesso em: 01 ago 2019.

TICONA, J. M. **A avaliação de Impacto econômico decorrente do processo de certificação de produtos**. Rio de Janeiro: PUC - Rio, 2003.

TRONCO, C. R. **Sistema de gestão integrado de qualidade, meio ambiente, saúde e segurança e responsabilidade social – SGI**: uma experiência de. XII SIMPEP. Bauru: Anais XII SIMPEP. 9 Novembro 2005. p. 12.

WAHLIN, D. **AngularJS in 60 Minutes**. Chandler(Arizona). 2014.

WILLIAMSON, K. **Introdução ao AngularJS**. São Paulo: Novatec Editora Ltda., 2015.

ZEMEL, T. Sobre Desenvolvimento para web.com. **Site Desenvolvementoparaweb.com**. Disponível em: <<http://desenvolvementoparaweb.com/javascript/json-javascript-object-notation/>>. Acesso em: 20 maio 2016.

Anexos

Relatório Final (parcial) Auditoria LAMEF UFRGS

30/06/2019

DbAuditoria

DbAudit Vr 1.0

Data: _/ _/ _

Relatório Final de Auditoria

Cliente

Razão Social: labortorio de metalurgia fisica

Nome Fantasia: lamef

CNPJ: 34.242.342/3423-42

Insc. Estadual: 234234

Fone 1: (51) 3308-7777

Fone 2: (51) 3308-3551

Email: lamef@ufrgs.br

Auditores

Auditor: Vanessa I

CPF: 111.111.111-11

Fone: (49) 9999-9999

Auditor: Fabiano I

CPF: 222.222.222-22

Fone: (51) 5555-5555

Auditor: Roberval

CPF: 017.397.039-39

Fone: (48) 9928-6711

Checklists

Checkpoint: 5c37f0515cd78d0004c008e0

Documento: Manual Qualidade Rev 04 Item 4.1

Data e Hora da Realização: 05/05/2017 11:00:00

Descrição: Equipe de Trabalho; Confidencialidade das informações e Imparcialidade; Controle de Registros e Dados(Rev 03)

Observação:

Não Conformidades

Data e Hora da Realização: 14/01/2019 10:04:04

Data e Hora Prevista Resolução: 10/05/2017 05:00:00

Descrição: Cadastro de : dos Santos no Equipa desatualizado(Titulação: Graduação).

Sugestão: Cadastro de : dos Santos no Equipa desatualizado(Titulação: Graduação).

Observação:

Ocorrências

Nenhuma ocorrência registrada

Checkpoint: 5c37f0515cd78d0004c008df

Documento: Manual Qualidade Rev 04 Item 4.2

Data e Hora da Realização: 05/05/2017 11:10:00

Descrição: Cronograma de Análise Crítica do sistema de gestão da qualidade no ano de 2016.

Observação:

Não Conformidades

Nenhuma não conformidade registrada