INSTITUTO FEDERAL
SANTA CATARINA

CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE SAÚDE E SEGURANÇA
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA TECNOLOGIA DA INFORMAÇÃO

TIAGO DA SILVA

# OS IMPACTOS DA IMPLEMENTAÇÃO DE UM PROCESSO DE QUALIDADE EM UMA EMPRESA DE APLICAÇÃO WEB

Florianópolis - SC
2022

17 de novembro de 2022

TIAGO DA SILVA

# OS IMPACTOS DA IMPLEMENTAÇÃO DE UM PROCESSO DE QUALIDADE EM UMA EMPRESA DE APLICAÇÃO WEB DE MÉDIO PORTE

Monografia apresentada ao (Gestão da tecnologia da informação) do (Florianópolis) do Instituto Federal de Santa Catarina para a obtenção do diploma de (tecnólogo)

Orientador: Felipe Cantório Soares, MSc.

Co-Orientador: Juliano Lucas Gonçalves, MSc.

Florianópolis

17 de novembro de 2022

Ficha de identificação da obra elaborada pelo autor.

Silva, Tiago
Os Impactos da Implementação de um Processo de Qualidade em uma Empresa de Aplicação Web / Tiago Silva; orientação de Felipe Cantorio Soares; coorientação de Juliano Lucas Gonçalves. – Florianópolis, SC, 2022.
43 p.

Trabalho de Conclusão de Curso (TCC) – Instituto Federal de Santa Catarina, Câmpus Florianópolis. CST em Gestão da Tecnologia da Informação. Departamento Acadêmico de Saúde e Serviços.
Inclui Referências.

1. Quality Assurance. 2. Aplicação Web. 3. Processo de Qualidade. 4. Testes. 5. Empresa de Software. I. Cantorio Soares, Felipe. II. Lucas Gonçalves, Juliano. III. Instituto Federal de Santa Catarina. IV. Os Impactos da Implementação de um Processo de Qualidade em uma Empresa de Aplicação Web.

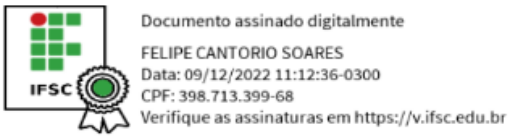
Tiago da Silva

OS IMPACTOS DA IMPLEMENTAÇÃO DE UM PROCESSO DE QUALIDADE EM UMA EMPRESA DE APLICAÇÃO WEB

Este trabalho foi julgado adequado para obtenção do título em (Nome da Habilitação), pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

Florianópolis, 17 de novembro de 2022.

Banca Examinadora:

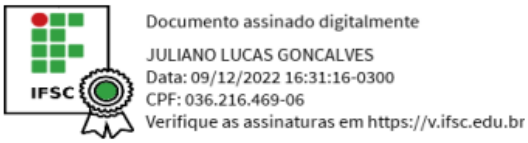



_____

Prof. Felipe Cantorio Soares, MSc.

Orientador

Instituto Federal de Santa Catarina

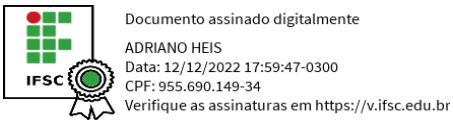



_____

Prof. Juliano Lucas Gonçalves, MSc.

Co-Orientador

Instituto Federal de Santa Catarina

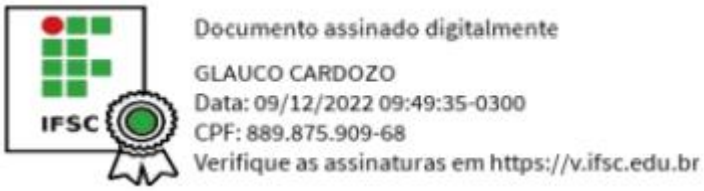



_____

Prof. Adriano Heis

Membro da banca

Instituto Federal de Santa Catarina

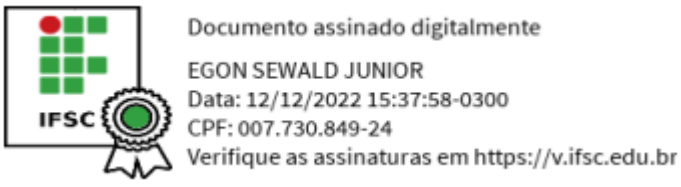



_____

Prof. Glauco Cardozo

Membro da banca

Instituto Federal de Santa Catarina




_____

Prof. Egon Junior

Coordenador do Curso

Instituto Federal de Santa Catarina

## RESUMO

O presente trabalho tem por objetivo apresentar os impactos da implementação de um processo de qualidade em uma empresa de aplicação web. Aborda uma breve contextualização do tema, justificando o desenvolvimento do mesmo. Tendo os objetivos pautados em mostrar os impactos da implementação de um processo de qualidade de aplicações web em uma empresa de software, elencando metodologias utilizadas, **identificando e relacionando** os métodos de qualidade utilizados na empresa, além de **reportar** quais os impactos do uso bem como **apresentar** estatísticas. Traz no referencial teórico a conceituação de Tecnologia da Informação, processo de qualidade, aprofundando no quesito aplicação web, e sobre os testes implementados na área. Os resultados demonstram que houve impactos positivos após a implantação de um processo de *Quality Assurance* (QA), principalmente ao apresentar uma diminuição percentual de bugs externos. Concluiu-se que o trabalho apresenta dados de um período razoável e diretamente retirados do mercado de trabalho, embasado na literatura o trabalho apresenta conteúdos pertinentes para estudantes, pesquisadores e profissionais da área.



Palavras-Chave: aplicação web. quality assurance. processo de qualidade. testes. empresa de software.

## ABSTRACT

The present work aims to present the impacts of implementing a quality process in a web application company. It addresses a brief contextualization of the theme, justifying its development. Having the objectives guided in showing the impacts of the implementation of a process of quality of web applications in a software company, listing methodologies used, identifying and relating the quality methods used in the company, in addition to reporting the impacts of use as well as presenting Statistics. It brings in the theoretical framework the conceptualization of Information Technology, quality process, delving into the web application question, and on the tests implemented in the area. The results show that there were positive impacts after the implementation of a Quality Assurance (QA) process, mainly when presenting a percentage decrease in external bugs. It was concluded that the work presents data from a reasonable period of time and directly taken from the job market, based on the literature the work presents relevant content for students, researchers and professionals in the area.



Keywords: web application. quality assurance. quality process. tests. software company.

## LISTA DE FIGURAS

Figura 1 - Fluxograma de um teste ..... 30
Figura 2 - Relação tempo x custo dos bugs ..... 37

## LISTA DE GRÁFICO

Gráfico 1 - Bugs encontrado no projeto antes de um processo de qualidade ..... 33
Gráfico 2 - Bugs encontrado no projeto depois de um processo de qualidade ..... 35
Gráfico 3 - Comparando Bugs antes e depois do processo de qualidade ..... 36

## LISTA DE ABREVIATURAS E SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

BSTQB – *Brazilian Software Testing Qualifications Board*

CTFL – *Certified Tester Foundation Level*

FTP – *File Transfer Protocol*

HTTP – *Hypertext Transfer Protocol*

IFSC – Instituto Federal de Santa Catarina

IMAP – Internet Message Access Protocol

QA – *Quality Assurance*

POP – *Post Office Protocol*

SMTP – *Simple Mail Transfer Protocol*

TI – Tecnologia da Informação

**SUMÁRIO**

**1. INTRODUÇÃO ..... 10**

1.1. OBJETIVOS ..... 11

**1.1.1. Objetivo Geral ..... 11**

**1.1.2. Objetivos Específicos ..... 11**

**2. REVISÃO DA LITERATURA ..... 12**

2.1. TECNOLOGIA DA INFORMAÇÃO ..... 12

2.2. PROCESSO DE QUALIDADE ..... 14

2.3. SOFTWARE ..... 15

**2.3.1. Qualidade de software ..... 15**

2.4. APLICAÇÃO WEB ..... 17

**2.4.1. Como funcionam as aplicações web ..... 20**

*2.4.1.1 Qualidade em aplicações web ..... 20*

2.5. IMPORTÂNCIA DOS TESTES ..... 21

**2.5.1. Teste de Caixa Branca ..... 22**

**2.5.2. Teste de Caixa Preta ..... 22**

**2.5.3. Teste de Regressão ..... 23**

**2.5.4. Teste de Carga ..... 23**

**2.5.5. Testes de Segurança ..... 23**

**2.5.6. Teste de Usabilidade ..... 23**

2.6. SCRUM ..... 23

2.7. QUALITY ASSURANCE (QA) ..... 24

**3. PROCEDIMENTOS MetodolÓGICOS ..... 26**

**4. ANÁLISE E DISCUSSÃO DOS RESULTADOS ..... 29**

4.1. DESENVOLVIMENTO SEM DEFINIÇÃO DE PROCESSO DE QUALIDADE ..... 32

4.2. DESENVOLVIMENTO COM A DEFINIÇÃO DE UM PROCESSO DE QUALIDADE . 34

4.3. ANÁLISE COMPARATIVA DOS RESULTADOS ..... 36
**5. CONCLUSÕES ..... 38**
**REFERÊNCIAS ..... 39**

## 1. INTRODUÇÃO

A falta de um processo de qualidade na etapa de desenvolvimento do produto afeta diversas empresas, existem algumas profissões que tentam contornar/corrigir isso, mas a profissão que vem ganhando destaque é o *Quality Assurance* (QA). Esses profissionais trabalham “por meio de um mapeamento das tarefas operacionais, visando a eliminação do desperdício por meio da redução de atividades redundantes.” (KALENDAE, 2019, p.1) Entende-se então que este profissional tem como objetivo garantir a qualidade do processo, tendo como principais atividades documentar seus processos a fim de criar um histórico e gerar provas do que foi feito. Outro aspecto é de fato o teste, que busca acompanhar a evolução do produto prestando atenção em cada detalhe, explorando vulnerabilidades e falhas, bem como as habilidades e competências dos profissionais que podem variar conforme as necessidades da empresa (Chiavenato, 2013).

A etapa de teste visa garantir a qualidade focando no refinamento do processo, a fim de evitar retrabalhos, custos não planejados, *bugs* etc. Com isso, é possível ter uma melhor experiência para o usuário final, desde o primeiro contato do *login* até o uso da ferramenta em si, tendo assim um produto mais assertivo, que vai agradar ainda mais quem for utilizar.

Em diversas situações podem ocorrer o descaso com a qualidade, isso ocorre quando há troca de eficácia (a qualidade do produto final) por eficiência (o gasto de recursos para alcançar o produto final), trazendo a curto prazo benefícios à empresa, porém impactando na qualidade, o que ocasiona problemas como retrabalho ou feedbacks negativos como comentado anteriormente.

Nesta pesquisa, será analisado o impacto da implementação do processo de qualidade em uma empresa de aplicação web para revelar que se faz necessário entender os reais impactos pela falta destes processos, que repercutem na eficiência e eficácia dos mesmos

O trabalho servirá como documento de estudos para a área da Tecnologia da Informação (TI), desenvolvendo um conteúdo que possa ser consultado por desenvolvedores, gestores e demais interessados. A associação da teoria e prática propostas, sobre os processos de qualidade, busca mostrar os impactos da qualidade nos processos de desenvolvimento das aplicações web, possibilitando empresas mais maduras, produtos de maior qualidade e uma diminuição de problemas em aplicações utilizadas pela sociedade como um todo.

O trabalho apresentará os impactos da implementação dos processos de qualidade, sendo os resultados relevantes principalmente para a empresa analisada bem como para as empresas que ainda não o fazem para que deem a devida importância ao uso e desenvolvimento destes.

A seguir, são apresentados os objetivos que se pretende alcançar neste estudo.

## 1.1. OBJETIVOS

Para o desenvolvimento deste trabalho foram definidos os seguintes objetivos: geral e específicos.

### 1.1.1. Objetivo Geral

Este trabalho tem o objetivo geral mostrar os impactos da implementação de um processo de qualidade de aplicações web em uma empresa de desenvolvimento.

### 1.1.2. Objetivos Específicos

Para alcançar o objetivo geral proposto neste trabalho, necessário alcançar os seguintes objetivos específicos:

- **Elencar** quais as metodologias utilizadas na literatura que estão envolvidas no processo de formação de habilidades e competências do QA;
- **Identificar e relacionar** os métodos de qualidade utilizados na empresa. que impactam no profissional de QA com a teoria;
- **Reportar** quais os impactos do uso de métodos de qualidade, visando transformar esse trabalho em um estudo teórico-prático útil a quaisquer interessados na temática, sendo os resultados e métodos utilizados servir de base para outras pesquisas em outras empresas
- **Apresentar** estatísticas com os resultados obtidos com a implementação

## 2. REVISÃO DA LITERATURA

Nesta sessão traremos uma fundamentação teórica, metodológica (materiais e métodos) sobre processo de qualidade, aplicação web, trazendo também estudos de caso.

### 2.1. TECNOLOGIA DA INFORMAÇÃO

A tecnologia da informação (TI) é uma área amplamente fragmentada e atuando em toda parte na atualidade, desde a vida cotidiana das pessoas, na utilização da internet, celulares, computadores etc. até as grandes indústrias que utilizam para automação. Por essa área ser tão vasta é difícil caracterizá-la, porém pode-se dizer que o a TI é “utilizada para fazer o tratamento da informação, auxiliando o usuário a alcançar um determinado objetivo.”(ALECRIM, 2019)

Para este trabalho abordaremos mais a fundo a parte da qualidade na área de TI, podemos falar de um dos processos que compõem esse processo maior, a testagem.

Assim, podemos começar citando sobre a origem do termo *bug* (termo em inglês para inseto) que ganhou, na área da informática, o significado para um erro de programação. Sabadoti (2010) conta um pouco sobre a história por trás do termo, esse evento ocorreu com os engenheiros que trabalhavam com Mark I em 1947, eles encontraram um inseto nos circuitos que estava causando um erro nos cálculos da máquina. Assim foi registrado como o primeiro *bug* encontrado. Por conta disso começou-se a pensar em processos para evitar que esse tipo de problema voltasse a acontecer.

Em 1979 os processos de testes começaram a ser mais refinados com a publicação de “*The art of software testing*” do autor Glenford Myers. O livro possui diversas informações sobre testes de software. Este livro é utilizado até os dias de hoje, onde muitos o consideram como o marco inicial de Testes de Software.

Sabadoti (2010) ainda informa que os modelos prescritivos de desenvolvimento de software surgiram na década de 1980. Alguns dos modelos foram o de cascata e espiral. Esses permitiram o início do desenvolvimento de padrões, além de consolidar e difundir o conceito de testes de softwares. Já em 1990 os fornecedores passaram a integrar ferramentas para testes funcionais e de gerenciamento. Alguns anos depois surgiram as primeiras ferramentas para automação de testes de performance para algumas

plataformas, permitindo organizar, manter dados, resultados e apresentar relatórios otimizando tempo, trazendo novas possibilidades e maior retorno financeiro.

No ano de 2002, surgiu no Brasil a Associação Latino-Americana de Teste de Software (ALATS). Esta é uma entidade sem fins lucrativos que foi criada com a finalidade de reunir profissionais das áreas de teste e de qualidade de sistemas com o intuito de servir de apoio à troca constante de informações objetivando a melhoria dos processos de teste, mantendo-se a sua integração como o processo de desenvolvimento de aplicações.

Atualmente podemos encontrar uma grande quantidade de certificações, ferramentas e conteúdo sobre testes de software. Livros, blogs, listas de discussões, são alguns exemplos de conteúdos que podem ser facilmente encontrados.

Em 2006 a *Brazilian Software Testing Qualifications Board* (BSTQB) começa a atuar no Brasil. Assim a *Certified Tester Foundation Level* (CTFL) começa a ser usada como referência, sendo a base das certificações do BSTQB. Nesta se encontram os 7 princípios do teste que é amplamente utilizado como padrão para estudos na área de qualidade.

Estes afirmam que os testes buscam mostrar a presença de defeitos e não, ao contrário do que muitos pensam, a sua ausência, assim os testes reduzem a probabilidade de defeitos não descobertos no software. Mas, não é uma prova de que o software esteja perfeito. Além disso, também lembra que é impossível testar tudo (todas as combinações de entradas e pré-condições), isso não seria viável, exceto em casos triviais.

Também pontua que quanto mais cedo for encontrado o *bug*, menor é o custo gerado para sua correção. Assim, um *bug* em produção demanda mais recursos do que um em fase de desenvolvimento. Além disso, afirma que defeitos se agrupam em um pequeno número de módulos que contém a maioria dos defeitos descobertos ou é responsável pela maioria das falhas operacionais.

Atenta para ter cuidado com o paradoxo do pesticida. Se os mesmos testes foram repetidos várias vezes, esses testes não encontraram novos defeitos, dando a errônea visão de que não há mais defeitos (a aplicação do pesticida impede o surgimento de pragas para sempre). Para detectar novos defeitos, os testes existentes e os dados de teste devem ser alterados frequentemente. Também pondera que os testes dependem do contexto em que está inserido e/ou objetivo a ser alcançado.

## 2.2. PROCESSO DE QUALIDADE

Primeiramente, é necessário entender o que é qualidade, pois trata-se de um elemento complexo e de difícil mensuração, já que é relativa ao objetivo, observador e valor atribuído na análise desta, é “resultante da interpretação de uma ou mais características das coisas ou pessoas.” (DE TOLEDO et al, 2012, p.1) Ou seja, a qualidade de um processo pode ser vista através de características como por exemplo funcionalidade, confiabilidade, desempenho, usabilidade etc.

Porém o aspecto subjetivo da qualidade existe, pois, a mensuração, interpretação, definição das características e as próprias características podem ser subjetivas. Por exemplo, a característica aparência em uma aplicação web tem um valor simbólico.

Lobo (2010, p.19), fala que “[...] a qualidade satisfatória envolve, de forma conjunta, todas as fases do ciclo da qualidade.” Entendendo assim que o processo de qualidade envolve diversos procedimentos, contínua:

> Os contributos dessas várias fases para a qualidade são por vezes identificados separadamente para distinguir, por exemplo, qualidade decorrente da definição das necessidades, qualidade decorrente da concepção do produto, qualidade decorrente da conformidade, qualidade decorrente da assistência ao produto prestada ao longo do seu ciclo de vida. (LOBO, 2010, p.19)

Sendo assim, podemos entender o processo de qualidade como um conjunto de medidas e critérios pensados visando garantir a qualidade final do produto. Esses processos variam de qual tarefa tem-se em mente, podem ser aplicadas desde as linhas de produção de uma fábrica até softwares e, no caso do presente trabalho, aplicações web.

Nas aplicações web são voltadas para as características das aplicações, sendo assim são analisadas as seguintes características:

- **Requisitos**: necessidades definidas pelo cliente junto com o analista.
- **Confiabilidade**: métrica que leva em consideração a disponibilidade e manutenção de um sistema, o quanto ele fica disponível e o quanto ele é renovado para as necessidades do cliente.
- **Funcionalidade**: Desenvolvimento das necessidades que foram definidas na etapa de requisitos.

- **Desempenho**: leva consideração o tempo de processamento de determinada funcionalidade.
- **Usabilidade**: Estudo para facilidade com que as pessoas podem empregar uma ferramenta ou objeto a fim de realizar.

## 2.3. SOFTWARE

Primeiramente se faz necessário o entendimento básico do que é um *Software*. De acordo com Gogoni (2019) *softwares* são conjuntos de instruções que devem ser seguidas e executadas por um mecanismo de modo a ditar o que um dispositivo deve fazer. São definidos como *softwares*: programas, apps, *scripts*, macros e instruções de código embarcado diretamente (*firmware*).

Os *softwares* são divididos em três principais categorias:

- De programação: São as ferramentas usadas para desenvolver outros softwares. Exemplos são VS code, IntelliJ, Spyder etc.
- De sistema: São os encarregados de fazer a comunicação entre a parte de hardware com os demais softwares e também é responsável pela maior parte da interface visual, facilitando a comunicação entre o dispositivo e o usuário. Exemplos são Windows, MacOS, Linux, Android, etc.
- De aplicação: Um *software* de aplicação tem como função executar tarefas das mais diversas buscando algum objetivo específico diretamente para o usuário. Exemplos são navegadores, editores de texto, jogos, *streaming* de vídeo etc.

Para que os *softwares* funcionem corretamente e executem seus objetivos de maneira eficaz, faz-se necessário procedimentos de análise de qualidade, estes serão apresentados na subseção a seguir.

### 2.3.1. Qualidade de software

Para o entendimento sobre processos de qualidade na aplicação web, é importante verificar anteriormente a literatura sobre qualidade de *software* que se pode utilizar como base para as aplicações web, sendo as suas naturezas parecidas.

Rocha (2001 apud Gatti e Werneck, 2004, p.1) define qualidade de *software* como “um conjunto de características que devem ser alcançadas em um determinado grau para que o produto atenda às necessidades de seus usuários.”

A NBR ISO/IEC 9126-1 traz um conjunto de requisitos divididos em qualidade interna, qualidade externa e qualidade de uso. Define qualidade interna como a todas as características do software do ponto de vista interno, que podem ser analisados na implementação do código, revisão e/ou teste. A qualidade externa é do ponto de vista externo, podendo ser medida em um ambiente simulado, utilizando dados e métricas externas (NBR ISO/IEC 9126-1, 2003). Para isso a norma traz um modelo de algumas características que devem ser mensuradas, resumidas abaixo:

- **Funcionalidade**: A capacidade de prover funções que atendam às necessidades explícitas e implícitas, divide-se ainda em adequação, acurácia, interoperabilidade e segurança de acesso;
- **Confiabilidade**: A capacidade de manter um nível de desempenho, traz ainda dentro disto maturidade, tolerância a falhas e recuperabilidade;
- **Usabilidade**: A capacidade de ser compreendido, aprendido, operado e atraente ao usuário, destrincha em inteligibilidade, apreensibilidade, operacionalidade e atratividade;
- **Eficiência**: Capacidade de apresentar desempenho apropriado, relativo à quantidade de recursos usados, segmenta em comportamento em relação ao tempo e utilização de recursos;
- **Manutenibilidade**: Capacidade de ser modificado, com intuito de fazer correções, melhorias ou adaptações devido a mudanças que ocorrem, separa em analisabilidade, modificabilidade, estabilidade e testabilidade;
- **Portabilidade**: Capacidade de ser migrado de um ambiente para outro caso necessário, desmembrar em adaptabilidade, capacidade para ser instalado, coexistência e capacidade para substituir;
- **Conformidade**: capacidade de todos os pontos acima estarem dentro das normas, convenções, regulamentações etc.

A qualidade de uso é definida como “a visão da qualidade sob a perspectiva do usuário” (NBR ISO/IEC 9126-1, 2003, p.11). Assim, busca verificar o nível de qualidade no ambiente do usuário (tendo em vista que existem distinções entre necessidades e

capacidades de diferentes usuários, como por exemplo o hardware de cada usuário. Além disso, avalia somente aqueles atributos do software que são usados nas tarefas dos usuários. E por fim traz o contraste que pode ocorrer algumas vezes, atributos de software especificados por um usuário final, inicialmente, não mais atendem ao que o usuário deseja quando o produto é usado (NBR ISO/IEC 9126-1, 2003). Para isso a norma NBR ISO/IEC 9126-1 (2003) traz um modelo de algumas características que devem ser mensuradas, resumidas abaixo:

- **Eficácia**: a capacidade de proporcionar que usuários atinjam metas especificadas com exatidão e em sua totalidade.
- **Produtividade**: a capacidade de permitir que os usuários empreguem quantidade apropriada de recursos em relação à eficácia obtida.
- **Segurança**: a capacidade de garantir níveis aceitáveis de riscos e danos a usuário e software.
- **Satisfação**: Capacidade de satisfazer usuários.

## 2.4. APLICAÇÃO WEB

Bianchini (2008) apresenta diversas classificações acerca da definição de aplicações web[1], trazidas por Pressman (2005), Conallen (2002), Araujo (2001), Powell et al (1998). A forma mais abrangente de classificar as aplicações web é defini-las como todo tipo de aplicação, desde sites de notícias a plataformas de streaming (Netflix, Youtube, Twitch, etc.), inseridas no ambiente web. Bianchini (2008) levanta as formas mais específicas de classificar essas aplicações como:

- **Site estático**: tipo de aplicação mais simples, apresenta conteúdos estáticos, geralmente por meio de HTML. Edições e alterações só podem ser realizadas diretamente no código fonte. É possível navegar entre páginas por meio de elos.
- **Site estático com formulários de entrada**: são idênticos aos sites estáticos, porém provém a possibilidade de implementar formulários de entrada para captação de dados dos usuários. Exemplos: questionários, comentários, pesquisa de cliente, etc.

[1] Web ou World Wide Web: é um sistema de documentos em hipermídia que são interligados e executados na Internet.

- **Site com acesso a dados dinâmicos**: “o usuário por meio de consultas pode ter acesso a bases de dados. As respostas das consultas são geradas dinamicamente e geralmente apresentadas no formato de documento HTML, DHTML ou XML” (BIANCHINI, 2008, p.7)
- **Site criado dinamicamente:** são sites desenvolvidos de forma com que permitam alterações no conteúdo sem necessidade de ter acesso ao código fonte, utilizando sistemas de gerenciamento de conteúdo. Como por exemplo: redes sociais, blogs e etc.
- **Aplicação web:** Traz os atributos anteriores e acrescenta a lógica de negócios, promovendo interação com as bases de dados. Assim, possibilitam a mesma funcionalidade de software[2] tradicional no contexto web.

[2] Software: são programas desenvolvidos para executar tarefas em uma máquina, ele é a parte lógica, um conjunto de instruções para a parte física da máquina. É desenvolvido por meio de uma linguagem de programação.

Abaixo apresenta-se uma tabela ilustrando as classificações segundo alguns autores

Tabela 1 **- Relação categorias de aplicação web**

| **CLASSIFICAÇÃO** | **CATEGORIAS** | | | | |
|---|---|---|---|---|---|
| **Powell et al. (1998)** | **Sites Web estáticos** | **Sites web estáticos com formulários de entrada de dados** | **Sites web com acesso a dados dinâmico** | **Sites Web criados dinamicamente** | **Aplicações Web** |
| **Araújo (2001)** | **Páginas de informações** | | **Aplicações Web centradas em banco de dados** | | **Aplicações Web que apoiam a realização de transação de negócios** |
| **Conallen (2002)** | **Sites Web** | | **Aplicações Web** | | |
| **Pressman (2005)** | **Aplicações Web** | | | | |

**Fonte:** Bianchini (2008, p.7)

Finkelstein et al. (2002, apud BIANCHINI, 2008), amplia a definição definindo aplicações web como “[...] uma aplicação que é executada na Web, envolve aspectos de hipermídia, como hipertextos e multimídia, em combinação com a lógica das aplicações convencionais.”

Segundo Gatti e Werneck (2004) uma aplicação Web com tantos recursos pode ser comparada a um software como outro qualquer. Como exemplo, temos que uma Aplicação de *Software* baseada na Web (às vezes implementada como uma Intranet ou Extranet) pode ser um sistema de controle de pagamentos, de ensino à distância, de geração de gráficos e de estatísticas, entre outros. Percebemos, então, que a Web é um ambiente complexo, além de dinâmico, o que dificulta a avaliação de produtos de software voltados para este domínio de aplicação. Uma aplicação Web com tantos recursos pode ser comparada a um *software* como outro qualquer.

Além disso, trazendo uma definição mais simples e atual de Noleto (2020) aplicações Web podem ser soluções executados diretamente no[3]'navegador', ou seja, estão sendo processadas em um servidor, assim não necessitam de ser instaladas pelo usuário final.

Para este trabalho o entendimento de aplicação web será dado como softwares aplicados na Web, ou seja, são aplicações desenvolvidas na Web com as funcionalidades dos softwares, além disso utilizam-se das características da web para inserir outras funcionalidades, por exemplo o uso de hipermídia.

### 2.4.1. Como funcionam as aplicações web

A Aplicação web funciona a partir de requisições do usuário, que são recebidas pelo servidor que processa e responde. Isso é possível pelo uso de protocolos como o HTTP (*Hypertext Transfer Protocol*) é um protocolo cliente-servidor, que permite a troca de dados na web entre a aplicação e o usuário, servindo como um padrão de troca de dados. Além disso, existem também o FTP (para transmissão de arquivos), SMTP (envio de mensagens para um servidor de e-mail), POP e IMAP (acesso a mensagens de e-mail eletrônico).

As vantagens do uso de aplicações Web, são que para o usuário final tudo fica mais simples e acessível, já que não há necessidade de instalação, a navegação é simplificada e intuitiva.

#### *2.4.1.1 Qualidade em aplicações web*

Gatti e Werneck (2004) apresentam diversos fatores para mensurar a qualidade de aplicações web. Estes selecionam as principais como:

- Visibilidade: facilidade de encontrar, acessar e/ou de efetuar downloads;
- Inteligibilidade: capacidade dos usuários absorverem e entender o conteúdo;
- Credibilidade: confiança que os usuários detêm sobre o conteúdo, sendo preciso e correto;

[3] Navegador: também conhecido como browser, são softwares desenvolvidos para acessarem o conteúdo na Web, são janelas para o mundo da internet.

- Atratividade: nível em que a aplicação consegue atrair a atenção de um usuário;
- Diferenciação: demonstra o diferencial da empresa, tendo relação com a imagem dela.

Além disso, deve-se levar em consideração a visão do usuário final, o objetivo da aplicação, aspectos técnicos como desenvolvimento baseado em componentes, segurança e padrões de Internet.

## 2.5. IMPORTÂNCIA DOS TESTES

se falar de qualidade dentro da área de TI é indispensável pensar na importância dos testes. A adoção de técnicas de teste contínuas e que permitam identificar o maior número de falhas possíveis é a chave para um projeto bem-sucedido ao evitar que esses problemas cheguem ao cliente.

Monitora (2021) elenca alguns pontos importantes dos testes, apresentados pelo autor com olhar para as aplicações web:

- Detecção e correção de problemas de performance: um desses testes pode verificar como a aplicação se comporta com um grande volume de usuários acessando continuamente.
- Otimização da gestão dos recursos: neste caso pode-se testar como estão sendo usados os recursos da empresa na ferramenta.
- Integração com outros sistemas: visam perceber se existem problemas de desempenho com a integração com outras aplicações/*softwares*/bases de dados.

Além disso, a testagem contínua permite o ajuste do projeto antes de chegar ao cliente. Monitora (2021) diz que quando esses erros surgem quando a aplicação já está em uso, é possível que uma parte do trabalho tenha que ser refeita e se houver a necessidade de suspensão da aplicação para as manutenções pode trazer ainda prejuízos diretamente ao cliente.

Monitora (2021) divide as técnicas de teste entre:

- Funcionais: são verificações nas funcionalidades, essas buscam erros de desempenho no decorrer da atividade de funções específicas;
- Não-funcionais: analisam características como a usabilidade, velocidade entre outros aspectos independente das ações executadas pelo sistema.

Além disso, os testes podem ser manuais, feitos diretamente por um profissional capacitado para executá-los, ou automatizados, através de *softwares* de testagem.

Um bom planejamento de TI é importante para acompanhar a execução do projeto e para mapear a realização dos testes, definindo quais serão feitos e em que etapa. Portanto, Monitora (2021) define alguns aspectos que precisam ser submetidos a testes, esses são: Funcionalidade; Confiabilidade; Usabilidade; Eficiência; Manutenibilidade; Portabilidade.

Tendo isso em vista, deve-se ter em mente que existem diversos testes para pontos ou necessidades específicas das fases de um projeto, a seguir serão apresentados alguns encontrados na literatura.

### 2.5.1. Teste de Caixa Branca

Esse teste, também chamado teste estrutural, segundo Monitora (2021) utiliza o código-fonte do sistema para analisar os componentes. Verificando o fluxo de dados, a qualidade da estruturação deste código, segurança, complexidade da manutenção, entre outros. De uma forma simplificada pode-se descrever o teste como sendo: “utilizado para testar o código. Garante a análise de todos os caminhos independentes de cada módulo, programa ou método” (SOARES, 2020, não paginado)

### 2.5.2. Teste de Caixa Preta

Segundo Monitora (2021), o teste, também chamado de teste funcional, visa verificar se a ferramenta cumpre as funções devidas, sendo as mesmas previamente definidas na etapa de especificação de requisitos. É importante frisar que nesse teste “É ignorado a estrutura interna do código, detalhes de implementação ou cenários de execução internos no software, focando apenas nos requisitos funcionais do sistema” (SOARES, 2020, não paginado)

### 2.5.3. Teste de Regressão

Soares (2020) diz que a proposta do teste de regressão é verificar o sistema sempre que algo for alterado, verificando assim evitando a recorrência de erros de versões anteriores de um aplicativo ou um *software*.

### 2.5.4. Teste de Carga

Também chamado de Teste de Volume, Segundo Soares (2020) analisa se o sistema vai funcionar medindo a quantidade máxima de usuários simultâneos. Monitora (2021) também diz que este teste verifica: “quanto ele suporta de tráfego, de informações, entre outros, até aparecerem travamentos ou bugs.”

### 2.5.5. Testes de Segurança

Segundo Soares (2020) nestes testes são utilizados diversos tipos de perfis e permissões, verificando até onde essas têm acesso às informações e funcionalidades do sistema.

### 2.5.6. Teste de Usabilidade

Segundo Monitora (2021), este teste, também chamado de teste de aceitação, é realizado diretamente por um grupo de usuários, buscando verificar o funcionamento da ferramenta, avaliando se o produto satisfaz as necessidades.

É importante ter em mente que não é necessário a implementação de todos os testes, porém uma estrutura de testagem é essencial para o desenvolvimento de qualquer projeto que preza pelo mínimo de qualidade.

## 2.6. SCRUM

O Scrum é um modelo de estrutura adaptável, desenvolvido para organizar demandas e executar tarefas, buscando permitir agilidade e alta qualidade nas tarefas realizadas.

Sydle (2020) aprofunda pontuando princípios que sintetizam o Scrum, como:

- Foco na satisfação do cliente
- Atenção contínua à excelência
- Designs auto-organizáveis
- Adaptabilidade

Além disso Sydle (2020) também elenca 3 pilares:

- Transparência: todos os envolvidos estão cientes sobre todas as etapas, de forma clara e definida.
- Inspeção: acompanhamento de todas as etapas. Visualizando as execuções de cada pessoa, permitindo visualizar impactos de cada um no todo.
- Adaptação: adaptação contínua de acordo com o escopo do projeto às necessidades. Assim, o projeto pode ser concluído com resultados que sejam mais produtivos.

Outra característica do Scrum, são seus ciclos, chamados de *sprints*. Ou seja, são períodos pré-determinados, geralmente duas semanas, dentro do qual a equipe completa conjuntos de tarefas do *Backlog*.

De acordo com Sydle (2020) Scrum utiliza ferramentas que garantem a execução do projeto, esses são:

- *Product Backlog*: são uma lista de requisitos, por meio do mapeamento de histórias, com documentação de processos para entendimento compartilhado do todo deve fazer parte da funcionalidade.
- *Sprint Backlog*: lista de requisitos para o time de desenvolvimento, criada a partir de prioridades da funcionalidade.
- Incremento: Uma funcionalidade que surge como resultado do trabalho de cada Sprint. Devendo ser capaz de gerar valor para seu cliente.
- *Release Burndown*: “Um gráfico que acompanha o progresso do time em comparação ao que foi planejado. [...] medida pode ser avaliada em dias, pontuação de histórias, etc.” (SYDLE, 2020, não paginado)

## 2.7. QUALITY ASSURANCE (QA)

*Quality Assurance* (QA) ou traduzindo pro literal “Garantia de Qualidade”, seria o profissional ou uma equipe com objetivo de garantir a qualidade no desenvolvimento de

um projeto. Kalendae (2019) diz que o QA é principalmente envolvido na checagem do cumprimento de certos critérios e métodos ao longo dos processos operacionais, com o objetivo de identificar desvios que possam comprometer a qualidade final do serviço, enquanto busca levantar oportunidades de melhoria direta no projeto a um aperfeiçoamento dos processos.

A atuação do profissional de QA busca identificar ganhos operacionais com base nas documentações e no diálogo com os demais setores, esses profissionais detêm conhecimentos sobre as atividades e metodologias da área de atuação da empresa.

Kalendae (2019) continua falando sobre o QA no setor de TI, diz que esse apresenta estratégias de regulação da qualidade de forma sistemática, buscando evitar problemas, “criando mecanismos para identificá-los e tratá-los com eficiência” (KALENDAE, 2019, não paginado).

Assim, pode-se entender que o *Quality Assurance* tem um papel fundamental ao garantir no desenvolvimento de projetos o nível de qualidade planejado, proporcionando assim produtos/serviços com a qualidade esperada pelos clientes e com menores custos e horas de retrabalho.

## 3. PROCEDIMENTOS METODOLÓGICOS

A pesquisa utilizará o método hipotético-dedutivo para análise do objeto de estudo e desenvolvimento de hipóteses para prognósticos. Segundo Prodanov e Freitas (2013) o método se inicia com a formulação de um problema, buscando informações e conhecimentos para auxiliar o pesquisador. Após, passa para a fase de observação, onde é observado determinado aspecto do universo. Então a formulação de hipóteses com base no que foi observado, desenvolvimento de prognósticos e teste deste por meio de observações mais detalhadas. Pode haver reformulações das hipóteses dependendo dos resultados.

Pretende-se utilizar as métricas de comparação em cima das entregas antes e depois do processo de qualidade, analisando retrabalho, bugs e feedbacks. Esse processo visa levantar dados para construir hipóteses e testá-las. Todos os dados para essas métricas serão levantados em parceria com uma empresa de desenvolvimento, que já tem esses dados estruturados, não tendo nenhum contato com funcionários ou clientes diretamente.

- ***Bugs***[4]: quantidade de bugs apresentada por funcionalidade;
- **Retrabalho**: o tempo a mais que a pessoa levou para corrigir os bugs do seu trabalho.

Além disso, pretende-se fazer um estudo de caso da empresa de desenvolvimento web, que atua em soluções governamentais, focadas nas áreas da saúde, educação e segurança, porém, os dados analisados nesta pesquisa serão focados na área de desenvolvimento educacional. Sendo estudo de caso uma abordagem metodológica de "investigação especialmente adequada quando procuramos compreender, explorar ou descrever acontecimentos e contextos complexos [...]"(BOGDAN; BILKEN, 1994:150, apud ARAÚJO et al., 2008, p.13). O estudo de caso é uma metodologia complexa que auxilia o pesquisador a entender complexos fenômenos sociais, no caso do presente trabalho esta metodologia auxiliará a visualizar o comportamento da empresa visualizando na prática o processo e as ocorrências do meio.

Assim, foi estruturado um roteiro de observação para a coleta dos dados, utilizado para registrar aquilo "que o investigador ouve, vê, experiencia e pensa no decurso da

[4] Erro que impede o uso da ferramenta desenvolvida.

recolha e refletindo sobre os dados [...]" (BOGDAN; BILKEN, 1994:150, apud ARAÚJO et al., 2008, p.14). Por meio deste serão levantados dados que serão analisados com as métricas e pesquisas visando uma melhor resposta da questão de pesquisa.

Yin (2015) cita que para a boa execução da metodologia deve-se ter em mente a formulação de boas questões, possibilitando a interpretação objetiva dessas. Ser um bom 'ouvinte 'e ser imparcial, não ficar atrelado a hipóteses e pressupostos, receber toda a informação de forma total, sem preconceitos ou ideologias. Ser adaptável, dependendo das circunstâncias, as mudanças podem favorecer a pesquisa. E por fim é necessário que se tenha o entendimento claro do assunto que está sendo analisado.

Yin (2015) informa que o estudo de caso pode trabalhar com diversas fontes de evidência, desde documentos, registros em arquivo, entrevistas, observações diretas, observação participante e artefatos físicos. Para esta pesquisa estaremos analisando apenas documentos e registros em arquivos, que Yin (2015) cita as características como:

- Estável: podendo ser conferida e/ou analisada sempre que necessário.
- Discreta: não tinha como foco o estudo de caso, sendo criada para outro fim específico.
- Exata, precisa e geralmente quantitativa: contém nomes, referências, valores e detalhes exatos.
- Ampla cobertura: pode cobrir longos períodos, possibilitando análise de longo prazo.

Nesta pesquisa será apresentado o estudo do processo envolvendo profissionais da área de qualidade de aplicação web, por meio da análise dos relatórios apresentados por esses funcionários. Isso possibilitará identificar o contexto da empresa analisando os pontos fortes e fracos do processo, reportar esses problemas e propor soluções.

Para isso serão analisados os dados coletados seguindo um roteiro, a fim determinar o que será pesquisado, elaborado e aplicado dentro do ambiente de trabalho. Dito isso, os principais dados recolhidos serão:

**Bugs:** Quantidade de reportes de defeitos feitos pelos QAs e pelo cliente

**Horas de retrabalho:** Quantidade de horas de retrabalho que foram exigidas para correção dos defeitos.

**Custo pela implementação e funcionamento da equipe de QA**: Levantamento do quanto custa para empresa e quanto ela ganha, sem levar em consideração o benefício da maturação do produto.

**Custo por bug interno e por bug externo**: A fim de mostrar quanto custa para empresa um bug na linha de produção e quanto custa na mão do cliente.

Esses vão ser algumas das métricas que serão levantadas conforme a evolução do trabalho e pesquisa, dito isso podem ser alteradas de acordo com o andamento do projeto.

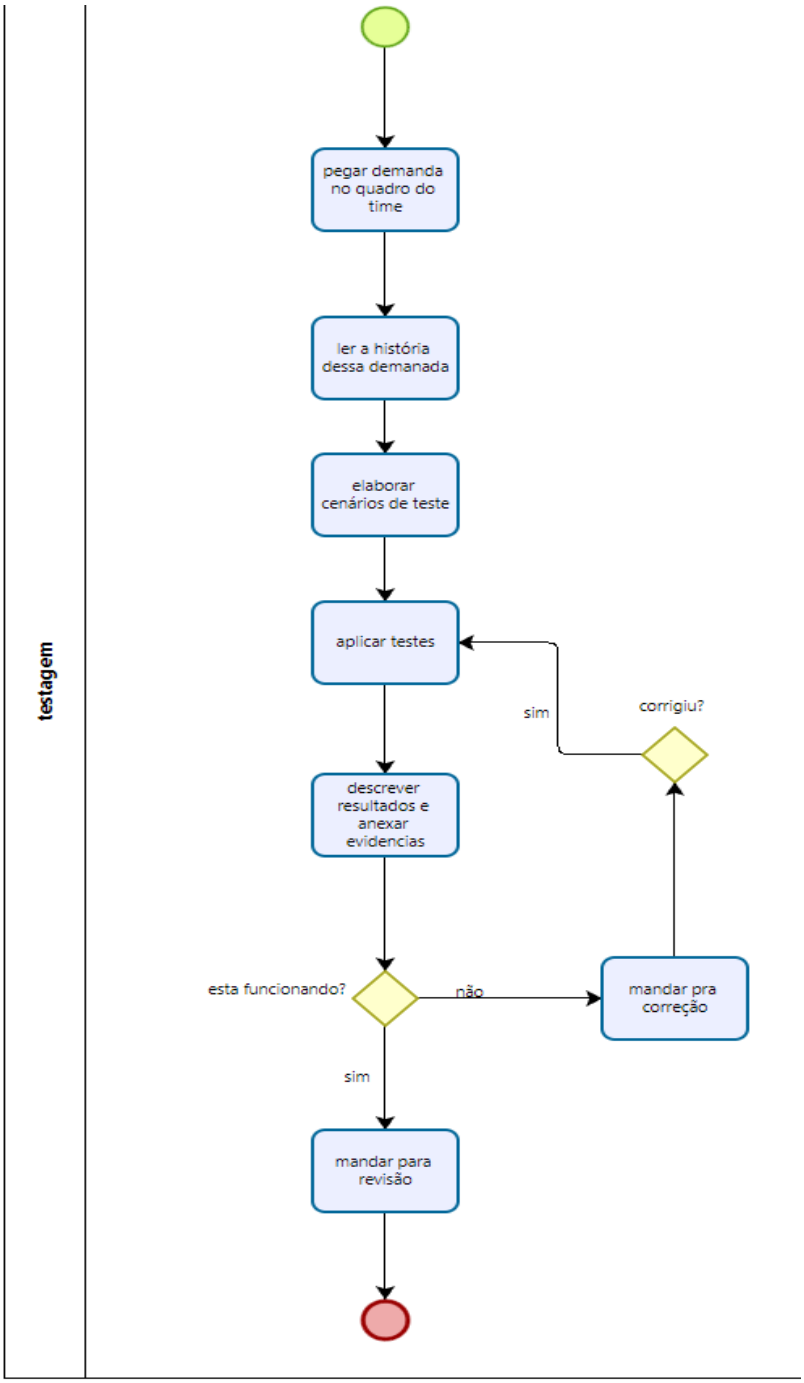
## 4. ANÁLISE E DISCUSSÃO DOS RESULTADOS

Nesta seção serão apresentados os resultados obtidos aplicando as metodologias de qualidade nos processos de desenvolvimento de aplicação web da empresa analisada. Os dados foram coletados em um período de 1 ano, sendo dados coletados 6 meses antes do QA e 6 meses depois do QA. os aspectos analisados serão *bug* internos e externos e custos de implementação.

Além disso, para a análise do período foram utilizados os dados referentes a uma metodologia ágil implementada pela empresa chamada SCRUM. Essa metodologia funciona por meio de Sprints, que foram definidos como prazos de 15 dias. Ao final desse prazo é planejada uma nova Sprint, analisando o trabalho realizado na anterior, quais as pendências que ficaram sendo transferidas para a Sprint seguinte, e são elaborados novos objetivos.

Esse processo de análise funciona em um ciclo apresentando a seguir:

Figura 1 - Fluxograma de um teste




Fonte: dados da pesquisa.

Como pode-se observar na figura 1, o teste é composto por 6 etapas principais. Porém antes há uma etapa que não está literalmente no processo de testagem, mas depende da mesma. A etapa consiste em entender a demanda e/ou elaborar a tarefa. Ela é desenvolvida pelo analista de negócio junto com clientes que definem qual a próxima

funcionalidade da aplicação. Assim, é criado uma estória[5] e após são divididas em pequenas tarefas para desenvolvimento, ficando disponível no quadro para ser executada de acordo com a etapa que ela se encontra, seja para desenvolvimento ou para teste.

**Etapas de testagem:**

A primeira etapa da testagem consiste em selecionar a tarefa demandada no quadro. Essa divisão permite que mais de um QA trabalhe em paralelo na mesma estória.

Após, é necessário entender e compreender toda a estória, e não apenas a funcionalidade, para ter ideia da dimensão que ela atua e o que pode afetar na aplicação como um todo.

Depois de ter o entendimento geral da funcionalidade e de sua dimensão, passa-se a etapa de elaboração dos casos de teste, baseada totalmente na estória, regra de negócio e nos protótipos desenvolvidos. Nesta etapa é comum passar despercebidos alguns cenários[6] por ainda estar na etapa de elaboração, porém na aplicação percebe-se esses deslizes.

Na etapa de aplicação dos testes elaborados, usou-se como apoio as regras de negócios e os casos de teste, como dito anteriormente, sempre acaba passando algum cenário quando se trata de uma funcionalidade mais complexa**.** Os testes são feitos em um ambiente separado do cliente, para que não haja impacto na experiência do usuário**.**

Conforme é testado, gera-se evidências do que foi feito com o intuito de facilitar caso surjam problemas futuramente. Assim, é possível mapear o problema, seja por algum erro feito na hora do teste, desde ambiente, versão da aplicação, versão do navegador, cenário mal interpretado ou até mesmo código.

Assim é verificado se está de acordo com as regras de negócios, se sim o protótipo é liberado para uso do cliente final, caso contrário, passa por uma revisão do desenvolvedor.

Nas subseções 4.1 e 4.2 serão apresentados, respectivamente, os resultados obtidos no processo de desenvolvimento sem a implementação e com a implementação de um processo de qualidade.

---

[5] O termo "estória" refere-se às características da funcionalidade, com isso, queremos dizer que ela vai explicar detalhe por detalhe, desde o porquê ela foi criada até o que ela precisa ter. Ela é elaborada com o analista de negócio junto ao cliente

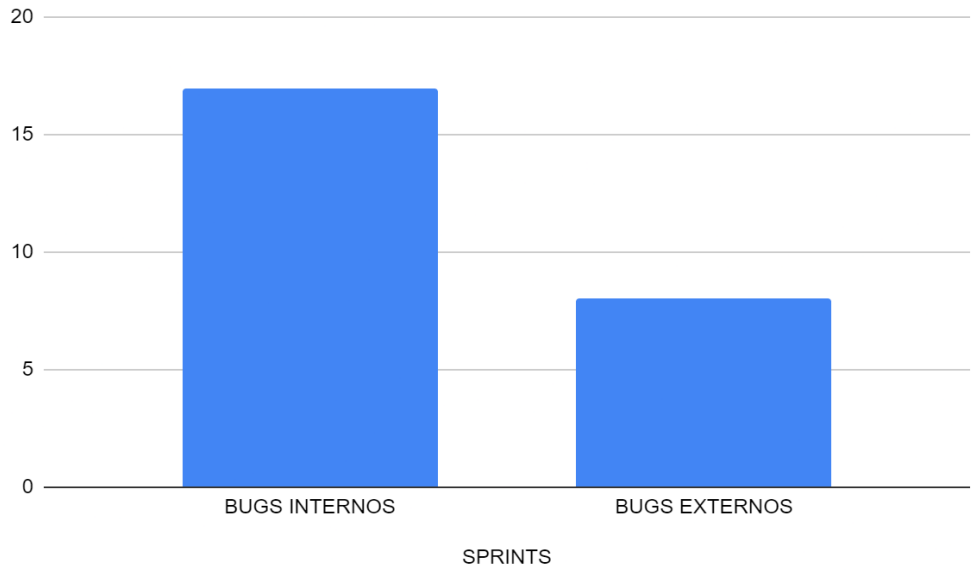
[6] Um cenário de teste é uma descrição de um objetivo que o usuário pode encontrar ao utilizar o programa. Um exemplo seria “Testar se um usuário consegue deslogar do programa ao fechá-lo”. Tipicamente, um cenário de teste vai precisar de diferentes tipos de testes para garantir que o objetivo tenha sido bem testado.

## 4.1. DESENVOLVIMENTO SEM DEFINIÇÃO DE PROCESSO DE QUALIDADE

Nesta seção é apresentado o teste unitário, sendo analisado cada componente individualmente. Esses testes foram realizados pela equipe de desenvolvimento, que além de implementarem a funcionalidade, tinham como responsabilidade testá-las também, porém, não é recomendado a mesma pessoa ficar encarregada de criar e analisar algo, pois para ela o que ela fez está correto e isso pode causar problemas no futuro. Isso pode impactar diretamente no produto final, tendo potencial de trazer um produto com menor qualidade e mais custoso de se manter.

Começando a análise dos dados anteriores a implementação do processo de QA, foram detectados 17 *bugs* internos e 8 *bugs* externos, em um período de 11 *Sprints*, como mostra o gráfico abaixo. Os *bugs* internos são os encontrados pela equipe de desenvolvimento, gerando custos menores para o projeto final, já que são solucionados antes da entrega final, enquanto os *bugs* externos já são encontrados pelos próprios clientes e os usuários desses, assim gerando maiores gastos além de um feedback negativo do cliente. O gráfico 1 apresenta o total de *bugs* internos e externos no projeto.

Gráfico 1 - Bugs encontrado no projeto antes de um processo de qualidade




Fonte: dados da pesquisa

Para solucionar os bugs internos foram totalizadas 55,9 horas, tendo uma média de 5,1 horas por *Sprint*. Além disso, foi feita uma média de *bugs* por hora que revelou sendo 2,2 horas para resolução de cada *bug* interno.

Os testes feitos antigamente pelo time no geral, eram muito simplórios, faziam apenas testes objetivos(comuns) e não levando em consideração aos caminhos mais incomuns, não se atentando a detalhes e experiência do usuário, o que pode interferir na usabilidade, trazendo mais falhas conforme o uso ou não trazendo uma experiência mais fluida, tornando a experiência com a ferramenta chata e desinteressante. Isso pode impactar diretamente no rendimento do usuário, seja momento de lazer ou até mesmo a trabalho, optando por outras soluções, isso dá margem para entrada de novos concorrentes ou as soluções consolidadas no mercado.

Na área de qualidade, sempre é priorizada a experiência do usuário, para que ele consiga enxergar valor naquilo que ele está pagando, do contrário ele pode se questionar. A falta de um processo pode resultar na perda de qualidade, visto que foram realizados somente testes unitários, que são considerados bem simples para um contexto da implementação de uma funcionalidade nova.

Como a empresa estava com os processos de desenvolvimento bem definidos, fez sentido investir na qualidade, através de profissionais qualificados de QA e processos de qualidade já estabelecidos pelo mercado. Assim conseguindo agregar ainda mais valor ao

produto final, com menos erros, trazendo uma experiência de usuário melhor e tendo uma disponibilidade maior do serviço.

O processo de desenvolvimento de *softwares* é uma prática que exige muita organização e planejamento. Sendo objetivo principal oferecer um produto de qualidade sem ter gastos exorbitantes em sua criação, é fundamental definir estratégias para otimizar o processo do início ao fim. Entretanto, na prática, isso pode ser um grande desafio no decorrer do projeto.

Kalendae (2019) estabelece que os principais benefícios oriundos da aplicação, são: aumento da satisfação do cliente; redução de custos operacionais; otimização da rotina de trabalho dos profissionais de TI; redução de retrabalhos; eliminação de gargalos; melhoria de previsibilidade aos projetos.

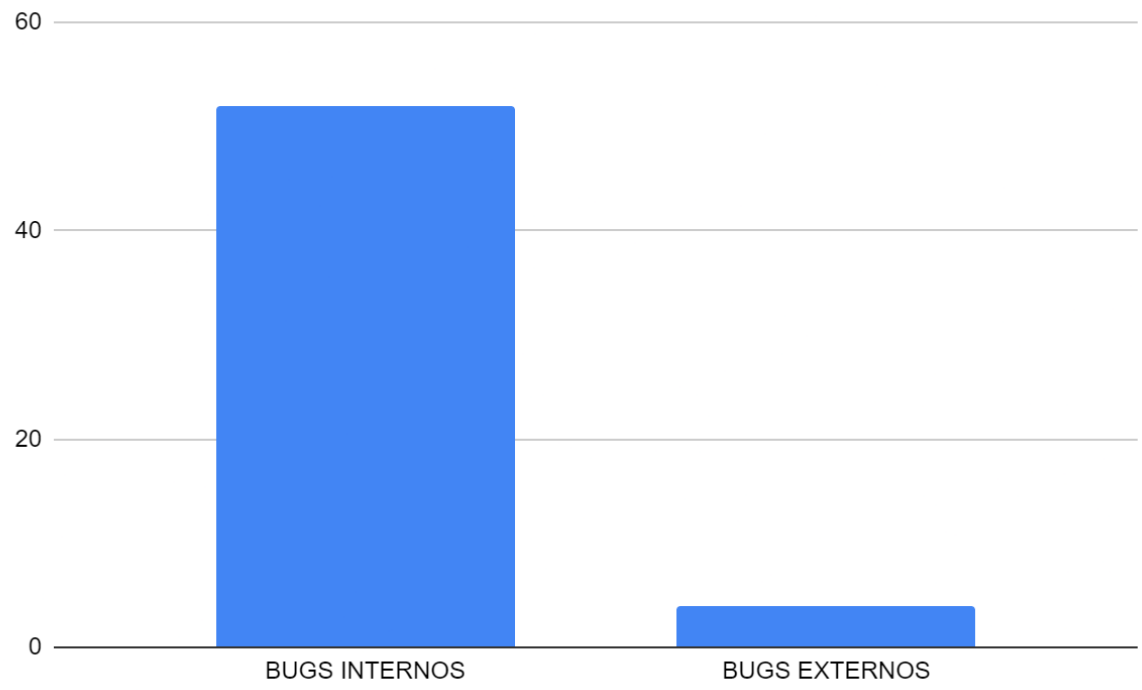
## 4.2. DESENVOLVIMENTO COM A DEFINIÇÃO DE UM PROCESSO DE QUALIDADE

Com a implementação de um processo de qualidade, começou a ser desenvolvido, além do teste unitário, foram implementados os testes de regressão, que analisa a aplicação após a junção de todas as funcionalidades e também teste exploratório, onde você busca testar as funcionalidades de forma a explorar suas vulnerabilidades, buscando encontrar casualmente algum *bug*. Além disso, começou a ser realizado relatórios com intuito de, se necessário, o mapeamento da origem dos *bugs*, esses registros dos testes continham registro da versão testada, capturas de telas, os responsáveis por cada tarefa, o status de cada teste e cenários de teste.

Vale a pena ressaltar que após a implementação do QA todos os testes começaram a ser feitos por eles, diferentemente de quando não se tinha um processo definido pois quem desenvolvia também era responsável pela qualidade.

Analisando os dados com o QA, vemos 52 *bugs* internos e 4 *bugs* externos, como mostra o gráfico abaixo.

Gráfico 2 - Bugs encontrado no projeto depois de um processo de qualidade




Fonte: dados da pesquisa.

Para solucionar os *bugs* internos foram totalizadas 106,8 horas, tendo uma média de 9,7 horas por *Sprint* em correções. Além disso, foi feita uma média de *bugs* por hora que revelou sendo 1,9 horas para resolução de cada *bug* interno.

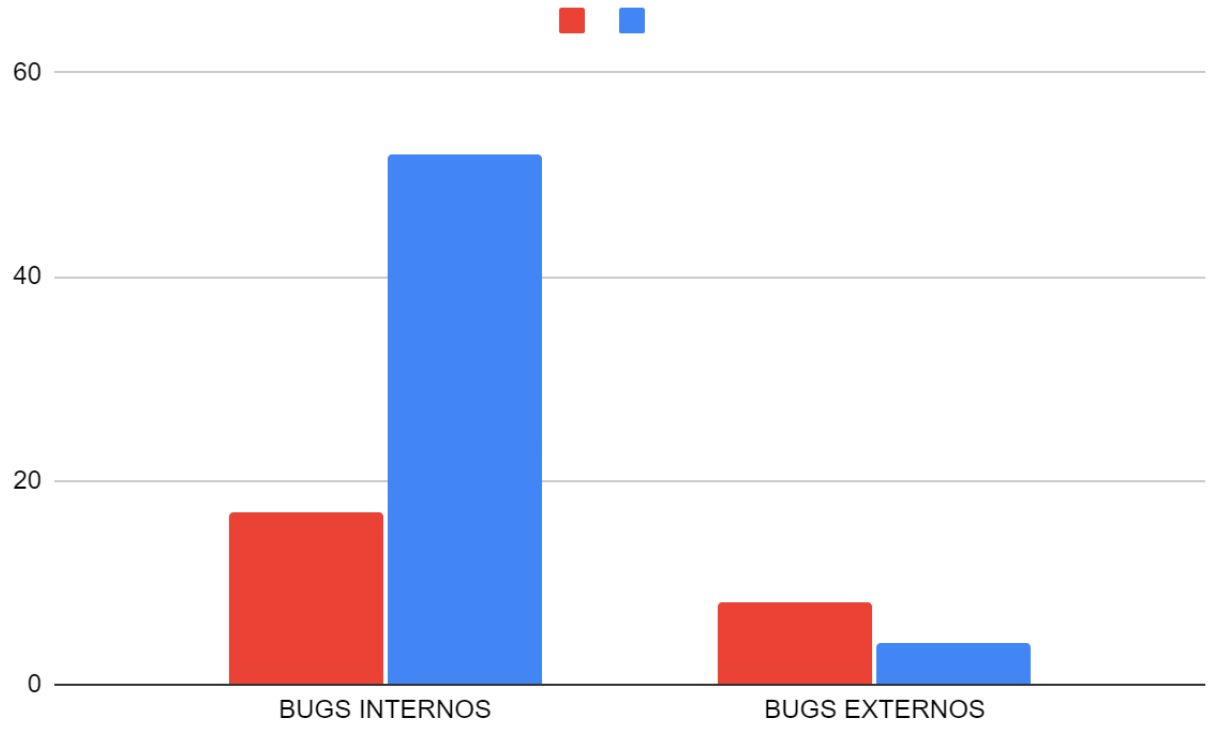
É importante analisarmos também os custos necessários para a implementação do processo de qualidade. Por serem funcionários contratados na modalidade de *home-office* o custo para a empresa é reduzido, tendo em vista que não foram necessários gastos com estrutura física, sendo necessário apenas a aquisição de equipamentos adequados e a contratação dos profissionais de QA, que geram custos mensais. Deve-se lembrar que toda a estrutura para implementação do processo de qualidade já existia, assim não foi necessário desenvolvimento e/ou adaptação da mesma.

Foram gastos em equipamentos cerca de R$12.645,00, na aquisição de notebooks, monitores, teclados, mouses etc. Na questão dos profissionais de QA foram contratados 2 profissionais com a média de custo hora de R$29,00, totalizando um salário mensal de R$10.467,00. Já a média de custo por hora de um dev gira em torno de R$80,00, pegando os bugs internos e multiplicando pelo custo de hora, temos um total de R$4.160,00.

## 4.3. ANÁLISE COMPARATIVA DOS RESULTADOS

Por fim, ao compararmos os dados (gráfico 3) pode-se verificar que houve um grande aumento de *bugs* internos e a diminuição de *bugs* externos em 50%, sendo as colunas vermelhas antes do QA e as azuis depois do QA.

Gráfico 3 - Comparando *Bugs* antes e depois do processo de qualidade



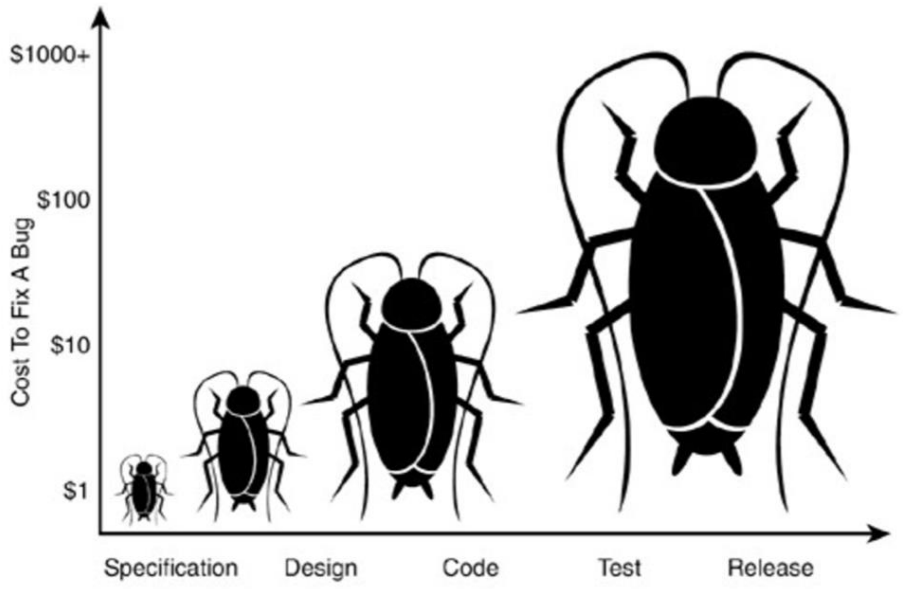

Fonte: dados da pesquisa.

Isso mostra que com o processo de qualidade foram encontrados mais *bugs* antes do lançamento da aplicação. Assim podemos assumir que a aplicação tende a ser finalizada com menos *bugs*. Não descartando a possibilidade de haver mais, pois o objetivo não é encontrar todas as falhas e sim entregar ao cliente esta aplicação, com a menor quantidade de erros possíveis, refinando o processo e trazendo um produto mais assertivo.

Além disso, é possível verificar que o tempo por hora gasto na correção depois do QA diminuiu aproximadamente 14.7%, mesmo havendo mais *bugs* encontrados. Isso pode ocorrer pois esses são encontrados previamente, facilitando a solução, já que o desenvolvedor vai estar com o contexto da funcionalidade “fresco” na memória e assim podendo atuar com mais velocidade de assertividade na correção.

Rodrigues (2017) fala que a relação de tempo e dinheiro dos *bugs* externos pode chegar até 30 vezes, tendo em conta os *bugs* encontrados nas fases de desenvolvimento (*bugs* internos) e os *bugs* encontrados após pelos clientes (*bugs* externos). Além disso, ele também estima que " a cada 1000 linhas de código criadas, 70 novos *bugs* são criados e 15 serão descobertos pelo seu cliente.”(RODRIGUES, 2017, não paginado) Para se evitar esses gastos várias vezes maiores os processos de QA são ideais.

Figura 2 - Relação tempo x custo dos bugs




Fonte: RODRIGUES, 2017, não paginado

## 5. CONCLUSÕES

Através da revisão da literatura foram apresentados os conceitos básicos e elencadas as metodologias encontradas, assim, facilitando o entendimento da área como um todo e das possibilidades de implementação. Além disso, foi identificado as metodologias utilizadas na empresa e quais foram utilizadas depois do profissional do QA relacionando a teoria com a prática de uma empresa de desenvolvimento de software. Por fim foi reportado quais os impactos dos métodos juntos com profissionais de QA analisando os dados da empresa, assim, constatando a melhora. Com o uso de gráficos foi possível apresentar estatisticamente os resultados da pesquisa. Assim este trabalho conseguiu alcançar seus objetivos apresentando os impactos da implementação de um processo de qualidade de aplicações web em uma empresa de software, através de dados quantitativos retirados diretamente da prática, durante um período de tempo razoável (1 ano) com dados estruturados e com análises concretas baseados nos mesmos. Assim, acredita-se que este trabalho tem grande importância para profissionais e estudantes da área, que por meio do mesmo, podem vislumbrar os impactos dos processos de qualidade.

Além disso, o autor tinha internamente a hipótese que a implementação de um processo de qualidade traria resultados positivos e expressivos ao projeto em que fazia parte. Assim, esse trabalho demonstrou, por meio dos dados analisados, os impactos positivos da implementação do processo de qualidade, quando bem planejada, fazendo jus ao investimento da empresa no novo setor.

Por conta da empresa analisada trabalhar apenas com testes manuais, este trabalho aprofundou os mesmos, entendendo como funcionam e quais os impactos.

Porém atualmente os testes automatizados vem crescendo buscando evitar falhas, agilidade e escalabilidade.

Assim é interessante desenvolver um trabalho analisando esses testes automatizados, talvez até comparando com os testes manuais.

## REFERÊNCIAS


ARAÚJO, Cidália et al. **Estudo de Caso**. 2008. Disponível em: http://nelsonreyes.com.br/Estudo%20de%20Caso%20-%20Doutora%20Clara%20Pereira%20Coutinho.pdf. Acesso em: 28 ago. 2021.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9126-1**: NBR ISO/IEC 9126-1. Rio de Janeiro: Associação Brasileira de Normas Técnicas, 2003. 21 p. Disponível em: https://jkolb.com.br/wp-content/uploads/2014/02/NBR-ISO_IEC-9126-1.pdf. Acesso em: 10 set. 2021.

BARRO, Bruna B.. **O Que São Frameworks e Quais os Mais Utilizados**. 2022. Disponível em: https://www.hostinger.com.br/tutoriais/frameworks. Acesso em: 11 out. 2022.

BIANCHINI, Sandro Lopes. **Avaliação de métodos de desenvolvimento de aplicações web**. 2008. 93 f. Dissertação (Mestrado) - Curso de Ciências da Computação e Matemática Computacional, Universidade de São Paulo, São Paulo, 2008.

CHIAVENATO, Idalberto. **Desempenho Humano nas Empresas**: como desenhar o trabalho e conduzir o desempenho. 8. ed. São Paulo: Atlas, 2021. 184 p. Disponível em: https://www.grupogen.com.br/e-book-desempenho-humano-nas-empresas-9786559771202. Acesso em: 05 out. 2022.

DE TOLEDO,.José. C.; AIRES, BORRÁS,.Miguel. Á.; COSER, MERGULHÃO,. R.; HENRIQU, MENDES,. G. *Qualidade - Gestão e Métodos*. [Digite o Local da Editora]: Grupo GEN, 2012. 978-85-216-2195-9. Disponível em: https://app.minhabiblioteca.com.br/#/books/978-85-216-2195-9/. Acesso em: 20 jan. 2022.

Domingues, A. L. S. **Aplicação web**: Definição e análise de recursos de teste e validação. Tese de doutorado, ICMC/USP, São Carlos/SP – Brasil, 2005.

DRUMOND, Claire. **Scrum**. Disponível em: https://www.atlassian.com/br/agile/scrum. Acesso em: 05 out. 2022.

GATTI, Maíra Athanázio de Cerqueira; WERNECK, Vera Maria Benjamim. **Requisitos de Qualidade de Aplicações Web**. Cadernos do IME: Série Informática, [s. l], v. 17, n. 15, dez. 2004.

GOGONI, Ronaldo. **O que é software?** 2019. Disponível em: https://tecnoblog.net/responde/o-que-e-software/. Acesso em: 01 set. 2022.

KALENDAE (Brasil) (org.). **Quality Assurance: entenda o que é e como aplicar na gestão de TI**. 2019. Disponível em: https://kalendae.com.br/blog/quality-assurance/. Acesso em: 05 out. 2022.

LOBO, Renato. N. Gestão da Qualidade. Editora Saraiva, 2010

MONITORA (Brasil). **Saiba quais são as técnicas de teste de software e como aplicá-las**. 2021. Disponível em: https://www.monitoratec.com.br/blog/tecnicas-de-teste-de-software/. Acesso em: 10 out. 2022.

PRODANOV, C. C.; FREITAS, E. C. D. **Metodologia do trabalho científico:** métodos e técnicas da pesquisa e do trabalho acadêmico. Novo Hamburgo: Editora da Feevale, 2013. 276 p.

RODRIGUES, André Cripa. **O custo efetivo dos bugs no desenvolvimento de softwares**. 2017. Disponível em: https://interaktiv.com.br/blog/post/o-custo-efetivo-dos-bugs-no-desenvolvimento-de-softwares. Acesso em: 27 set. 2022.

SABADOTI, Vinicius. **Histórico sobre Testes de Software**. 2010. Disponível em: https://viniciussabadoti.wordpress.com/2010/08/03/historico-so

SIGNIFICADOS (Brasil). **Tecnologia da Informação (TI)**. Disponível em: https://www.significados.com.br/tecnologia-da-informacao/. Acesso em: 05 out. 2022.

SOARES, João Paulo. **Importância dos testes de software na qualidade do sistema**. 2020. Disponível em: https://www.treinaweb.com.br/blog/importancia-dos-testes-de-software-na-qualidade-do-sistema#:~:text=Permite%20identificar%20erros%20durante%20as,reputa%C3%A7%C3%A3o%20do%20neg%C3%B3cio%20no%20setor. Acesso em: 17 out. 2022.

SYDLE (Brasil) (org.). **Framework Scrum: o que é e como funciona?** 2020. Disponível em: https://www.sydle.com/br/blog/framework-scrum-5f6dc45f320703787497f887/#:~:text=um%20objetivo%20comum.-,O%20Scrum%20como%20conhecemos%20hoje,foi%20criado%20nos%20anos%201990. Acesso em: 10 out. 2022.

YIN, Robert K. **Estudo de Caso**. [Digite o Local da Editora]: Grupo A, 2015. 9788582602324. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788582602324/. Acesso em: 04 fev. 2022.