

CENTRO UNIVERSITÁRIO ATENAS

GUSTAVO HENRIQUE MACHADO DA SILVA

**SISTEMA DE ORDEM DE SERVIÇO PARA TÉCNICOS EM
INFORMÁTICA**

Paracatu

2019

GUSTAVO HENRIQUE MACHADO DA SILVA

SISTEMA DE ORDEM DE SERVIÇO PARA TÉCNICOS EM INFORMÁTICA

Monografia apresentada ao Curso de Sistemas de Informação do Centro Universitário Atenas, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Área de Concentração: Sistemas de informação.

Orientador: Prof. Msc. Felipe Neto Vasconcelos.

Paracatu

2019

GUSTAVO HENRIQUE MACHADO DA SILVA

SISTEMA DE ORDEM DE SERVIÇO PARA TÉCNICOS EM INFORMÁTICA

Monografia apresentada ao Curso de Sistemas de Informação do Centro Universitário Atenas, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Área de Concentração: Sistemas de informação.

Orientador: Prof. Msc. Felipe Neto Vasconcelos.

Banca Examinadora:

Paracatu – MG, ____ de _____ de _____.

Prof. Msc. Felipe Neto Vasconcelos
Centro Universitário Atenas

Prof.
Centro Universitário Atenas

Prof.
Centro Universitário Atenas

Dedico a Deus e a todos aqueles que de alguma maneira me ajudaram a chegar até aqui.

AGRADECIMENTOS

Agradeço primeiramente a Deus por me conceber a oportunidade e a capacidade de chegar até aqui.

A minha família que sempre me incentivou em buscar o melhor para mim. Pelo incentivo e confiança que sempre depositaram em mim.

A minha namorada que sempre está ao meu lado com total empenho e carinho imensurável.

A todos professores por nos ensinar com total excelência e ser um perfeito exemplo de profissional e ser humano.

Por último, mas não menos importante ao meu professor e orientador Felipe Neto Vasconcelos, que sempre me atendeu com dedicação e apontou onde eu poderia melhorar, assim me fazendo melhor.

“A perfeição é uma montanha impossível que deve ser escalada um pouco a cada dia”

Código Samurai

RESUMO

Com a evolução tecnológica que ocorre globalmente a manutenção de aparatos desse meio se torna cada vez mais necessária para que se consiga manter a continuidade de serviços e operações, portanto para que isso ocorra é necessário a existência do técnico de informática para as devidas soluções. Com um grande volume de serviços o técnico necessita de uma forma para se organizar e assim aproveitar melhor o seu tempo, obtendo uma maior produtividade em relação aos seus serviços prestados. Com o uso de um sistema de informação o técnico pode garantir essa organização de seus serviços assim como ter um documento do serviço, tornando mais profissional a sua atuação. O presente trabalho procura mostrar todos os preceitos e fundamentos além das características do desenvolvimento de um sistema de informação, demonstrando desde a definição de um problema que seria a falta de um sistema de ordens de serviços para técnicos em informática e como esse problema seria solucionado com a construção do devido sistema.

Palavras-chave: Técnicos de informática, sistema, ordem de serviço.

ABSTRACT

With a technological evolution that occurs globally, the maintenance of devices of this type is becoming increasingly necessary to be able to maintain the maintenance of services and operations, so that the use of computer technicians as unmotivated solutions occurs or is necessary. With a high volume of services or the technician who uses a way to better organize and make the most of your time, you get a better recovery from your services. The use of an information system or a technician can guarantee this organization of your services as a service document, making your performance more professional. The present paper seeks to show all the precepts and fundamentals beyond the capabilities of developing an information system, demonstrating from a definition of a problem that would be the lack of a service request system for computer technicians and how this problem is seriously solved. The construction of the proper system.

Keywords: Computer technicians, system, order of Service.

LISTA DE FIGURAS

Figura 1 - Modelo conceitual de banco de dados	37
Figura 2 - Modelo lógico de banco de dados	38
Figura 3 - Modelo de caso de uso	49
Figura 4 - Tela de acesso ao sistema	50
Figura 5 - Tela com menu do sistema	51
Figura 6 - Tela de consulta de OS	52

LISTA DE SIGLAS E ABREVIATURAS

BD – Banco de dados

COBIT – Control Objectives for Information and related Technology

CSS – Cascading Style Sheets

DER – Diagrama de entidade e relacionamento

HTML – Hipertext Markup Language

ITIL – Information Technology Infrastructure Library

JS – JavaScript

OS – Ordem de Serviço

PDCA – Plan, Do, Check and Act

PDF – Portable Document Format

PHP – Hipertext Preprocessor

SGBD – Sistema de Gerenciamento de Banco de Dados

SQL – Structured Query Language

TI – Tecnologia da Informação

UML – Unified Modeling Language

SUMÁRIO

1 INTRODUÇÃO	12
1.2 PROBLEMA	13
1.3 HIPÓTESES	13
1.4 OBJETIVOS	13
1.4.1 OBJETIVO GERAL	13
1.4.2 OBJETIVO ESPECÍFICO	13
1.5 JUSTIFICATIVA DO ESTUDO	13
1.6 METODOLOGIA DO ESTUDO	14
1.7 ESTRUTURA DO TRABALHO	16
2 CONCEITOS DE SISTEMAS DE INFORMAÇÃO E DESENVOLVIMENTO DE SOFTWARE.	17
2.1 DEFININDO UM SISTEMA	17
2.2 ENGENHARIA DE SOFTWARE	18
2.3 LINGUAGEM DE PROGRAMAÇÃO	19
2.4 PHP	20
2.5 BANCO DE DADOS	20
2.6 FRAMEWORK	21
3 ELUCIDAR O PROCESSO DE GESTÃO DE ORDEM DE SERVIÇO POR UM TÉCNICO DE INFORMÁTICA.	22
3.1 A TI COMO PRESTADORA DE SERVIÇOS	22
3.2 GOVERNANÇA DE TI	23
3.3 COBIT	23
3.4 ITIL	24
3.5 MELHORIA DO SERVIÇO DE TI COM BASE NA ITIL	24
3.5.1 ESTRATÉGIA DE SERVIÇO	24
3.5.2 DESENHO DE SERVIÇO	25
3.5.3 TRANSIÇÃO DE SERVIÇO	25
3.5.4 OPERAÇÃO DE SERVIÇO	25
3.5.5 MELHORIA CONTINUA DE SERVIÇO	26
4 APRESENTAR UM SISTEMA DE INFORMAÇÃO DE ORDEM DE SERVIÇO E SUAS IMPLICAÇÕES.	27

4.1 LEVANTAMENTO DE REQUISITOS	27
4.1.1 REQUISITOS FUNCIONAIS	28
4.1.2 REQUISITOS NÃO FUNCIONAIS	28
4.2 LARAVEL	28
4.2.1 AUTH	29
4.2.2 BLADE	29
4.2.3 MIGRATIONS	29
4.2.4 ADMIN LTE	30
4.2.5 BOOTSTRAP	30
4.2.6 DOMPDF	30
4.3 MODELAGEM DE SOFTWARE	30
4.3.1 UML	31
4.3.2 CASO DE USO	31
4.4 FORMAS DE AUXÍLIO DO SOFTWARE	31
5 CONSIDERAÇÕES FINAIS	32
5.1 TRABALHOS FUTUROS	32
REFERÊNCIAS	33
APÊNDICE A – Termo de Abertura de Projeto	35
APÊNDICE B – Modelo conceitual de banco de dados	37
APÊNDICE C – Modelo lógico de banco de dados	38
APÊNDICE D – Modelo físico de banco de dados	39
APÊNDICE E – Modelo de caso de uso	49
APÊNDICE F – Tela de acesso do sistema	50
APÊNDICE G – Tela do menu inicial do sistema	51
APÊNDICE H – Tela consulta de ordens de serviço	52
APÊNDICE I – Requisitos Funcionais	53
APÊNDICE J – Requisitos não funcionais	54
APÊNDICE K –Termo de Fechamento de Projeto	55

1 INTRODUÇÃO

Os sistemas de informação já demonstram em nosso cotidiano a sua enorme importância, a sua utilidade é imensa e pode gerar inúmeros ganhos para as empresas e organizações que utilizam de seus meios. Um sistema conforme diz Rezende (2006) se dá pelo conjunto de informações ou partes que compartilham entre si os dados para a obtenção de um objetivo em comum. Os sistemas estão presentes em nosso cotidiano mesmo antes de entendermos o seu conceito, como citado por Rezende (2006) hoje é comum associarmos a uma empresa, ou uma organização a um sistema, pelos processos que são feitos por ela, pois a tecnologia da informação está grandemente atrelada a todos os processos atuais.

Em um contexto moderno onde as informações são cruciais para a resolução de problemas com eficácia, Batista (2017) relata que os sistemas de informação se fazem necessários, para empresas que demandam um grande número de dados constantemente. Para essas empresas segundo Batista (2017) a assertividade de informações que um sistema pode proporcionar é imensamente maior que os problemas e custos que os sistemas de informação podem gerar ao decorrer de seu uso e fabricação.

Atualmente um técnico em informática para realizar o seu serviço necessita entender o problema que está ocorrendo com seu cliente, levantando todos os pontos e eliminando problemas simples, como falta de energia, falta de internet, um cabo mau conectado, após analisar os defeitos relatados cabe ao técnico decidir a melhor forma para resolver aquele problema, ou seja, qual a estratégia ele pode escolher para solucionar aquele problema, juntamente com quais ferramentas como relatado por Cohen (2008).

Nesse cenário um sistema de ordem de serviço pode ser bem utilizado conforme enfatiza Rezende (2006) para facilitar as atividades de empresas e prestadores de serviço da área de tecnologia, compreendendo dados de suma importância para resolução de problemas com clareza e objetividade, além de poder proporcionar um controle para ambas as partes, e podendo automatizar muitas vezes alguns problemas simples.

1.2 PROBLEMA

Quais os benefícios da utilização de um sistema de informação de ordem de serviço para um profissional de informática?

1.3 HIPÓTESES

- a) Um sistema de informação de ordem de serviço pode permitir que se relate todos os dados fornecidos pelo cliente para que o técnico consiga resolver o problema da maneira mais eficiente, assim no final gerando um relatório permitindo com que técnico e a empresa fiquem resguardados com um documento comprobatório.
- b) Os usuários podem não se adequar ao sistema, pois para se utilizar um sistema eles devem entender de processos informacionais e princípios básicos de informática, para assim relatar os problemas ocorridos de uma forma que o técnico compreenda, nesse sentido espera-se que o funcionário tenha essa base para que o sistema funcione com êxito, caso o usuário não tenha noção alguma o sistema poderá não ser executado com êxito, pode-se gerar uma ineficiência no serviço prestado.

1.4 OBJETIVOS

1.4.1 OBJETIVO GERAL

Apresentar um software que poderia ser utilizado para ação de ordem de serviço para técnicos de informática autônomos.

1.4.2 OBJETIVO ESPECÍFICO

- a) Conceituar os sistemas de informação e desenvolvimento de software.
- b) Elucidar o processo de gestão de ordem de serviço por um profissional de informática.
- c) Apresentar um sistema de informação de ordem de serviço e suas implicações.

1.5 JUSTIFICATIVA DO ESTUDO

A justificativa de implantação de um sistema de Ordem de Serviço se dá em virtude da necessidade de se consubstanciar, todas as informações dos serviços técnicos de informática prestados, e sobre o estado dos equipamentos. A falta de um

sistema de informação para profissionais da área técnica de tecnologia, abre espaço para erros, dúvidas e atrasos na realização dos serviços, além de não permitir uma comprovação dos serviços contratados e prestados.

Um sistema de ordem de serviço permitiria um maior controle sobre os serviços prestados, pois os relatórios armazenados poderiam ser constantemente requeridos em diversas ocasiões, como reuniões, consultas externas, verificação de serviços, etc.

Com o sistema poderia se ter um maior controle dos usuários, maior facilidade e agilidade nos processos assim como uma conservação de tempo e recursos que poderiam gerar uma facilidade de documentação dos processos assim evitando problemas quanto a mal-entendidos, e deixando sempre explícito os serviços que foram executados e para quem foram executados, assim sendo capaz de gerar maior confiança e assegurar a documentação de todo o processo feito no atendimento.

1.6 METODOLOGIA DO ESTUDO

A ciência se baseia em fatos para poder afirmar algumas causas e fenômenos, que segundo Rampazzo (2005) e para se confirmar essas causas existem métodos que se concretizam nas etapas ou processos que devem ser dados para solucionar um problema. Um método científico se caracteriza como tal segundo Gil (2008) pois o mesmo tem como claro objetivo a veracidade de fatos, possuindo esse forte ponto como sua característica fundamental, fazendo-se necessário também para que um conhecimento se torne científico a identificação e a determinação do método ao qual foi aplicado para se chegar a esse conhecimento ou estudo. Assim conforme citado por Marconi e Lakatos (2007) o conhecimento científico se trata de algo real que se ocasiona de fatos, têm a sua veracidade por se tratar de fatos ou então sua falsidade, cabendo ao estudo relatar cada caso através de uma metodologia.

Nesse contexto, o presente estudo é caracterizado como uma pesquisa de revisão exploratória, que para Gil (2008) tem como propósito ampliar, elucidar e transformar conteúdo e idealizações, destinando-se a elaboração de problemas mais precisos ou hipóteses pesquisáveis para estudos decorrentes, e que para Ruiz (1996) a pesquisa exploratória também estabelece métodos, critérios e técnicas com o intuito de se obter informações sobre o fator da pesquisa e orientar a formulação de

hipóteses, envolvendo um levantamento bibliográfico que de acordo com Gil (2008) é exposto com base em material já produzido, elaborado principalmente de livros e artigos científicos sobre as ações da equipe científica.

1.7 ESTRUTURA DO TRABALHO

Este trabalho apresenta cinco capítulos, sendo o primeiro capítulo a exposição do tema, dos objetivos geral e específicos bem como a metodologia utilizada, e a estrutura do trabalho.

O capítulo dois apresenta conceitos de sistemas de informação e desenvolvimento de software.

O capítulo três elucida o processo de gestão de ordem de serviço por um técnico de informática.

O quarto capítulo apresentar um sistema de informação de ordem de serviço e suas implicações.

O quinto capítulo conta com as considerações finais do trabalho e as referências.

2 CONCEITOS DE SISTEMAS DE INFORMAÇÃO E DESENVOLVIMENTO DE SOFTWARE.

Com o advento da revolução industrial o uso de máquinas substituindo a mão de obra braçal, trouxe avanços e melhorias para as tarefas repetitivas e de grande esforço, modificando para sempre a forma com que as tarefas eram feitas e criando um novo rumo para a história geral. (BATISTA, 2017)

Na utilização de tecnologia nas empresas houve o surgimento dos sistemas de informação que permitiam automatizar e melhorar processos e que segundo Pressman (2016) para se desenvolver um software é necessário realizar um conjunto de atividades, tarefas e ações; esse conceito também é abordado por Wazlawick (2013) que ainda complementa dizendo que as tarefas iniciais de avaliar, observar e até mesmo alterar processos produtivos empresariais quando necessários é de suma importância para a construção de um software.

Assim, fazendo uma analogia com as limitações que as empresas pré-revolução industrial tiveram conforme citado por Batista (2017) e as empresas que antecederam os sistemas de informação vivenciaram, pode-se apontar fatores como globalização, evolução, praticidade, tecnologia como pontos que hoje são cruciais para o crescimento de qualquer tipo de negócio que necessite de uma interação tecnológica, assim como enfatiza AL. Albertin e RMM Albertin (2008) a tecnologia da informação se tornou uma das partes de maior importância do meio empresarial, atingindo desde níveis operacionais até estratégicos com sua capacidade, assim melhorando o processo produtivo das organizações e conseqüentemente o controle de operações, tomadas de decisão e fluxos de trabalho entre outros aspectos.

2.1 DEFININDO UM SISTEMA

Um sistema pode ser definido por conter dois elementos principais, podendo ser eles um aglomerado de objetos, e a relação lógica entre eles, como citado por Mattos (2017), formando assim um sistema, não necessariamente computacional, a informação se dá por um conjunto de dados como afirma Côrtes (2017) que mantém um relacionamento entre si gerando algo que tenha sentido ou não.

Um sistema de informação faz com que os dados coletados sejam processados para assim gerar uma informação conforme afirma Côrtes (2017), e que segundo Mattos (2017) os sistemas de informação realizam uma tarefa específica no processamento dos dados, sendo construído por inúmeros módulos que fazem o papel de comunicação, controle, armazenamento, processamento e que estão ligados a si mesmo em um comum acordo chamado de protocolo comum.

Ao se transformar dados em informações é possível se tomar decisões, a quantidade a qualidade e a relação entre os dados obtidos definirão o nível de qualidade da informação que surgirá segundo Cortes (2017), sistemas de informação são fundamentais para um bom ambiente empresarial, tanto no lado de máquinas quanto no âmbito de pessoas. (MATTOS, 2017)

Exemplificando um grande sistema de informação que talvez possa ser considerado o mais complexo e perfeito que exista, Mattos (2017) cita o sistema nervoso do corpo humano, que comumente é chamado de rede neural humana, podendo se considerar que as informações que chegam ao nosso corpo por meios externos como odores, e sensações geram os comportamentos externos que seriam o output por exemplo a linguagem humana tanto literal quanto corporal diante das entradas. Para que um sistema de informação consiga resolver um problema com eficácia usa-se a engenharia de software para ordenar o seu desenvolvimento.

2.2 ENGENHARIA DE SOFTWARE

A construção de um software usando a engenharia de software baseia-se na ideia da construção civil onde temos uma base forte e pilares que fazem uma estrutura para outros artefatos conforme Pressman (2016), assim ocorre a construção de um software, onde precisamos criar pilares baseados em métodos e técnicas para se obter um produto de software.

Com isso Pressman (2016) explicita que deve se iniciar um projeto de software pela comunicação, ou seja, antes de existir alguma atividade técnica, deve existir uma boa comunicação entre as partes, podendo ser entendido os reais objetivos do projeto, assim como os requisitos e funcionalidades necessárias. Nesse contexto, Wazlawick (2013) faz uma comparação em relação aos componentes do projeto, onde pode-se ter na equipe, um engenheiro de software e um desenvolvedor e deve-se conhecer quais as funções de cada um dentro de um projeto de software,

em sua comparação ele propõe que o engenheiro de software está para mestre de obras, assim como o pedreiro está para o desenvolvedor, é necessário haver essa distinção para o melhor andamento do projeto, uma comunicação efetiva dentro da equipe permite um andamento sólido do projeto.

Sendo assim após a fase de comunicação conforme Pressman (2016) passa-se para o planejamento onde pode-se quebrar em pequenas tarefas uma atividade grande, facilitando a conclusão da mesma e o trabalho em equipe, e podendo gerar um mapa com cronograma e relação de resultados esperados. Após o planejamento o processo de modelagem faz a criação de um esboço que pode gerar uma ideia de como o projeto irá ficar na sua conclusão essa parte é crucial pois nela pode-se evitar erros e até perda de tempo com processos errados. Partindo da modelagem inicia-se a construção do software, que se baseando no projeto e no modelo, é feita a criação dos códigos e os testes que se julgarem necessários. E finalmente a parte da entrega que é feita após o feedback do cliente, segundo Pressman (2016) concretizando superficialmente os passos da engenharia de software.

Os sistemas de informação, além de terem um grande campo de atuação permitem que várias tecnologias sejam usadas para sua composição, criando assim sistemas altamente eficazes e necessários para solução de grandes problemas Pressman (2016), dentre os vários pontos para a criação de um sistema de informação a linguagem de programação é o que dá a vida ao sistema.

2.3 LINGUAGEM DE PROGRAMAÇÃO

Uma linguagem de programação é a forma para com que o programador atue como criador de comandos para que um sistema ou máquina os execute de acordo com as regras previstas e situações problemas dadas a ele, segundo Sebasta (2018) dando assim vida aos conhecidos sistemas de informação.

Para fazer a junção de todas essas tecnologias e conceitos utiliza-se uma sequência lógica para se dar instruções que a máquina deve realizar de acordo com certos eventos e ações feitas pelo usuário ou não, conceituado como algoritmo segundo Medina e Ferting (2006). Com a necessidade de grande utilização dos algoritmos necessita-se de formas que permitam com que um algoritmo seja feito de forma prática e que seja extremamente funcional para isso existem as linguagens de

programação que permitem segundo Sebesta (2018) a construção de softwares consistentes e que realizam com êxito as funções computacionais.

As linguagens de programação podem ser vistas como ferramentas, sendo assim cabe ao operante denominado programador ou desenvolvedor, ter a lógica necessária para resolver os problemas com as ferramentas (linguagens) adequadas, a lógica de programação é uma habilidade crucial tanto para a programação como para a vida na era da informação afirma Machado (2016), dentro das linguagens de programação temos linguagens voltadas para o desenvolvimento web onde se destaca a consistente linguagem PHP.

2.4 PHP

O PHP que nos primórdios de sua criação era denominada de Personal Home Pages Tool teve seu desenvolvimento realizado no outono de 1994 pelo seu criador Rasmus Lerdorf, era uma linguagem baseada em scripts (pequenos blocos de código) da linguagem C, porém em seu início somente Rasmums Lerdorf a utilizava, sendo a mesma só liberada amplamente seu código-fonte em 1995 após algumas melhorias conforme Dall'oglio (2018) foi adicionada a interação com o banco de dados, passando por várias melhorias até sua última versão disponibilizada que ocorreu em 2015 denominada de PHP 7.

Um programa em PHP pode ser codificado em qualquer editor de texto como exemplo pode-se citar o bloco de notas, entre várias ferramentas mais atuais que permitem uma facilidade no desenvolvimento a delimitação de um trecho em PHP se dá entre as tags `<?php ?>` sendo escrita dessa maneira os comandos encontrados no meio das tags serão reconhecidos pelo servidor afirma Niederauer (2017), para se ter mais utilidade nos códigos adiciona-se a ferramenta de banco de dados.

2.5 BANCO DE DADOS

O banco de dados se faz necessário para que as informações sejam armazenadas utilizando-se um sistema de banco de dados, que segundo Date (2004) é um sistema computadorizado de manutenção de registros, podendo ser comparado a um armário só que do meio eletrônico, onde se armazena dados que podem ser requeridos, modificados e excluídos de acordo com a sequência da aplicação,

complementado pela de ideia de Silberschatz e Sundarshan (2016) o banco de dados é definido por uma coleção de dados que armazenam informações relevantes para a empresa, os sistemas de banco de dados são construídos para gerenciar grandes volumes de informação e garantir a segurança das mesmas. Conforme Bean (2015) o uso de frameworks atrelado ao grande poder de bancos de dados permitem que o foco seja na construção da lógica do negócio e não em reinventar a roda.

2.6 FRAMEWORK

A sociedade avança cada vez mais no âmbito informacional, fazendo com que mais vezes se necessite de melhores estruturas e respostas rápidas para a resolução de problemas, para que o desenvolvedor se preocupe somente com a lógica do problema e não perca tanto tempo reinventando a roda utilizam-se os frameworks, ferramentas que segundo Bean (2015) ajudam o desenvolvedor evitando repetições de códigos dentre vários frameworks do mercado o LARAVEL se destaca pela sua solidez e grande aceitação pelos desenvolvedores, conforme será apresentado nos próximos capítulos.

3 ELUCIDAR O PROCESSO DE GESTÃO DE ORDEM DE SERVIÇO POR UM TÉCNICO DE INFORMÁTICA.

No cenário das organizações, vigora os diversos investimentos em busca de melhores formas de produtividade, uma das áreas de maior enfoque é a da Tecnologia da Informação que faz com que as organizações tenham inovações constantes para que elas consigam se manter no mercado, conforme cita (DA SILVEIRA DUARTE, 2014). De acordo com De Almeida Moraes e Terence (2004) o cenário de grande competitividade das organizações mesmo que em pequenas empresas demandam transformações, as organizações precisam se adequar aos eventos e tendências do mercado. Com base nesses princípios um técnico de informática visto como uma organização que segundo Maximiano (2000) se trata de uma unificação de esforços individuais que têm como propósito uma finalidade coletiva, ou seja, um objetivo comum, que presta um serviço e têm de pensar em fatores como o crescimento de sua produtividade, podendo focalizar na redução de tempo dos processos, o que pode ser realizado com um sistema de informação para ordem de serviços.

As organizações atualmente têm em seu poder uma grande gama de sistemas de informações que segundo Rainer e Cegielski (2016) permitem auxiliá-los em tomadas de decisão, gerenciamento, planejamento e na execução de inúmeras tarefas, afim de tornar o processo mais produtivo, econômico e enxuto, podendo até mesmo em alguns casos fazer a exclusão da figura humana como fonte de trabalho.

3.1 A TI COMO PRESTADORA DE SERVIÇOS

Quando um usuário solicita um serviço na área de tecnologia ele espera que vários pontos sejam alcançados para que se consiga sua satisfação, dentre esses pontos podem ser citados como exemplo: o prazo definido na hora do contrato, assim como o orçamento gerando uma maior confiança do cliente para a contratação de um certo serviço o que se consegue transferir através de uma ordem de serviço, trazendo maior produtividade fazendo com que o preço final abaixe. Para que todas essas exigências sejam atendidas adequadamente exige-se uma postura objetivada a solução de problemas e prestação de serviços, assim afirma (FERNANDES e DE ABREU, 2014). Para Mansur (2007), a governança de TI é uma área que permite um

grande auxílio a TI como prestadora de serviços, com o uso da governança voltada para a prestação de serviços pode-se obter grandes vantagens tais como aumento de produtividade, separação de serviços em processos, tendo grande impacto geral na empresa.

3.2 GOVERNANÇA DE TI

Baseia-se na ideia de conseguir avaliar e guiar o uso da tecnologia dentro da organização, para que se possa dar suporte e monitorar seus processos e ações, fazendo com que os gestores e o setor estratégico possam verificar através dos dados obtidos se os resultados até então mensurados estão dentro da conformidade pré-estabelecida pelo setor estratégico e caso não estejam dentro dos padrões servem de alerta para demonstrar as áreas que não estão em total desenvolvimento segundo (FERNANDES e DE ABREU, 2014). A governança em TI vem baseada no preceito de que o que não se pode medir dificilmente poderá ser controlado, através de uma dificuldade de organizações de controlar e mensurar seus esforços surgiu essa tecnologia capaz de suprir várias necessidades, e também de guiar decisões a serem tomadas, quem deverá ser o detentor dessas decisões e como monitorar para ver se realmente as escolhas feitas estão saindo conforme o esperado se tornando algo essencial para o crescimento e produtividade seja do setor, da empresa ou mesmo do indivíduo afirma (MANSUR, 2007). Como ferramenta da Governança em TI tem-se o COBIT e o ITIL que permitem alcançar esse patamar de controle de processos e gestão do mesmo para a melhoria em qualidade.

3.3 COBIT

COBIT (Control Objectives for Information and related Technology) teve sua criação em 1994 com o intuito de se conseguir um maior controle baseado em padrões internacionais que envolvem técnicas profissionais regulatórias para processos em TI acompanhando as várias evoluções do mundo tecnológico sempre em constante atualização o COBIT 5 veio em 2012 com a ideia de não ser mais um modelo e sim um framework de negócios e passa a não ser somente uma sigla o nome COBIT se torna um nome de uma marca forte de boas práticas em TI segundo (FERNANDES e DE ABREU, 2014). O COBIT conforme Mansur (2007) é uma forma de se decompor

a TI em domínios e processos facilitando o controle de um todo por partes, podendo assim o controle ser dividido entre vários gestores e as métricas também, gerando um maior nível de produtividade quando implementado.

3.4 ITIL

O Information Technology Infrastructure Library (ITIL) se descreve por ser um aglomerado de sugestões e orientações que informam as melhores práticas para um gerenciamento de serviços de TI a sua preocupação base é com o suporte e entrega dos serviços de maneira apropriada respeitando os requisitos do negócio, é a forma mais aceita mundialmente, dentre as preocupações que a ITIL atende estão infraestrutura de hardware, softwares, comunicações, pessoas, documentação entre outros segundo (MANSUR, 2007). A ITIL se baseia na ideia de que o serviço é uma forma para a qual um serviço seja entregue gerando valor para seu cliente, permitindo que ele atinja os resultados esperados, e tire deles a preocupação de custos e riscos específicos, sendo utilizada como framework segundo Fernandes e De Abreu (2014) visa reunir um conjunto de práticas para o gerenciamento de TI já previamente testadas e validadas no mercado, fazendo com que a organização chegue a um grau de maturidade e qualidade bem maior.

3.5 MELHORIA DO SERVIÇO DE TI COM BASE NA ITIL

Os avanços da ITIL em empresas bilionárias refletem nos cenários das menores empresas que se espelham no ambiente inovador e eficiente que a implantação da ITIL proporciona nas grandes corporações e assim aplicam as técnicas em seus ambientes corporativos para conseguirem chegar a um melhor desempenho e, portanto, transforma a ITIL em um padrão global afirma (CESTARI FILHO, 2012). Para que se aplique com eficácia a ITIL em uma organização segue-se um núcleo de ITIL como afirma Fernandes e De Abreu (2014) que é composto por cinco publicações sendo elas a Estratégia de serviço, Projeto de serviço, Transição de serviço, Operação de serviço e melhoria contínua de serviço.

3.5.1 ESTRATÉGIA DE SERVIÇO

Se concentra em prever e conceituar um aglomerado de serviços que podem vir a ajudar a organização ou negócio a alcançar seus objetivos afirma (CESTARI FILHO, 2012). Além desse aspecto a estratégia serve como forma de orientação e visualização segundo Fernandes e De Abreu (2014) permitindo uma visão de todo o ativo estratégico sendo abordado nesse ponto diretrizes como provedores, tipos de serviço, riscos estratégicos, portfólio de serviços e outros pontos.

3.5.2 DESENHO DE SERVIÇO

O desenho de serviço conforme Fernandes e De Abreu (2014) se torna uma forma de se conseguir modelar um serviço ou o desenvolvimento do serviço visando a melhoria de práticas, dentro dessa publicação consegue-se obter o nível do serviço, e detalhamento de outros aspectos como a segurança da informação, a disponibilidade, continuidade e capacidade. É o segundo passo para se seguir um conceito de ITIL afirma Cestari Filho (2012) que dentro do ambiente empresarial, tem como entrada as decisões e escolhas feitas na estratégia e projeta-se o que foi anteriormente decidido.

3.5.3 TRANSIÇÃO DE SERVIÇO

Para que o processo continue em andamento após criação do projeto a fase de transição tem o papel de mover os serviços para a produção, onde serão desenvolvidos analisados, testados e liberados segundo (CESTARI FILHO, 2012). Além de direcionar os novos serviços essa área exerce a função de redirecionar serviços modificados como afirma Fernandes e De Abreu (2014) e também é responsável pelo gerenciamento da liberação e distribuição dos mesmos.

3.5.4 OPERAÇÃO DE SERVIÇO

Essa fase é a que orienta todas as atividades diárias que serão realizadas pela organização ou mesmo pelo indivíduo afirma Fernandes e De Abreu (2014) assim fazendo com que as entregas sejam garantidas e o suporte aos serviços continue de maneiras eficientes e eficazes e que sejam feitas as devidas anotações sobre ocorridos durante o dia, acidentes, problemas e cumprimento das requisições. A

operação de serviços se trata de uma fase prolongada dentro do ciclo de vida da ITIL conforme Cestari Filho (2012), pois o mesmo serviço pode ser utilizado até que uma nova tecnologia ou processo o substitua, e cabe ao planejamento das fases anteriores fazer com que a migração de novos serviços seja simples e que não ocorra tumultos entre outros serviços.

3.5.5 MELHORIA CONTINUA DE SERVIÇO

Essa fase se dedica ao aprimoramento de um certo serviço ou processo realizado, por meio de uma avaliação prática visando que cada fase esteja sempre em seu maior nível de otimização, assim atendendo as necessidades de visão do negócio, essa fase, porém não pode ser classificada como individual pois as atividades presentes nela devem ser executadas durante todo ciclo de vida afirma (CESTARI FILHO, 2012).

A melhoria contínua é a forma de se estar sempre gerenciando a qualidade, incrementando as atividades afim de proporcionar maior eficiência segundo afirma Fernandes e De Abreu (2014) permitindo o gerenciamento desde setores operacionais quanto em setores estratégicos, baseando-se no modelo PDCA (Planejar-Fazer-Conferir-Ajustar ou Executar uma ação) para um melhor rendimento das tomadas de decisões e tarefas realizadas.

A ideia de um sistema de ordens de serviço se baseia em princípios citados acima para que se crie um sistema que melhore a eficiência dos serviços prestados e consiga documentar os mesmos.

4 APRESENTAR UM SISTEMA DE INFORMAÇÃO DE ORDEM DE SERVIÇO E SUAS IMPLICAÇÕES.

O software na sociedade atual tem um papel crucial de fazer a distribuição de informações, desde as informações mais úteis até as mais inúteis que variam de acordo com o ponto de vista de cada pessoa afirma (PRESSMAN, 2016).

O sistema de informação para ordem de serviço é caracterizado como um sistema web, ou seja que segundo Costa (2007) se baseia na utilização da internet para transmissão de dados, e que a rede internet é constituída de várias redes que se interligam por todo o mundo, permitindo assim a comunicação entre vários dispositivos, conforme cita Miletto e Bertagnolli (2014), o sistema web é um sistema distribuído pois tem a característica de permitir acessos de várias localidades através de uma conexão com a internet que tramitam dados.

Os dados, conforme enfatiza Costa (2007) são transmitidos via protocolos, para que seja feita de forma organizada e padronizada o traslado das informações de um dispositivo para o outro, Miletto e Bertagnolli (2014) cita o modelo seguido pela web chamado de cliente-servidor, onde um computador ou dispositivo faz requisições ao servidor e o mesmo entrega os dados de volta. Essa forma de utilização como afirma Marini (2012) permite com que os dispositivos não necessitem armazenar todas as informações, pois as mesmas podem ser requeridas a qualquer instante por uma requisição ao servidor, ou a vários servidores de acordo com a arquitetura pensada para a aplicação.

4.1 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é uma fase importante no início do projeto pois ela dita o ritmo das funcionalidades e permite que o desenvolvedor entenda o real problema de seu cliente conforme afirma Sommerville (2011), o levantamento de requisitos pode ser definido como a descrição das ações que o sistema deve executar para chegar a um fim desejado que também deve ser relatado para fim de documentação do projeto. Conforme afirma Pressman (2016) a fase de levantamento de requisitos engloba os requisitos funcionais e não funcionais, que permitem uma maior clareza dos aspectos que o software deverá atingir e assim de forma eficiente

e econômica realizar as tarefas que o cliente necessita atingindo o conceito de qualidade de software.

4.1.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais são uma categoria dentro da engenharia de requisitos onde se tem a descrição clara de um parâmetro que o sistema deve realizar conforme Pressman (2016), os requisitos funcionais podem ser de usuário, por exemplo, um bastante usado é o de autenticação que exprime a ideia de que o usuário deve estar autenticado ou seja que tenha seus dados verificados garantindo ser um usuário autentico para realizar tarefas dentro do sistema, isso se caracteriza como um requisito funcional, conforme apresentado no apêndice I. Em alguns casos os requisitos funcionais segundo Sommerville (2011) fazem o papel contrário, ou seja, explicita atividades ou ações que o sistema não deve realizar ajudando assim a definir o escopo da aplicação.

4.1.2 REQUISITOS NÃO FUNCIONAIS

Os requisitos podem se restringir ou até mesmo gerar novos requisitos, para Sommerville (2011) os requisitos funcionais podem ser descritos como funções ou restrições de serviços que podem conter dentro de um software, conforme apresentado no apêndice J e que também os requisitos não funcionais têm um aspecto mais amplo ou seja suas implicações se estendem por todo o software e não por alguma parte específica somente.

4.2 LARAVEL

A busca por formas mais simples e eficazes de se resolver um problema é um grande paradigma da humanidade, não seria diferente no desenvolvimento de *softwares*, os *frameworks* permitem que o foco com a infraestrutura seja menor, ou seja, o processo de desenvolvimento é agilizado fazendo com que o desenvolvedor ganhe vantagens para assim focar em outras partes aplicação ou até mesmo em novos projetos conforme afirma (BEAN, 2015).

O *framework* Laravel permite que não seja necessário copiar uma estrutura anteriormente feita, ele com pequenos comandos permite que seja iniciado um projeto e já seja codificado sem se preocupar com pastas e outras configurações, o Laravel conforme afirma Turini (2015) nos oferece uma estrutura padrão de uma forma bem organizada e de fácil entendimento, seguindo as melhores práticas de desenvolvimento do mercado. Dentre suas funcionalidades que poupam muito tempo de trabalho se destacam a conexão com o banco de dados, o envio de e-mails, e a autenticação de usuários que já vêm prontos ou pré-configurados.

4.2.1 AUTH

Ao se desenvolver um *software* um dos pontos de maior importância e que se deve dar um maior grau de atenção, é em relação a autenticação e segurança daquele *software*, para isso o Laravel simplifica bastante a sua implementação, deixando o arquivo já pré-configurado, localizado em `config/auth.php` conforme cita Bean (2015) contendo várias opções de configurações para a aplicação.

4.2.2 BLADE

Para conseguir ajudar o desenvolvedor ainda mais no desenvolvimento o Laravel conta com o Blade, o motor de telas do Laravel, uma forma de evitar com que repetições de códigos de HTML sejam feitas segundo Bean (2015) e assim tornando as páginas mais dinâmicas. É possível utilizar sequencias lógicas de maneiras mais fáceis e com sua utilização ganha-se bastante flexibilidade, por exemplo com uma definição simples de uma estrutura de decisão dentro do texto da aplicação conforme Turini (2015), para se utilizar o Blade basta colocar como extensão do arquivo antes da extensão php, por exemplo “teste.blade.php”.

4.2.3 MIGRATIONS

Dentre as grandes funcionalidades do Laravel também se destaca o uso de *migrations*, sem um *framework* você deve abrir algum aplicativo de gerenciamento de banco de dados e inserir todas as tabelas e inserções que o seu sistema precisa ter no banco de dados, com as *migrations* o laravel permite que essa tarefa seja configura

apenas uma vez e chamada quando necessário através de um comando de linha única conforme (TURINI, 2015).

4.2.4 ADMIN LTE

O Admin LTE se trata de um painel de controle de modelos de telas totalmente responsivo e gratuito, que é disponibilizado no site <https://adminlte.io> e tem sua construção baseada no *framework Bootstrap 3* o que faz com que o sistema tenha um design atual e responsivo.

4.2.5 BOOTSTRAP

Bootstrap se trata de um framework com código aberto com foco no desenvolvimento de componentes de interface utilizando HTML, CSS e Javascript, baseando-se em princípios do design da atualidade segundo Bean (2015) tais como responsividade e experiência do usuário.

4.2.6 DOMPDF

Para se armazenar dados e que se tenha um arquivo leve e suportado pela maioria das mídias digitais se faz necessário o uso de PDF (Formato Portátil de Documento), que permite que uma sequência de dados seja armazenada no mesmo documento e pode ser configurado no padrão A4 de folhas, para irem diretamente para a impressão gerando um documento para o usuário, a biblioteca Dompdf permite uma integração simples entre Laravel e a emissão de PDF.

4.3 MODELAGEM DE SOFTWARE

A modelagem de sistemas é um processo em que se cria modelos abstratos de um *software*, fazendo com que cada modelo demonstre uma visão ou perspectiva diferente do sistema. Fazendo parte dos processos de engenharia de requisitos pois ajuda a extrair e até mesmo demonstrar de maneira clara os requisitos do sistema segundo (SOMMERVILLE, 2011).

A modelagem de sistemas geralmente é representada por alguma notação gráfica, atualmente o modelo mais utilizado é o de UML (linguagem de modelagem unificada).

4.3.1 UML

Para que se consiga visualizar o escopo de um *software* utiliza-se a UML, assim como na engenharia tem-se as plantas baixas, no desenvolvimento aplica-se a UML. A UML se trata de uma linguagem visual que se baseia no paradigma da orientação a objetos, a UML se tornou uma linguagem padrão de modelagem, sendo adotada internacionalmente na indústria de engenharia de *software* (GUEDES,2018).

4.3.2 CASO DE USO

O caso de uso entra dentro da UML como uma forma de se demonstrar ações que poderão vir a ser feitas dentro de um sistema, o caso de uso se concentra em dois princípios os atores e o casos de uso, onde os atores desempenham o papel do usuário e o caso de uso reflete, conforme o apêndice “E” onde mostra que o sistema deverá dar como resposta para cada ação que for executada conforme (PRESSMAN, 2016).

4.4 FORMAS DE AUXÍLIO DO SOFTWARE

As principais funcionalidades contidas no *software* de ordem de serviços para técnicos em informática são para poder organizar o dia a dia do técnico, assim facilitando o entendimento para a realização de determinados serviços ao cliente. Assim tendo um documento registrado com o defeito reclamado inicialmente pelo cliente, fazendo com que caso o cliente não esteja no local ou até mesmo não pertença mais a empresa o técnico tenha uma forma de saber o problema que estava ocorrendo pelo relato inicial do cliente. O sistema permite armazenar todos os clientes já cadastrados podendo assim o técnico ter um maior controle de suas informações, para quando necessário abrir chamado para clientes já cadastrados ter uma maior agilidade nos processos, as ordens de serviços armazenadas permitem ao técnico ter uma dimensão de seu faturamento no decorrer dos atendimentos.

5 CONSIDERAÇÕES FINAIS

A partir das informações levantadas e também de aspectos bibliográficos analisados para este trabalho estabelece-se que a hipótese de que um sistema permite que se relate os serviços juntamente com os dados de seu cliente, através da necessidade de se ter um documento onde fosse relatado os serviços prestados e peças utilizadas, observa-se que o OS *System* – Ordem de Serviço para técnicos em Informática cumpre com a necessidade prevista.

Através desse trabalho conseguiu-se demonstrar todos os objetivos propostos como conceituar um sistema de informação, elucidar o processo de gestão de ordem de serviço por um profissional de informática e apresentar um sistema de informação de ordem de serviço e suas implicações.

O sistema permitirá que o usuário tenha um documento e também o técnico para que assim fique firmado por meio de assinatura das duas partes o que foi realizado e que estão de acordo com o recebimento dos equipamentos, assim também estão cientes de suas responsabilidades, do cliente de realizar o pagamento e do técnico de realizar o reparo.

Utilizando desse sistema o técnico consegue ter um armazenamento ordenado em banco de dados dos dados de seus clientes, assim como de serviços realizados para balanceamentos e relatórios mensais de seus serviços, o que se comparado com o processo manual feito através de cadernetas estaria propenso a demora e também problemas como o não entendimento de alguma informação, a caderneta poderia sumir, molhar ser queimada etc.

Sabe-se que o sistema pode ter melhorias para melhor atender e assim captar mais clientes e outras formas de atendimento também.

5.1 TRABALHOS FUTUROS

O sistema futuramente pode permitir que o usuário comum (cliente) entre e abra um chamado e já dê um início automático a ordem de serviço, fazendo com que o técnico recebesse via e-mail o chamado e assim já deslocasse para o local da empresa com sua localização via smartphone, agilizando mais ainda o processo de atendimento por meio do técnico.

REFERÊNCIAS

- ALBERTIN, Alberto Luiz; ALBERTIN, Rosa Maria de Moura. **Benefícios do uso de tecnologia de informação para o desempenho empresarial**. 2008.
- BATISTA, Emerson de Oliveira. **Sistemas de informação**. Editora Saraiva, 2017.
- BAZZOTTI, Cristiane; GARCIA, Elias. **A importância do sistema de informação gerencial na gestão empresarial para tomada de decisões**. Ciências Sociais Aplicadas em Revista, v. 6, n. 11, 2006.
- BEAN, Martin. **Laravel 5 essentials**. Packt Publishing Ltd, 2015.
- CESTARI FILHO, Felício. **ITIL v3 Fundamentos**. Rio de Janeiro: RNP/ESR, 2012.
- CÔRTEZ, Pedro Luiz. **Administração de sistemas de informação**. Editora Saraiva, 2017.
- COHEN, Roberto. **Implantação de help desk e service desk**. Novatec Editora, 2008.
- COSTA, Carlos J. **Desenvolvimento para web**. ITML press/Lusocredito, 2007.
- DA SILVEIRA DUARTE, Fernanda et al. **A contribuição da TI como ferramenta na gestão de negócios: um estudo de caso de uma empresa no ramo tecnológico localizada no município de Dom Pedrito/RS**. Tecno-Lógica, v. 18, n. 2, p. 103-114, 2014.
- DATE, Christopher J. **Introdução a sistemas de bancos de dados**. Elsevier Brasil, 2004.
- DALL'OGGIO, Pablo. **PHP Programando com orientação a Objetos**. Novatec Editora, 2018.
- DE ALMEIDA MORAES, Giseli Diniz; TERENCE, Ana Cláudia Fernandes; ESCRIVÃO FILHO, Edmundo. **A tecnologia da informação como suporte à gestão estratégica da informação na pequena empresa**. JISTEM-Journal of Information Systems and Technology Management, v. 1, n. 1, p. 28-44, 2004.
- FERNANDES, Aguinaldo Aragon; DE ABREU; Vladimir Ferraz. **Implantando a Governança de TI-: Da estratégia a gestão de Processos e Serviços**. Brasport, 2014.
- GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social**. 6. ed. Editora Atlas SA, 2008.
- GUEDES, Gilleanes TA. **UML 2-Uma abordagem prática**. Novatec Editora, 2018.

MACHADO, Raquel et al. **Ler, escrever e programar: atividades essenciais para o desenvolvimento cognitivo na era da informação**. Sánchez, J.(2016) Editor. Nuevas Ideas en Informática Educativa, v. 12, p. 511-516, 2016.

MANSUR, Ricardo. **Governança de TI: metodologias, frameworks e melhores práticas**. Brasport, 2007.

MARINI, Emiliano. **El Modelo Cliente/Servidor**. Recuperado el, v. 5, 2012.

MATTOS, Antonio Carlos Marques. **Sistemas de informação**. Editora Saraiva, 2017.

MAXIMIANO, Antonio Cesar Amaru. **Introdução à administração**. Atlas, 2000.

MEDINA, Marco; FERTING, Cristina. **Algoritmos e programação: teoria e prática**. Novatec Editora, 2006.

MILETTO, Evandro Manara; DE CASTRO BERTAGNOLLI, Sílvia. **Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP-Eixo: Informação e Comunicação-Série Tekne**. Bookman Editora, 2014.

NIEDERAUER, Juliano. **PHP para quem conhece PHP**. Novatec Editora, 2017.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software-8ª Edição**. McGraw Hill Brasil, 2016.

RAINER JR, R. Ke; CEGIELSKI, Casey. **Introdução a sistemas de informação: apoiando e transformando negócios na era da mobilidade**. Elsevier Brasil, 2016.

RAMPAZZO, Lino. **Metodologia científica**. Edições Loyola, 2005.

REZENDE, Denis Alcides. **Engenharia de software e sistemas de informação**. Brasport, 2006.

RUIZ, João Álvaro. Metodologia científica. **Guia para eficiência nos estudos**, v. 13, p. 131, 1996.

SEBESTA, Robert W. **Conceitos de Linguagens de Programação-11**. Bookman Editora, 2018.

SILBERSCHATZ, Abraham; SUNDARSHAN, S.; KORTH, Henry F. **Sistema de banco de dados**. Elsevier Brasil, 2016.

SOMMERVILLE, Ian. **Software engineering** 9th Edition. ISBN-10137035152, 2011.

TURINI, Rodrigo. **PHP e Laravel: Crie aplicações web como um verdadeiro artesão**. Editora Casa do Código, 2015.

WAZLAWICK, Raul. **Engenharia de software: conceitos e práticas**. Elsevier Brasil, 2013.

APÊNDICE A – Termo de Abertura de Projeto

Termo de Abertura de Projeto

NOME DO PROJETO: OS SYSTEM: Sistema de ordem de serviço para técnicos em informática.

DATA DE ELABORAÇÃO: 12/08/2019

1 ESCOPO PRELIMINAR DO PROJETO

1.1 ESCOPO PRODUTO

O sistema irá fornecer para o técnico uma forma de ter um documento comprobatório da realização de seu serviço, com funcionalidades como cadastro de clientes, serviços, peças e também os cadastros de ordens de serviço permitindo que se possa analisar a quantidade de serviços realizados, os materiais utilizados e gerar um comprovante que serve para que o cliente assine e confirme a realização do serviço.

1.2 OBJETIVOS DO PROJETO

- a) Conceituar os sistemas de informação e desenvolvimento de software.
- b) Elucidar o processo de gestão de ordem de serviço por um profissional de informática.
- c) Apresentar um sistema de informação de ordem de serviço e suas implicações.

1.3 PREMISSAS DO PROJETO

- a) Falta de tempo para desenvolvimento.

1.4 RESTRIÇÕES DO PROJETO

- a) O Projeto deverá estar em conformidade com os parâmetros instituídos pelo Centro Universitário Atenas.
- b) Todas as ferramentas utilizadas deverão constar na parte escrita do projeto.
- c) O usuário receberá uma senha mestra para seu uso.

d) O sistema será desenvolvido somente em versão web.

2 EQUIPE PRELIMINAR

Este projeto está sendo elaborado por Gustavo Henrique Machado da Silva

3 RISCOS PRELIMINARES

TABELA 1: RISCOS PLELIMINARES

Identificação	Riscos
RP 01	Projeto não ser aceito
RP 02	Projeto não ser entregue em tempo
RP 03	Indisponibilidade de integrantes na participação do projeto
RP 04	Mudanças no cronograma do projeto
RP 05	Mudanças no escopo do projeto

Fonte: Elaborado pelo autor.

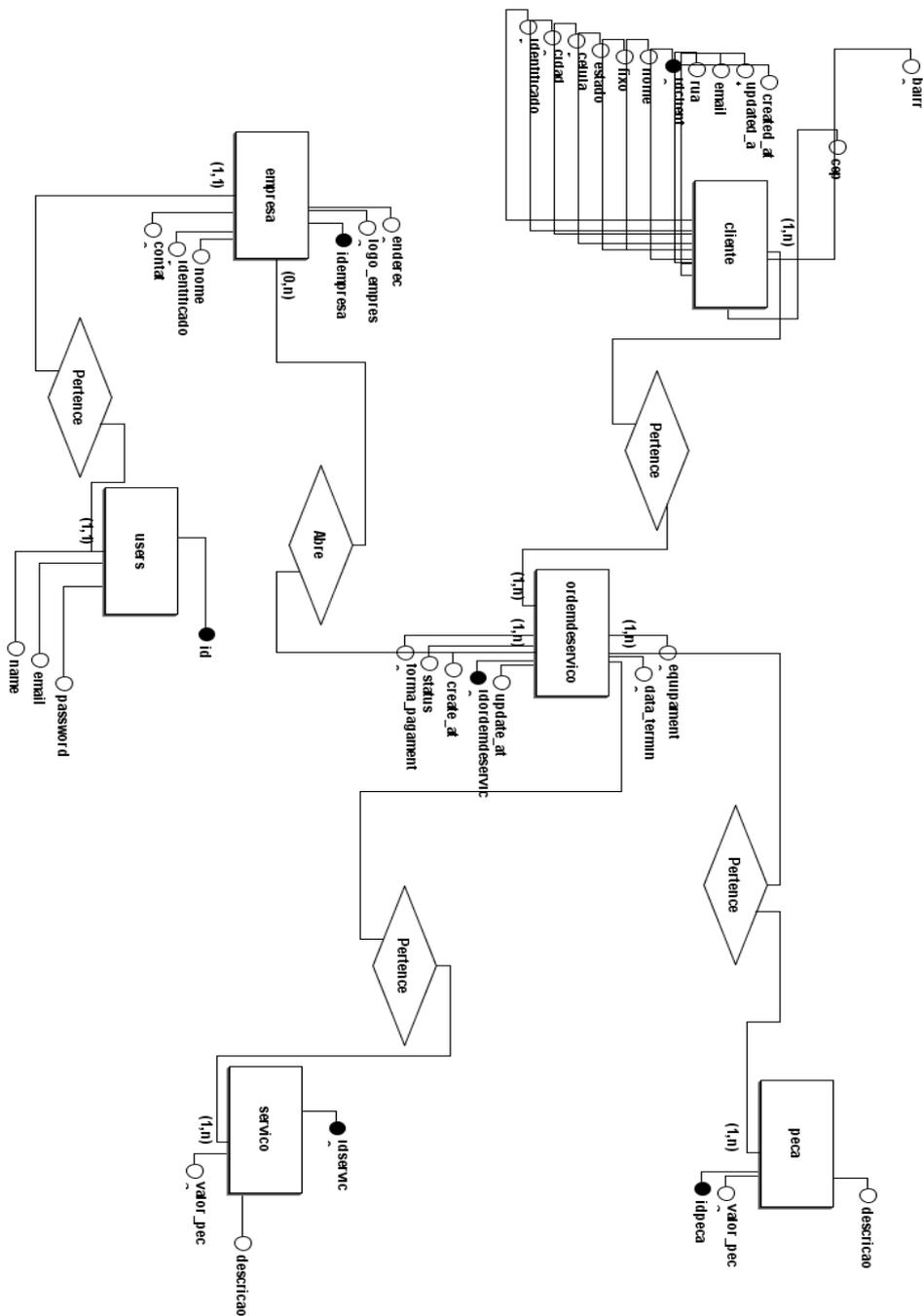
4 APROVAÇÃO

Data de Aprovação: 31 de Outubro de 2019.

Felipe Neto Vasconcelos

APÊNDICE B – Modelo conceitual de banco de dados

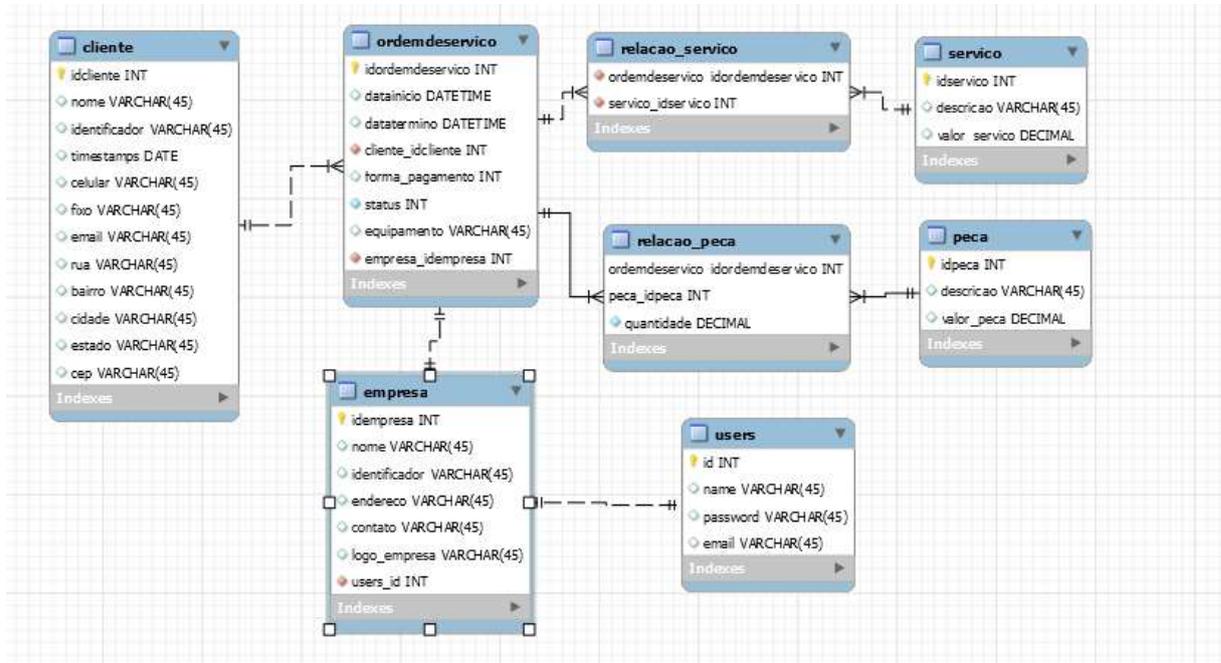
Figura 1 - Modelo conceitual de banco de dados



Fonte: Elaborado pelo autor

APÊNDICE C – Modelo lógico de banco de dados

Figura 2 - Modelo lógico de banco de dados



Fonte: Elaborado pelo autor

APÊNDICE D – Modelo físico de banco de dados

Modelo físico de banco de dados

```
-- Banco de dados: `ordemdeservico`
```

```
-- -----
```

```
--
```

```
-- Estrutura da tabela `clientes`
```

```
--
```

```
CREATE TABLE `clientes` (
  `idcliente` bigint(20) UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  `nome` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `identificador` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `celular` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `fixo` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `email` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `rua` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `bairro` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `cidade` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `estado` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `cep` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
--
```

```
-- Extraindo dados da tabela `clientes`
```

```
--
```

```
INSERT INTO `clientes` (`idcliente`, `created_at`, `updated_at`, `nome`,
`identificador`, `celular`, `fixo`, `email`, `rua`, `bairro`, `cidade`, `estado`, `cep`)
VALUES
(1, '2019-10-31 17:33:32', '2019-10-31 17:33:32', 'Gualter da Silva Pedra',
'01.225.573/0001-00', '6136121245', '6136121122', 'teste@gmail.com', 'Rua do
Doce', 'Praça José Adamian, S/N', 'CRISTALINA', 'GOIÁS', '73850-000');
```

```

-----
--
-- Estrutura da tabela `empresa`
--
CREATE TABLE `empresa` (
  `idempresa` bigint(20) UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  `nome` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `identificador` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `endereco` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `contato` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `logo_empresa` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `fk_users` bigint(20) UNSIGNED NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
--
-- Extrair dados da tabela `empresa`
--
INSERT INTO `empresa` (`idempresa`, `created_at`, `updated_at`, `nome`,
`identificador`, `endereco`, `contato`, `logo_empresa`, `fk_users`) VALUES
(3, '2019-10-31 17:34:45', '2019-10-31 17:34:45', 'OS COMPANY', '13196325407',
'Rua 25 de Dezembro Qd 03 Lt 1/14 AP 102', '6136122672', 'asdasdasdasds', 1);
-----
--
-- Estrutura da tabela `migrations`
--
CREATE TABLE `migrations` (
  `id` int(10) UNSIGNED NOT NULL,
  `migration` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `batch` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
--
-- Extrair dados da tabela `migrations`
--

```

```

INSERT INTO `migrations` (`id`, `migration`, `batch`) VALUES
(1, '2014_10_12_000000_create_users_table', 1),
(2, '2014_10_12_100000_create_password_resets_table', 1),
(3, '2019_07_17_134443_create_clientes_table', 1),
(4, '2019_09_16_172550_create_empresa_table', 1),
(5, '2019_09_16_172612_create_peca_table', 1),
(6, '2019_09_16_173627_create_servico_table', 1),
(7, '2019_09_18_210002_create_ordem_servico_table', 1),
(8, '2019_09_18_222529_create_relacao_peca_table', 1),
(9, '2019_10_24_084043_create_relacao_servico_table', 1);
-----
--
-- Estrutura da tabela `ordemservico`
--
CREATE TABLE `ordemservico` (
  `idordemdeservico` bigint(20) UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  `data_inicio` date NOT NULL,
  `data_termino` date DEFAULT NULL,
  `forma_pagamento` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `status` int(11) NOT NULL,
  `equipamento` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `fk_cliente` bigint(20) UNSIGNED NOT NULL,
  `fk_empresa` bigint(20) UNSIGNED NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
--
-- Extraindo dados da tabela `ordemservico`
--
INSERT INTO `ordemservico` (`idordemdeservico`, `created_at`, `updated_at`,
`data_inicio`, `data_termino`, `forma_pagamento`, `status`, `equipamento`,
`fk_cliente`, `fk_empresa`) VALUES

```

```
(1, '2019-10-31 17:35:00', '2019-10-31 17:35:00', '2019-10-31', NULL, 'BOLETO', 2,
'Notebook Positivo', 1, 3);
```

```
-----
```

```
--
```

```
-- Estrutura da tabela `password_resets`
```

```
--
```

```
CREATE TABLE `password_resets` (
  `email` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `token` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-----
```

```
--
```

```
-- Estrutura da tabela `peca`
```

```
--
```

```
CREATE TABLE `peca` (
  `idpeca` bigint(20) UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  `descricao` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `valor_peca` decimal(8,2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
--
```

```
-- Extraindo dados da tabela `peca`
```

```
--
```

```
INSERT INTO `peca` (`idpeca`, `created_at`, `updated_at`, `descricao`, `valor_peca`)
VALUES
```

```
(1, '2019-10-31 17:33:43', '2019-10-31 17:33:43', 'Junta do Cabeçote do Notebook
636', '180.00');
```

```
-----
```

```
--
```

```
-- Estrutura da tabela `relacao_peca`
```

```
--
```

```

CREATE TABLE `relacao_peca` (
  `fk_ordemdeservico` bigint(20) UNSIGNED NOT NULL,
  `fk_peca` bigint(20) UNSIGNED NOT NULL,
  `quantidade` decimal(8,2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-----
--
-- Estrutura da tabela `relacao_servico`
--
CREATE TABLE `relacao_servico` (
  `fk_ordemdeservico` bigint(20) UNSIGNED NOT NULL,
  `fk_servico` bigint(20) UNSIGNED NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-----
--
-- Estrutura da tabela `servico`
--
CREATE TABLE `servico` (
  `idservico` bigint(20) UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  `descricao` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `valor_servico` decimal(8,2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
--
-- Extraindo dados da tabela `servico`
--
INSERT INTO `servico` (`idservico`, `created_at`, `updated_at`, `descricao`,
`valor_servico`) VALUES
(1, '2019-10-31 17:33:55', '2019-10-31 17:33:55', 'Formatação com Backup',
'120.00');
-----
--

```

```

-- Estrutura da tabela `users`
--
CREATE TABLE `users` (
  `id` bigint(20) UNSIGNED NOT NULL,
  `name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `email` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `email_verified_at` timestamp NULL DEFAULT NULL,
  `password` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `remember_token` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
--
-- Extraindo dados da tabela `users`
--
INSERT INTO `users` (`id`, `name`, `email`, `email_verified_at`, `password`,
`remember_token`, `created_at`, `updated_at`) VALUES
(1, 'Gustavo Henrique', 'gustavohenrique.atenas@gmail.com', NULL,
'$2y$10$xjfPDyv/F6qQuXwXJcY8deWMwRd6GGCdB.ngk9I37p1bjEmNP9aWy',
NULL, '2019-10-31 15:45:57', '2019-10-31 15:45:57');
--
-- Índices para tabelas despejadas
--
--
-- Índices para tabela `clientes`
--
ALTER TABLE `clientes`
  ADD PRIMARY KEY (`idcliente`),
  ADD UNIQUE KEY `clientes_identificador_unique` (`identificador`);
--
-- Índices para tabela `empresa`
--
ALTER TABLE `empresa`
  ADD PRIMARY KEY (`idempresa`),

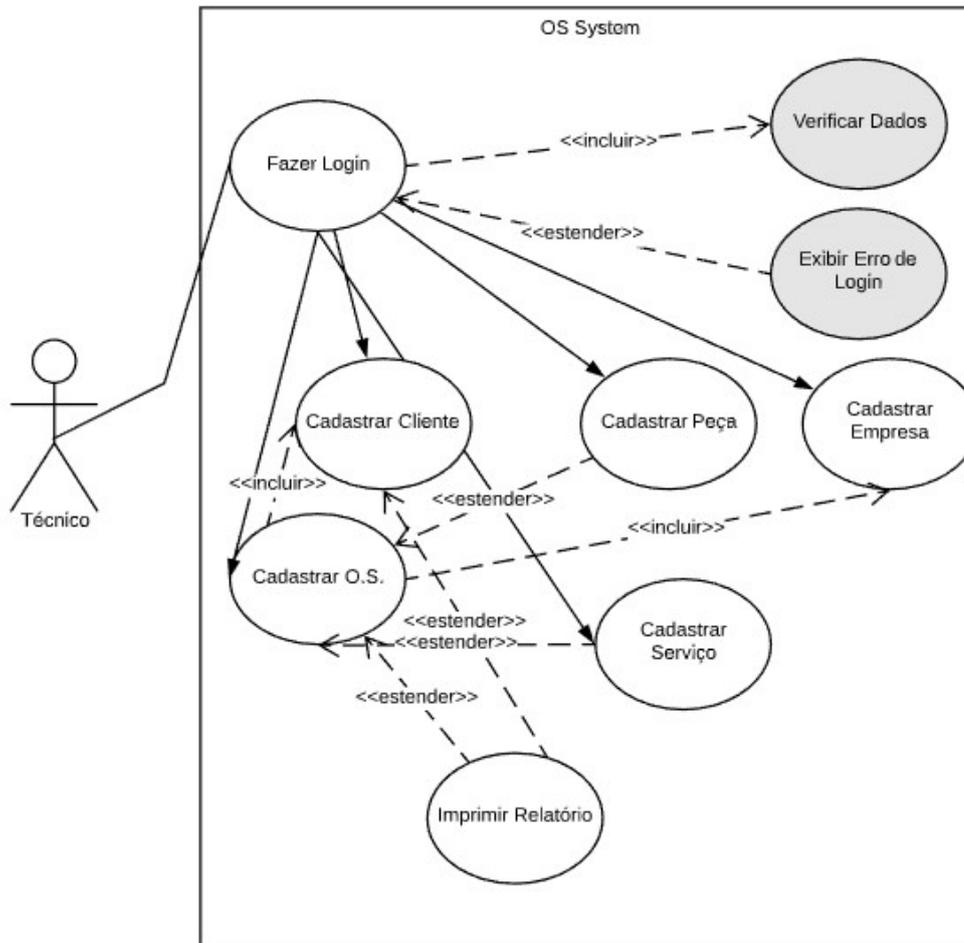
```

```
ADD UNIQUE KEY `empresa_identificador_unique` (`identificador`),
ADD KEY `empresa_fk_users_foreign` (`fk_users`);
--
-- Índices para tabela `migrations`
--
ALTER TABLE `migrations`
  ADD PRIMARY KEY (`id`);
--
-- Índices para tabela `ordemservico`
--
ALTER TABLE `ordemservico`
  ADD PRIMARY KEY (`idordemdeservico`),
  ADD KEY `ordemservico_fk_cliente_foreign` (`fk_cliente`),
  ADD KEY `ordemservico_fk_empresa_foreign` (`fk_empresa`);
--
-- Índices para tabela `password_resets`
--
ALTER TABLE `password_resets`
  ADD KEY `password_resets_email_index` (`email`);
--
-- Índices para tabela `peca`
--
ALTER TABLE `peca`
  ADD PRIMARY KEY (`idpeca`);
--
-- Índices para tabela `relacao_peca`
--
ALTER TABLE `relacao_peca`
  ADD PRIMARY KEY (`fk_ordemdeservico`,`fk_peca`),
  ADD KEY `relacao_peca_fk_peca_foreign` (`fk_peca`);
--
-- Índices para tabela `relacao_servico`
--
ALTER TABLE `relacao_servico`
```

```
ADD PRIMARY KEY (`fk_ordemdeservico`,`fk_servico`),
ADD KEY `relacao_servico_fk_servico_foreign` (`fk_servico`);
--
-- Índices para tabela `servico`
--
ALTER TABLE `servico`
  ADD PRIMARY KEY (`idservico`);
--
-- Índices para tabela `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `users_email_unique` (`email`);
--
-- AUTO_INCREMENT de tabelas despejadas
--
-- AUTO_INCREMENT de tabela `clientes`
--
ALTER TABLE `clientes`
  MODIFY `idcliente` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=2;
--
-- AUTO_INCREMENT de tabela `empresa`
--
ALTER TABLE `empresa`
  MODIFY `idempresa` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=4;
--
-- AUTO_INCREMENT de tabela `migrations`
--
ALTER TABLE `migrations`
  MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=10;
```

```
--  
-- AUTO_INCREMENT de tabela `ordemservico`  
--  
ALTER TABLE `ordemservico`  
  MODIFY `idordemdeservico` bigint(20) UNSIGNED NOT NULL  
  AUTO_INCREMENT, AUTO_INCREMENT=2;  
--  
-- AUTO_INCREMENT de tabela `peca`  
--  
ALTER TABLE `peca`  
  MODIFY `idpeca` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=2;  
--  
-- AUTO_INCREMENT de tabela `servico`  
--  
ALTER TABLE `servico`  
  MODIFY `idservico` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=2;  
--  
-- AUTO_INCREMENT de tabela `users`  
--  
ALTER TABLE `users`  
  MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=2;  
--  
-- Restrições para despejos de tabelas  
--  
-- Limitadores para a tabela `empresa`  
--  
ALTER TABLE `empresa`  
  ADD CONSTRAINT `empresa_fk_users_foreign` FOREIGN KEY (`fk_users`)  
  REFERENCES `users` (`id`);  
--
```

```
-- Limitadores para a tabela `ordemservico`
--
ALTER TABLE `ordemservico`
  ADD CONSTRAINT `ordemservico_fk_cliente_foreign` FOREIGN KEY (`fk_cliente`)
REFERENCES `clientes` (`idcliente`),
  ADD CONSTRAINT `ordemservico_fk_empresa_foreign` FOREIGN KEY
(`fk_empresa`) REFERENCES `empresa` (`idempresa`);
--
-- Limitadores para a tabela `relacao_peca`
--
ALTER TABLE `relacao_peca`
  ADD CONSTRAINT `relacao_peca_fk_ordemdeservico_foreign` FOREIGN KEY
(`fk_ordemdeservico`) REFERENCES `ordemservico` (`idordemdeservico`) ON
DELETE CASCADE,
  ADD CONSTRAINT `relacao_peca_fk_peca_foreign` FOREIGN KEY (`fk_peca`)
REFERENCES `peca` (`idpeca`) ON DELETE CASCADE;
--
-- Limitadores para a tabela `relacao_servico`
--
ALTER TABLE `relacao_servico`
  ADD CONSTRAINT `relacao_servico_fk_ordemdeservico_foreign` FOREIGN KEY
(`fk_ordemdeservico`) REFERENCES `ordemservico` (`idordemdeservico`) ON
DELETE CASCADE,
  ADD CONSTRAINT `relacao_servico_fk_servico_foreign` FOREIGN KEY
(`fk_servico`) REFERENCES `servico` (`idservico`) ON DELETE CASCADE;
COMMIT;
Fonte: Elaborado pelo autor
```

APÊNDICE E – Modelo de caso de uso**Figura 3 - Modelo de caso de uso**

Fonte: Elaborado pelo autor

APÊNDICE F – Tela de acesso do sistema

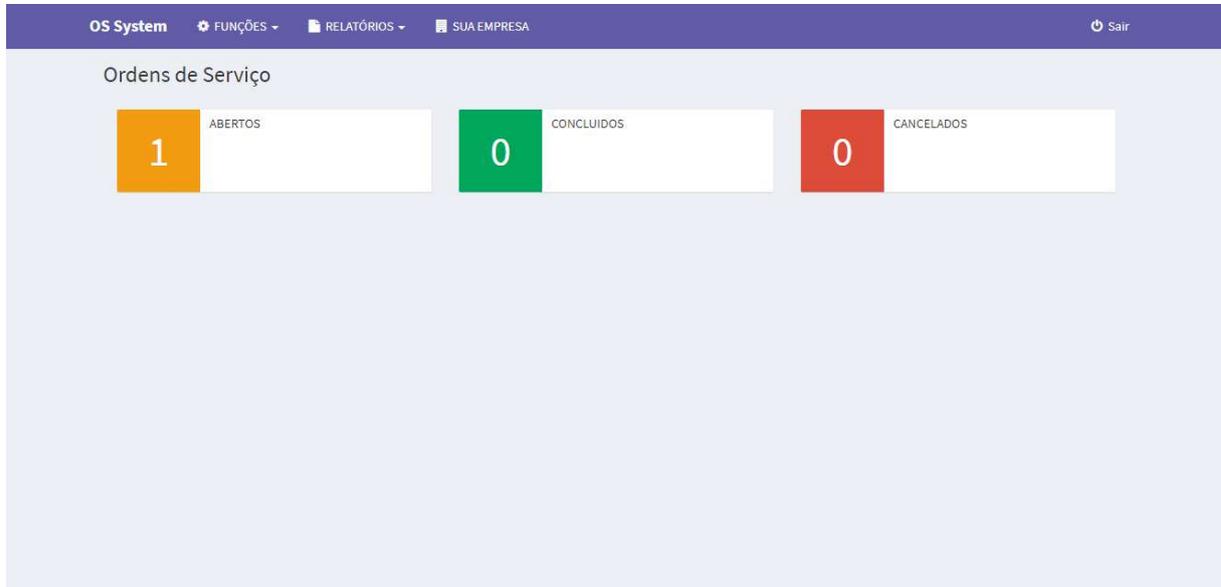
Figura 4 - Tela de acesso ao sistema



Fonte: Elaborado pelo autor

APÊNDICE G – Tela do menu inicial do sistema

Figura 5 - Tela com menu do sistema



Fonte: Elaborado pelo autor

APÊNDICE H – Tela consulta de ordens de serviço

Figura 6 - Tela de consulta de OS

ID	Data Abertura	Status	Cliente	Ações
1	2019-10-31	Aberto	1	Adc. Serviço Adc. Peça Editar Deletar Imprimir

Fonte: Elaborado pelo autor

APÊNDICE I – Requisitos Funcionais

TABELA 2 – Requisitos Funcionais

Identificador	Descrição dos requisitos	RNF
RF-01	Usuário deve fazer login	RNF-01
RF-02	Usuário efetuará cadastro de sua empresa	RNF-03
RF-03	Usuário efetuará cadastro de seus clientes	RNF-03
RF-04	Usuário efetuará cadastro de seus serviços prestados	RNF-03
RF-05	Usuário efetuará cadastro de peças utilizadas para serviços.	RNF-03
RF-06	Usuário cadastra ordem de serviço utilizando de dados buscados no banco de dados do sistema	RNF-03
RF-07	Usuário retira relação de clientes cadastrados	RNF - 03
RF-08	Usuário imprimir ordem de serviço gerada para fins comprobatórios de realização de serviços.	RNF - 03

Fonte: Elaborado pelo autor

APÊNDICE J – Requisitos não funcionais**TABELA 3 – Requisitos não funcionais**

Identificação	Descrição
RNF – 01	Para que seja liberado o acesso o usuário deve entrar no sistema com seu login e senha de acesso.
RNF – 02	O desenvolvimento ocorrerá na linguagem PHP.
RNF – 03	O sistema deverá ter um banco de dados para armazenar informações.
RNF – 04	O sistema deverá conter cores e ícones que facilitem a utilização e o aprendizado.
RNF – 05	Caso o usuário tente cadastrar um formulário sem um campo obrigatório o sistema deve avisá-lo e não permitir a continuidade.
RNF – 06	A disponibilidade do sistema é algo importante.

Fonte: Elaborado pelo autor

APÊNDICE K –Termo de Fechamento de Projeto

Termo de Fechamento de Projeto

NOME DO PROJETO: OS SYSTEM: Sistema de ordem de serviço para técnicos em informática.

PERÍODO DE EXECUÇÃO: 12/08/2019 a 11/11/2019

1 Produtos entregues

Um software de gerenciamento de ordens de serviços com as seguintes funcionalidades: Manter cadastros de clientes e usuários, registrar ordens de serviço, registro de peças e serviços com seus respectivos valores, emissão de relatórios de clientes e ordens de serviços.

2 Aceite do projeto

Confirmo o recebimento das configurações de software descrita neste documento e declaro que esta configuração atende plenamente os objetivos definidos para o projeto.

Portanto, considero o projeto concluído com sucesso, e assumo, a partir dessa data, a posse e a propriedade do software produzido no projeto.

Paracatu, 11 de Novembro de 2019.

Felipe Neto Vasconcelos