

HELPER: ANÁLISE DO FUNCIONAMENTO DE UM SISTEMA DE TICKET

EDUARDO LAMARCA DE OLIVEIRA SANTOS PIRES¹, LUCIANA ROCHA CARDOSO²; EZEQUIAS FERREIRA DE SOUZA³; LUDMILA FURTADO BREDER⁴, ANDRÉIA ALMEIDA MENDES⁵

1 Acadêmico do 3º período de Tecnologia em Análise e Desenvolvimento de Sistemas do Centro Universitário UNIFACIG, Manhuaçu – MG, 2010563@sempre.unifacig.edu.br.

2 Mestre em Ciência da Computação pela Universidade Federal de Viçosa (UFV), Centro Universitário UNIFACIG, Manhuaçu – MG, luroca@sempre.unifacig.edu.br.

3 Mestre em Desenvolvimento Local pelo Centro Universitário Augusto Motta (UNISUAM), Centro Universitário UNIFACIG, Manhuaçu - MG, ezequias.souza@sempre.unifacig.edu.br.

4 Mestre em Informática pela Pontifícia Universidade Católica de Minas Gerais (PUC Minas), Centro Universitário UNIFACIG, Manhuaçu – MG, Ludmila@sempre.unifacig.edu.br

5 Doutora em Linguística pela Universidade Federal de Minas Gerais (UFMG), Centro Universitário UNIFACIG, Manhuaçu-MG, andreialetras@yahoo.com.br.

RESUMO

O artigo tem como objetivo detalhar o desenvolvimento do sistema Helper descrevendo as escolhas feitas no decorrer do crescimento deste sistema, bem como a linguagem de programação e as informações sobre diagramas UML, diagramas de classes e diagramas de casos de uso. Por meio desses diagramas, obtém-se noção do ambiente em que o sistema será desenvolvido e os conhecimentos mais detalhados de classes que serão criados. Com o intuito de analisar as dificuldades enfrentadas por uma empresa na hora de uma organização adequada em frente a determinado problema, foi escolhida a metodologia exploratória; buscando-se informações do que falta e o que pode melhorar na hora de um bom atendimento e na resolução de um problema. Assim, explica-se, neste artigo, como um sistema de ticket funciona, demonstrando suas vantagens e como ele pode, de modo ágil, auxiliar na resolução dos infortúnios gerados na empresa, gerando mais organização e comunicação entre as entidades como também oferecendo relatórios e informações que poderão ser utilizados para melhorias futuras nos atendimentos, minimizando, assim, o tempo gasto para resolver esses imprevistos e evitando trabalho desnecessário. Com todas as adversidades enfrentadas, comprovou-se que o sistema de ticket é uma boa solução para gerar um ambiente de trabalho mais produtivo.

Palavras-chave: Sistema de Ticket; Helper; Linguagem de Programação.

HELPER: ANALYSIS OF THE FUNCTIONING OF A TICKET SYSTEM

ABSTRACT

The article aims to detail the development of the Helper system describing the choices made during the growth of this system, as well as the programming language and information about UML diagrams, class diagrams and use case diagrams. Through these diagrams, it is possible to get an idea of the environment in which the system will be developed and a more detailed knowledge of the classes that will be created. In order to analyze the difficulties faced by a

company at the time of an adequate organization facing a given problem, the exploratory methodology was chosen; looking for information about what is missing and what can be improved when it comes to providing a good service and solving a problem. Thus, this article explains how a ticket system works, demonstrating its advantages and how it can, in an agile way, help in the resolution of misfortunes generated in the company, generating more organization and communication between entities as well as offering reports and information that can be used for future improvements in care, thus minimizing the time spent to resolve these unforeseen events and avoiding unnecessary work. With all the adversities faced, the ticket system proved to be a good solution to generate a more productive work environment.

Keywords: Ticket System; Helper; Programming Language.

1 INTRODUÇÃO

Toda empresa busca otimizar, ao máximo, todas as suas atividades, sempre procurando formas de executá-las do modo mais rápido e eficiente possível; porém, problemas e imprevistos são inevitáveis e sempre irão aparecer em qualquer local. Assim, na hora de resolver esses problemas, a organização se torna um fator de extrema importância para que as inconveniências sejam solucionadas de forma ágil, evitando prejuízos e situações que possam atrasar as atividades da empresa.

Este artigo tem o intuito de analisar esses obstáculos que surgem no ambiente de trabalho e oferecer uma solução, por meio de um sistema de tickets, demonstrando o funcionamento deste sistema e como seu uso poderá ajudar no processo de resolução de problemas, como também apresentar informações sobre diagrama UML e a importância de uma documentação bem elaborada no seu processo de criação. Assim, fica claro que este tipo de sistema oferece uma melhor comunicação entre os usuários, que necessitam que algum problema seja resolvido e a área que será responsável por resolver este problema, oferecendo, de forma estruturada, uma boa resposta às situações inesperadas.

Um sistema de ticket gera uma área melhor tanto para quem necessita de ajuda, quanto para quem irá prover esta ajuda, organizando os infortúnios de forma que os de maior importância com um risco maior, tenham maior prioridade, ao mesmo tempo que gera um ambiente mais controlado para coleta de informações a fim de agilizar processos futuros por meio de dados adquiridos pelo próprio sistema de problemas passados, auxiliando a solução e levando a alcançar os melhores resultados por meio de organização, comunicação e coleta de dados.

2 METODOLOGIA

Para a realização deste artigo, a metodologia escolhida foi a exploratória com o intuito de levantar informações sobre a importância da organização na hora da resolução de problemas em uma empresa; utilizou-se também a pesquisa bibliográfica como forma de entender o suporte adequado a qualquer funcionário de uma empresa. Para detalhar o funcionamento do sistema e seu processo de criação, utilizou-se da pesquisa descritiva.

3 REFERENCIAL TEÓRICO

3.1 Gestão da Informação

A gestão da informação é entendida como a gestão eficaz de todos os recursos de informação relacionados com a empresa, incluindo os recursos gerados internamente e externamente, utilizando-se da tecnologia da informação sempre que necessário. Almeida *et al* (2015, p. 2821) define a gestão da informação como “um conjunto de conceitos, princípios, métodos e técnicas usadas na prática administrativa que auxiliam no processo de tomada de decisão e alcance da missão e objetivos” de forma a criar uma organização melhor para a tomada de decisão visando cumprir uma missão ou objetivo, gerando, assim, um ambiente favorável para o empregado.

É nesse ambiente que os empregados da empresa constroem vínculos, que melhoram a confiança da organização, como afirmado por Batista e Oliveira (2012, *on-line*), que relata que “O estudo da confiança do empregado na organização contribui para a compreensão da formação de vínculos com a organização empregadora”. Muito mais que melhorar a relação empregado e organização, a gestão de informação gera diversas vantagens como explicado a seguir.

Ter a informação correta no momento em que ela se faz necessária pode ajudar o negócio a atingir todos os resultados esperados. Com os dados precisos, o negócio pode avaliar a sua posição no mercado com mais certeza e otimizar a sua cadeia operacional por meio de mudanças de maior impacto.

Em outras palavras, as informações que o negócio possui podem ser um dos seus principais ativos. Basta ver o impacto causado pelas ferramentas de *Business Intelligence* e Big Data: elas permitiram que empresas de vários setores conseguissem obter insights inovadores cruzando registros diversificados para encontrar a melhor forma de atingir suas metas (BLOG TECLÓGICA, 2017, *on-line*).

Mostrando o valor da informação, que pode ser capaz de muito mais, demonstrando a importância de um bom controle de dados nos dias atuais, quem organiza e sintetiza melhor o conhecimento obtém uma enorme vantagem.

A informação deve ser considerada como diferencial de negócios, quando proporciona alternativas de lucratividade e retornos profícuos para a empresa, seja sedimentando atuações e implementando os atuais negócios, seja criando novas oportunidades de negócios, devendo, portanto, ser gerenciada de forma estratégica. (TORRES, NEVES, 2008, p. 5).

3.2 Gestão de Projetos

Explicado por Justo (2018, on-line), “a gestão de projetos é um conjunto de práticas e competências utilizadas para planejar, executar, monitorar e controlar os projetos de uma organização, independentemente do tamanho ou da complexidade desses projetos”. Oferecendo também vantagens, como comentado por Sbardelotto *et al* (2017, p.53), “a gestão de projetos visa à transparência das etapas do trabalho, o que beneficia a comunicação nos métodos de delegar tarefas, primando para a vantagem competitiva sustentável das organizações”. Trazendo algo que, de acordo com Filho (2011, p.36), é de extrema importância.

É de suma importância entender as necessidades do cliente para saber como e onde você e sua equipe podem ajudar na solução do problema. Sem o entendimento completo do problema a ser tratado e um planejamento em mãos, você (gerente e líder) e sua equipe não saberão onde querem e precisam chegar. Uma possível consequência é deparar-se com a inserção de erros logo cedo no desenvolvimento do projeto, os quais virão, possivelmente, apenas bem mais tarde, a serem descobertos. Isto geralmente acontece quando não há qualquer ‘preocupação’ com a gestão. Essas, dentre outras, são razões pelas quais muitos projetos se transformam em casos de insucesso. Portanto, a primeira lição é:(saber) ouvir o cliente e (re)conhecer o problema (a ser resolvido) (FILHO, 2011, p.36).

As diversas vantagens oferecidas por uma boa gestão de projetos é um dos motivos de inúmeras empresas realizarem investimentos nessa área, como defendido por Project Builder (2021, on-line): “Na atualidade, a maioria das empresas investe nessa área, contratando gestores de projetos, treinando os seus colaboradores e promovendo e participando de eventos. Cursos, palestras e especializações são oferecidos aos interessados[...]”, confirmando assim a importância da gestão de projetos para a avaliação do trabalho mais simples até o mais complexo e com o intuito de resolver problemas, “[...]pois o maior desafio de um gerente de projetos é construir um relacionamento positivo com as partes interessadas de seus projetos, garantindo o suporte um ambiente favorável a realização do projeto” (NORO, 2012, on-line).

3.3 Um sistema de Tickets

Conforme assevera Ramalho (2018),

Desde que existem serviços de atendimento ao público, pelo menos em algum momento houve um período de espera. Essa situação ocorre sempre que existe um número de pessoas por atender superior ao número de pessoas para atender. Quando essa situação ocorre são utilizadas duas formas de lidar com o problema, criando uma fila de espera física, em que a pessoa avança na fila até chegar a sua vez, ou criando uma fila de espera virtual (RAMALHO, 2018, p.1).

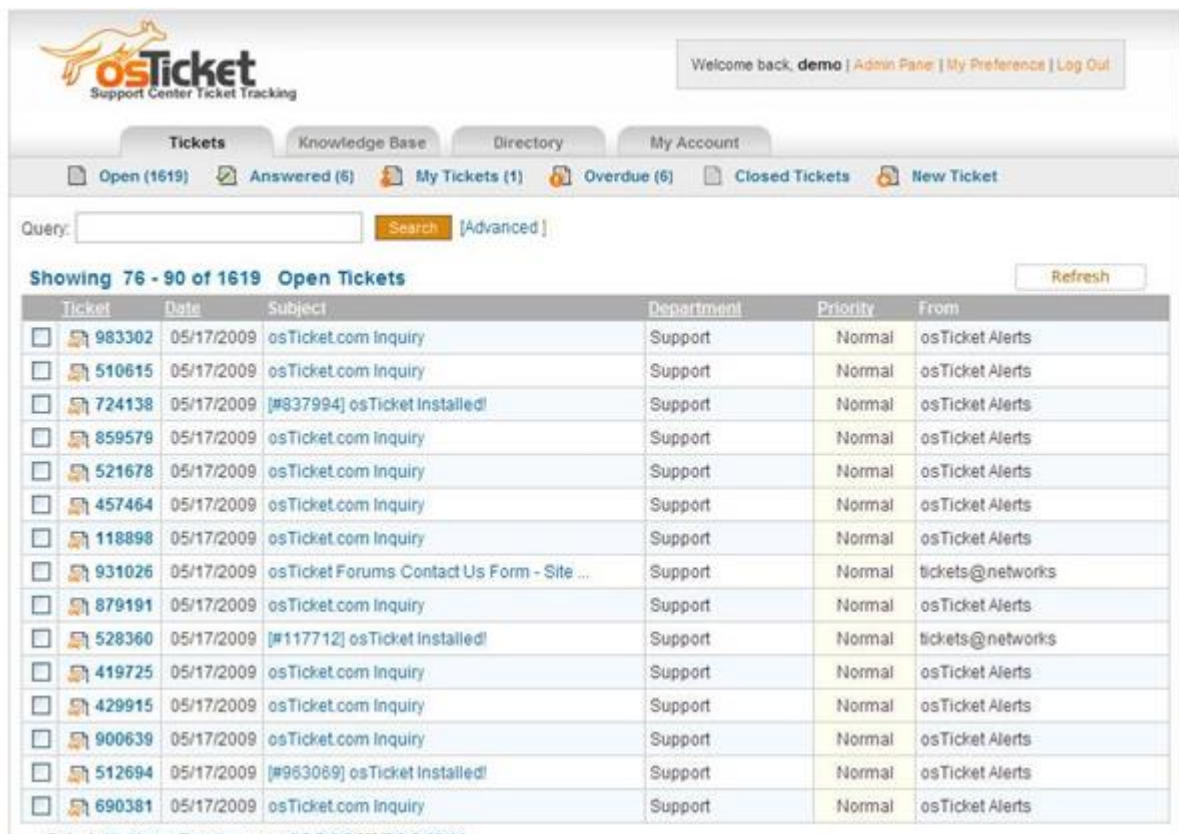
E, para criar essa fila virtual, foi escolhido um sistema de ticket no qual as chamadas para o suporte serão automaticamente transformadas em tickets com as informações necessárias, como descrito a seguir, podendo-se usar a Imagem 1 como referência de um sistema de ticket.

Os tickets representam os chamados de atendimento. Cada solicitação realizada no suporte é automaticamente transformada em um ticket dentro da plataforma, independentemente do canal de comunicação utilizado: chat, e-mail, telefone ou formulário.

E as vantagens não param por aí! Utilizar um sistema de tickets possibilita um atendimento *omnichannel*, assegurando uma experiência única ao cliente, por meio da integração entre todos os canais de contato.

Nesse sentido, em cada ticket aberto, constará a informação do canal de origem. Além disso, é possível identificar o solicitante, o tipo de serviço requisitado e a categoria (dúvida, problema, solicitação de serviço ou sugestão) (CASARIL, 2019, on-line).

IMAGEM 1: Exemplo de um Sistema de Ticket



Ticket	Date	Subject	Department	Priority	From
983302	05/17/2009	osTicket.com Inquiry	Support	Normal	osTicket Alerts
510615	05/17/2009	osTicket.com Inquiry	Support	Normal	osTicket Alerts
724138	05/17/2009	[#837994] osTicket Installed!	Support	Normal	osTicket Alerts
859579	05/17/2009	osTicket.com Inquiry	Support	Normal	osTicket Alerts
521678	05/17/2009	osTicket.com Inquiry	Support	Normal	osTicket Alerts
457464	05/17/2009	osTicket.com Inquiry	Support	Normal	osTicket Alerts
118898	05/17/2009	osTicket.com Inquiry	Support	Normal	osTicket Alerts
931026	05/17/2009	osTicket Forums Contact Us Form - Site ...	Support	Normal	tickets@networks
879191	05/17/2009	osTicket.com Inquiry	Support	Normal	osTicket Alerts
528360	05/17/2009	[#117712] osTicket Installed!	Support	Normal	tickets@networks
419725	05/17/2009	osTicket.com Inquiry	Support	Normal	osTicket Alerts
429915	05/17/2009	osTicket.com Inquiry	Support	Normal	osTicket Alerts
900639	05/17/2009	osTicket.com Inquiry	Support	Normal	osTicket Alerts
512694	05/17/2009	[#963089] osTicket Installed!	Support	Normal	osTicket Alerts
690381	05/17/2009	osTicket.com Inquiry	Support	Normal	osTicket Alerts

Fonte: Sapo pplware.

Assim sendo, o sistema de ticket oferece o ambiente favorável que todos procuram para a solução de problemas; além disso, segundo Casaril (2019, on-line), “o ponto é: quanto menos complexo e burocrático for o atendimento para o usuário, mais satisfeito ele ficará. Um sistema de tickets proporciona a facilidade no relacionamento entre agentes e clientes”, trazendo a simplicidade necessária nesse relacionamento, atendendo a necessidade de organização, gerando confiança no empregado, enquanto auxilia na gestão de informação, com o objetivo de otimizar o processo de auxílio ao funcionário no seu problema, assim gerando um local favorável para ambos os lados, o que precisa de ajuda e o que vai providenciar a ajuda.

Para a criação deste projeto, foi escolhido a linguagem de programação php.

3.4 Linguagem de programação utilizada

Segundo Soares *et al.* (2013),

A partir de 1994, o desenvolvedor canadense-dinamarquês Rasmus Lerdorf, começou o desenvolvimento da linguagem de programação PHP (Hypertext Preprocessor), para auxiliar no desenvolvimento de sua página pessoal. Aliberdade de código fonte aberto e softwaregratuito do PHP garantida por um grupo de esquemas de licença, denominadas GPL (General Public License), colocou o PHP como uma das linguagens mais usadas para desenvolvimento web, tendo grande reconhecimento mundial [Converse and Park 2003] (SOARES *et al.*, 2013, p.47).

E de forma a utilizar essa linguagem, o modo ou paradigma de programação escolhido foi o orientado a objeto, segundo Henrique (2019), a programação orientando a objetos é um paradigma de programação ou modo de programar, que surgiu como alternativa a características de programação estruturadas com o intuito de aproximar o manuseio de estruturas do mundo real, justamente por isso o nome “objeto”. Trazendo, assim, o sistema de ticket para uma forma mais real, para um entendimento melhor do programa.

3.5 Diagramas UML

Conforme Oliveira (2020),

O desenvolvimento de software, também conhecido como programas de computador, aplicativos ou sistemas, é uma atividade complexa, exige do projetista uma série de habilidades que vão além da compreensão das linguagens de programação e bancos de dados. Na Engenharia de Software uma das etapas fundamentais é chamada de análise de requisitos, onde o projetista vai até o cliente e faz o levantamento das necessidades deste para então iniciar o projeto de um software. Durante esta etapa, toda a informação deve ser registrada, pois são elas que vão determinar o quanto o software será eficiente e sua qualidade para o cliente final. [...]

Após a etapa inicial, o projetista faz uma documentação técnica dos dados coletados utilizando uma série de representações esquemáticas (Diagramas), que auxiliam na compreensão das relações entre as informações, norteados os programadores durante o desenvolvimento do software. Estes diagramas são elaborados quase sempre a partir de uma linguagem conhecida como UML (Unified Modeling Language), que não é uma linguagem de programação, mas sim um conjunto de diagramas compostos por uma série de símbolos gráficos com significados bem específicos (OLIVEIRA, 2020, pp.116-117).

Como explicado por Oliveira (2020, pp.116-117), a documentação técnica na etapa inicial é extremamente importante para a realização de um projeto e levantamento de requisitos de um sistema. Para o levantamento de requisitos deste projeto, foi feito um Diagrama UML que é explicado a seguir.

A Linguagem de modelagem unificada (UML) foi criada para estabelecer uma linguagem de modelagem visual comum, semanticamente e sintaticamente rica, para arquitetura, design e implementação de sistemas de software complexos, tanto estruturalmente quanto para comportamentos. Além do desenvolvimento de software, a UML tem aplicações em fluxos do processo na fabricação. É análoga aos modelos utilizados em outros campos, e é composta por diferentes tipos de diagramas. De modo geral, diagramas UML descrevem o limite, a estrutura e o comportamento do sistema e os objetos nele contidos (LUCIDCHART, [s.d.], on-line).

Um diagrama UML oferece uma imagem de seu projeto antes mesmo de construí-lo, fazendo assim que seja definida a estrutura do sistema como seus limites e o comportamento, mostrando como cada usuário vai interagir com ele e quais objetos terá no sistema. A visualização da ideia de um sistema, antes mesmo dele ser criado, é algo que, de acordo com Heredia (2012, p.27), ajuda a aperfeiçoar os processos de negócios das organizações.

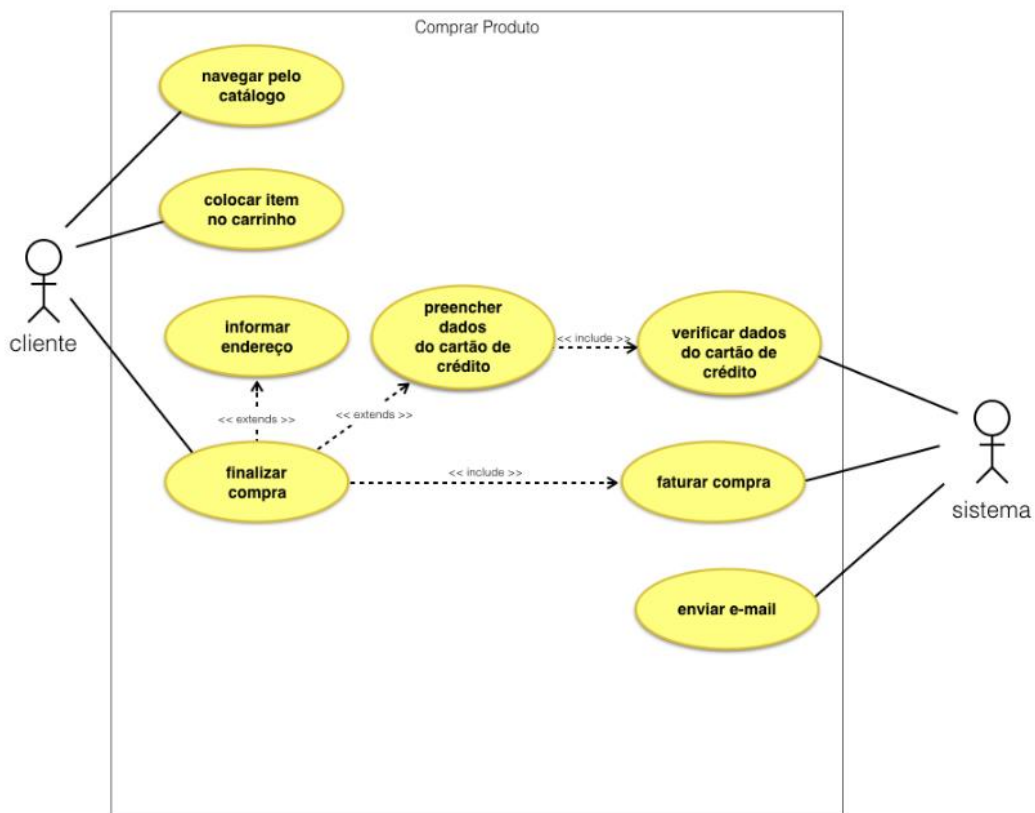
Atualmente as organizações necessitam aperfeiçoar os seus processos de negócio para permanecerem competitivas. Uma das formas de visualizar ou definir esses processos é através da construção de modelos. A modelagem dos processos de negócio ajuda na identificação e entendimento das atividades realizadas por diferentes participantes de uma organização. O processo do negócio consiste em um conjunto de um ou mais procedimentos ou atividades que, coletivamente, realizam um objetivo do negócio dentro de um contexto organizacional [WMC99]. Um modelo de processo de negócio é uma representação visual dos elementos que formam esse processo. Esses elementos podem ser tarefas, participantes, transições, documentos, arquivos, entre outros. [...] Para a modelagem de sistemas a linguagem utilizada como padrão atual na indústria é a UML [UML11]. A UML possui diferentes diagramas que podem ser usados para modelagem dos diversos aspectos do sistema. Dentre esses diagramas pode-se destacar o diagrama de casos de uso. Esse diagrama permite obter uma modelagem da visão dos casos de uso do sistema mostrando os casos de uso, os atores e seus relacionamentos (HEREDIA, 2012, p.27).

O que comprova o quão vantajoso é um diagrama UML e demonstrado sua preferência pela indústria. Para o estudo do desenvolvimento do sistema de ticket, foi feito um diagrama de casos de uso, conforme explicado a seguir:

O diagrama de Casos de Uso é uma técnica que modela os requisitos de um sistema, ou seja, demonstra quais são as operações previstas. Esse diagrama é composto por atores (elementos externos ao sistema que inicia ou recebe um valor de um caso de uso), casos de uso (serviços fornecidos aos usuários) e interações (estímulos recebidos pelo sistema) (VITAL R.; VITAL T. 2015, p.68).

Mostrando uma imagem de como cada usuário vai interagir com um sistema, ofereceu-se uma ideia melhor do projeto, podendo prepará-lo para realizar devida tarefa da maneira correta. Utiliza-se o Diagrama 1 como exemplo de um diagrama de casos de uso em uma compra de produto.

DIAGRAMA 1: Exemplo de um Diagrama de Casos de Uso



Fonte: OperacionalTI, UML.

Utilizou-se, também, um diagrama das classes, porém, antes de se explicar o que é o diagrama de classes, primeiramente, será explicado o que é uma classe para um sistema.

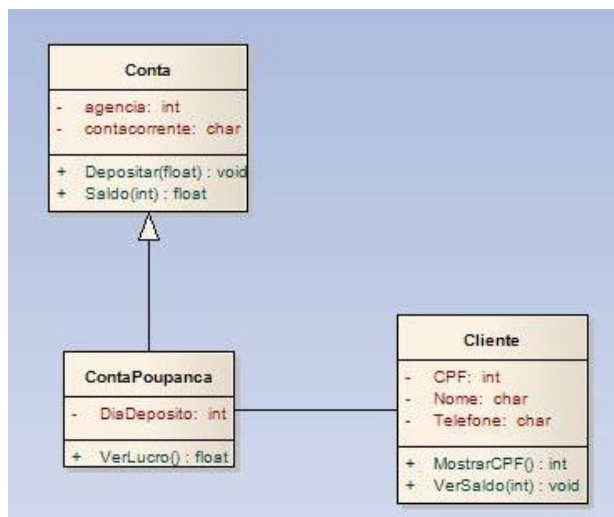
Considerando a realidade onde o conceito de Classes surgiu, no contexto de produção software, podemos entender que é uma Classe é uma abstração de um objeto da vida real (vida real que será tratada via software), que agrupa dados (atributos) e procedimentos (operações) relacionados ao seu contexto (VENTURA, 2018, on-line).

Abstraindo assim objetos da vida real para o *software* com o intuito de agrupar dados e realizar operações dentro do sistema, justamente para ter uma visão melhor dessas classes, foi criado um diagrama, explicado por Tybel (2016) a seguir:

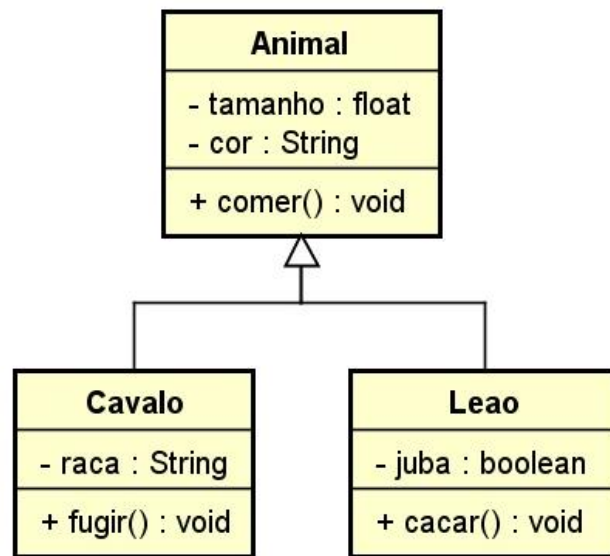
Em programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. Podemos afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes. Na Unified Modeling Language (UML) em diagrama de classe, uma classe é representada por um retângulo com três divisões, são elas: O nome da classe, seus atributos e por fim os métodos (TYBEL, 2016, on-line).

Tendo como exemplo um diagrama de classes, o Diagrama 2, no qual se pode observar o nome de cada classe na parte de cima de cada quadrado, seus atributos, as informações que cada um armazena, na parte do meio e no final, seus métodos e o que será que cada classe vai realizar. Podemos ver também, no Diagrama 3, a realização de uma herança, na qual as classes “Leão” e “Cavalo” compartilham o atributo tamanho e cor e ambos precisam se alimentar de alguma forma, então foi criado uma classe mais geral, chamada animal, que irá passar esses atributos e métodos para as classes “Leão” e “Cavalo” que, por sua vez, terão seus próprios atributos e métodos, além daqueles que herdaram.

DIAGRAMA 2 – Exemplo de um diagrama de classes



Fonte: Pleon's Blog, UML – Diagrama de Classes.

DIAGRAMA 3 – Exemplo de uma herança em um diagrama de classes

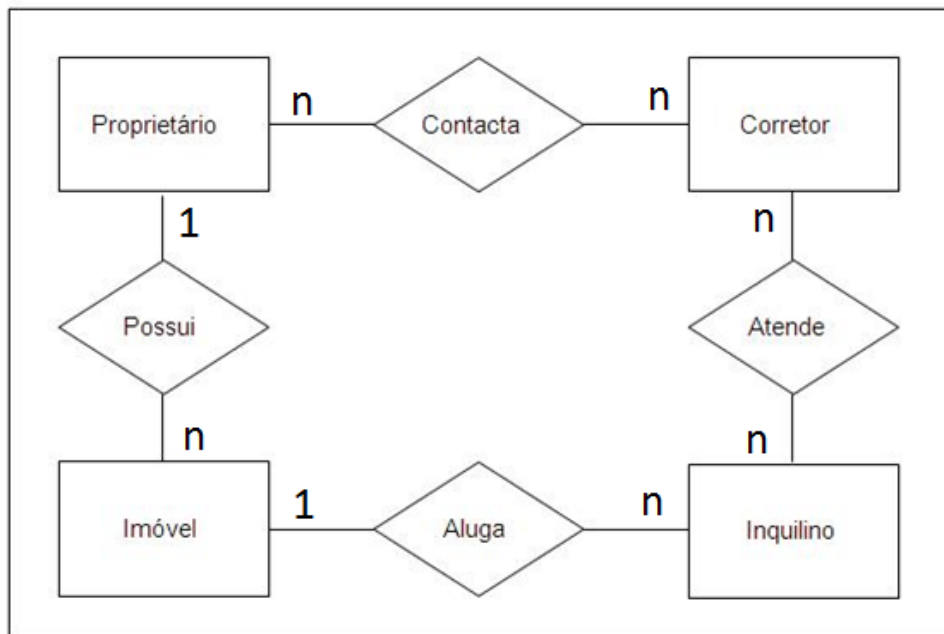
Fonte: Space Programmer. Introdução as Classes Associações e Generalizações.

E, por final, um diagrama de Entidade relacionamento.

O Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos). [...]

Enquanto o MER é um modelo conceitual, o Diagrama Entidade Relacionamento (Diagrama ER ou ainda DER) é a sua representação gráfica e principal ferramenta. Em situações práticas, o diagrama é tido muitas vezes como sinônimo de modelo, uma vez que sem uma forma de visualizar as informações, o modelo pode ficar abstrato demais para auxiliar no desenvolvimento do sistema. Dessa forma, quando se está modelando um domínio, o mais comum é já criar sua representação gráfica, seguindo algumas regras (RODRIGUES, 2014, on-line).

Utilizando justamente esse diagrama para poder criar uma ideia das entidades no sistema e como elas irão relacionar entre si, de forma clara, podendo analisar e fazer as mudanças necessárias enquanto em fase de planejamento. Para um exemplo de um diagrama entidade relacionamento, pode -e utilizar o Diagrama 4, no qual vários imóveis possuem um proprietário, nele, vários proprietários contatam vários corretores e vários corretores atendem vários inquilinos e esses inquilinos alugam um imóvel.

DIAGRAMA 4 – Exemplo de uma Diagrama Entidade-relacionamento

Fonte: Aluno Pragmático, Conceitos e arquitetura de bancos de dados relacional.

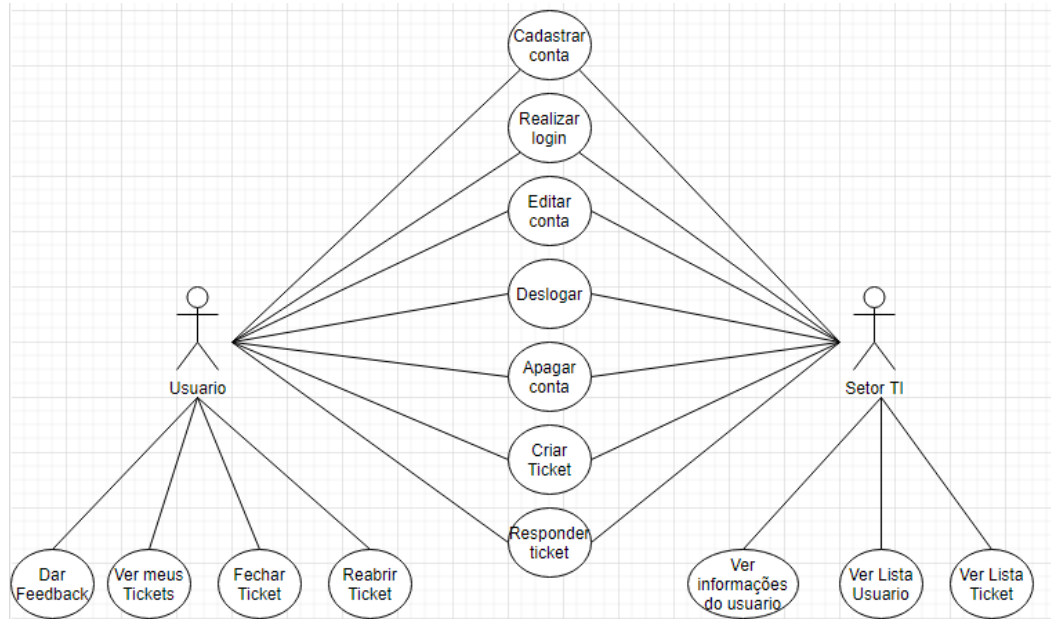
Por meio desses diagramas, é possível realizar um levantamento de requisitos do sistema visualizando a estrutura do sistema por meio dos esquemas criados assim, tornando a etapa de desenvolvimento mais simples e amigável ao programador. E, apesar de ser contraintuitivo, o desenvolvedor deve dar ao trabalho de criar esses diagramas para acelerar o processo, os diagramas em si auxiliam a fazer um trabalho bem feito em uma primeira tentativa, ou seja, evitam retrabalho em cima de projetos mal feitos, resultando assim em um sistema pronto de forma mais rápida.

3.6 Requisitos e Modelagem do Sistema

Com as informações adquiridas através das pesquisas, foi vista a importância de um sistema organizado na hora de solucionar um problema e como um ticket é capaz de fornecer essas necessidades. Então, com o intuito de analisar melhor o ambiente em que o sistema será desenvolvido, foi criado um diagrama de casos de uso (Diagrama 5), resumindo os detalhes dos usuários do sistema e também mostrando suas interações com o sistema. A partir desse diagrama, criou-se o Diagrama de Classes (Diagrama 6) para obter informações mais detalhadas da própria programação já trazendo uma ideia das classes que deverão ser criadas e como irão ser criadas, com quais atributos e quais métodos. E também, ao final, um Diagrama de Entidade-Relacionamento (Diagrama 7) que servirá para visualizar melhor como as

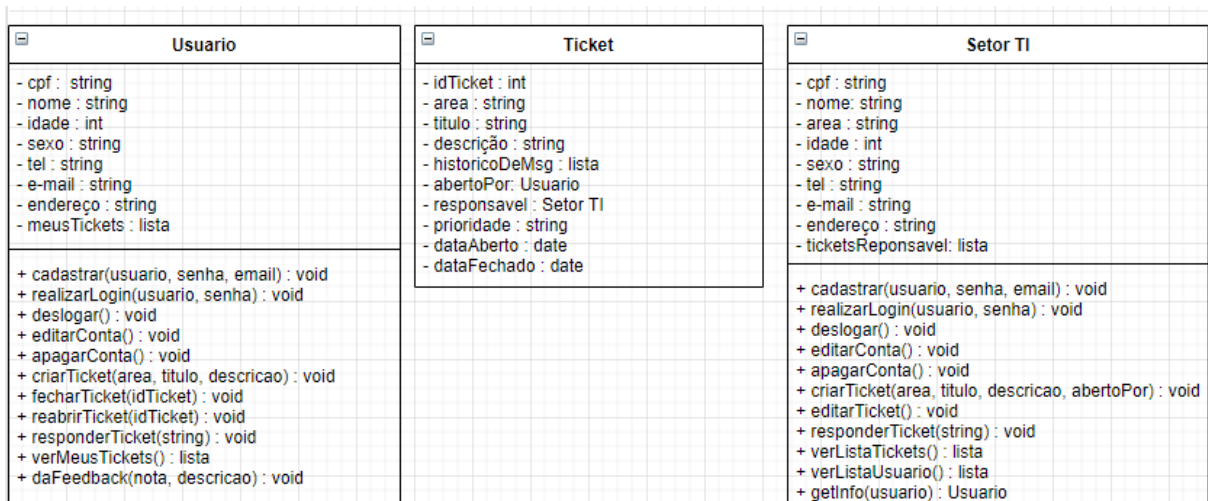
entidades do sistema irão relacionar entre si, permitindo um sistema construído com mais precisão devido a análise prévia.

DIAGRAMA 5 – Diagrama de casos de uso do sistema Helper

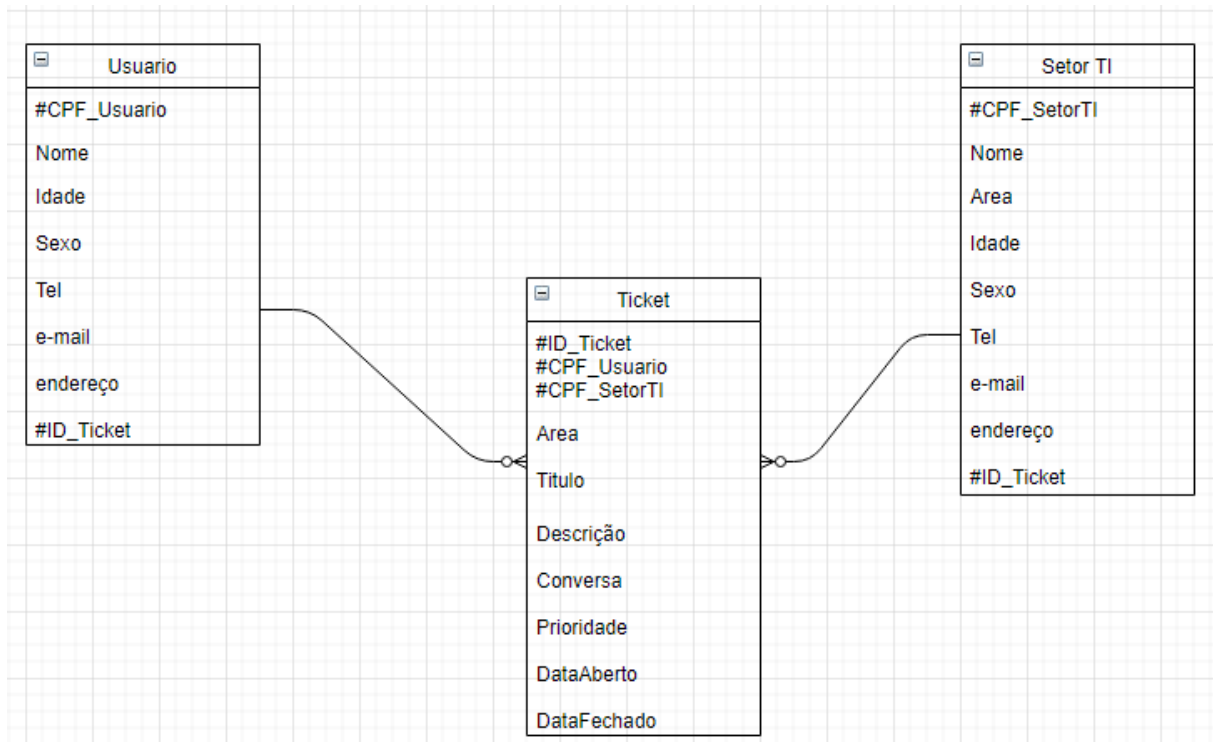


Fonte: Acervo Pessoal.

DIAGRAMA 6 – Diagrama de classes do sistema Helper



Fonte: Acervo Pessoal.

DIAGRAMA 7 – Diagrama Entidade-Relacionamento do sistema Helper

Fonte: Acervo Pessoal.

Por meio dos diagramas criados a partir do sistema Helper (Apêndice A), tem-se uma melhor visão do que será construído, conhecendo melhor os limites do software e o que ele vai fazer já podendo acrescentar funções ou retirar funções do sistema antes mesmo de sua criação, evitando a necessidade de retrabalho.

4 CONSIDERAÇÕES FINAIS

Como forma de enfrentar os diversos problemas de organização na hora de relatar um problema e também de endireitar este problema, o sistema de ticket oferece uma grande vantagem em frente a essa dificuldade por dar suporte a ambos os lados como também auxiliar na aquisição de dados, tanto de *feedback* dos clientes, que promoverá um sistema de avaliação com o intuito de identificar as partes boas e as partes ruins de um atendimento, como de informações dos infortúnios já resolvidos para, caso o problema volte a ocorrer, ou haja um derivado do mesmo problema, o setor responsável, tenha acesso as resoluções feitas anteriormente, fornecendo conhecimentos que podem resultar em uma resolução rápida do novo problema.

5 REFERÊNCIAS

ALMEIDA, M. R.; BRITO C. S.; RODRIGUES, H. D. I.; GOMES, É. V. B.; CARVALHO, A. K. P.; OLIVEIRA, B. D.; GALDINO, S. D. A. V.; SANTOS, F. E. R. Gestão da informação: revisão integrativa de literatura, Brasília. **Revista Gestão & saúde**, v.6, n.3, pp.2819-2850, 2015. Disponível em: <<https://periodicos.unb.br/index.php/rgs/article/view/3269/2954>>. Acesso em: 24 jun. 2021.

BASTISTA, R. L.; OLIVEIRA, A. F. Antecedentes da confiança do empregado na organização, Natal. **SciELO**, v.17, n.2, pp.247-254, 2012. Disponível em: <<https://www.scielo.br/j/epsic/a/K7LFW3KpbTdh4j3PRTH4CVx/?format=html>>. Acesso em: 24 jun. 2021.

BLOG TECLÓGICA, **Gestão da informação: entenda a importância e saiba como colocar em prática**. 2017. Disponível em: <<https://blog.teclogica.com.br/gestao-da-informacao-como-colocar-em-pratica/>>. Acesso em: 1 jul. 2021.

CASARIL, M. Tudo o que você precisa saber sobre um sistema de tickets. **Movidesk Blog**, 2019. Disponível em: <<https://conteudo.movidesk.com/tudo-sobre-sistema-de-tickets/>>. Acesso em: 24 jun. 2021.

FILHO, A. M. S. Ouvir o cliente e reconhecer o problema: ingredientes essenciais à gestão de projetos. **Revista Espaço Acadêmico**, v.11, n.125, pp. 35 – 40. Disponível em: <<https://periodicos.uem.br/ojs/index.php/EspacoAcademico/article/view/14926/7976>>. Acesso em: 2 jul. 2021.

HENRRIQUE, J. **POO**: o que é programação orientada a objetos?. Alura, 2019. Disponível em: <https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos?gclid=Cj0KCQjw_dWGBhDAARIsAMcYuJxhPfpEQdy_SYfmLVM3cftuNGgypuo3Y_ky2pX1up_Ueth0QsMK8mMaAr6IEALw_wcB>. Acesso em: 24 jun. 2021.

HEREDIA, L. R. **Transformação de modelos de processos de negócio em BPMN para modelos de sistema utilizando casos de uso da UML**. Pontifícia Universidade Católica do Rio Grande do Sul. 2012. Disponível em: <<https://repositorio.pucrs.br/dspace/bitstream/10923/1621/1/000438602-Texto%2bCompleto-0.pdf>>. Acesso em: 3 jul. 2021.

JUSTO, A. S. **Gestão de projetos**: o que é, principais conceitos e benefícios e como fazer em 5 passos. EUAX Blog 2018. Disponível em: <euax.com.br/2018/08/o-que-e-gestao-de-projetos/>. Acesso em: 01 jul. 2021.

LUCIDCHART. **O que é um diagrama UML?** s.d. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-uml>>. Acesso em: 3 jul. 2021.

NORO, G. B. A Gestão de Stakeholders em Gestão de Projetos. **Revista de Gestão e Projetos**, v.3, n.1, pp.127-158, 2012. Disponível em: <<https://doaj.org/article/319d5faf45df49f2b91878aa17fba32>>. Acesso em: 24 jun. 2021.

OLIVEIRA, R. D. Análise do uso da cor no Diagrama de Classes da Linguagem Unificada de Modelagem (UML). **INFODESIGN: BRAZILIAN JOURNAL OF INFORMATION**

DESIGN. 2020, v.17, n.1, pp.116–130. Disponível em: <<https://infodesign.org.br/infodesign/article/view/783/469>>. Acesso em: 4 jul. 2021.

PROJECT BUILDER. Gestão de Projeto: O que é e para que serve? 2021. Disponível em: <<https://www.projectbuilder.com.br/blog/gestao-de-projeto-o-que-e-e-para-que-serve/>>. Acesso em: 2 jul. 2021.

RAMALHO, J. F. F. R. Sistema de e-Tickets para Serviços de Atendimento. Universidade de Évora. pp.1-62, 2018. Disponível em: <<https://dspace.uevora.pt/rdpc/bitstream/10174/23217/1/Mestrado%20-%20Engenharia%20Inform%C3%A1tica%20-%20Jo%C3%A3o%20Filipe%20Fernandes%20Fonseca%20Ribeiro%20Ramalho%20-%20Sistema%20de%20e-Tickets%20para%20servi%C3%A7os%20de%20atendimento.pdf>>. Acesso em: 04 jul. 2021.

RODRIGUES, J. Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER). **DEV Media**. 2014. Disponível em: <<https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>>. Acesso em: 5 jul. 2021

SBARDELOTTO, B.; SEVERO, E. A.; GUIMARÃES, J. C. F.; CAPITANIO, R. P. R. Uma exploração bibliométrica das práticas de gestão de projetos e a vantagem competitiva sustentável. **Revista Produção e Desenvolvimento**, v.3, n.2 pp.52–64. Disponível em: <<https://revistas.cefet-rj.br/index.php/producaoedesarrollo/article/view/e232/168>>. Acesso em: 2 jul. 2021

SOARES, J. L. P.; ROSA, M. V. C.; SILVA, F. A. UMA FERRAMENTA WEB PARA AUXILIAR O DESENVOLVIMENTO DE RELATÓRIOS EM PDF PARA SISTEMAS EM PHP. **Colloquium Exactarum**, v.3, n.2, on-line, 2013. Disponível em: <<http://revistas.unoeste.br/index.php/ce/article/view/709/917>>. Acesso em: 24 jun. 2021

TYBEL, D. Orientações básicas na elaboração de um diagrama de classes. **DEV Media**. 2016. Disponível em: <<https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>>. Acesso em: 4 jul. 2021.

TORRES, R. F.; NEVES J. T. R. Gestão estratégica da informação: estudo de caso em uma prestadora de serviços de tecnologia da informação. **DataGramaZero**, v.9, n.1, pp.1-18, 2008 Disponível em: <<https://www.brapci.inf.br/index.php/article/download/7577>>. Acesso em: 1 jul. 2021.

VENTURA, P. Entendendo o Diagrama de Classes da UML. **Até o Momento**. 2018. Disponível em: <<https://www.ateomomento.com.br/uml-diagrama-de-classes/>>. Acesso em: 4 jul. 2021.

VITAL R. B. N.; VITAL T. M. Utilização da modelagem UML em um sistema de gerenciamento de uma franquia do setor de alimentação. **Revista Teccen**. 2015. v.8, n.2, pp.65-72. Disponível em: <<https://doaj.org/article/374801d5c15740d9b16da2d7bf9bcce8?frbrVersion=2>>. Acesso em: 2 jul. 2021.