



TICKETMASTER: Sistema de Atendimentos

Walison de Melo Miranda

Ezequias Ferreira de Souza

Curso: Análise e Desenvolvimento de Sistemas Período: 6°

Área de Pesquisa: Ciências Exatas e da Terra

Resumo: Atualmente, toda empresa deve procura ter controle de tudo que está sendo feito para que possa ser gerenciado, para saber a demanda e o que mais está sendo feito dentro do ambiente de trabalho. Com isso, nada melhor que um sistema de atendimentos. O Sistema irá funcionar no registro de atendimentos solicitados pelos clientes da organização. A ideia é que ele funcione de forma segura e consistente para a organização tenha controle de seus atendimentos, deixando atendimentos documentados e registrados para consulta do cliente e posteriormente dos atendentes. O cliente poderá acompanhar a situação do atendimento pelo portal do cliente, onde terá acesso ao *status* do atendimento e o retorno do atendente/analista, onde também poderá ver todos os atendimentos solicitados por ele podendo responder e finalizar, se necessário. Os atendentes e administradores terão o poder de manipular o chamado, ou seja, responder, finalizar, atualizar o *status* e buscar sempre solucionar a solicitação de forma rápida e eficiente. Os atendentes também poderão acessar o controle dos usuários do sistema, cadastrando, desativando e atualizando dados cadastrais, podendo também realizar o cadastro de departamentos, categorias e o *status* do ticket, além de gerar relatórios e ver gráficos de atendimentos em tempo real.

Palavras-chave: Atendimento, Sistemas da Informação, *Ticket*, *Web*.

1.INTRODUÇÃO

Atualmente, toda empresa deve procura ter controle de tudo que está sendo feito para que possa ser gerenciado, para saber a demanda e o que mais está sendo feito dentro do ambiente de trabalho. Com isso, nada melhor que um sistema de atendimentos. Com ele é possível registrar todo um processo de atendimento ao cliente tendo então um controle de tudo que está sendo solicitado.

Saber a quantidade de atendimento gerado na empresa ajuda na tomada de decisão de sua organização, fazendo que os processos melhorem e que tudo seja ajustado e dividido de forma integral, sem sobrecarregar os colaboradores. Com um sistema ao seu lado, auxiliando-o no processo de administração, os processos ficarão mais otimizados.

Nesse processo a forma de administrar fica mais otimizada, pois os atendimentos podem ser divididos entre a equipe fazendo que todos trabalhem de forma produtiva e organizada.

No ambiente de trabalho, em muitas vezes nos deparamos com várias pessoas ligando e/ou mandando *e-mail* solicitando algum serviço e como forma de organização, em muitas das vezes, são criadas planilhas ou até mesmo anotadas em papéis as solicitações enviadas pelos clientes. Imagine, se quisesse verificar um atendimento de um ano atrás para ver qual serviço foi prestado, como foi prestado, tempo e etc., como iria fazer? Possivelmente se estivesse isso anotado em um papel

ou planilha dificilmente acharia e mesmo que achasse levaria um bom tempo procurando.

Já com o um *software* isso não aconteceria, pois os atendimentos seriam armazenados em uma base de dados e a qualquer momento e lugar você poderia ter acesso a eles, de forma rápida, prática e segura, com poucos cliques podendo acessar qualquer atendimento, de qualquer data sem riscos de deixar de atender os clientes.

Assim, pergunta-se: Seria possível a construção de um *software* que auxilie na gestão de atendimentos uma empresa local de tecnologia?

2.DESENVOLVIMENTO

A evolução da tecnologia mudou a forma como as coisas são tratadas. No atendimento ao cliente não foi diferente, com os avanços tecnológicos fez com que a comunicação ficasse ainda mais rápida pelo fato de estar a todo tempo conectado na rede mundial de computadores, a Internet. Existem muitos sistemas que oferecem o atendimento ao cliente de forma rápida e prática fazendo com que a resolução de problemas seja eficaz.

2.1. Atendimento ao Cliente

A evolução da tecnologia transformou tudo, a forma que vivemos, agimos e até a forma em que pensamos. O atendimento realizado de forma virtual diminuiu a distância entre atendentes e clientes, fazendo-se então que todo processo seja agilizado.

Para William H. Davidow, “quanto mais tempo o cliente esperar, mais difícil de providenciar o serviço ao cliente excepcional” (DAVIDOW, 2013). Desse modo, o atendimento rápido assegura a satisfação do cliente e o maior índice de aprovação da empresa.

Já Horst Schulze afirma que “até que você não tenha 100% de seus clientes satisfeitos, você deve aperfeiçoar”. (SCHULZE, 2019)

Assim, deve-se empenhar ao máximo em agradar os clientes, sempre colocando-os em primeiro lugar para que se sintam confortáveis em expressarem seus problemas para que possamos buscar uma solução.

2.1.1 Sistema de Atendimento

Hoje em dia praticamente tudo se faz *online*, dessa forma, tudo que se pede é gerado um atendimento/pedido, como exemplo simples disso, é quando é feito um pedido no *delivery*, onde é feita a requisição ao mercado para seja entregue algum tipo de produto. Um atendimento geralmente contém o nome do solicitante, o assunto, descrição e quem vai dar prosseguimento com a solicitação, o atendente, etc.

O blog Conta Azul (CONTA AZUL, 2021, *online*) define um atendimento (*ticket*) da seguinte maneira:

A ordem de serviço é um documento que formaliza o serviço a ser prestado para um cliente e serve como ponto de partida para a organização do trabalho. Com ela, as empresas prestadoras de serviços conseguem se planejar e organizar as demandas para o atendimento, mantendo o fluxo de trabalho sob controle.

Uma OS pode oficializar atividades como instalação de *software*, limpeza, transporte, manutenção de equipamentos, criação de campanhas publicitárias, consultoria e qualquer outra área de atuação. Ela pode ser criada a partir do pedido de um cliente ou pela determinação de um contrato, por exemplo, e é obrigatória para iniciar qualquer prestação de serviços.

De forma geral, um sistema de atendimento trabalha na coleta de informações das requisições do cliente, processa (atende) e retorna ao cliente a solução e/ou como ele deve proceder sob o retorno cedido. Um simples exemplo é um sistema de atendimento que o cliente realiza uma solicitação, caracterizando a entrada de dados. Logo após, o sistema verifica com base no que foi fornecido pelo cliente onde esse atendimento deve ser enviado (departamento / setor) e, por fim, o atendente verifica a solicitação e o retorna com a solução do problema.

Durante o funcionamento do sistema, o *feedback* do usuário em relação ao sistema é indispensável.

De acordo com Peter Drucker a “qualidade de serviço não é o quanto nos dedicamos ao serviço. É o que o cliente obtém com isso” (DRUCKER, 2020). Ao analisar essa afirmação, pode-se refletir que mesmo que a preparação dos atendentes seja grande a satisfação do cliente é o que importa.

2.3. Ferramentas de Desenvolvimento

Para que ter um *software* de alta qualidade deve-se escolher com precisão qual linguagem e/ou ferramenta para o desenvolvimento. Ao escolher uma linguagem de programação devemos levar em conta se ela é eficiente, confiável, legível, portátil, de rápido aprendizado, e claro se a mesma é orientada a objetos.

A fim de desenvolver o sistema de atendimento TICKETMASTER, as ferramentas de desenvolvimento escolhidas são: o PHP (PHP: *Hypertext Preprocessor*.), *framework Bootstrap*, biblioteca *JQuery*, *JavaScript* e *Ajax (Asynchronous JavaScript and XML)*. Como banco de dados iremos usar o MySQL juntamente com o *phpMyAdmin*.

2.3.1. PHP

Criado em 1994 por Rasmus Lerdof, o PHP (*Hypertext Preprocessor*) é uma linguagem de programação para *web*, de *script open-source*. As páginas PHP contêm HTML (*Hypertext Markup Language*) em código mesclado e é executado no lado servidor, o que possibilita que o *site* seja dinâmico, e garanta mais segurança nas informações, já que o código PHP é todo processado no lado do servidor e o lado do cliente fica responsável apenas por apresentar os dados em tela para o usuário. (PHP.NET, 2021)

Quando foi desenvolvido, o PHP era uma solução para utilização em páginas pessoais de Rasmus Lerdof, seu criador, contudo necessidades foram surgindo e novas versões foram lançadas, contendo novos recursos, como, interpretador de comandos SQL (Linguagem de consulta estruturada), orientação a objetos, herança, polimorfismo, apontadores de memória e outros (PHP.NET, 2021).

Hoje essa linguagem é utilizada em massa, em aplicações referência como *MediaWiki*, *Facebook*, *Drupal*, *Joomla*, *WordPress*, *Magento* e o *Oscommerce*.

Segundo Bento (2014) alguns motivos de usar o PHP e o MySQL são: PHP nasceu para a *web* e sua integração com os servidores é simples, tem uma curva de aprendizado suave, comparada a outras linguagens. PHP e MySQL são tecnologias livres, é fácil de encontrar serviços de hospedagem que oferecem PHP e MySQL. Serviços de hospedagem que as oferecem são mais baratos que serviços semelhantes para outras tecnologias e ainda, o MySQL é leve e rápido, mesmo para quantidades razoavelmente grandes de dados.

2.3.2. *Bootstrap*

O *Bootstrap* é um *framework* de código aberto para desenvolvimento *front-end* de uma aplicação, ou seja, a parte gráfica da página *web*. Com ele é possível estilizar páginas *web* e torná-las responsivas com poucas linhas de código, é possível estilizar botões, textos, ícones, e muitos outros elementos de tela.

Basicamente, o *Bootstrap* é um *framework* CSS para ser utilizado no *front-end* de aplicações *web*. Ele utiliza *JavaScript* e CSS para estilizar as páginas e adicionar funcionalidades que vão além de apenas proporcionar um visual bonito ao *site* (TRYBE, 2021).

Junto ao *JQuery* podemos criar vários efeitos de tela, como mostrar e esconder elementos com um simples clique, mostrar janelas modais e muitas outras funcionalidades desse *framework* poderoso, que é o mais utilizado quando o assunto é desenvolvimento *web*.

2.2.3. *Ajax*

O *Ajax* é um conjunto de técnicas de desenvolvimento voltado para a *web*, com ele é possível criar aplicações que trabalhem de forma assíncrona, processando requisições ao servidor em segundo plano (HOSTINGER, 2021).

Com o *Ajax* pode-se fazer requisições ao servidor sem que precise recarregar a página. Um exemplo simples do uso do *Ajax* é quando se digita na barra de pesquisa do navegador, ele nos traz outras possíveis formas de pesquisa (HOSTINGER, 2021).

2.2.4. Banco de Dados: *MySQL*

MySQL é o banco de dados de código aberto mais popular do mundo, que possibilita a entrega econômica de aplicações de banco de dados confiáveis, de alto desempenho e escaláveis, com base na *web* e incorporadas (ORACLE, 2014).

Segundo o vice-presidente e analista diretor da *Constellation Research*, *Holger Mueller*, em seu depoimento ao *Oracle* fez a seguinte afirmação:

A introdução do novo *MySQL Database Service* reduz tarefas não produtivas e demoradas de administração do banco de dados e permite que os desenvolvedores se concentrem na inovação. O *MySQL Database Service* agora adiciona um *Analytics Engine* integrado, que fornece uma única plataforma unificada para OLTP e OLAP, permitindo um conjunto completo de novos aplicativos de última geração, eliminando a necessidade de desenvolvedores dependerem de vários bancos de dados e ferramentas para ETL entre bancos de dados conforme exigido com outros fornecedores de nuvem. (*Constellation Research, Holger Mueller, 2021*)

No *MySQL* ficam armazenadas todas as informações dos usuários, documentos e caminhos de imagens mantidos pelo sistema proposto. As operações suportadas por esse banco são inclusão, exclusão, atualização de registros, alterações de estrutura e outras.

O *MySQL* possui um excelente desempenho e possui uma fácil integração com o PHP, tornando o desenvolvimento de aplicações *web* simples e robusto.

2.2.4. PhpSpreadsheet

O *PhpSpreadsheet* é um *framework* desenvolvido pela PHPOffice para geração de planilhas usando o PHP. Com ele é possível gerar planilhas das mais diferentes formas, podendo adicionar fórmulas, cores, filtros e entre outras operações que é possível fazer com uma planilha. O *PhpSpreadsheet* é compatível com os mais diversos tipos de planilhas, incluindo o *Excel*, *Google Sheets* e *LibreCalc*.

2.2.5. Dompdf

Dompdf é um conversor de HTML para PDF. Na documentação disposta no *GitHub* define a ferramenta da seguinte forma:

Em sua essência, *dompdf* é (principalmente) um mecanismo de renderização e layout HTML compatível com CSS 2.1 escrito em PHP. É um renderizador baseado em estilo: ele baixa e lê folhas de estilo externas, *tags* de estilo embutidas e os atributos de estilo de elementos HTML individuais. Ele também suporta a maioria dos atributos HTML de apresentação. (DOMPDF,)

Com essa ferramenta pode ser gerada as mais variadas formas de relatórios em PDF passando apenas o código HTML e CSS para geração da interface. Ele possui várias configurações, como margens, tamanho da página, estilo da página (horizontal ou vertical), cores, fontes entre outras configurações (GITHUB.COM, 2021).

2.2.6. ApexCharts.js

O *ApexCharts* é um a biblioteca *JavaScript* para criação de gráficos personalizados. Em sua documentação oficial no *GitHub* eles definem a ferramenta da seguinte forma:

“Uma moderna biblioteca de gráficos *JavaScript* que permite criar visualizações de dados interativos com API simples e mais de 100 amostras prontas para uso. Embalado com os recursos que você espera, *ApexCharts* inclui mais de uma dúzia de tipos de gráficos que fornecem visualizações atraentes e responsivas em seus aplicativos e painéis” (GITHUB.COM, 2021).

Alguns dos gráficos suportados são: Gráficos de linha, rosca, *pizza*, área, coluna, linha do tempo, misturado (faz a junção de alguns gráficos), bolha, castiçal, entre outros.

Na Figura 1 temos imagens de algumas opções de gráficos:

Figura 1 – Exemplos de gráficos.



Fonte: <https://apexcharts.com/javascript-chart-demos/>

2.2.7. SweetAlert2

O *SweetAlert2* é uma das bibliotecas mais utilizadas do *JavaScript*. Com ela é possível criar alertas e mensagens personalizados. Com fácil instalação, em poucos minutos já é possível usar os benefícios dessa biblioteca.

Na Figura 2, temos um exemplo de alerta que é possível criar usando o *SweetAlert*.

Figura 2 - Exemplo de alerta.



Fonte: <https://sweetalert2.github.io/>

3. METODOLOGIA

Nesta seção será abordado os tipos de pesquisa e métodos usados para construção do sistema.

3.1. Tipo de Pesquisa

Foi adotado para desenvolvimento deste trabalho a metodologia de pesquisa exploratória com base em levantamento bibliográfico, a fim de compreender os processos e formatos de atendimento. Esse tipo de pesquisa tem por objetivo desenvolver, esclarecer e modificar conceitos e ideias, tendo em vista a formulação

de problemas mais precisos ou hipóteses pesquisáveis para estudos posteriores. (GIL, 2008, p. 27).

3.2. WAMP Server (*Windows, Apache, MySQL e PHP*)

Para que o Sistema de atendimento TICKETMASTER funcione, é necessário ter instalado e rodando um servidor para trabalhar com PHP e MySQL. O WAMP Server é um servidor que contém em um único pacote o Apache, MySQL e PHP para o sistema operacional Windows. Existem outros servidores como XAMP e EasyPHP.

Segundo Erick Eduardo Petrucelli (Linguagem de programação PHP, p.21)

O WAMP se diferencia por oferecer tanto o MySQL quanto o MariaDB, o que pode ser bem interessante para quem quer mais versatilidade, ou ocupar espaço para quem não quer. O bom é que, através de menus simples, permite desativar qual não estiver sendo utilizado. Também permite alternar entre versões PHP e até remover aquela que você realmente não pretenda mais utilizar. Também possui uma interface Web padrão com alguns recursos úteis. [...] ele é o único deles que vem nativamente com tradução para o português, embora algumas coisas continuem em inglês, facilitando compreender os menus e as configurações existentes. (Erick Eduardo Petrucelli, 2019)

Usando o WAMP é permitido configurar o servidor da forma que for necessário, podendo trocar as versões do PHP e MySQL e até mesmo mudar o nome do *host*.

3.3. JSON

JSON (JavaScript Object Notation - Notação de Objetos JavaScript) é uma formatação leve de troca de dados. Para seres humanos, é fácil de ler e escrever. Para máquinas, é fácil de interpretar e gerar. *JSON* é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens C e familiares, incluindo C++, C#, *Java*, *JavaScript*, *Perl*, *Python* e muitas outras. Estas propriedades fazem com que *JSON* seja um formato ideal de troca de dados (JSON.ORG, 2021).

A cerca de envio e recebimento de requisições de dados via *Ajax* e PHP o uso do *JSON* é uma forma simples e fácil e manipular.

4. RESULTADOS

4.1. Visão geral do Sistema

O Sistema de atendimento TICKETMASTER será dividido em duas partes: Administrativa/Atendente, onde conterá a lista de atendimentos, gráficos, relatórios e administração de usuários/atendentes. E a de acesso do cliente, onde o mesmo poderá abrir seus *tickets* e acompanhá-los.

Ambas partes do sistema é restrita por *login*. Na parte administrativa haverá restrições de acordo com o nível hierárquico do usuário, onde somente o administrador poderá criar, alterar e deletar usuários enquanto os atendentes terão

acesso somente a parte de atendimentos. Já os clientes terão acesso somente para criar, responder e finalizar os atendimentos.

Para que se tenha uma visão mais ampla do sistema, iremos listar alguns passos para que fique mais claro o funcionamento do sistema seguindo as normas da UML (*Unified Modeling Language*).

4.2. Diagramas da UML

Segundo a LucidChart (2021),

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.

4.2.1. Diagrama de caso de uso

O Diagrama de caso de uso apresentado na Figura 10 do APÊNDICE A, mostra as funcionalidades propostas pelo sistema. Exibe qual parte cada usuário deve ter acesso. Administrador tem acesso liberado para acessar todos os módulos do sistema, como cadastro de usuário e manipulação de tickets (criar, responder e finalizar).

4.2.2. Diagrama de Atividade

O Diagrama de Atividade apresentado na Figura 11 do APÊNDICE A, apresenta a interação entre o usuário e a máquina. Serve para descrever o fluxo de controle de uma atividade para outra. No exemplo da Figura 11 é descrito o processo de login, onde o usuário deve fornecer as credencias de acesso. Se as credencias forem validas o acesso é liberado, se não, novas credencias devem ser informadas.

4.2.3. Diagrama de Componentes

Serve para ilustrar as componentes do sistema como devem ser encontradas no sistema, trazendo uma maior organização ao desenvolver. Para melhor entendimento veja a Figura 12 no APÊNDICE A. Na figura é notável a separação das bibliotecas do banco, e do banco da aplicação, através desse diagrama fica mais fácil de ver os componentes e como são distribuídos no sistema.

4.2.4. Diagrama de Sequência

O Diagrama de sequência, nos mostra a sequência que o programa deve fazer de acordo com a interação do usuário para chegar a um objetivo. Na Figura 13 do APÊNDICE A vemos o processo de visualizar um ticket. O usuário informa as credencias de login, se aprovadas segue ao modulo de ticket onde realiza sua tarefa.

4.2.5. Diagrama de Classes

O Diagrama de Classes descreve como devem ser as classes do sistema, será muito útil na construção de tabelas do banco de dados. Ele é um dos diagramas mais importantes de um sistema, na Figura 14 do APÊNDICE A é possível ver uma parte desse diagrama do sistema.

4.2.6. Diagrama de Objetos

Na Figura 15 do APÊNDICE A é possível ver como funciona o diagrama de objetos, nada mais é que a representação dos objetos do diagrama de classe em um determinado estado variado. Nele pode-se ver os tipos de dados e tamanho de um campo de cada objeto do banco.

4.2.7. Diagrama de Máquina de Estado

O diagrama de máquina de estado da Figura 16 do APÊNDICE A revela como deve ser o comportamento ou estado de um componente do sistema. No Exemplo da Figura 16, é descrito o comportamento da máquina para realizar o atendimento de um ticket.

5 - SISTEMA

Nesta sessão será listado requisitos para que seja possível desenvolver o Sistema atendimento TICKETMASTER.

5.1. Requisitos do Sistema

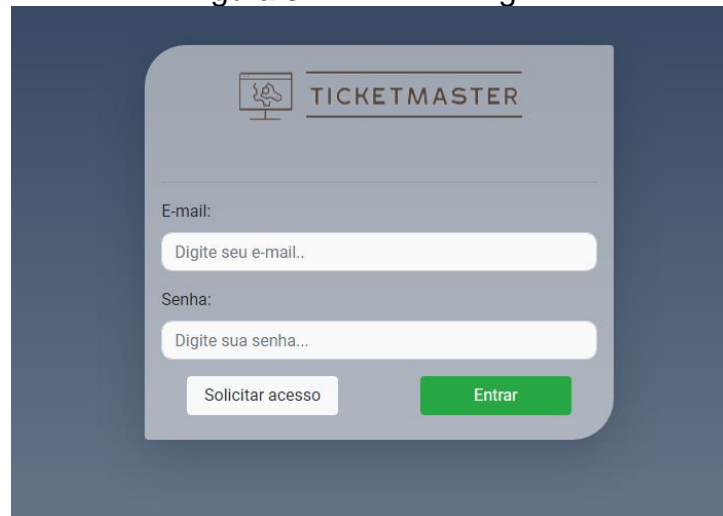
Para que o Sistema de atendimentos, TICKETMASTER, funcione corretamente, existem alguns requisitos que devem ser atendidos. Para que o sistema funcione é necessário que se tenha um servidor PHP com versão superior a 7.0, tenha suporte a MySQL e que tenha conexão com a Internet, para usar bibliotecas externas, como o *Bootstrap*, *Jquery* e *Ajax*.

Também é necessário a inclusão das bibliotecas *Dompdf* e *PhpSpreadsheet*, ambas podem ser instaladas via *Composer* ou baixando o código fonte em seus *sites* oficiais.

5.2. Funcionamento do sistema

Para o eventual uso do sistema, o usuário deve fazer o *login*. Para isso, basta que o usuário forneça seu *e-mail* e senha.

Figura 3 – Tela de “Login”.

A imagem mostra a interface de login do sistema Ticketmaster. No topo, há o ícone de uma pessoa e o nome "TICKETMASTER". Abaixo, há dois campos de entrada: "E-mail:" com o placeholder "Digite seu e-mail.." e "Senha:" com o placeholder "Digite sua senha...". Na base da interface, há dois botões: "Solicitar acesso" (branco) e "Entrar" (verde).

Fonte: Acervo pessoal.

Caso o usuário ainda não tenha acesso ao sistema, ele pode solicitar clicando em “Solicitar acesso” na parte inferior esquerda da tela e preencher os campos solicitados, conforme a Figura 4.

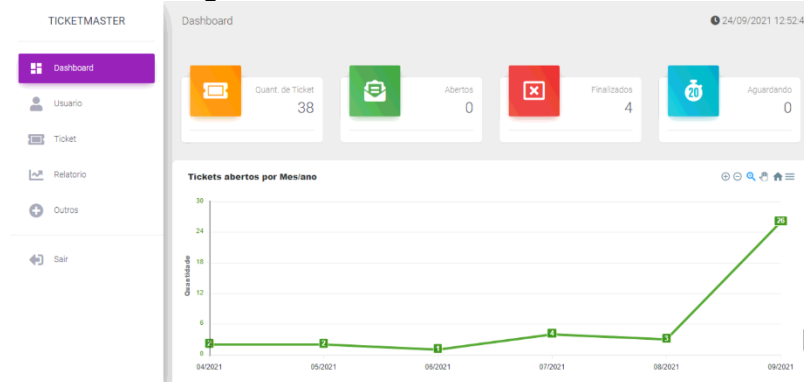
Figura 4 – Tela de “Solicitação de acesso”.

A imagem mostra uma janela modal intitulada "Solicitação de acesso". Ela contém três campos de entrada: "Nome:" com o placeholder "Insira seu nome completo...", "E-mail:" com o placeholder "Digite seu e-mail..." e "Função/cargo:" com o placeholder "Digite o cargo/função de trabalho...". Na base da janela, há dois botões: "Fechar" (branco) e "Solicitar" (verde).

Fonte: Acervo pessoal.

Após a realização do *login*, o usuário será levado a página inicial do sistema, como mostra a Figura 5. Agora cabe ao usuário escolher qual ação ele quer tomar, ele pode adicionar um novo usuário (Figura 6), ou ver se tem algum *ticket* aberto (Figura 5), por exemplo.

Figura 5 – Tela inicial do sistema.



Fonte: Acervo pessoal.

Na Figura 6, é onde é feito o controle de usuários do sistema. Nela é possível adicionar, editar e desativar um usuário, além de trocar o departamento e senha.

Figura 6 – Cadastro de usuário.

The "Controle de Usuário" page includes a search bar and a "NOVO USUÁRIO" button. The user list is as follows:

Nome	E-mail	Departamento	Tipo	Situação	Ações
TICKETMASTER	ticketmaster@ticketmaster.com	Administração	Administrador	Ativo	[Ver] [Desativar]
Joaquim Almorim Santos	joaquim@gmail.com	Tecnologia Da Informação	Cliente	Ativo	[Ver] [Desativar]
Matheus Santos	matheus_birl@seudominio.com.br	Administração	Cliente	Ativo	[Ver] [Desativar]
Alex Freitas	alex.freitas@mail.com	Contabilidade	Cliente	Ativo	[Ver] [Desativar]
Joao Maria	joao.maria@mail.com	Administração	Administrador	Ativo	[Ver] [Desativar]
Marcio Dos Anjos	marcim@gmail.com	Tecnologia Da Informação	Cliente	Ativo	[Ver] [Desativar]
Walison Miranda	walison.miranda@ticketmaster.com	Administração	Administrador	Ativo	[Ver] [Desativar]

Fonte: Acervo pessoal.

Na Figura 7, é a página onde é realizado os atendimentos, essa página possui uma navegação de abas, onde cada aba apresenta um tipo o *status* do *ticket*.

- Aba novos: Apresenta os *tickets* recém abertos, sem nenhum atendente.
- Aba Vinculados: Após realizar o atendimento do *ticket* ele irá aparecer nessa aba.
- Aba Respondidos: Quando o cliente responder ao *ticket*, irá aparecer nessa aba.
- Aba Todos: Apresenta os *tickets* vinculados e respondidos.
- Aba Finalizados: Apresenta todos os *tickets* finalizados.

Figura 7 – Tela de atendimento.

Protocolo	Cliente	Assunto	Status	Data/Hora	Atendente
54	Walison Miranda	Demissão de funcionario	Aberto	25/09/2021 16:35:50	Sem Atendente
53	Walison Miranda	Custo Medio	Aberto	25/09/2021 16:34:50	Sem Atendente
52	Walison Miranda	Computador esta travando	Aberto	25/09/2021 16:34:11	Sem Atendente

Fonte: Acervo pessoal.

Na página de relatórios (Figura 8), é possível ver e exportar em Excel, CSV e PDF alguns relatórios previamente cadastrado no sistema. O cadastro dos relatórios deve ser solicitado ao administrador do sistema.

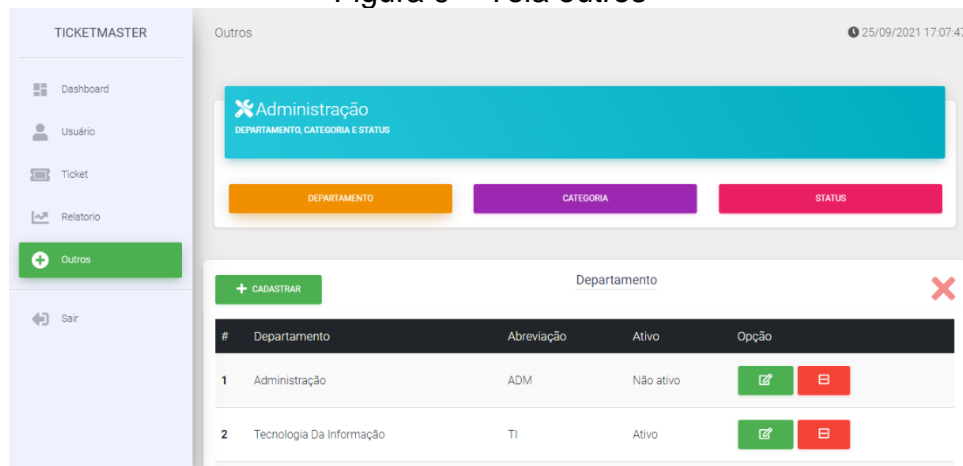
Figura 8 – Tela de relatórios.

Nome	E-mail	Departamento	tipo	Ativo	Criado em	desativado em
Marcelo Marcelino Silva	Marcelo.Silva@Outlook.Com	Administração	Administrador	0	17/08/2021 11:48	00/00/0000 00:00
Rayane Franklin	Rayane.Franklin@Email.Com	Administração	Atendente	0	17/08/2021 11:48	00/00/0000 00:00
Roberval De Souza Nunes	Robervalnunes324@Hotmail.Com	Telefonia	Atendente	0	25/08/2021 20:08	23/09/2021 12:09
Usuario Teste	Teste.Usuario@Mai.Com.Br	Juridico	Atendente	0	21/09/2021 11:09	00/00/0000 00:00

Fonte: Acervo pessoal.

Na página outros (Figura 9), onde é realizado o cadastro de departamentos, categorias e status. Na tela fica disposta em abas as opções. Todas as abas apresentam o mesmo designer, botão de cadastrar, editar e desativar.

Figura 9 – Tela outros



Fonte: Acervo pessoal.

6. CONCLUSÃO

Poder gerenciar, ter controle de tudo que está sendo solicitado a sua empresa de forma simples é de grande vantagem. Mas a grande questão é: Seria possível a construção de um *software* que auxilie na gestão de atendimentos uma empresa local de tecnologia? Sim, é possível. Com o sistema apresentado anteriormente é possível realizar isto, sem dúvidas. Com isso, o presente trabalho até aqui apresentado nos mostrou como funciona um sistema de atendimento, que tem como objetivo solucionar seguintes problemas: falta de controle de responsabilidades e serviços para os atendentes e administradores, e maior facilidade de interação e solução de problemas para os clientes.

Assim, espera-se que o sistema de atendimento TICKETMASTER permita um maior controle sobre aquilo que está sendo solicitado, podendo então ajudar na tomada de decisão de uma empresa perante os problemas solicitados pelos clientes, facilidade de atendimento com plataforma intuitiva, rápida e segura. E para os clientes a satisfação de se ter uma aplicação moderna, interativa e segura para realizar suas solicitações.

Pensado em melhorar sempre o atendimento aos clientes, as próximas versões da aplicação irão conter melhorias como painel de mensagens instantânea, API para que você possa gerar seus próprios relatórios e *dashboard* customizável, correção de *bugs*, isso tudo sempre prezando a segurança, rapidez e interatividade. Após essas implementações, projeta-se fazer com que o sistema seja disponibilizado para teste publicamente na web, onde quaisquer pessoas podem realizar o cadastro e realizar o teste da aplicação por um determinado período de tempo.

Acerca da pesquisa realizada, destaca-se aprendizagem em todos processos de desenvolvimento, técnicas implementadas e atendimento ao cliente. Com o método de pesquisa apresentado, destaca-se o amplo aprendizado e compreensão da forma que ela é aplicada, possibilitando uma facilidade para obter informações fazendo-se então primordial para todo projeto e escrita de artigos/documentação.

7. REFERÊNCIAS BIBLIOGRÁFICAS

A beautiful, responsive, customizable, accessible (wai-aria) replacement for javascript's popup boxes. **SweetAlert**. 2021. Disponível em: <<https://sweetalert2.github.io/>>. Acesso em: 20 abr. 2021.

BENTO, Evaldo Junior. **Desenvolvimento web com PHP e MySQL**. São Paulo: Casa do Código, 2014.

Bootstrap: o que é, como usar e para que serve esse framework? **Trybe**. 2021. Disponível em: <<https://blog.betrybe.com/framework-de-programacao/o-que-e-bootstrap/>>. Acesso em 10 mar. 2021.

COBRA, Marcos Henrique Nogueira. **Marketing Básico**. São Paulo: Atlas, 1997.

DAVIDOW, William H., UTTAL, **Bro. Serviço Total ao Cliente, a Arma Decisiva**. Rio de Janeiro: Campus, 1991.

Dompdf is an HTML to PDF converter. **Dompdf**. 2021. Disponível em: <<https://github.com/dompdf/dompdf>>. Acesso em: 10 mai. 2021.

Introdução ao JSON. **Json**. 2021. Disponível em: <<https://www.json.org/json-pt.html>>. Acesso em: 26 mai. 2021.

KOTLER, Philip e KELLER, Kevin Lane. **Administração de Marketing**. São Paulo: Pearson Prentice Hall, 2006.

Modern & Interactive Open-source Charts. **ApexCharts**. 2021. Disponível em: <<https://apexcharts.com/>>. Acesso em: 15 fev. 2021.

MySQL Database Service. **Oracle**. 2021. Disponível em: <<https://www.oracle.com/br/mysql/>>. Acesso em: 25 mai. 2021.

O Que é AJAX e Como Funciona? **Hostinger**. 2021. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-ajax>>. Acesso em: 30 mar. 2021.

O que é o PHP? **PHP**. 2021. Disponível em: <https://www.php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 12 mar. 2021.

O que é um diagrama UML? **LucidChart**. 2021. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-uml>>. Acesso em: 25 abr. 2021.

Spitaliere, Juliana. **Ordem de serviço: o que é, para que serve e como emitir**. 2020. Disponível em: <<https://blog.contaazul.com/ordem-de-servico>>. Acesso em: 15 mar. 2021.

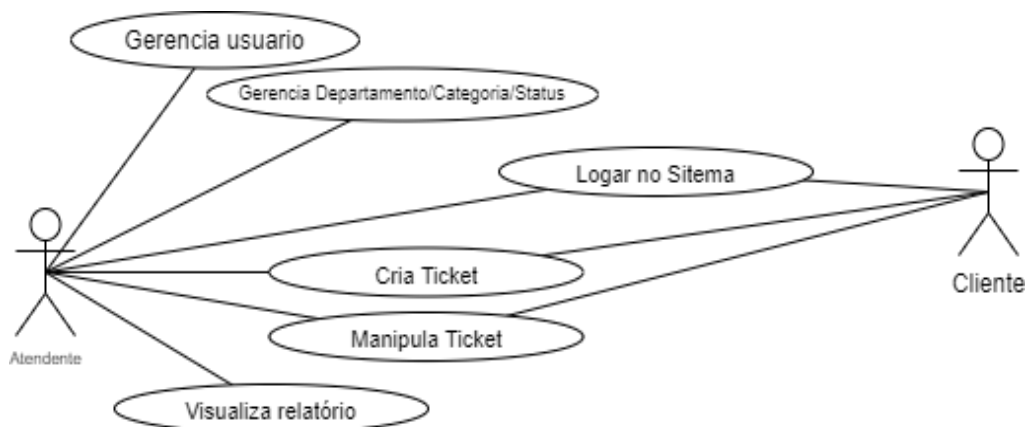
Welcome to PhpSpreadsheet's documentation. **PhpSpreadsheet's**. 2021. Disponível em: <<https://phpspreadsheet.readthedocs.io/en/latest/>>. Acesso em: 8 mai. 2021.

_____. **Administração de Marketing no Brasil**. São Paulo: Cobra Editora de Marketing, 2003.

APÊNDICE A: Diagramas

Serão apresentados nessa sessão os diagramas utilizados para o desenvolvimento do Sistema atendimento TICKETMASTER.

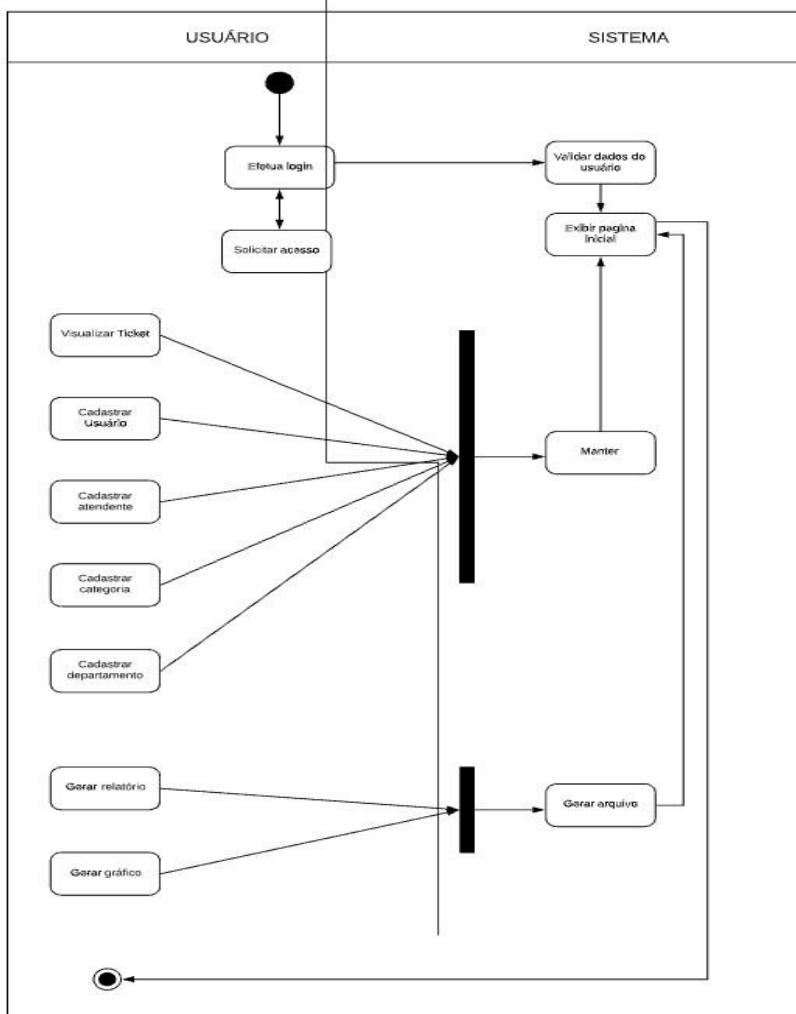
Figura 10 – Diagrama de Casos de Uso.



Fonte: Acervo pessoal.

Na figura 10 podemos ver as principais entidades do sistema, como podemos notar o administrador tem controle total sob o sistema, podendo fazer todas as ações, como gerenciar usuário, departamentos, status e categoria, e manipular os tickets. O cliente pode apenas criar e manipular os tickets.

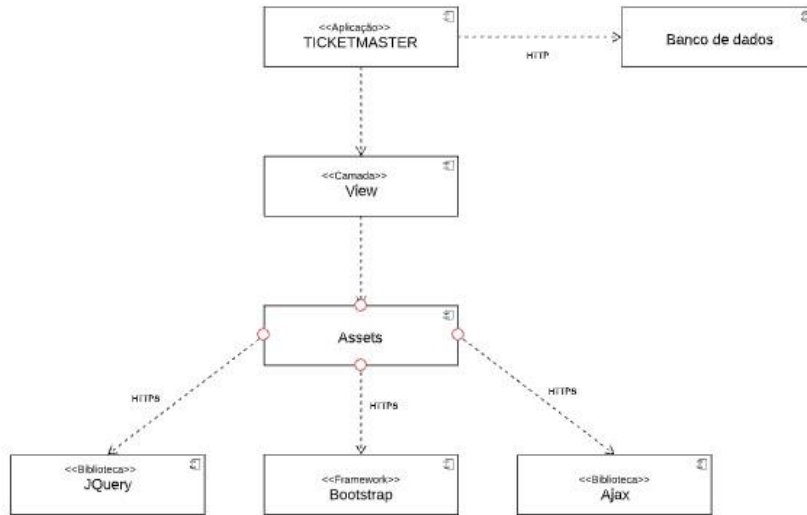
Figura 11 – Diagrama de Atividade.



Fonte: Acervo pessoal.

Na figura 11 podemos acompanhar uma das atividades do sistema, o processo mostrado é da tela de *login*, onde antes do usuário ter acesso aos conteúdos do sistema o mesmo deve informar as credenciais validas.

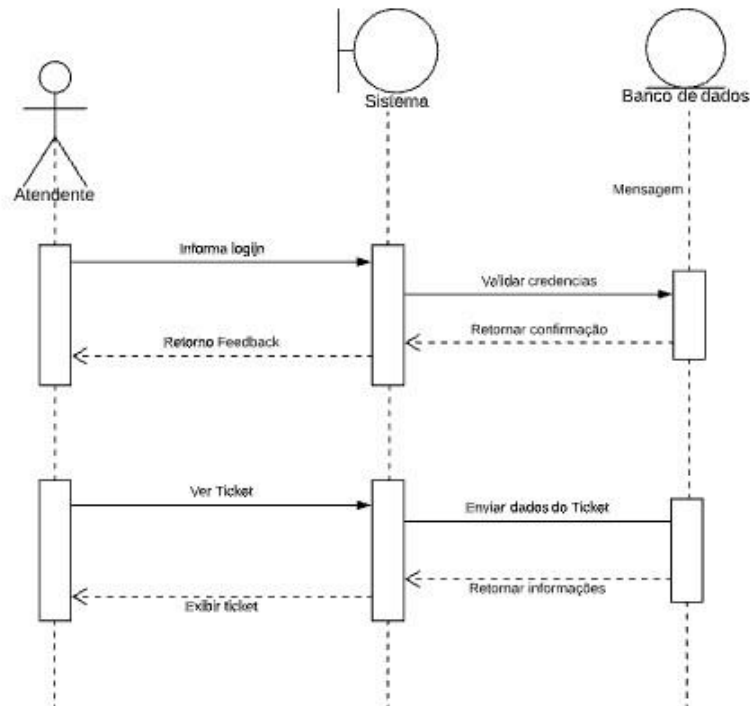
Figura 12: Diagrama de Componentes.



Fonte: Acervo Pessoal.

O Diagrama apresentado pela Figura 12 nos mostra alguns componentes do sistema, como a biblioteca *Jquery*, *Bootstrap* e *Ajax* que são primordiais no funcionamento do sistema juntamente ao banco de dados.

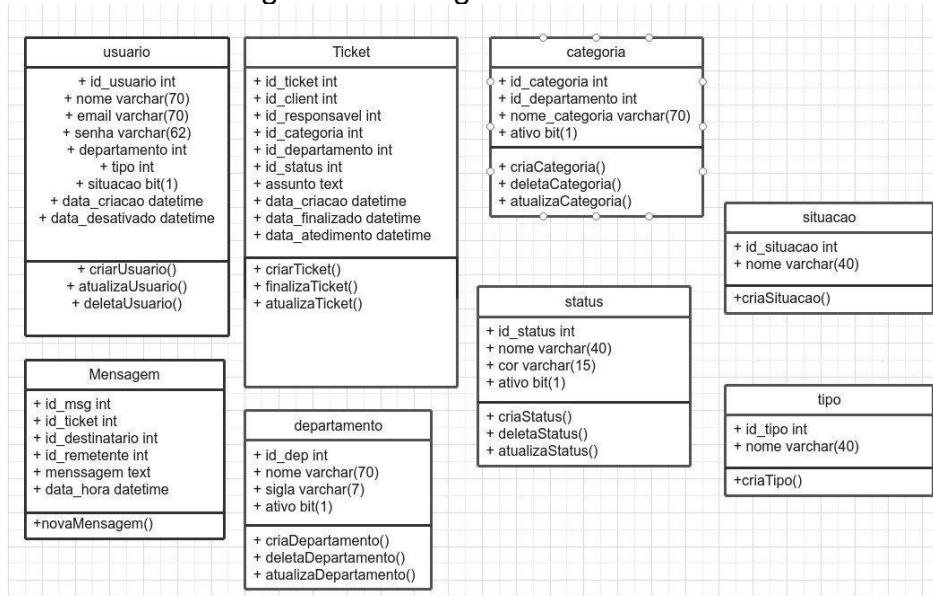
Figura 13 – Diagrama de Sequência.



Fonte: Acervo Pessoal.

Na Figura 13 é mostrado o processo para visualizar um ticket, é possível ver toda ação necessária para o atendente solicitar ao sistema a visualização do *ticket*. Vemos que o sistema sempre retorna algum feedback da ação solicitada.

Figura 14 – Diagrama de Classes.



Fonte: Acervo pessoal.

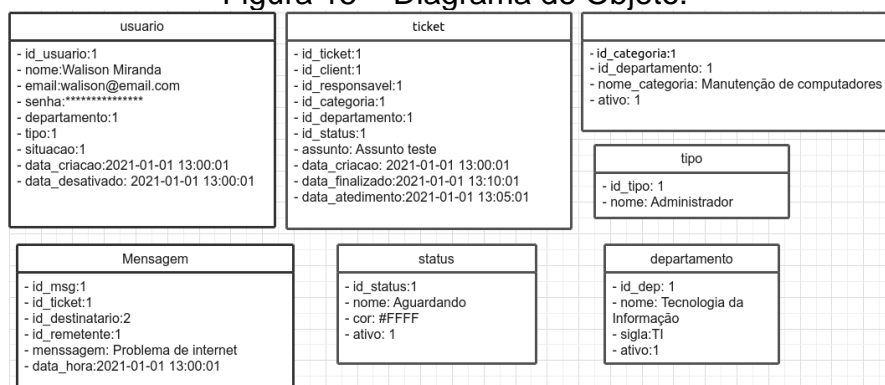
A Figura 14 podemos verificar algumas de nossas classes, atributos e suas respectivas características no sistema. Nela podemos ver as visibilidades e métodos a serem usados no sistema. As principais tabelas são a de usuario, ticket_ e mensagem, descrição delas abaixo.

Usuario: É a tabela onde fica armazenado todas as informações do usuário, informações como e-mail, nome, senha entre outras. A tabela usuario é interligada com as tabelas departamento e tipo, onde indica o departamento e tipo do usuário.

Ticket_: A tabela ticket_ armazena todas informações do ticket, assunto, responsável, cliente, horário que foi aberto, atendido e finalizado.

Mensagem: Essa tabela armazena dados da mensagem, como quem enviou, data, e a mensagem em si. As mensagens são apresentadas ao usuário de acordo com o ticket. Um ticket pode ter várias mensagens vinculadas a ele, mas as mensagens podem pertencer a somente um ticket.

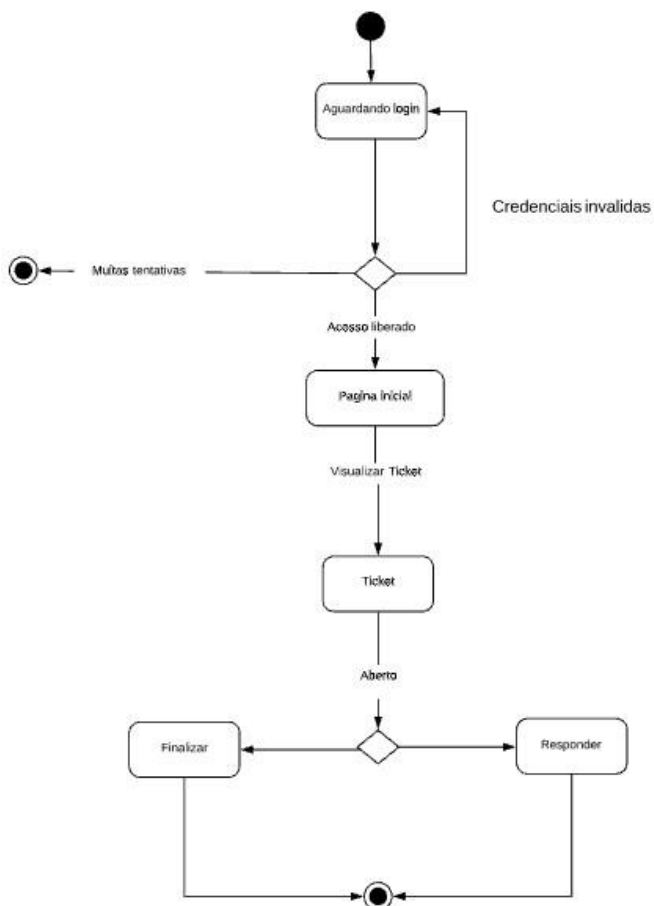
Figura 15 – Diagrama de Objeto.



Fonte: Acervo pessoal.

Na Figura 15 é representado o instanciados pelo diagrama de classes. Na figura podemos ter uma prévia de como eles devem funcionar.

Figura 16 – Diagrama de Máquina de Estado.



Fonte: Acervo pessoal.

Na Figura 16 vemos como é o processo do atendente ao abrir o *ticket*. Também podemos ver que ele tem a opção de responder ou finalizar o atendimento.