

## **DESENVOLVIMENTO DO SISTEMA DE GERENCIAMENTO SERVIÇOS MECÂNICOS – GERENCIE JÁ**

Walesson Batista Gonçalves<sup>1</sup>  
Romério Ribeiro Da Silva<sup>2</sup>

### **RESUMO**

O avanço tecnológico está tomando conta do cotidiano das pessoas tornando um fator essencial para auxílio de suas tarefas é de grande importância que todas as pessoas estejam sempre atualizadas das tecnologias que existem ao redor do mundo, as empresas não ficaram para trás sempre buscam uma inovação para aperfeiçoar e organizar as tarefas do seu dia a dia. Sendo assim, este trabalho é um resultado de estudos bibliográficos sobre TIC (tecnologia da informação e comunicação) voltadas para ambiente empresarial, que tem como problema de pesquisa a falta de ferramenta para auxiliar empresas voltadas ao ramo de prestação de serviços mecânicos, o objetivo desse trabalho é auxiliar empresas no seu gerenciamento de serviços mecânicos prestados a clientes. O resultado deste trabalho é o sistema (*software*) web denominada “Gerencia Já”.

**Palavras-chave:** Serviço. Agendamento. Cliente. *Software*.

### **ABSTRACT**

*The technological advancement is contemplated with the aim of everyday people who become essential to the maintenance of their tasks of great importance that everyone keeps up-to-date the technologies that exist in the world, how companies do not lag behind and seek innovation To perfect and organize as tasks of your day to day. Thus, this work is a result of bibliographic studies on TIC (aimed at the business environment), which has as a research problem, objective of this work is to assist companies not managing their mechanical services to clients. The result of this work is the system (software) called "Management Now".*

**Keyword:** Service. Scheduling. Client. *Software*.

---

<sup>1</sup> Acadêmico do curso de Sistemas de Informação – Faculdade Atenas

<sup>2</sup> Docente do curso de Sistemas de Informação – Faculdade Atenas

## **INTRODUÇÃO**

Com a globalização e a evolução tecnológica no mundo, a sociedade acabou se tornando dependente do uso de tecnologias em suas tarefas no seu dia a dia, sejam elas simples ou complexas. A interação das pessoas sofreu uma grande migração para área tecnológica, gerando um aumento exponencial no fluxo de informação, contidas a atender essa crescente demanda. Com esse avanço empresas estão cada dia mais adotando ferramentas para auxiliarem suas tarefas na organização (MARTINS, 2007).

Esse crescimento de tecnológico subsidia as tomadas de decisões e também torna os processos mais fáceis e rápidos dentro das organizações. Empresas estão buscando cada dia mais melhorias nos seus métodos de atendimento para ter mais qualidade nos serviços prestados aos clientes. Todas as empresas que não buscam melhorar no atendimento ficaram para trás (hospitais, bancos, telefônicas, mecânicas, etc).

O presente projeto visa refletir sobre a tecnologia e como ela pode contribuir nos processos de gerenciamento de serviços mecânicos prestados a clientes de uma determinada empresa, a partir daí essa reflexão como base teórica para construir um sistema computacional será capaz de gerenciar os serviços prestados

## **TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO (TIC) NAS EMPRESAS**

Com crescente aumento da tecnologia da informação e comunicação - TIC no mundo atual as organizações deparam com a constante necessidade de melhorar suas tarefas do cotidiano organizacional. Nesse sentido empresas buscam por tecnologias para otimizar e agilizar suas tarefas do dia a dia (OLIVEIRA, 2009).

A tendência é cada vez mais de ter um aumento de adoção das tecnologias de informação e comunicação não apenas por instituições do governo, mas por empresas de diversas áreas. Em um ambiente corporativo um grupo de pessoas tem certos objetivos a serem cumpridos, sendo assim ocorre a necessidade de se comunicarem para relatarem os acontecimentos no cotidiano da

corporação (PINOCHET, 2014).

A TIC é responsável por todas as tecnologias que auxiliam os processos informatizados e de comunicação envolvidos na organização. Pode ser entendido como um conjunto de recursos de tecnologias que trabalham juntos por meio de *hardware* e *software*, sobre tudo com espalhamento dos aparelhos tecnológicos a cultura da organização se torna outra cultura conhecida como a cultura digital, gerando mais otimização e qualidade nas tarefas do dia (TOPLAB, 2012).

E assim, estima-se que as TICs no ambiente corporativo sejam capazes de melhorar a qualidade das tarefas realizadas e criar soluções rápidas, pois com ela é possível fazer o uso de recursos como *software* que possa gerar relatórios demonstrativos, além dessas facilidades, podem-se associar outros fatores, como segurança, confiabilidade e rapidez nas informações prestadas. Por esse sentido os envolvidos estejam cientes que quando incluírem o uso de TIC terão que desenvolver novos conceitos de aplicar organização e disciplinas dentro da corporação (RAMOS, 2010).

## **SISTEMAS DE INFORMAÇÃO**

Os sistemas de informação no mundo atual tornaram-se muito importantes para o ambiente empresarial, principalmente a partir do surgimento de novas ferramentas e tecnologias, o que transformou modo de organização, bem como a forma de pensar e trabalhar, tendo em vista a diversidade de ferramentas, técnicas e tecnologias. Cabe ressaltar ainda que, as empresas para poderem aproveitar todo o potencial tecnológico devem fazer uma atualização em sua infraestrutura, e ter seu diferencial na maneira de trabalhar e de se organizar, para que possa resolver com mais agilidade seus problemas e atender com mais qualidade as necessidades de seus clientes; gerando qualidade em seus serviços e satisfação dos seus consumidores (MORAN, 2004).

Percebe-se que a TIC é uma grande aliada no mundo atual, sendo um recurso de extrema importância para as organizações e pessoas, principalmente para as que ainda não se adaptaram às novas tendências do mercado. É necessário que as empresas comecem a reavaliar os conceitos de suas atividades e processos, pois certamente podem perder controle sobre sua organização, pela ausência ou administração incorreta desses recursos, acarretando prejuízos para a

empresa. Percebe-se então que essa ferramenta é de grande importância no processo de tomadas de decisões (desde pessoais à corporativa) (BALTZAN; PHILLIPS, 2012).

Dessa forma, observa-se que as empresas que não utilizam a tecnologia da informação, correm um sério risco de serem deixadas para trás pelas que fazem o uso da mesma. Atualmente as organizações devem procurar se adaptar ao avanço da tecnologia para que consigam acompanhar o ritmo de crescimento que acontece no mundo atual (MUÑOZ, 2012).

## **GESTÃO DE DADOS E INFORMAÇÕES**

Quando se fala em banco de dados, observa-se nos últimos anos o grande uso do termo *Big Data*, item que passou a ser do interesse não somente de profissionais de TI, mas das empresas que estão adotando a todo momento inovação tecnológica. Nesse sentido, é importante compreender que os dados podem ser considerados matérias-primas dentro das empresas, principalmente no cenário atual dentro contexto empresarial. Observa-se que há muitos anos gestores tomam decisões baseadas nos dados, por isso é importante que essas decisões proporcionem benefícios, sendo necessário realizar a gestão dos dados. Tem-se que a gestão de dados e informações, traz um alinhamento da área de TI e de negócios, busca um aumento a segurança da informação, gerenciar e destinar as informações para usuários que realmente tem a necessidade do uso, evitar informações com redundâncias, possibilita a compreensão da necessidade de dados e informações para planejamentos e traz uma certa melhoria na qualidade e confiabilidade dos dados (DATASTORM, 2016).

## **DESENVOLVIMENTO DE SOFTWARE**

### **SOFTWARE**

*Software* é uma sequência de instruções escritas para serem interpretadas por um computador com objetivo de executar comandos e tarefas específicas (somar, dividir, subtrair e multiplicar), também é considerado e definido como os programas que comandam o funcionamento de um computador. A classificação do *software* entra como parte lógica, cuja sua função é fornecer

instruções para o *hardware*. *Software* é um termo inglês surgiu no início dos anos 50, quando empresas começaram a adquirir os primeiros computadores que surgiram na época para uso comercial, primeiros softwares foram os administrativos e financeiros básicos mais específicos para contabilidade que gerava folha de pagamento e controle de estoques (PRIKLADNICKI, AUDY, 2008).

Um *software* é construído para solucionar um determinado problema, possuindo diversas funções, bibliotecas e módulos, trabalha através de comandos como entrada de informações que é processada por uns algoritmos que gera uma saída que demonstra um resultado das informações processadas. Observando-se que nos últimos anos a população de algumas linguagens se destacam no cenário do desenvolvimento de *software*, dentre elas está o Java Web (PRESSMAN; MAXIM, 2016).

## JAVA WEB

A linguagem Java nasceu nos anos 90 na *Sun Microsystems* e a grande diferença da plataforma Java de outras linguagens de programação é que o Java é executado sobre uma máquina virtual chamada JVM (Java Virtual Machine). Assim, tem-se que, qualquer plataforma de *hardware* que possa executar uma máquina virtual conseguira executar Java (LUCKOW; MELO, 2015).

A características da linguagem de programação Java representa uma linguagem orientada a objeto, oferece programação robusta, pois permite que o desenvolvimento de sistemas seja desenvolvido em diferentes tipos sistemas operacionais e diferentes arquiteturas de *hardware*, oferecendo segurança e alto desempenho (MENDES, 2009).

O Java no mundo atual tornou-se uma das linguagens mais usadas para desenvolvimentos de aplicativos baseados na *web* e *softwares* para dispositivos que fazem o uso de uma rede para se comunicar. Nos desenvolvimentos de aplicações Java a engenharia de *software* é de extrema importância, pois é uma área que contém técnicas e métodos de desenvolvimento de *software* (DEITEL; DEITEL, 2010).

## **ENGENHARIA DE SOFTWARE**

A engenharia de *software* cresce dia após dia, por meio de somatórias das experiências dos profissionais, de forma moderna a engenharia de *software* tem uma caminhada paralela com sistemas de informação os mesmos são destinados as organizações, para auxiliar as mesmas a tomarem decisões sob o foco de seu negócio empresarial ou de atividade pública. Engenharia de *software* é a arte da construção com base em conhecimentos científicos, assim tento a arte de engenhar que é a ideia de maquirar, inventar e produzir (REZENDE, 2005).

A engenharia de *software* tem como objetivo apoiar o desenvolvimento profissional de *software* incluindo técnicas que ajudam na evolução de projetos. Ela é uma área da computação que envolve diversas áreas de conhecimento voltada à manutenção e desenvolvimento de sistemas com uso de levantamento de requisitos, tecnologias e ferramentas para chegar em um determinado objetivo. Além da engenharia de *software* proporcionar que o *software* será executado exatamente de acordo com seu objetivo planejado (SOMMERVILLE, 2011).

Neste sentido tem-se um contexto do desenvolvimento de *software*, a onde se tem os requisitos que visam a construção do *software* que pode trazer diversas soluções. Por isso, é importante compreender que o desenvolvimento de *software* segue etapas, como levantamento de requisitos, modelagem e metodologia de desenvolvimento (WAZLAWICK, 2013).

## **LEVANTAMENTO DE REQUISITOS**

Para desenvolver um *software* e necessário saber dos detalhes, do objetivo que o *software* vai realizar, neste contexto tem-se um levantamento de requisitos busca auxiliar e facilitar o desenvolvimento do *software*. Levantamento de requisitos é examinar e entender as informações básica de processos e atividades que vão gerar o resultado final do sistema computacional. O levantamento de requisitos não somente examina as informações básicas do sistema, mas também entender aquilo que o cliente deseja ou o que o cliente acredita que precisa e as regras do negócio ou processos do negócio (BASSANI, 2006).

E assim, é importante compreender a definição de requisitos, que é uma

coleção de sentenças que devem ser descritas de modo claro, sem duplos sentidos, considerar todos os aspectos significativos do sistema proposto, deve conter informações suficientes para permitir que os envolvidos no desenvolvimento do projeto construam um sistema com qualidade e que vai realizar o objetivo desejado (VAZQUEZ; SIMÕES, 2016).

O levantamento de requisitos, e a fase que se deve analisar todos os detalhes do *software* buscando entender os objetivos que o *software* irá realizar, documentar o mesmo para que possa gerenciar os requisitos levantados na hora do desenvolvimento, dentro dessa atividade, pode analisar os requisitos, validar, para que não ocorra falhas e atraso no desenvolvimento (BARBOSA, 2009).

## **MODELAGEM DE SOFTWARE**

Modelagem de *software* é uma representação simplificada de algo real, fazendo semelhança com uma planta de uma construção civil, ele representa todos os detalhes de estrutura que terá o projeto, uma atividade de construir modelos que expliquem as características e o comportamento do *software*, na identificação das funcionalidades que o *software* deverá cumprir na sua execução. A modelagem de *software* não só permite que o *software* seja projetado com os requisitos necessários para que seu funcionamento cumpra seus objetivos com qualidade, mas que use uma arquitetura apropriada no projeto (LOBO, 2009).

Na criação de projetos são utilizados modelos desenhados que tem objetivo de representar as características relevantes do projeto. Em cada engenharia existe uma forma técnica de fazer a utilização de símbolos e diagramas para criar modelos e peças ou do sistema a ser construído. Na modelagem de *software* não é diferente por meio de diagramas que e construído modelos que vai representar o *software* de forma direta (LOBO, 2008).

## **METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE**

Uma metodologia de *software* é um conjunto estruturado de práticas que trabalha no lado interior da engenharia de *software*, serve para não tornar tarefas do desenvolvimento, complexas por natureza. Os métodos tradicionais podem deixar a equipe de desenvolvimento preso a requisitos desatualizados nesse sentido o desenvolvimento não vai ter um final de projeto bem-sucedido

(CAETANO, 2016).

O *software* desenvolvido sem fazer o uso de uma metodologia de desenvolvimento apropriada poderá apresentar um resultado negativo do objetivo que foi esperado, gerando insatisfação do cliente, e afetando a imagem da equipe de desenvolvimento (KOSCIANSKI; SOARES, 2007).

A metodologia de desenvolvimento permite que a equipe do projeto faça definições de ferramentas a serem utilizadas no desenvolvimento do projeto, a equipe de desenvolvimento, distribuição de tarefas a serem cumpridas, prazos de entrega. Ao fazer o uso de uma metodologia de desenvolvimento o projeto será finalizado dentro do prazo que foi definido no escopo do projeto, trazendo satisfação para toda equipe envolvida no projeto (POMPILHO, 2002).

## **LINGUAGEM DE MODELAGEM UNIFICADA (UML)**

Linguagem de modelagem unificada (UML), é uma linguagem padronizada para modelagem orientada a objeto, surgiu da união de três metodologias de modelagem (método de *booch*, método de *OMT* e método *OOSE*). É um registro gráfico para projetar sistemas, não é considerada uma linguagem de programação, sua definição é de uma ferramenta para desenvolver diagramas padronizados e complexos. Modelagem é uma atividade de construir modelos que explique as características e comportamentos de um sistema (BRAUDE, 2008).

No planejamento do sistema a equipe de desenvolvimento tem como objetivo entender o funcionamento do sistema, a UML é uma ferramenta que tem a função de desenhar plantas de sistemas, disponibiliza conceitos, objetos, símbolos e diagramas, uma forma simples, mas objetiva e funcional. Nessas circunstâncias entende-se que UML é uma linguagem gráfica para visualizar, especificar a construção e a documentação de um sistema, tem a forma proporcional de padronização do início ao fim do sistema (BEZERRA, 2006).

## **DIAGRAMAS**

### **CASO DE USO**

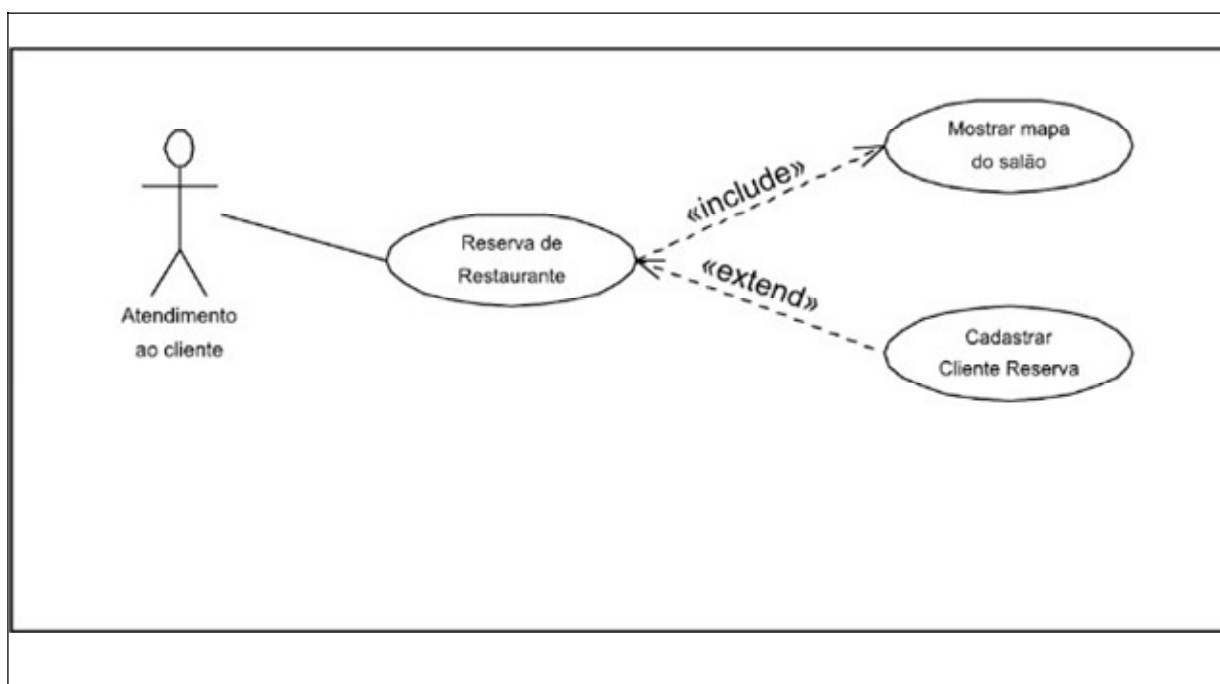
Um caso de uso é uma descrição de relato de uma sequência de eventos que ocorre quando um ator usa um sistema para realizar umas funções, especificando o comportamento de um sistema. Para interpretar e considerado um



caso de uso do sistema, alguns elementos devem ser levados em consideração, tais como: identificar objetivos, funcionalidades, informações que poderiam ser consultadas, cadastro, alterações (REZENDE, 2005).

A especificação do caso de uso é um documento funcional, ou seja, ele descreve as funcionalidades que um sistema deve ter. No caso de uso é apresentado como o sistema deve funcionar e reagir com conjunto de entradas que o sistema vai receber, o comportamento desse sistema e as saídas que vão ser geradas. A utilização de caso de uso para expressar a fronteira do sistema, e modelar os requisitos do mesmo, sua existência permite uma visão gráfica dos relacionamentos que o sistema vai operar. As especificações de caso de uso envolvem, por exemplo: cálculos que o sistema deve executar, manipulação de dados entre outras informações, que pode ser melhor visualizado na figura 01 a seguir (MELO, 2010).

**FIGURA 01** – Exemplo de caso de uso.



Fonte: Elaborado por MELO, 2010.

## CLASSES

Uma classe é a descrição de um determinado objeto, esses objetos são instâncias de classes. As classes descrevem as propriedades e os comportamentos do determinado objeto, os objetos só podem ser instanciados a classes. As classes são usadas para classificar os objetos que são identificados no

mundo real. Uma classe pode ser a descrição de um determinado objeto em qualquer sistema (BEZERRA, 2005).

Na programação um diagrama de classes é a representação da estrutura das relações das classes que servem como descrição de modelo de objetos. Os conceitos de classe e de um elemento abstrato que é representado como conjuntos de objetos, contendo suas especificações suas características, atributos (define características da classe), métodos (ações, comportamentos). Os diagramas de classes são utilizados não somente para representar objetos, representam as estruturas estáticas de sistemas que são compostas por classes de negócio, interface, controle. No desenvolvimento de sistemas o uso do diagrama de classe não pode ser deixado de lado, pois sua utilização torna o desenvolvimento ágil e finalizado com qualidade trazendo satisfação para todos envolvidos no projeto (MARTINS, 2007).

## **ENTIDADE E RELACIONAMENTO**

Modelo de Entidade e Relacionamento também conhecido como modelo ER ou MER, é utilizado na engenharia de *software* para demonstrar através de uma descrição as entidades que estariam envolvidas com seus atributos demonstrando como eles se relacionar. Esse modelo é representado de forma geral é basicamente a estrutura do banco de dados, que vai ser desenvolvido para um determinado sistema. O modelo de entidade e relacionamento possui níveis de raciocínio (abstração) que podem ser utilizados (conceitual, lógico e físico) (RÊGO, 2013).

O modelo conceitual se baseia na etapa de análise de requisito, é usado para desenvolver uma descrição de alto nível dos dados juntamente com suas restrições. O objetivo é criar uma descrição gráfica bem simples dos dados que correspondem a ideia de como vai ser o seu funcionamento. É um diagrama em blocos sua demonstração e descrever as relações entre as entidades e seus atributos (RAMAKRISHNAN, 2011).

O modelo conceitual é um objeto (artefato) muito importante durante a análise de construção do sistema, seu objetivo é ilustrar os conceitos importantes do problema, suas combinações, atributos corretos e incorretos, adicionar conceitos e entender os termos interface e classes (HEUSER, 2009).

O modelo lógico é feito a partir do modelo conceitual, sua atividade e demonstra a estrutura básica de entidades, atributos e relacionamentos. Descreve toda estrutura do banco de dados, conforme vista pelos usuários do SGBD, demonstra as ligações entre as tabelas do banco de dados e suas chaves primárias, e seus componentes (MACÊDO, 2016).

O modelo físico e a inclusão da análise das características e os recursos necessários para armazenamentos e manipulação da estrutura de dados, o seu funcionamento e através de comandos executados em SQL a fim de criar as tabelas e as estruturas, ligações que foram projetadas e assim se cria o banco de dados. Os modelos citados acima são de extrema importância no desenvolvimento de sistemas, pois através deles a visão de desenvolvimento é mais ampla. (MANNINO, 2014).

## **BANCO DE DADOS**

Banco de dados é uma coleção de dados que são inter-relacionados capaz de representar informações sobre um domínio específico, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo objetivo é considerado um banco de dados (KORTH, 1994).

Nesse contexto entende-se que banco de dados são conjunto de dados que se relacionam entre si, são utilizados em empresas para gerenciar suas informações. Existem definições fundamentais para entender melhor o que é banco de dados, como dados são fatos ou eventos brutos, em sua forma primária, muitas vezes os dados não fazem sentido quando estão sozinhos. Podem ser arquivos, imagens, nomes, textos, datas, áudio e vídeo, e também informações consistem no agrupamento de dados de forma organizada para fazer sentido e gerar conhecimento (FEITOSA, 2013).

Há tempos atrás o banco de dados era visto como um sistema de arquivo a ser utilizado e manipulado por aplicativos, nos dias atuais é um conjunto de *softwares* associados a um sistema de arquivos que não precisam entender de arquivos pois eles entendem uma linguagem padronizada para se comunicar com o banco de dados. Esses sistemas são chamados de Sistema de Gerenciamento de Banco de Dados (SGBD) que servem para dar gerenciar o banco de dados, facilitam a manutenção das bases de dados (DATE, 2003).

## **SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS**

Um sistema de gerenciamento de banco de dados (*Data Base Mangement System*) conhecido como SGBD, é basicamente um conjunto de programas (*software*) que é responsável por gerenciar e administrar banco de dados. O SGDB possui recursos para que os usuários possam apenas efetuar atividades relacionadas aos dados (criar, editar, atualizar, armazenar e recuperar dados em tabelas). As vantagens de usar um SGDB, é maior disponibilidade dos dados, minimização de redundância de dados, precisão e outros. Os tópicos mais importantes de um SGBD é a capacidade de organizar e recuperar os dados, usar os pontos fortes e fracos do SGDB, as tomadas de decisões a serem mais eficientes na questão de organizar e utilizar os dados (LEITE, 2007).

## **STRUCTURED QUERY LANGUAGE (SQL)**

A Linguagem de Consulta Estruturada (*Structured Query Language*) conhecido como SQL, surgiu nos anos 70 na empresa IBM (*International Business Machines*), o seu objetivo era demonstrar a viabilidade da implementação do modelo relacional de banco de dados. SQL é uma linguagem exclusivamente projetada para desenvolvimento de banco de dados, permite que desenvolvedores adicione e modifique os dados armazenados em um determinado banco de dados (TAYLOR, 2016).

## **IDE**

IDE (*Integreted Develpment Environment*) é basicamente o ambiente de desenvolvimento de programas (*softwares*), ou seja, IDE é um programa (*software*) que integra as várias ferramentas que são necessárias no desenvolvimento de software e todos os processos de codificação. Sua função é ajudar e auxiliar o programador a escrever o código fonte do *software*. Os IDEs também são compiladores, interpretadores esse parâmetro e a execução do código fazendo a criação do software se não ocorrer erros (WILKER, 2016).

A IDE escolhida para desenvolver do projeto foi o ECLIPSE na versão *JeeNeon*, devido a sua simplicidade e usabilidade.

## FRAMEWORK

O *framework* é um conjunto de classes que trabalham juntas para proporcionar a melhoria na prática de desenvolvimento de *software*. Conhecido como biblioteca de desenvolvimento de *software*, sua funcionalidade é genérica, específica, abstrato, seu uso no desenvolvimento de *software* torna códigos otimizados, simplificados e diminuição de repetição nas tarefas aplicadas no desenvolvimento (PRESSMAN, 2006).

No desenvolvimento do *software* o uso de *frameworks* não foi deixado de lado, os frameworks escolhidos para desenvolver o *software* foram: JSF, *bootsface* e *hibernate*.

O JSF é basicamente um *JavaServer Faces* é um *framework* que permite desenvolver interfaces web, o JSF permite a separação entre lógico e regras de negócio, navegação e conexões com serviços externos e gerenciamento das configurações. O JSF possui diversos componentes para ser utilizados no desenvolvimento de interface (design), simples e flexíveis (LUCKOW; MELO, 2015).

O *bootsface* é um *framework* desenvolvido especialmente para desenvolvimento de aplicações Java web, foi criado pela equipe do *twitter* baseado no *framework bootstrap*. O *bootsface* é voltado para ser utilizado no desenvolvimento de *front-end (interface)* responsivo de código aberto, possui diversos componentes que podem ser modificados facilmente tornando o desenvolvimento otimizado (GOMES, 2016).

*Hibernate* é um *framework* de mapeamento de objetos relacional escritos na linguagem Java, onde seu principal objetivo é diminuir a dificuldade e a complexidade envolvido no desenvolvimento de softwares que necessitam trabalhar com banco de dados. O desenvolvimento de banco de dados no *hibernate* tem a base do SQL (LUCKOW; MELO, 2015).

## APRESENTAÇÃO E RECURSOS DO DESENVOLVIMENTO DO SISTEMA

Através das bibliografias apresentadas anteriormente percebe-se que o uso de sistemas para auxiliar empresas em tarefas do cotidiano e de extrema importância pois seu benefício é claro. Com base nas bibliografias descritas no

projeto, buscou-se o desenvolvimento do sistema que seja capaz de auxiliar empresas voltadas ao ramo de serviços mecânicos, tendo como objetivo gerenciar serviços prestados aos clientes.

O tópico a seguir será apresentado os requisitos do sistema.

## LISTA DE REQUISITOS FUNCIONAIS

Com base as bibliografias apresentadas anteriormente, sabemos que levantamento de requisito e de extrema importância no desenvolvimento de *software*, nesse contexto apresenta-se o quadro 01 que é a descrição dos requisitos funcionais, foi desenvolvido através da funcionalidade apresentada nas telas do *software*, quadro 02 são os requisitos não funcionais, desenvolvido com base nos requisitos funcionais apresentados.

**QUADRO 01** – Requisitos funcionais.

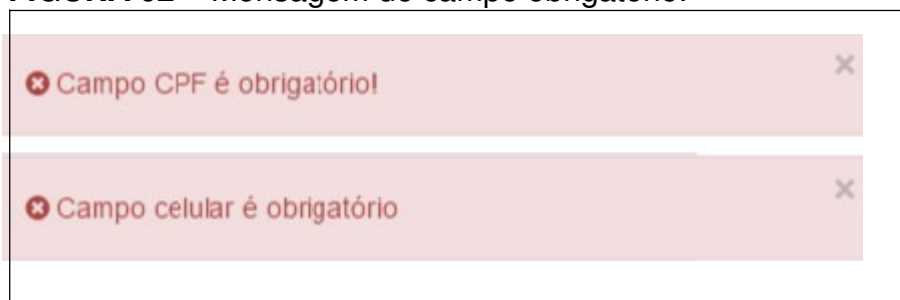
Requisitos funcionais	
Identificação	Descrição
RN01	Permitir que o usuário efetuar <i>login</i> .
RN02	Permitir que o usuário escolher a opção desejada no menu após o efetuar <i>login</i> .
RN03	Permitir que o usuário efetue cadastro de funcionários.
RN04	Permitir que o usuário efetue o cadastro de clientes.
RN05	Permitir que o usuário efetue o agendamento de serviços
RN06	Permitir que o usuário visualize cadastros no banco de dados.
RN07	Permitir que o usuário visualize clientes cadastrados no banco de dados.
RN08	Permitir que o usuário visualize agendamentos cadastrados no banco de dados.
RN09	Permitir que o usuário visualize funcionários cadastrados no banco de dados.
RN10	Permitir que o usuário efetue <i>logout</i>

**Fonte:** Elaborado pelo autor.

## REQUISITOS NÃO FUNCIONAIS QUADRO 02 – Requisitos não funcionais.

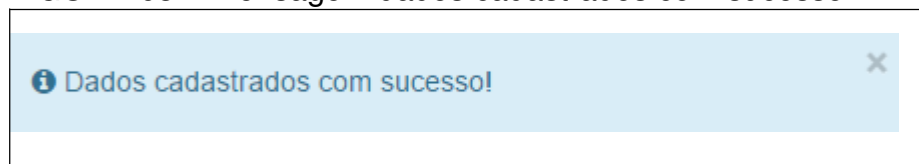
Requisitos Não Funcionais	
Identificação	Descrição
RNF01	Deve utilizar senhas de acesso para o controle seguro da aplicação.
RNF02	Deve ser desenvolvido na linguagem Java para <i>Web</i> .
RNF03	Deve possuir banco de dados.
RNF04	Deve possuir cores e fontes que facilitem a leitura e evitem o cansaço da visão.
RNF05	Precisa informar ao usuário os campos dos formulários que são obrigatórios para preenchimento.
RNF06	Deve ser <i>web</i> e compatível com qualquer <i>browser</i> .
RNF07	Deve permitir acesso a qualquer momento.
RNF08	Deve ser hospedado em um servidor <i>web</i> .
RNF09	Poderá ser acessado por qualquer aparelho tecnológico tais como, <i>smartphone</i> , <i>tablet</i> , <i>notebook</i> etc.
RNF10	Deve mostrar mensagem de confirmação ao realizar alguma operação.
<b>OPERACIONALIDADE DO SISTEMA</b>	
RNF11	Deve possuir uma interface gráfica com ícones representativos para facilitar a sua utilização.
RNF12	O cliente deve ligar ou comparecer na prestadora de serviços para agendar o serviço.

Esse tópico é responsável por apresentar as principais telas do sistema desenvolvido. O sistema pode ser acessado de qualquer dispositivo que tenha acesso a rede como exemplo: *smartphone*, *tablet* e outros. Como na informação anteriormente sobre *bootsface*, o sistema trabalha com *design* responsivo das telas e seus componentes, tornando a sua usabilidade mais prática e agradável. Todos os campos que possui (\*) são campos obrigatórios, caso não seja preenchido uma mensagem será mostrada conforme na figura 02, após todos os campos obrigatórios serem preenchidos uma mensagem será mostrada conforme figura 03. As figuras a seguir serão explicadas com descrição a baixo de cada figura.

**FIGURA 02** – Mensagem de campo obrigatório.

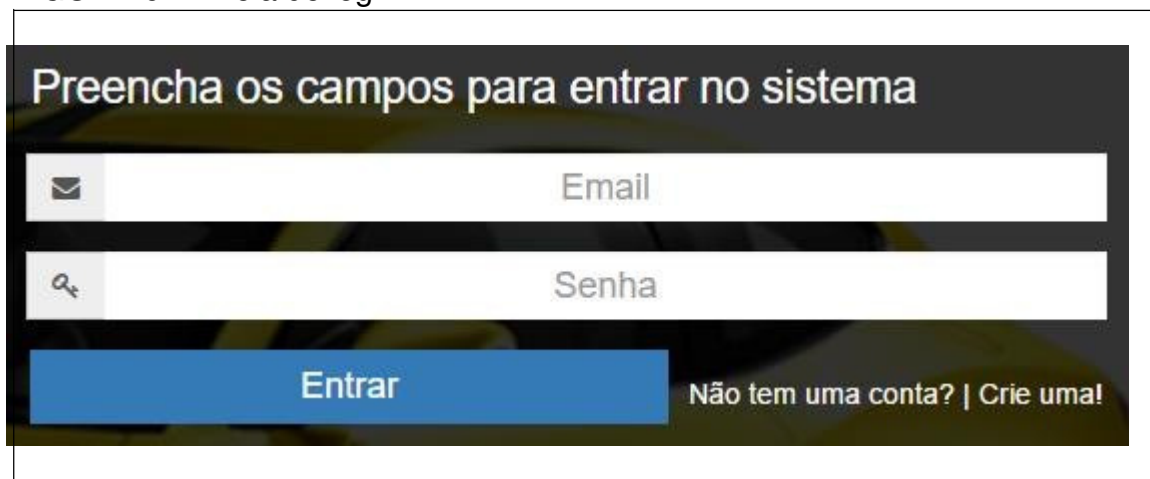
**Fonte:** Elaborada pelo autor.

Os campos obrigatórios que não foram preenchidos serão mostrados na mensagem conforme a figura 02.

**FIGURA 03** – Mensagem dados cadastrados com sucesso.

**Fonte:** Elaborada pelo autor.

A imagem a cima e a forma de dizer que as informações foram cadastradas no banco de dados com sucesso.

**FIGURA 04** – Tela de login.

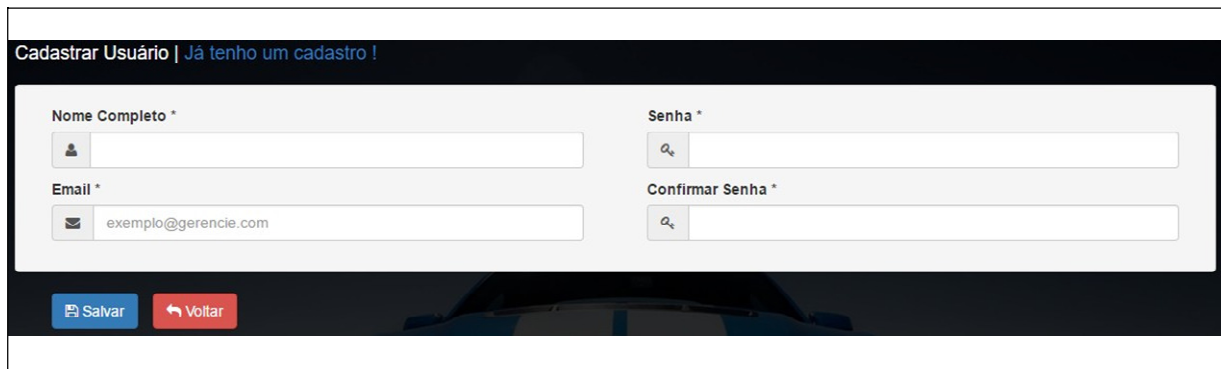
**Fonte:** Elaborado pelo autor.

A imagem a cima trata se da tela de efetuar o *login* no sistema caso o usuário não tenha um cadastro, e queria se cadastrar no sistema é só clicar no link



“Não tem uma conta? Crie uma!”, que o mesmo será redirecionado para tela de cadastro de usuário.

**FIGURA 05** – Tela de cadastro de usuário.

The image shows a web form for user registration. At the top, there is a header with the text "Cadastrar Usuário" and a link "Já tenho um cadastro!". Below this, the form is divided into two columns. The left column has two input fields: "Nome Completo \*" with a person icon and "Email \*" with an email icon and the placeholder text "exemplo@gerencie.com". The right column has two input fields: "Senha \*" with a key icon and "Confirmar Senha \*" with a key icon. At the bottom of the form, there are two buttons: "Salvar" (blue) and "Voltar" (red). The background of the form is dark with a faint image of a car.

**Fonte:** Elaborado pelo autor.

A imagem acima e chamada através do link descrito na imagem anteriormente sua funcionalidade e de cadastrar usuários para ter acesso ao sistema, caso o usuário já seja cadastro ele pode clicar no link “Já tenho um cadastro!”, o mesmo será redirecionado para pagina de *login*.

**FIGURA 06** – Menu do sistema.



**Fonte:** Elaborado pelo autor.

A imagem a cima trata se do menu do sistema onde o usuário pode acessar a outra tela do sistema clicando em início, clientes, serviços, funcionário e configuração para sair do sistema.


**FIGURA 07 – Tela de cadastro de cliente.**

## Cadastrar Cliente

Cliente Físico


Cliente Jurídico


Nome \*




Sexo \*

Selecion






Nº \*




Bairro \*




Data Nascimento

clique aqui




CPF \*




Telefone

(99) 9999-9999




Celular \*

(99) 99999-9999




Cidade \*




Complemento


Cep \*




Estado \*

Selecione






Endereço \*



Email \*



Salvar

Voltar

**Fonte:** Elaborado pelo autor.

A imagem a cima trata se de efetuar o cadastro de clientes, sua usabilidade e simples e pratica, o sistema tem opção de cadastro de cliente físico e jurídico, o usuário poderá escolher através dos botões “Cliente Físico”, “Cliente Jurídico”, qual cliente deseja efetuar o cadastro.

**FIGURA 08 – Tela consultar clientes**

Consultar Clientes

Cliente Físico

Cliente Jurídico

10

resultados por página

Pesquisar

Nome	CPF	Telefone	Celular	Visualizar	Editar	Excluir
edss	323.444.444-44	(34) 3434-3434	(43) 4 3434-3434			
Walesson Batista Gonçalves	129.460.316-75	(38) 3561-9348	(34) 9 9837-1044			
Eduardo Gonçalves	213.124.124-14	(12) 4123-1231	(12) 3 1231-2312			

Mostrando de 1 até 3 de 3 registros

Anterior

1

Próximo

**Fonte:** Elaborada pelo autor.

A imagem a cima busca no banco de dados todos os clientes cadastrados no sistema, possui as seguintes funções: visualizar dados do cliente, editar e excluir. O usuário pode selecionar qual o tipo de cliente ele deseja consultar clicando nos botões “Cliente Físico”, “Cliente Jurídico”.

**FIGURA 09** – Tela de agendar serviço.

Agendar Serviço

**Cliente Físico** **Cliente Jurídico**

**Cliente**

edss. CPF: 323.444.444-44

**Veículo \*** **Placa \***

**Data \*** **Hora de Inicio \*** **Hora de Termina \***

Clique Aqui

**Serviço \***

**Responsável pelo serviço**

sdsdsd

**Descrição**

**Fonte:** Elaborado pelo autor.

A imagem a cima trata se de efetuar o agendamento de serviço para clientes já cadastrado no sistema, caso o cliente não esteja cadastrado no sistema o usuário deverá efetuar o cadastro do mesmo de acordo com a figura 07.

**FIGURA 10** – Tela de consultar serviço agendado.

Consultar Serviço Agendado

**Cliente Físico** **Cliente Jurídico**

10 resultados por página

Pesquisar

Nº	Nome	CPF	Data	Serviço	Status	Visualizar	Editar	Excluir
1	edss	323.444.444-44	01/11/2016	5454545	Realizado ✓			
2	edss	323.444.444-44	01/11/2016	efertt	Não realizado ⚠			

Mostrando de 1 até 2 de 2 registros

Anterior 1 Próximo

**Fonte:** Elaborado pelo autor.

A imagem a cima representa a tela de consultar os serviços agendados para clientes, possui as seguintes funcionalidades: *status* onde o usuário do sistema pode clicar no campo (Realizado, Não Realizado) que representa se o serviço foi executado ou não, visualizar dados do cliente, editar e excluir. O usuário pode selecionar qual o tipo de cliente ele deseja consultar clicando nos botões “Cliente Físico”, “Cliente Jurídico”.

## CONCLUSÃO

Por meio do desenvolvimento do trabalho conclui-se que o avanço tecnológico no mundo atual é constante, e ainda à ausência de ferramentas que possam auxiliar as empresas em suas tarefas. Sendo assim, a necessidade de desenvolver uma ferramenta com recursos que sejam capazes de auxiliar empresas na otimização de suas tarefas é clara.

Por fim, após todas as etapas de desenvolvimento neste trabalho, gerou um resultado do sistema (*software*) *web* que agrega funcionalidade de cadastrar clientes e agendar serviços mecânicos a serem prestados, com propósito de responder a hipótese proposta no trabalho.

A hipótese do trabalho foi validada pois o *software* desenvolvido tem como função cadastrar cliente e agendar serviços a serem prestados ao mesmo, sendo assim o software auxilia no gerenciamento de clientes e de serviços prestados aos mesmos.

## REFERÊNCIAS

- BALTZAN, Paige; PHILLIPS, Amy. **Sistemas de informação**. Amgh Editora, 2012.
- BARBOSA, Glívia et al. Um processo de elicitação de requisitos com foco na seleção da técnica de elicitação. **VIII Simpósio Brasileiro de Qualidade de Software**, p. 159-173, 2009.
- BASSANI, Patricia Scherer et al. **Em busca de uma proposta metodológica para o desenvolvimento de software educativo colaborativo**. Renote, v. 4, n. 1, 2006.
- BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 3 ed. Rio de Janeiro: Elsevier, 2005.
- BEZERRA, Eduardo. **Princípios De Análise E Projeto De Sistemas Com Uml**. 3 ed. Rio de Janeiro: Elsevier, 2006.
- BRAUDE, Eric. **Projeto de Software da Programação à Arquitetura: Uma abordagem baseada em Java**. Porto Alegre: Bookman, 2008.
- CAETANO, Rodrigo. **Metodologias de desenvolvimento: qual a mais adequada**. Disponível em: <<http://computerworld.com.br/gestao/2009/08/05/metodologias-de-desenvolvimento-qual-a-mais-adequada>>. Acesso em 15 novembro 2016.
- CERVO, Arnaldo Luiz; PEDRO Alcino; DA SILVA, Roberto. **Metodologia**

**científica.**

2 ed. Person Education, 2007.

DATASTORM. **Gestão de dados e informação.** Disponível em: <<http://datastorm.com.br/gestao-de-dados-e-informacoes-em-big-data/>>. Acesso em 09 novembro de 2016.

DATE, C. **Introdução a sistemas de banco de dados:** Tradução de Daniel Vieira. Rio de Janeiro: Elsevir, 2003.

DEITEL, Harvey M.; DEITEL, Paul J. **Java Como Programar.** 8 ed. São Paulo: Person Prentice Hall, 2010.

FEITOSA, Marcia Porto. **Fundamentos de banco de dados.** 1 ed. São Paulo: Edição do Autor, 2013.

GOMES, Fabio. **Conheça o Bootsface: Bootstrap para Java Server Face.** Disponível em: <<http://www.devmedia.com.br/conheca-o-bootsface-bootstrap-para-java-server-face/31070>>. Acesso em 25 novembro 2016.

HEUSER, Carlos Alberto. **Projeto de Banco de Dados.** 6 ed. Porto Alegre: Bookman, 2009.

KORTH, H.F. e SILBERSCHATZ, A. **Sistemas de Bancos de Dados, Makron Books.** 2 ed. edição revisada, 1994.

KOSCIANSKI, André. SOARES, Michel dos Santos. **Qualidade de Software: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software.** 2 ed. São Paulo: Novatec 2007.

LEITE, Mário. **Acessando Bancos de Dados com ferramentas RAD: Aplicações em Visual Basic.** Rio de Janeiro: Brasport, 2007.

LOBO, Edson J.R. **Curso de Engenharia de Software.** São Paulo: Digerati Books, 2008.

LOBO, Edson Junio Rodrigues. **Guia Prático de Engenharia de Software.** São Paulo: Digerati Books, 2009.

LUCKOW, Décio Heinzelmann; MELO, Alexandre Altair. **Programação Java para web.** 2 ed. São Paulo: Novatec Editora, 2015.

MACÊDO, Diego. **Modelagem Conceitual, Lógica e Física de Dados.** Disponível em <<http://www.diegomacedo.com.br/>>. Acesso em 26 novembro 2016.

MANNINO, Michael V. **Projeto, Desenvolvimento de Aplicações e Administração de Banco de Dados.** 3 ed. São Paulo: Amgh, 2014.

MARTINS, José Carlos Cordeiro. **Técnicas para gerenciamento de projetos de software.** Rio de Janeiro: Brasport, 2007.

MELO, Ana Cristina. **Desenvolvendo aplicações com UML 2.2: do conceitual à implementação**. 3 ed. Rio de Janeiro: Brasport, 2010.

MENDES, Douglas Rocha. **Programação Java com ênfase em orientação a objetos**. São Paulo: Novatec Editora, 2009.

MORAN, José Manuel. **Os novos espaços de atuação do professor com as tecnologias**: Conhecimento local e conhecimento universal: diversidade, mídias e tecnologias na educação. Curitiba: Champagnat, v. 2, 2004.

MUÑOZ, Paula Andrea Zuñiga. **Sistemas de Informação**. TCC em Revista, n. 15, 2012.

OLIVEIRA, FÁTIMA BAYMA DE. **Tecnologia da Informação e da Comunicação: articulando processos, métodos e aplicações**. Rio de Janeiro: E-papers: Fundação Getúlio Vargas, 2009.

PINOCHET, Luis Hernan Contreras. **Tecnologia da Informação e Comunicação**. 1 ed. Rio de Janeiro: Elsevir, 2014.

POMPILHO, S. Analise **Essencial: guia prático de análise de sistemas**. Rio de Janeiro: Ciência Moderna, 2002.

PRESSMAN, Roger S., MAXIM, Bruce R. **Engenharia de Software uma abordagem profissional**. 8 ed. São Paulo: Amgh, 2016.

PRESSMAN, Roger. S., **Engenharia de Software**. 6 ed. McGraw-Hill, 2006.

PRIKLADNICKI, Rafael, AUDY, Jorge. **Desenvolvimento distribuído de software**. Rio de Janeiro: Elvier, 2008.

RAMAKRISHNAN, Raghu. **Sistemas de Gerenciamento de Banco de Dados**. 3 ed. Porto Alegre: Amgh, 2011.

RAMOS, Marília do Rosário Delgado. **A importância da tecnologia da informação e comunicação nas Organizações de Serviços de Contabilidade**. Mindelo, 2010.

RÊGO, Bergson Lopes. **Gestão e governança de dados: promovendo os dados como ativo de valor nas empresas**. Rio de Janeiro: Braspost, 2013.

REZENDE, Denis Alcides. **Engenharia de software e Sistemas de Informação**. 3 ed rev. e ampl. Rio de Janeiro: Brasport, 2005.

SOMMERLILLE, Ian. **Engenharia de software**. 9 ed. São Paulo: Person Prentice Hall, 2011.

TAYLOR, Allen G. **SQL Para Leigos**. 8 ed. Rio de Janeiro: Alta Books, 2016.

TOTLAB, **O que é TIC**. Disponível em: <<http://totlab.com.br/noticias/o-que-e-tic-tecnologias-da-informacao-e-comunicacao/>>. Acesso em 08 de setembro de 2016.

VAZQUEZ, Carlos Eduardo. SIMÕES, Guilherme Siqueira. **Engenharia de requisitos: Software orientado ao negócio**. Rio de Janeiro: Brasport, 2016.

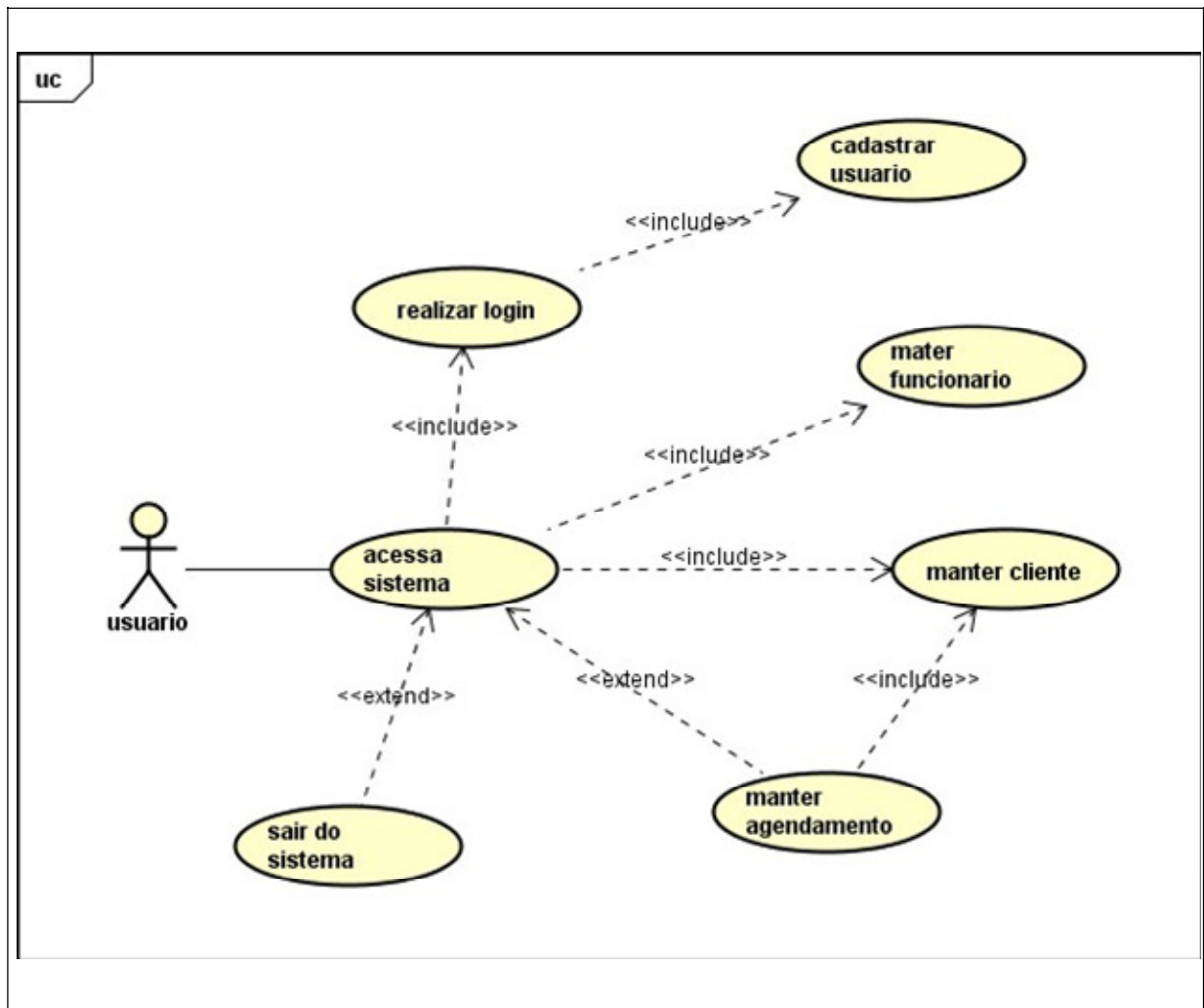
WAZLAWICK, Raul Sidnei. **Engenharia de Software Conceitos e Práticas**. Rio de Janeiro: Elsevier, 2013.

WILKER. **Qual a melhor IDE para se começar na programação**. Disponível em <<http://www.blog.wlconsultoria.net/qual-melhor-ide/>>. Acesso em 25 novembro 2016.

## APÊNDICE A – DIAGRAMA DE CASO DE USO

No desenvolver do sistema foi criado para auxiliar no desenvolvimento o diagrama de caso de uso pelo motivo do mesmo ser de extrema importância no desenvolvimento de sistemas.

**FIGURA 08** – Diagrama de caso de uso



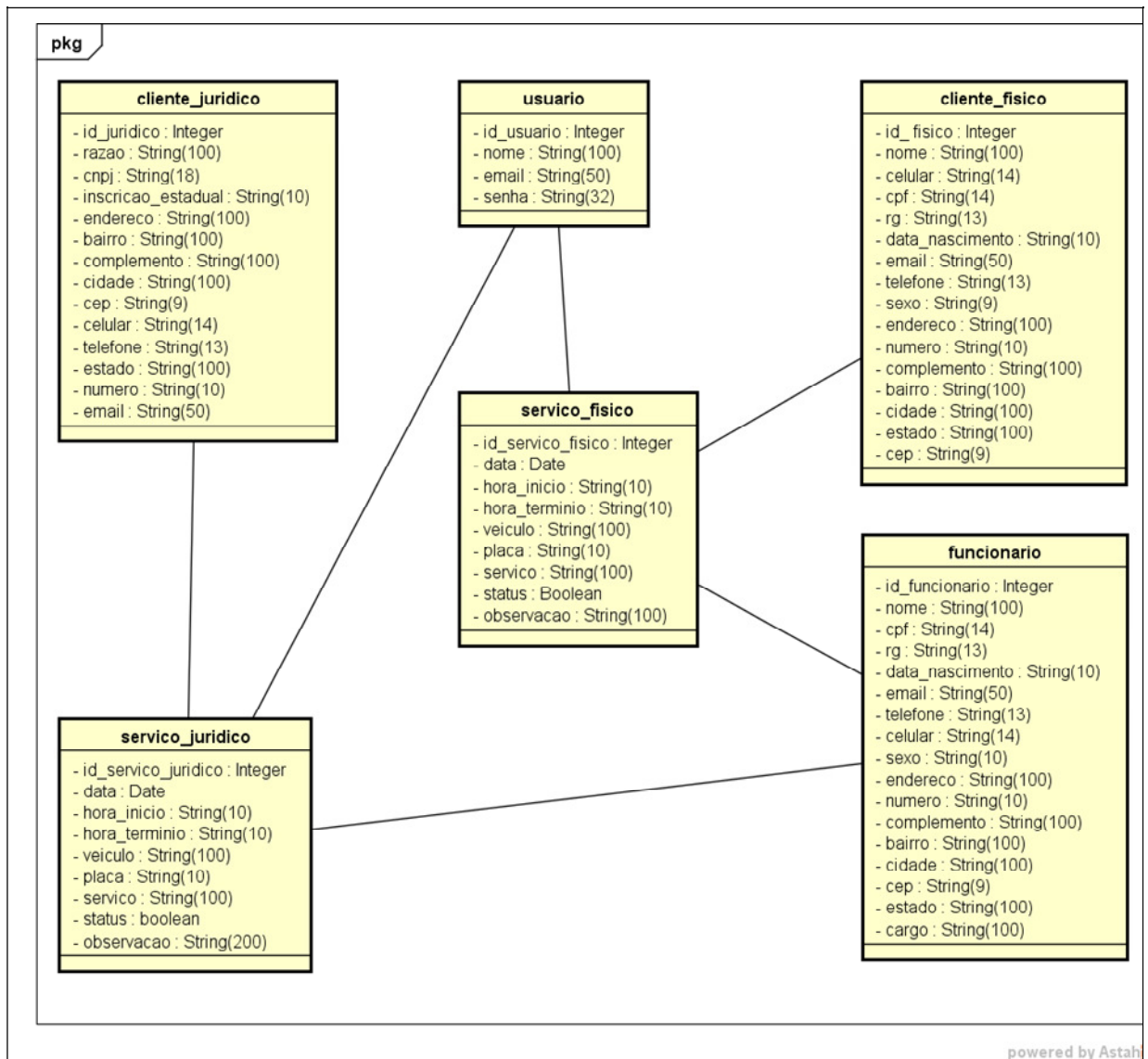
**Fonte:** Elaborado pelo autor.



## APÊNDICE B – DIAGRAMA DE CLASSE

Diagrama de classe foi desenvolvido para auxiliar na construção das classes e do banco de dados.

**FIGURA 09** – Diagrama de classe.



**Fonte:** Elaborado pelo autor.