

**COLÉGIO ESTADUAL DE EDUCAÇÃO PROFISSIONAL  
PEDRO BOARETTO NETO  
CURSO TÉCNICO EM INFORMÁTICA**

**KAMYLI RENATA RIBEIRO DE SOUZA  
MARIA CRISTINA BECKER**

**TODA BELLA**

**CASCADEL - PR  
2023**

**KAMYLI RENATA RIBEIRO DE SOUZA**

**MARIA CRISTINA BECKER**

**TODA BELLA**

Projeto de Desenvolvimento de Software  
do Curso Técnico em Informática do  
Colégio Estadual de Educação  
Profissional Pedro Boaretto Neto –  
Cascavel, Paraná.

Orientadores: Prof<sup>a</sup> Aparecida S.Ferreira<sup>1</sup>  
Prof. Reinaldo C. da Silva<sup>2</sup>  
Prof. Célia K.Cabral<sup>3</sup>

**CASCADEL - PR**  
**2023**

---

<sup>1</sup>Especialista em Educação Permanente: Saúde e educação pela FioCruz – Fundação Osvaldo Cruz. Especialista em tecnologias da Informação pela UNIVEL – União Educacional de Cascavel. Pedagoga formada pela UNIPAR – Universidade Paranaense. Professora do núcleo técnico do Estado do Paraná – Ensino médio técnico.

<sup>2</sup>Especialização em Educação Profissional Tecnológica. Faculdade Origenes Lessa, FACOL, Brasil. Especialização em Tecnologias e Educação a Distância. Faculdade Origenes Lessa, FACOL, Brasil. Especialização em MBA em Data Warehouse e Business Intelligence. UNYLEYA EDITORA E CURSOS S/A, Unyleya, Brasil. Graduação em Programa Especial de Formação Pedagógica - Docência em Informática. Faculdade de Administração, Ciências, Educação e Letras, FACEL, Brasil. Graduação em Gestão da Tecnologia da Informação. Universidade Estácio de Sá, UNESA, Brasil.

<sup>3</sup>Graduação em Sistemas Distribuídos para Internet JAVA. Universidade Federal do Paraná, UTFPR, Brasil. Graduação em Tecnólogo em Processamento de Dados. União Educacional de Cascavel, UNIVEL, Brasil.

**KAMYLI RENATA RIBEIRO DE SOUZA**

**MARIA CRISTINA BACKER**

**TODA BELLA**

Este Projeto de Conclusão de Curso foi julgado e aprovado pelo Curso Técnico em Informática do Colégio Estadual de Educação Profissional Pedro Boaretto Neto.

Cascavel, Pr., xx de Xxxxx de 2023

**COMISSÃO EXAMINADOR**

---

Prof<sup>a</sup>. Aparecida da S. Ferreira<sup>1</sup>  
Especialista em Tecnologia da  
Informação  
*Faculdade de Ciências Sociais*  
*Aplicadas de Cascavel*  
Orientadora

---

Prof. Reinaldo

Web Design

---

Prof<sup>a</sup>. Célia Kouth Cabral  
Pós-graduada em Sistemas  
Distribuídos JAVA.  
Universidade Tecnológica Federal  
do Paraná - UTFPR  
Banco de dados

---

Prof<sup>a</sup> Ana Cristina Santana  
Especialista em Gestão e Docência  
no ensino superior, médio e técnico.  
Coordenadora de curso

## Sumário

1	INTRODUÇÃO.....	5
1.1	Apresentação do Problema.....	6
2	OBJETIVOS.....	7
3	METODOLOGIA.....	8
4	REFERENCIAL TEÓRICO.....	9
5	DOCUMENTAÇÃO DO PROJETO.....	11
5.1	Requisitos.....	13
5.1.1	Requisitos funcionais.....	13
5.1.2	<b>Requisitos não funcionais.....</b>	<b>15</b>
5.2	Diagrama de Contexto.....	17
5.3	Diagrama de Fluxo de dados.....	18
5.4	Diagrama de Entidade e relacionamento.....	19
5.5	Dicionário de Dados.....	20
5.6	Diagrama de Caso de Uso.....	24
5.7	Diagrama de Classe.....	28
5.8	Diagrama de Sequência.....	34
5.9	Diagrama de Atividade.....	35
6	Telas.....	39
7	CONCLUSÃO.....	45
8	REFERÊNCIAS.....	46

## 1 INTRODUÇÃO

Este site tem o objetivo de fazer agendamentos de procedimentos estéticos sem sair de casa. Já se perguntou, por que a melhor forma de agendar procedimentos em um salão é online? O objetivo é o desenvolvimento de um sistema de informação capaz de facilitar o agendamento de horário e a comunicação entre clientes e gestores de salões de beleza.

O sistema foi desenvolvido direcionado para o controle de agendamento de horários, cadastro de clientes e funcionários, gerenciamento de serviços, visando o controle de um salão de beleza de maneira simples, trazendo ferramentas que facilitam o trabalho diário, de modo a aprimorar o seu desempenho, sendo compatível com a realidade dos estabelecimentos de pequeno porte.

Ao adquirir o sistema de agendamento eletrônico a clínica pode obter diversos benefícios. Entre eles, obter a credibilidade dos clientes. Tanto que muitos pacientes que estão por dentro das novidades preferem empresas que oferecem esse tipo de serviço. Mas por que o agendamento online tem ganhado tanto espaço? Passou-se a época em que tudo era feito à mão. Hoje a tecnologia tem sido cada vez mais frequente na vida dos indivíduos.

Podendo realizar o agendamento por conta própria a qualquer momento pela internet, a pessoa fica satisfeita por não precisar gastar tempo, agilidade seu procedimento.

Ao solicitar o agendamento online de uma consulta, o paciente deverá preencher todos os seus dados corretamente, como nome completo, documentos, endereços, telefones, entre outros. Então a credibilidade começa desse ponto. Quando uma consulta é marcada pelo meio convencional, muitas vezes esses mesmos dados fornecidos podem correr o risco de estarem incorretos. Isso muitas vezes pode acabar deixando o paciente desconfortável. Quando o próprio é responsável pelo preenchimento diminui que esse deslize aconteça.

O sistema de agendamento online permite que o paciente escolha a melhor data e horário de consultas que estiverem disponíveis. Isso acarreta a uma maior credibilidade, já que é muito difícil que com esse sistema aconteça uma confusão com os horários. Se o próprio paciente escolheu, ele tem o conhecimento de que

esse será o horário aproximado de atendimento.

Com a acelerada evolução tecnológica nos últimos anos, impactou diretamente no processo de informatização nas empresas que buscam produzir mais com menos recursos, consequentemente, proliferando em todos os setores da sociedade. O setor de beleza, em especial, salão de beleza é uma atividade que está se adequando ao processo de informatização, uma vez, que o setor está em constantes mudanças em função da sua clientela. Neste caso, exigindo agilidade nos processos de atendimento, fazendo-se necessário o uso de sistemas informatizados para controlar o gerenciamento das informações. (GONTIJO,2018).

Considerando-se o modo como cada salão de beleza gerencia os seus produtos, serviços e sua clientela, encontram-se sistemas já prontos para a área, geralmente generalistas, outros requer que o salão tenha um suporte técnico, o que acarreta em elevação dos custos. Em ALMEIDA e MACHADO (2010) destacam a importância da informatização nas pequenas empresas e inclui os salões de beleza nesta classe, ressaltando o impacto das tecnologias. As empresas do setor de beleza precisam adequar a uma clientela cada vez mais exigente, pela qualidade dos serviços ofertados nos salões de beleza, seja eles pequenos ou médios e o uso da informática tornou-se estratégica para qualquer negócio, o diferencial entre elas será o investimento, melhorias na qualidade da prestação dos serviços e na capacitação dos seus funcionários. Apesar da existência de softwares no mercado os mesmos não se adequam a necessidades de muitos salões. Essas premissas permitem que as empresas definam possíveis fortalecimentos do processo de gestão, garantindo o diferencial de atuação e por consequência, vantagem competitiva em relação aos concorrentes, e os gestores podem tomar decisões rápidas e de fontes seguras. Atualmente, é fundamental para qualquer empresa a busca por competitividade, pois dela depende a sua permanência e destaque no mercado. No desenvolvimento de sistemas envolve o uso de linguagens, banco de dados.

### **1.1 Apresentação do Problema**

Hoje em dia as pessoas buscam mais praticidade ao resolver seus compromissos do cotidiano, então para que você não tenha que se deslocar de casa até o estabelecimento, ou ter que fazer uma ligação, você terá uma opção mais prática e rápida que será fazer um agendamento online.

## **2 OBJETIVOS**

Fazer agendamentos de procedimentos estéticos online.

- Agendamento de clientes: faz o agendamento dos clientes.
- Cadastro de serviços: cadastra e armazena os dados dos serviços;
- Cadastro de funcionários: cadastra e armazena os dados dos funcionários;
- Cadastro clientes: cadastra e armazena os dados dos clientes;
- Cadastro de usuários: cadastra os usuários do sistema;
- Cadastro de produto e tipo de produto: cadastra e armazena os dados dos produtos e tipos de produtos;
- Cadastro de fornecedores: cadastra os fornecedores do salão;
- Lançamento de vendas: gerenciar a venda dos produtos e/ou serviços
- Lançamento de compras: gerencia compras e estoques;
- Emissão relatórios; emite os relatórios que dão suporte a gerência.



### **3 METODOLOGIA**

A metodologia adotada neste trabalho buscou delinear o processo de construção de um sistema para automatizar salões de beleza. Considera-se trabalho prático, caracterizado como pesquisa aplicada, direcionada para a construção de um sistema para a solução de um problema (GIL, 2002, Jung 2004). Utilizou-se a pesquisa bibliográfica documental para embasar a sua construção, trazendo as experiências relatadas em trabalhos publicados e conhecidos sobre a construção de sistema para salão de beleza. Para diagnosticar os problemas nos pequenos salões, desenvolveu-se um questionário que foi aplicado para identificar as necessidades reais dos estabelecimentos para levantamento dos requisitos de software: negócio e funcionalidade. Na sequência, analisaram-se os dados coletados com a pesquisa e definiu o roteiro de desenvolvimento do sistema, quais os requisitos necessários, a descrição do minimundo e a descrição das regras de negócio. Definido o roteiro, foram escolhidas as técnicas e recursos tecnológicos a serem utilizados, como o ambiente de edição e desenvolvimento, o banco de dados, a linguagem de programação e as ferramentas e frameworks. Após toda a análise de requisitos e a escolha das ferramentas a serem utilizadas deu-se início a implementação, a tarefa de “codificação” começando do “zero” e auxílio das bibliotecas, tendo como base a documentação oriunda da fase de design junto com a documentação de requisitos e pôr fim a realização de testes tendo por resultado um sistema para gerenciamento de salões de beleza de acordo com demanda de mercado, atendendo setor de beleza que deseja se sobressair e permanecer no mercado competitivo por meio da informatização.

#### 4 REFERENCIAL TEÓRICO

O referencial teórico é a base teórica que fundamenta e dá suporte ao estudo ou pesquisa que está sendo desenvolvido. É a revisão bibliográfica que tem como objetivo situar o tema em questão dentro de um contexto mais amplo e relacionar com os conhecimentos e teorias existentes na área.

Para iniciar o referencial teórico é necessário definir os conceitos do que estamos utilizando, definiremos aqui, o que nos ajudou a constituir esse projeto com um breve resumo sobre cada item utilizado.

Sistemas de informação são sistemas que utilizam tecnologia da informação para gerenciar, organizar e processar informações. Eles são usados em empresas e organizações para automatizar processos de negócios, gerenciar bancos de dados, criar aplicativos e gerar relatórios.

Front end: É a camada do sistema com a qual o usuário interage diretamente. É responsável pela apresentação e interação visual com o usuário. Incluir interfaces gráficas, o design de elementos visuais, a programação em linguagens como HTML, CSS e JavaScript, e componentes como botões, formulários, menus e animações. De acordo com Viana (2017), O desenvolvedor front-end é responsável por “dar vida” à interface. Trabalha com a parte da aplicação que interage diretamente com o usuário. Por isso, é importante que esse desenvolvedor também se preocupe com a experiência do usuário.

Back end: É a parte do sistema responsável pelo processamento e pela lógica das operações. Engloba o gerenciamento e a manipulação de dados, a implementação das regras de negócio, a interação com bancos de dados, a autenticação de usuários, a segurança e a implementação dos recursos do sistema. Desenvolvido utilizando linguagens de programação como Python, Java, PHP, C#. De acordo com Viana (2017), como o nome sugere, o desenvolvedor back-end trabalha na parte de “trás” da aplicação. Ele é o responsável, em termos gerais, pela implementação da regra de negócio.

HTML, sigla para HyperText Markup Language, é uma linguagem de marcação utilizada para criar e estruturar páginas na web. De acordo com Mishra et al. (2015), o HTML é considerado uma linguagem básica para o desenvolvimento web, sendo a base para a maioria das páginas web disponíveis na internet.

CSS, sigla para Cascading Style Sheets, é uma linguagem de estilo utilizada para definir a aparência de uma página web. Segundo Faria e Zandonadi (2019), o CSS permite a criação de estilos personalizados para cada elemento HTML, como fontes, cores, tamanhos, espaçamento, entre outros.

JS, sigla para JavaScript, é uma linguagem de programação utilizada para adicionar interatividade e dinamismo às páginas web. De acordo com Flanagan (2011), o JavaScript permite a criação de animações, validação de formulários, interação com o usuário, dentre outras funcionalidades.

PHP, sigla para Hypertext Preprocessor, é uma linguagem de programação utilizada para criar páginas dinâmicas e interativas na web. Segundo Welling e Thomson (2016), o PHP é amplamente utilizado em conjunto com o banco de dados MySQL para criar aplicações web robustas e escaláveis.

O XAMPP é um pacote de software que inclui o Apache, o PHP, o MySQL e outras ferramentas necessárias para executar um servidor web localmente em um computador. É usado para desenvolvimento web e testes antes de colocar um site ou aplicativo em produção. Segundo Faraone e Oliveira (2018), o XAMPP é amplamente utilizado para o desenvolvimento local de aplicações web, permitindo que os desenvolvedores testem suas aplicações em um ambiente seguro e isolado.

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Ele inclui suporte para depuração, controle de versionamento, realce de sintaxe, complementação inteligente de código, e refatoração de código.

O MySQL é um sistema de gerenciamento de banco de dados relacional usado para armazenar e organizar dados. É frequentemente usado em aplicativos da web para armazenar informações do usuário, como nome, endereço e informações de Login. O PHP é uma linguagem de programação de servidor popular usada com o MySQL para criar aplicativos da web dinâmicos e interativos.

O Drawio é um software de diagramação gratuito usado para criar diagramas e fluxogramas. O Drawio pode ser usado para criar diagramas de fluxo para o desenvolvimento de sites e aplicativos da web, para visualizar a estrutura e o fluxo de dados em um sistema de gerenciamento de banco de dados.

## 5 DOCUMENTAÇÃO DO PROJETO

A documentação de um projeto é um conjunto de informações e registros que descrevem todos os aspectos relevantes do projeto, desde o seu conceito inicial até a sua implementação e manutenção. Ela pode incluir documentos, manuais, especificações técnicas, diagramas, fluxogramas e outros recursos que fornecem detalhes sobre o projeto.

Para uma documentação de projetos ideal é necessário ser completa, precisa, atualizada e de fácil compreensão para todos os envolvidos no projeto. Ela deve abranger todos os aspectos relevantes do projeto, como requisitos, objetivos, escopo, cronograma, recursos utilizados, decisões tomadas, testes realizados, problemas encontrados e soluções implementadas. Além disso, a documentação ideal deve ser organizada de forma lógica e estar disponível para consulta e referência futura.

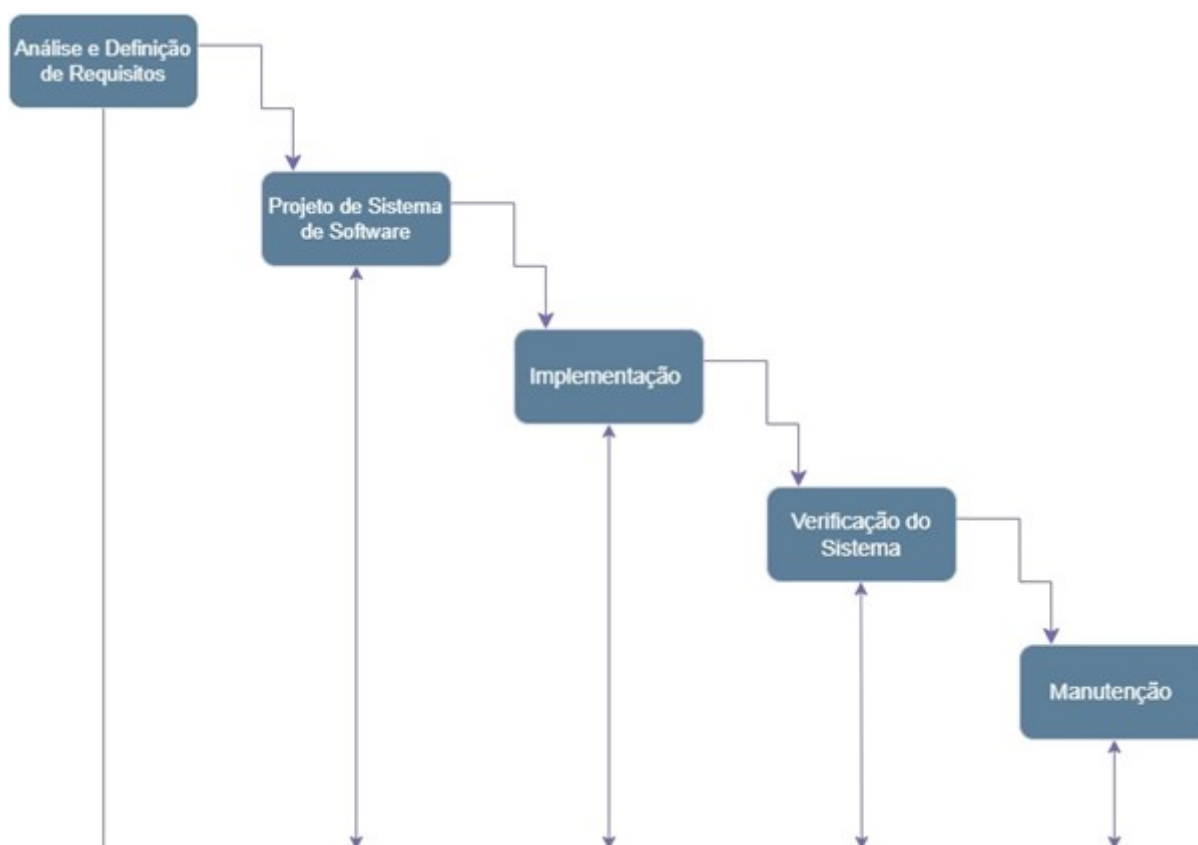
Documentamos nosso projeto por várias razões importantes. Primeiro, a documentação é essencial para garantir a continuidade do projeto ao longo do tempo, permitindo que pessoas diferentes compreendam e trabalhem nele, mesmo que a equipe mude. Além disso, a documentação ajuda a evitar retrabalho, pois fornece informações detalhadas sobre as decisões tomadas, evitando que erros se repitam no futuro. A documentação também facilita a comunicação entre os membros da equipe e outras partes interessadas, permitindo uma melhor colaboração e compreensão mútua. Além disso, a documentação pode servir como um recurso de aprendizado, permitindo que novos membros da equipe se familiarizem rapidamente com o projeto e seus detalhes.

Para documentar um programa, é importante seguir algumas práticas recomendadas. Primeiro, é necessário descrever o propósito e os objetivos do programa, destacando os problemas que ele se propõe a resolver. Em seguida, é importante documentar os requisitos do programa, ou seja, as funcionalidades esperadas e as restrições que devem ser consideradas. Além disso, é útil criar diagramas ou fluxogramas que ilustrem a estrutura do programa, como os módulos e suas interações. Também é importante fornecer instruções detalhadas sobre a instalação, configuração e uso do programa, além de exemplos de código e explicações sobre sua lógica interna. Por fim, é recomendado manter a

documentação atualizada à medida que o programa evolui, refletindo alterações e atualizações relevantes.

A elaboração de uma documentação eficaz envolve algumas etapas importantes. Primeiro, é necessário planejar o escopo e a estrutura da documentação, identificando os elementos-chave a serem incluídos. Em seguida, é preciso coletar informações relevantes sobre o projeto, como requisitos, decisões de design, testes realizados e soluções implementadas. Essas informações devem ser organizadas de forma lógica e clara, seguindo uma sequência compreensível. É importante utilizar uma linguagem clara e concisa ao escrever a documentação, evitando jargões técnicos excessivos. Além disso, é recomendado incluir exemplos, ilustrações e diagramas para facilitar a compreensão. Por fim, a documentação deve ser revisada e atualizada regularmente para refletir as mudanças e melhorias no projeto.

O propósito da documentação é fornecer um registro completo e preciso do projeto, permitindo que os envolvidos compreendam e trabalhem nele de forma eficiente. Ela serve como uma fonte confiável de informações sobre o projeto, facilitando a manutenção, a colaboração e a continuidade. A documentação também ajuda a evitar retrabalho, pois fornece orientações claras sobre as decisões tomadas e as soluções implementadas. Além disso, a documentação pode ser utilizada como um recurso de aprendizado para novos membros da equipe, permitindo que eles se familiarizem rapidamente com o projeto. Por fim, a documentação ajuda a melhorar a comunicação entre os membros da equipe e outras partes interessadas, garantindo uma compreensão mútua e eficaz do projeto.



Fonte: Becker,Souza, 2023

## 5.1 Requisitos

### 5.1.1 Requisitos funcionais

Parte da etapa de licitação, os requisitos funcionais são todos os problemas e necessidades que devem ser atendidos e resolvidos pelo software por meio de funções ou serviços. São exemplos de requisitos funcionais: inserir dados em um formulário, buscar pratos específicos em um cardápio, consultar o status de um pedido, realizar compras, comunicar-se com um atendente, alterar informações de um registro, elaborar relatórios. Tudo o que for relacionado a uma ação a ser feita é considerado uma função. Também é importante lembrar que quanto menos ambíguos e mais objetivos forem os requisitos funcionais, maior será a qualidade do software gerado.

Código	Identificação	Objetivo
RF01	Cadastro de funcionário	Cadastro de funcionário é realizado pela empresa, preenchendo um formulário com os dados exigidos.
RF02	Cadastro de usuário	Cadastro de usuário é realizado pelo mesmo, preenchendo o formulário com dados exigidos pelo site.
RF03	Login de usuário	Autenticação do usuário pelos dados cadastrados.
RF04	Edição do cadastro	Alteração de dados que já foram inseridos no site.
RF05	Escolha de funcionário	Escolha de funcionário desejado.
RF06	Agendamento	Escolha da data e horário desejado.
RF07	Confirmação	O usuário confirma o agendamento realizado.

Fonte: Becker,Souza, 2023

### **5.1.2 Requisitos não funcionais**

Requisitos são solicitações, desejos, necessidades. Um requisito é a propriedade que um software exhibe para solucionar problemas reais, é a conjuntura indispensável para satisfazer um objeto. Quando se trata de um software sob demanda, por exemplo, um requisito é uma maneira pelo qual o sistema oferecido deve fazer, ou um condicionamento no desenvolvimento do sistema.

Lembrando que, em ambas as ações, embora o programador ou o arquiteto de software tenha suas opiniões, é importante chegar em um acordo para resolver o problema do cliente.

Esse sempre será o foco. É muito importante frisar que manter uma concordância com os clientes e àqueles envolvidos é um dos principais objetivos dos requisitos. Um dos principais responsáveis pelo sucesso dos softwares, os requisitos, são a base para estimativas, modelagem, projeto, execução, testes e até mesmo para a manutenção dos mesmos.

Assim, os requisitos estão presentes ao longo de todo o ciclo de vida de um software. Ao começar um projeto, os requisitos já devem ser levantados, entendidos e documentados. Assim como realizar atividades de controle de qualidade para verificar, validar e garantir a qualidade dos mesmos para criar um aplicativo.

E isso vale para todos os modelos de software, como aplicativo de marketplace, aplicativo de mobilidade, aplicativo de entregas, aplicativo de streaming, e vários outros.

Gerenciar a evolução dos requisitos é importante, estando cientes de que os negócios com sua dinâmica não garantem estabilidade e podem vir a sofrer alterações. Desse modo é necessário manter a rastreabilidade entre os requisitos e as outras peças do projeto.



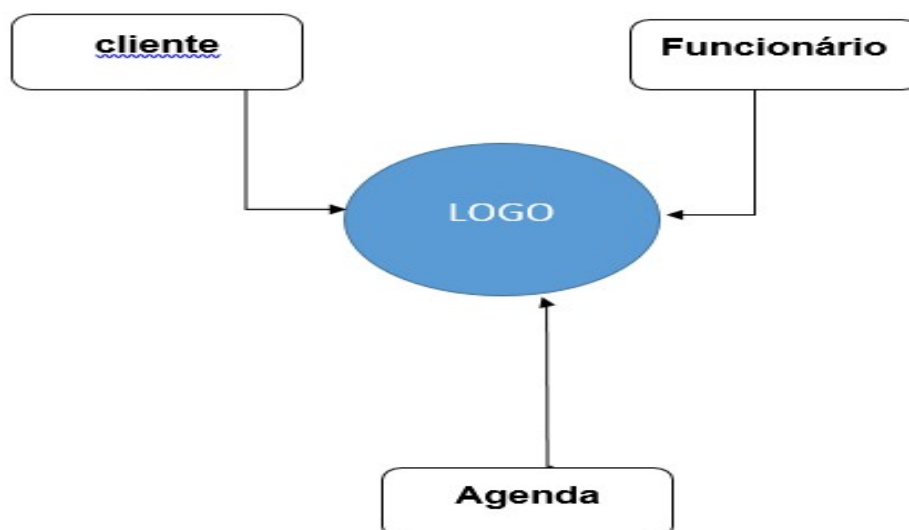
Código	Identificação	Objetivo
RNF01	Restrição de usuário	O usuário tem acesso ao site com o perfil de usuário, não sendo visível campos de administradores.
RNF02	Linguagem de programação	A linguagem que o site será desenvolvido.
RNF03	Banco de dados	MYSQL.
RNF04	Desempenho	o site deve ser capaz de processar um grande volume de dados e transações sem atrasos ou interrupções.
RNF05	Facilidade de uso	O site deve ser de fácil acesso.
RNF06	Confiabilidade	Alta.
RNF07	Ético	Manter um perfil ético.
RNF08	Erro	Caso não tenha cadastro, informar uma mensagem de erro.
RNF09	Mensagem de Confirmação	Mensagem de confirmação para o agendamento realizado.

Fonte: Becker,Souza, 2023

## 5.2 Diagrama de Contexto

Conforme Tech (2021), Os Diagramas de Contexto são apresentados no Guia PMBOK® do PMI como uma das ferramentas e técnicas para coletar os requisitos do projeto. O diagrama de contexto descreve visualmente o escopo do produto, mostrando um sistema de negócios (um processo, equipamentos, sistema computacional, etc.) e, em seguida, mostra a relação que este tem com outras entidades externas (sistemas, atores, grupos organizacionais, repositórios de dados, etc.).

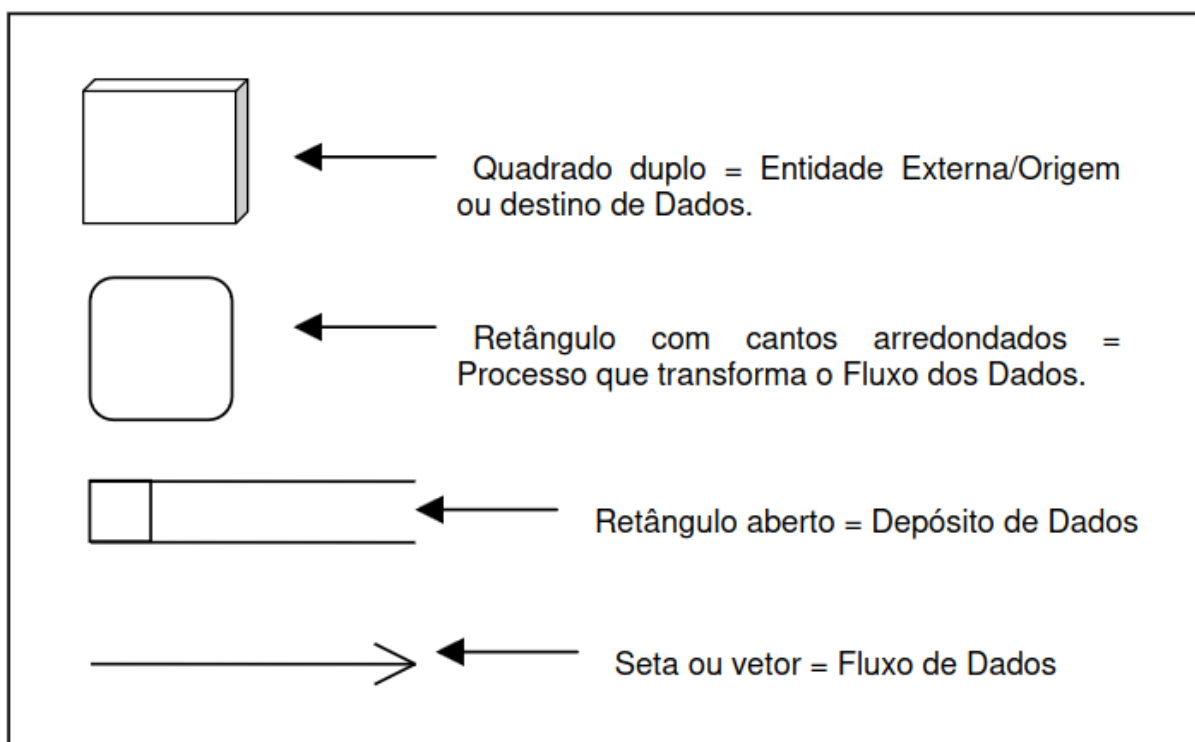
No gerenciamento de projetos o diagrama de contexto é utilizado para a modelagem do escopo, descrevendo graficamente o escopo do produto. Essa técnica já vem sendo utilizada há muito tempo em outras áreas. Um diagrama de contexto pode ser encarado como uma versão especializada do Diagrama de Fluxo de Dados (DFD) utilizado em análise de sistemas, que é uma visualização gráfica do movimento de dados através de um sistema. Em análise de negócios estes diagramas também podem ser utilizados para capturar e comunicar as interações e o fluxo de dados entre processos de negócios.



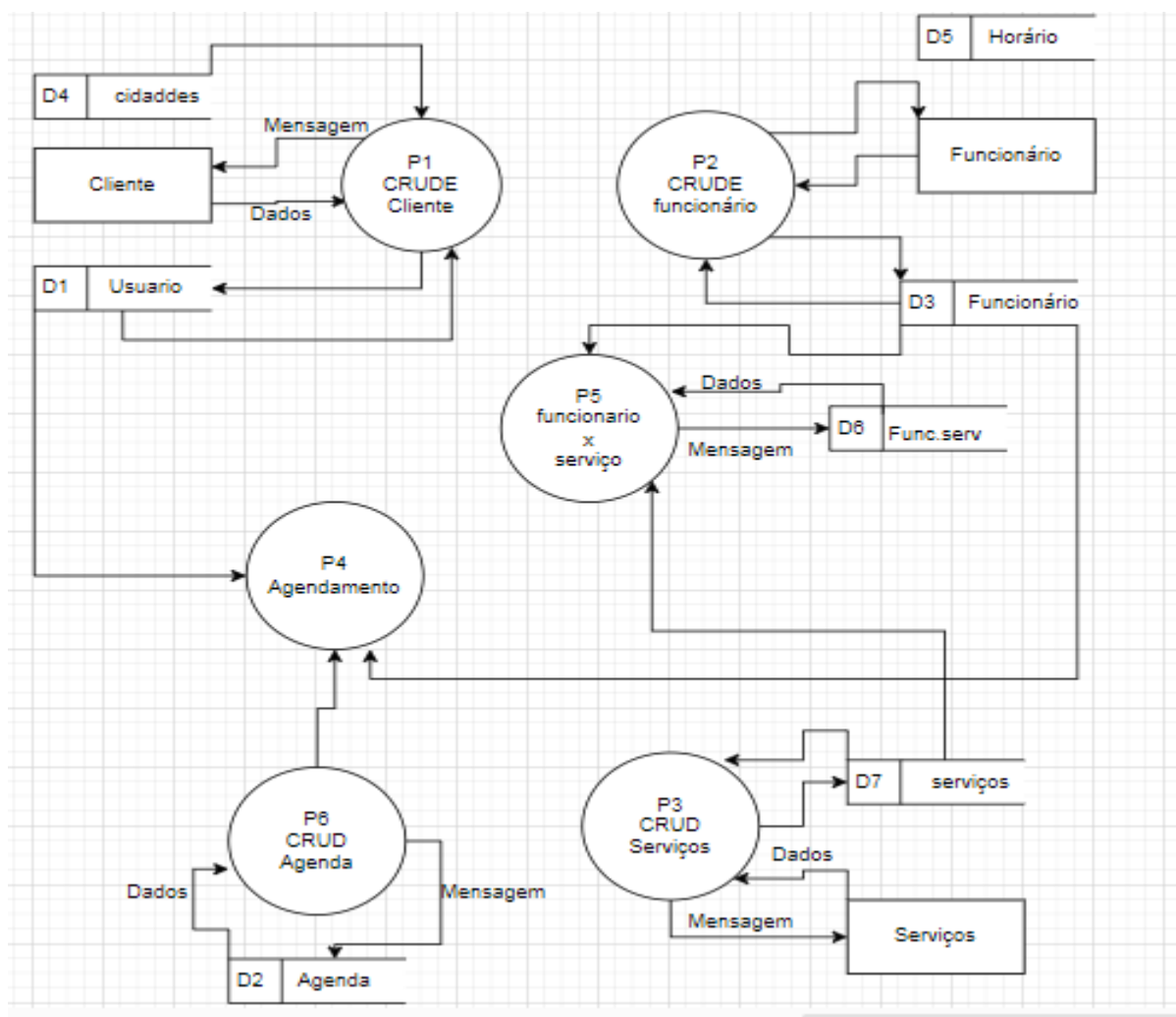
Fonte: Becker, Souza, 2023

### 5.3 Diagrama de Fluxo de dados

De Barros (2013) O Diagrama de Fluxo de Dados (DFD) é uma das principais ferramentas utilizadas no projeto de sistemas de informação. O DFD é um diagrama gráfico, baseado apenas em quatro símbolos, que mostra a estrutura do sistema e sua fronteira, ou seja, todas as relações entre os dados, os processos que transformam esses dados e o limite entre o que pertence ao sistema e o que está fora dele. DFD é uma representação em rede dos processos (funções) do sistema e dos dados que ligam esses processos. Ele mostra o que o sistema faz e não como é feito. É a ferramenta de demonstração central da análise estruturada. Um DFD apresenta as partes componentes de um sistema e as interfaces entre elas. É um conjunto integrado de procedimentos, sendo que as partes do computador poderão estar inseridos ou não. Na elaboração de um DFD, utilizaremos quatro símbolos que nos permitirão, debater e apresentar ao usuário todo o processo, sem assumir nenhum compromisso com implementações e demonstrar a sua fluência, sem a preocupação com a hierarquização e tomadas de decisão. São os seguintes símbolos utilizados na elaboração de um DFD:



Fonte: Barros (2013)



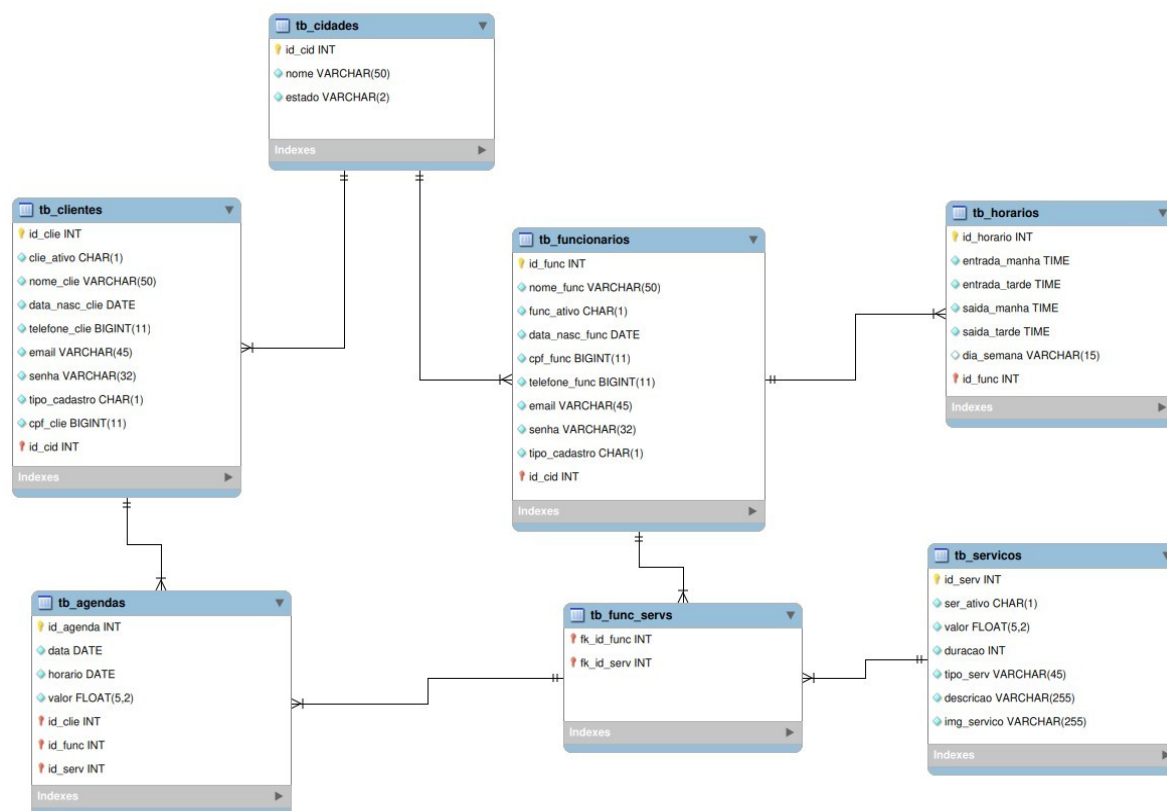
Fonte:Becker,Souza, 2023

#### 5.4 Diagrama de Entidade e relacionamento

Diagrama de Entidade e Relacionamento (DER) é uma representação gráfica do Modelo de Entidade e Relacionamento. Enfim, isso evita excesso de abstração, trazendo para a realidade informações pertinentes de uma forma mais visual e, conseqüentemente, mais intuitiva.

Saindo do micro para o macro, temos todas as informações pertinentes para modelar o nosso Diagrama de Entidade e Relacionamento. Existem diversas

maneiras de fazer isso, sugiro modelar do modo que achar mais confortável e que os outros entendam.



Fonte: Backer, Souza, 2023

## 5.5 Dicionário de Dados

Segundo Costa (2021), um dicionário de dados armazena e comunica informações sobre um banco de dados. É considerado uma lista de dados com os principais termos e métricas do banco de dados que está vinculado. Esse material também pode ser entendido como um glossário dos seus dados. Embora pareça simples, ele é crucial para alinhar o funcionamento da equipe que irá trabalhar com os dados e manter todas as informações padronizadas. O conteúdo do dicionário de dados pode variar, mas normalmente inclui alguns itens como: Nomes e definições

das variáveis, Propriedades detalhadas dos dados (tipo de dados, tamanho, se há valores nulos), Sequência das perguntas ou da coleta das variáveis, Dados de referência (para fazer classificações e domínios descritivos), Codificação para dados ausentes.

Variável	Nome Variável	Tipo de Variável	Descrição	Valores Permitidos	Possui valores nulos?	Anotações
ID do participante	ID	numérica	Número de identificação dos usuários em ordem de sequência	001-999	não	
Idade em anos	idade	numérica	Idade do paciente em anos	18-65	não	
Gênero	genero	categorica	Gênero informado pelo participantes	1 = masculino 2 = Feminino	sim	Essa categorização tem como referencia o IBGE

Figura 1. Exemplo de um dicionário de dados. Criado por Costa(2021).

## Informações relevantes em um dicionário de dados

### Variável

Um dos primeiros passos na confecção de uma pesquisa ou levantamento de dados é definir o que será avaliado, ou seja, quais são as variáveis que irão ajudar a responder a minha pergunta principal. Ao determinar quais são as variáveis que irão compor o seu banco de dados, você já pode iniciar a confecção do seu dicionário de variáveis.

### Nome da Variável

O nome da variável representa o rótulo\* (ou label) que você quer atribuir a sua variável e que representará a variável no banco de dados. Ao contrário do item anterior, nesse item você deve colocar um nome simples, sem acentuação, caracteres especiais e sem espaços entre palavras.

Por exemplo, imagine que você irá utilizar uma escala psicométrica e cada item dessa escala (E) será uma variável. No nome da variável você pode colocar um código simples que presente a variável, como: E1 e E2.

### Tipo de Variável

Saber o tipo de variável que contém no banco é fundamental para analisar os dados. Muitas vezes é necessário recalcular uma variável para deixá-la no formato ideal, esse cenário poderia ser evitado com um bom planejamento da coleta de dados e a utilização de dicionários de dados.

Uma variável pode ser numérica ou categórica. Em especial nas variáveis

categóricas, é importante especificar as categorias e se essas categorias possuem uma ordem de importância ou não. Então, sinalize sempre no tipo de variável se ela é numérica ou categórica, especifique ainda se as categorias possuem uma ordem de importância ou hierárquica (como, escolaridade ou grau de uma doença). Também há variáveis do tipo tempo (ou time), que representa alguma informação de tempo, como: segundos, minutos, horas, datas. Nesses casos, seria importante discriminar qual das medidas de tempo estão representadas na variável.

### **Descrição**

O espaço para descrição é destinado para incluir maiores detalhes sobre determinada variável. Uma descrição detalhada é encorajada para que todos possam entender o que realmente aquela variável representa, qual a razão dela está ali e como deve ser tratada durante as análises.

Você pode acrescentar também detalhes de como e quando essa variável foi coletada ou em que momentos ela não será coletada. Com essas informações em mãos, você pode, por exemplo, ter maiores detalhes sobre possíveis *missings* no seu banco de dados.

Ao mesmo tempo, evite descrições desnecessárias e circulares, como por exemplo: “a idade do paciente representa quanto tempo o paciente está vivo”. Diga informações diretas e relevantes, como: “a idade fornecida pelo paciente em anos”. Dessa forma, você sabe que a ideia foi fornecida diretamente os dados da variável em questão foi fornecido pelo paciente e está em anos.

### **Valores Permitidos**

No campo de valores permitidos você deve fornecer todos os valores que são possíveis de serem coletados. Para as variáveis numéricas indique o valor mínimo e máximo que podem ser informados no seu banco. Já para as variáveis categóricas informe o número da categoria, bem como o rótulo associada a ela.

Por exemplo, se a variável gênero tiver codificações de 1 e 2, indique no dicionário o que significa esses valores respectivamente (1 = masculino e 2 = feminino).

### **Possui Valores Nulos**

Durante o processo de análise de dados os valores faltantes (*missings*) sempre são um problema. Há momentos em que não é possível coletar determinada

variável e os dados daquele caso ficarão em branco ou vazio. Isso é perfeitamente possível no mundo real, mas é um problema durante o processo de análise de dados. Portanto, ao fazer o exercício da construção de um dicionário de variáveis você pode evitar dados nulos ou *missings*. Mas se não há saída, sinalize no seu dicionário de dados que aquela variável pode ter valores nulos ou faltantes.

### **Anotações**

Ao final do seu dicionário de dados você pode criar um campo de anotações. Coloque nesse campo demais informações que não foram até então declaradas. Como, por exemplo, a referência que você utilizou para criar as categorias de uma variável ou o estudo que tem uma variável parecida e que fomenta a discussão dos seus resultados.

### **Considerações Finais**

Vimos nesse artigo a importância de se criar dicionários de dados desde o início do planejamento da sua pesquisa. Ao criar um dicionário de dados você facilitará a comunicação entre membros da equipe e evitará erros comuns da coleta de dados. Você estará ainda melhorando a qualidade dos seus dados, bem como obtendo resultados mais fidedignos na sua análise de dados.

Você pode criar um dicionário de dados utilizando o *Excel*, *Spreadsheet* ou o *Word*. O dicionário de dados deve ser um documento simples e democrático, que facilitará o seu trabalho e dos seus colegas.



tb_agendas	DataType	PK	FK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_agenda	INT	✓		✓							
data	DATE			✓	✓						
horario	VARCHAR(50)			✓							
valor	FLOAT(5,2)			✓							
tb_clientes_id_cliente	INT	✓	✓	✓							
id_func	INT	✓	✓	✓							
id_serv	INT	✓		✓							

tb_cidades	DataType	PK	FK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_cid	INT	✓		✓							
nome	VARCHAR(50)			✓							
estado	VARCHAR(2)			✓						PR	

tb_clientes	DataType	PK	FK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_cliente	INT	✓		✓							
cliente_ativo	CHAR(1)			✓						S	
nome_cliente	VARCHAR(50)			✓							
data_nascimento_cliente	DATE			✓							
tipo_cadastro	char(1)			✓						C	
telefone_cliente	BIGINT(11)			✓	✓			✓			
email	VARCHAR(45)			✓	✓			✓			
senha	VARCHAR(32)			✓	✓						
cpf_cliente	BIGINT(11)			✓	✓			✓			
id_cid	INT			✓							

tb_funcionarios	DataType	PK	FK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_func	INT	✓		✓							
nome_func	VARCHAR(50)			✓							
func_ativo	CHAR(1)			✓						S	
data_nascimento_func	DATE			✓							
cpf_func	BIGINT(11)			✓	✓			✓			
telefone_func	BIGINT(11)			✓	✓			✓			
email	VARCHAR(45)			✓	✓						
senha	VARCHAR(32)			✓	✓						
imagem_func	VARCCHAR(255)			✓							
id_cid	INT		✓	✓							

tb_horarios	DataType	PK	FK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_horario	INT	✓		✓							
entrada_tarde	TIME			✓							
saida_manha	TIME			✓							
saida_tarde	TIME			✓							

tb_agendas	DataType	PK	FK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_agenda	INT	✓		✓							
data	DATE			✓	✓						
horario	VARCHAR(50)			✓							
valor	FLOAT(5,2)			✓							
tb_clientes_id_cliente	INT	✓	✓	✓							
id_func	INT	✓	✓	✓							
id_serv	INT	✓		✓							

tb_cidades	DataType	PK	FK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_cid	INT	✓		✓							
nome	VARCHAR(50)			✓							
estado	VARCHAR(2)			✓						PR	

tb_clientes	DataType	PK	FK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_cliente	INT	✓		✓							
cliente_ativo	CHAR(1)			✓						S	
nome_cliente	VARCHAR(50)			✓							
data_nascimento_cliente	DATE			✓							
tipo_cadastro	char(1)			✓						C	
telefone_cliente	BIGINT(11)			✓	✓			✓			
email	VARCHAR(45)			✓	✓			✓			
senha	VARCHAR(32)			✓	✓						
cpf_cliente	BIGINT(11)			✓	✓			✓			
id_cid	INT			✓							

tb_funcionarios	DataType	PK	FK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_func	INT	✓		✓							
nome_func	VARCHAR(50)			✓							
func_ativo	CHAR(1)			✓						S	
data_nascimento_func	DATE			✓							
cpf_func	BIGINT(11)			✓	✓			✓			
telefone_func	BIGINT(11)			✓	✓			✓			
email	VARCHAR(45)			✓	✓						
senha	VARCHAR(32)			✓	✓						
imagem_func	VARCCHAR(255)			✓							
id_cid	INT		✓	✓							

tb_horarios	DataType	PK	FK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_horario	INT	✓		✓							
entrada_tarde	TIME			✓							
saida_manha	TIME			✓							
saida_tarde	TIME			✓							

dia_semana	VARCHAR(15)			✓								
id_func	INT		✓	✓								
tb_servicos	DataType	PK	FK	NN	UQ	BIN	UN	ZF	AI	Default	Comment	
id_serv	INT	✓		✓						S		
ser_ativo	CHAR(1)			✓								
duracao	INT			✓								
valor	FLOAT(5,2)			✓								
tipo_serv	VARCHAR(45)			✓								
descricao	VARCHAR(255)			✓								
img_servicos	VARCHAR(255)			✓								
tb_funcos_servs	DataType	PK	FK	NN	UQ	BIN	UN	ZF	AI	Default	Comment	
id_func	INT		✓	✓								
id_serv	INT		✓	✓								

Fonte: Becker,Souza, 2023

## 5.6 Diagrama de Caso de Uso

Unidade funcional provida pelo sistema, subsistema, ou classe manifestada por sequências de mensagens entre o sistema e um ou mais atores representa uma possível utilização do sistema por um ator, que pode ser uma pessoa, dispositivo físico, mecanismo ou subsistema que interage com o sistema alvo, utilizando algum de seus serviços. Um caso de uso narra a interação entre o sistema e os atores envolvidos, para atingir um ou mais objetivos Deve estar relacionado a um processo bem definido, com começo, meio e fim. Exemplos: Emprestar Livro Vender Produtos Incluir ordem de serviço Casos de Uso. Muitas vezes, são utilizados como um contrato entre desenvolvedor e cliente. Podem ser identificados com base no Documento de requisitos Podem ser utilizados como forma de identificar os requisitos, para depois escrever o documento de requisitos.

### O que são atores?

Responsáveis, direta ou indiretamente, pela interação com o sistema Uma pessoa, dispositivo físico, mecanismo ou subsistema.

### Como identificar os atores?

Tipos de ator:Ator principal: interage diretamente com o sistema computacional Ator secundário: interage com outros atores.

Exemplos de atores:

funcionário de um banco sensor de fumaça subsistema de autorização de crédito

### Como identificar os atores?

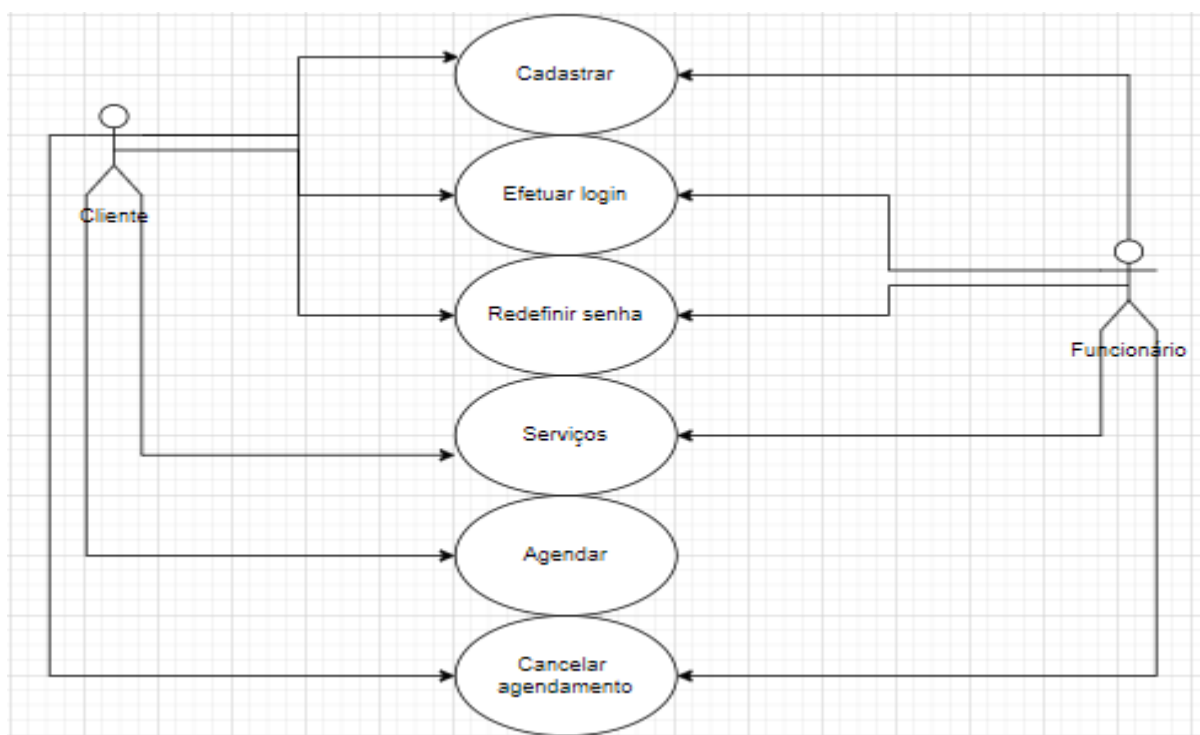
Exemplo (Sistema de Biblioteca): Ao emprestar um livro, o Atendente é quem opera

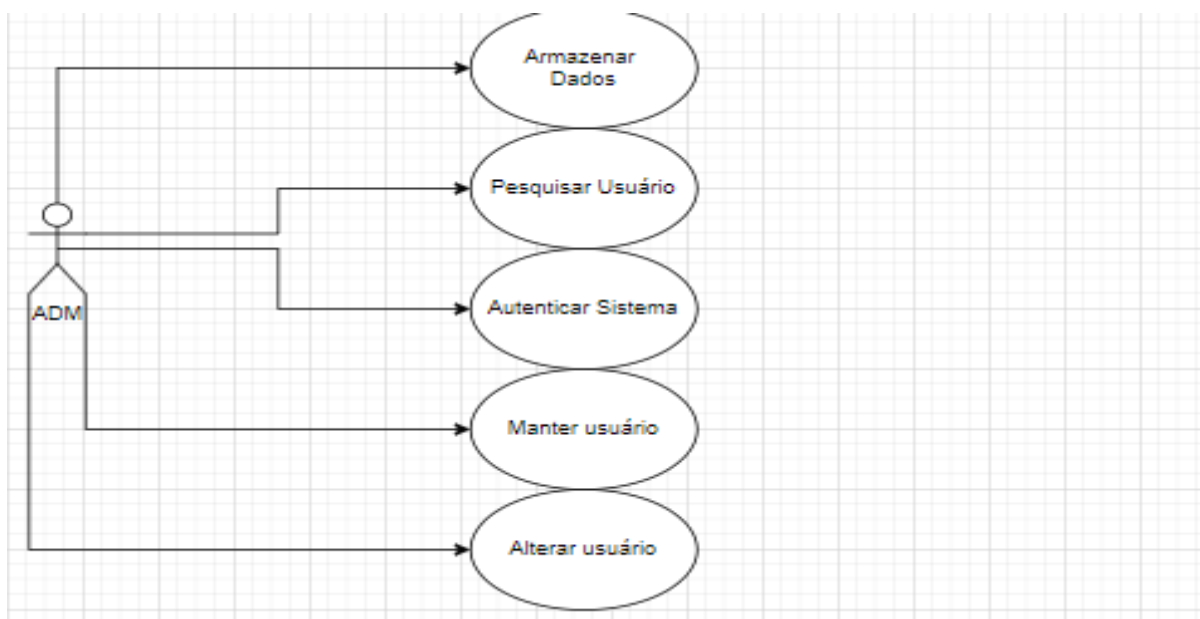
o computador e realiza a transação, portanto, é o ator principal Já o Leitor, interage com o atendente, sendo um ator secundário

### Como identificar os casos de uso?

Analisar cada requisito do sistema em busca dos grandes eventos que ocorrem no mundo real e que dão origem a uma interação entre um ator e o sistema.

Um diagrama de caso de uso é excelente para: Mostrar a fronteira do sistema o que está dentro ou fora dele Dar uma visão geral do comportamento do sistema como ele é usado e por quem





Fonte: Becker,Souza, 2023

Cenário de agendamento para um site de salão de beleza:

**\*\*Nome do Site:\*\* TODA BELLA**

**\*\*Descrição do Site:\*\***

Toda Bella é um site dedicado ao agendamento de serviços de beleza em salões de todo o país. Oferecemos uma plataforma fácil de usar que permite aos clientes encontrar e agendar serviços de beleza, economizando tempo e simplificando o processo de marcação.

**\*\*Público-alvo:\*\***

Nosso site atende a pessoas que buscam serviços de beleza de alta qualidade e conveniência. Nosso público-alvo inclui homens e mulheres de todas as idades que desejam agendar serviços de salão, como cortes de cabelo, coloração, manicure, pedicure, massagens e muito mais.

**\*\*Cenário de Agendamento:\*\***

Aqui está como funciona o cenário de agendamento no Toda Bella:

1. **\*\*Pesquisa de Serviços:\*\*** Os clientes visitam o site BelezaAgenda.com e têm a

opção de pesquisar por serviços específicos, salões próximos ou profissionais de beleza de sua escolha.

2. **\*\*Seleção de Data e Hora:\*\*** Após encontrar o serviço desejado, os clientes podem selecionar uma data e hora disponível que se adapte à sua programação. Eles também podem ver a disponibilidade de estilistas ou terapeutas específicos, se desejarem.

3. **\*\*Confirmação de Reserva:\*\*** Uma vez selecionada a data e hora, os clientes podem revisar os detalhes de sua reserva e confirmá-la com um único clique.

4. **\*\*Lembranças por E-mail e SMS:\*\*** Os clientes recebem automaticamente lembretes por e-mail e SMS alguns dias antes de seu compromisso para garantir que não se esqueçam.

5. **\*\*Política de Cancelamento:\*\*** BelezaAgenda.com implementa uma política de cancelamento que permite aos clientes reagendar ou cancelar compromissos com um aviso prévio, proporcionando flexibilidade.

6. **\*\*Feedback e Avaliações:\*\*** Após cada visita ao salão, os clientes têm a oportunidade de fornecer feedback e avaliações sobre a experiência, contribuindo para a melhoria contínua dos serviços.

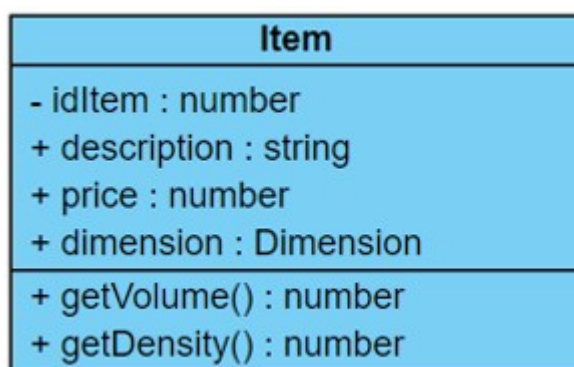
7. **\*\*Conta do Cliente:\*\*** Os clientes podem criar uma conta Toda Bella para armazenar seu histórico de agendamentos, preferências e informações de contato para agilizar futuras reservas.

Este cenário de agendamento online no site Toda Bella tem como objetivo proporcionar uma experiência de reserva fácil e conveniente, ao mesmo tempo em que conecta os clientes a salões de beleza de qualidade em sua região.

## 5.7 Diagrama de Classe

Os diagramas de classes nos permitem modelar o conteúdo estático e os relacionamentos entre as classes. Em um diagrama de classes podemos mostrar as variáveis membro e as funções de uma classe específica. Ademais, há a possibilidade de mostrar se uma classe herda de outra ou se contém uma referência para outra.

As classes são símbolos da classe no diagrama pode ser subdividido em compartimentos. O nível superior contém o nome da classe, enquanto o segundo possui as suas variáveis e o terceiro apresenta os seus métodos.



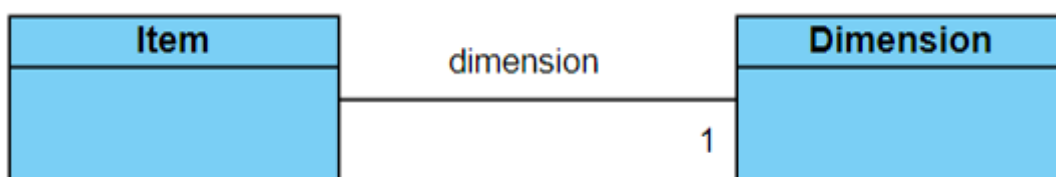
Fonte: Dirani,2022

Diagrama de classes mostrando os atributos e métodos de um tipo identificado como Item. Os atributos e funções costumam ter caracteres na frente de seus nomes. Um traço denota acesso privado, ou seja, somente integrantes dentro da própria classe estão autorizados a consultar o membro em questão. Por outro lado, o jogo da velha corresponde ao acesso protegido, que é bastante similar ao privado, com a diferença de que as subclasses também podem utilizar as variáveis e métodos correspondentes. Enfim, o sinal de adição significa que o campo ou método são públicos, expostos para qualquer cliente consumir.

Esse tipo de detalhe às vezes é útil, mas alguns autores como Robert C. Martin (Uncle Bob) aconselham a não usá-lo com muita frequência. As declarações de variáveis e funções devem ser aplicadas apenas quando elas forem essenciais para o propósito do diagrama.

### Associação:

As associações entre classes reproduzem variáveis de instância que contêm referências para outros objetos. A direção da seta indica que uma classe possui uma referência para outra classe. O nome próximo à seta é a identificação da variável de instância, e o número revela quantas referências são armazenadas.

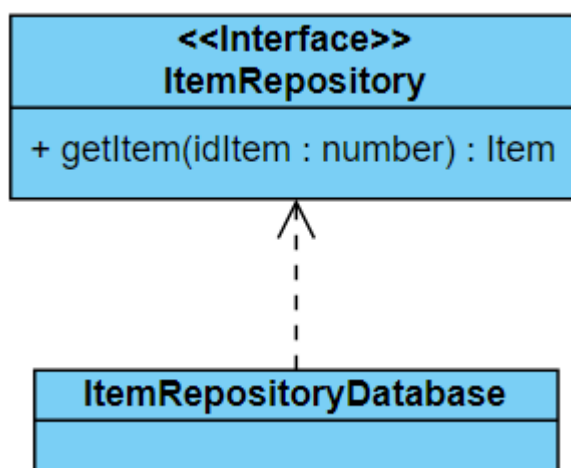


Fonte: Dirani,2022

A classe Item possui uma associação direta com a classe Dimension

### Herança

As setas de herança apontam diretamente para a classe base, de acordo com as especificações da UML. No caso de interfaces, as setas são desenhadas com um tracejado, embora esta especificidade não seja tão relevante de ser seguida pelo desenvolvedor.

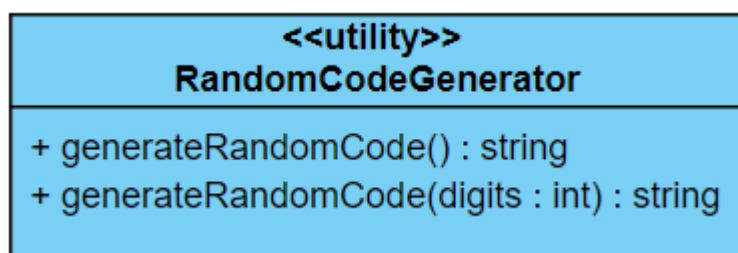


Fonte: Dirani,2022

A classe ItemRepositoryDatabase implementa (herda) da interface ItemRepository

### Estereótipos de classe:

Os estereótipos de classe aparecem entre dois caracteres semelhantes aos sinais de menor e maior, normalmente acima do nome da classe.



Fonte: Dirani,2022

Classe RandomCodeGenerator com o estereótipo utility

O estereótipo interface indica que todos os métodos de uma classe são abstratos. Já o estereótipo utility indica que todos os métodos e variáveis são estáticos.

### Classes abstratas:



Fonte: Dirani,2022

A classe OrderCode compõem de maneira exclusiva a classe Order

A composição, do mesmo modo que a agregação, compartilha das características propostas pela associação. A regra aplicada à agregação também



está presente na composição: não pode haver ciclos de instâncias. Entretanto, a UML expõe muito mais detalhes para este outro tipo de associação.

Uma instância de objeto não pode pertencer simultaneamente a mais de um objeto do mesmo tipo;

O objeto contêiner é responsável pela duração do objeto de composição. Se o objeto contêiner é destruído, o objeto de composição deve ser destruído com ele.

Multiplicidade



Fonte: [Dirani](#),2022

A multiplicidade da associação entre Order e OrderItem é de zero para muitos. Os objetos podem conter arrays ou coleções de outros objetos, ou podem conter muitos objetos do mesmo tipo em variáveis de instância distintas. Na UML, isso pode ser mostrado colocando-se uma expressão de multiplicidade na extremidade da associação.

As formas de multiplicidade permitidas são as seguintes:

Dígito: o número exato de elementos;

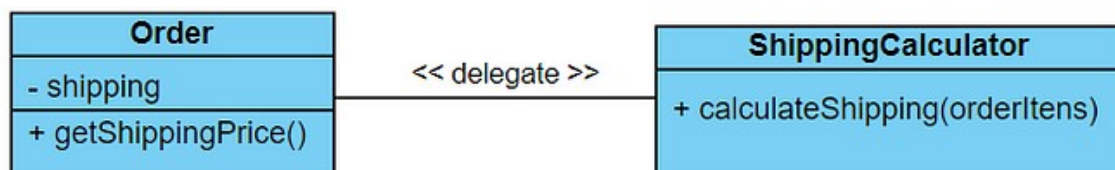
\* ou 0..\*: de zero para muitos;

0..1: zero ou um;

1..\*: de um para muitos;

3..5: de três a cinco.

Estereótipo de associação



Fonte: Dirani,2022

A classe Order delega a execução para a classe ShippingCalculato. As associações podem ser rotuladas com estereótipos que alteram os seus significados.

O estereótipo create indica que o destino da associação é criado pela origem;

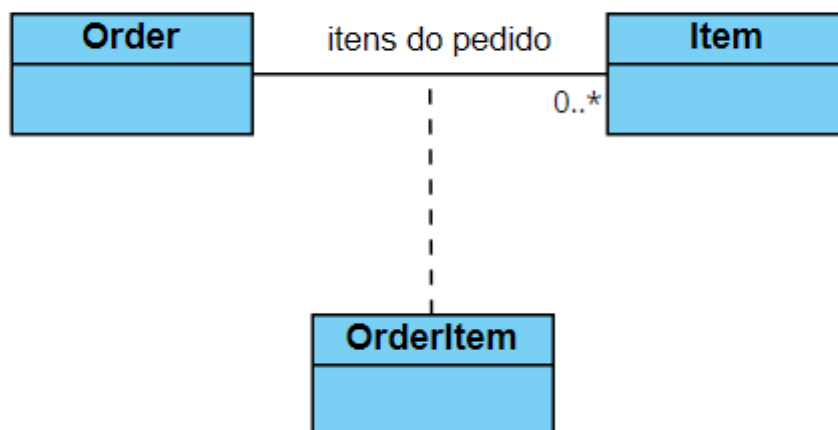
O estereótipo local é usado quando a classe de origem cria uma instância do destino e a mantém em uma variável local;

O estereótipo parameter mostra que a classe de origem obtém acesso à instância de destino por meio do parâmetro de uma de suas funções;

O estereótipo delegate é usado quando a classe de origem encaminha a chamada de uma função membro para o destino.

#### Classes associativas

As associações com multiplicidade nos informam que a origem está ligada a muitas instâncias do destino, mas o diagrama não nos informa que tipo de classe contêiner é usada. Isso pode ser representado pelo uso de uma classe associativa.



Fonte: Dirani,2022

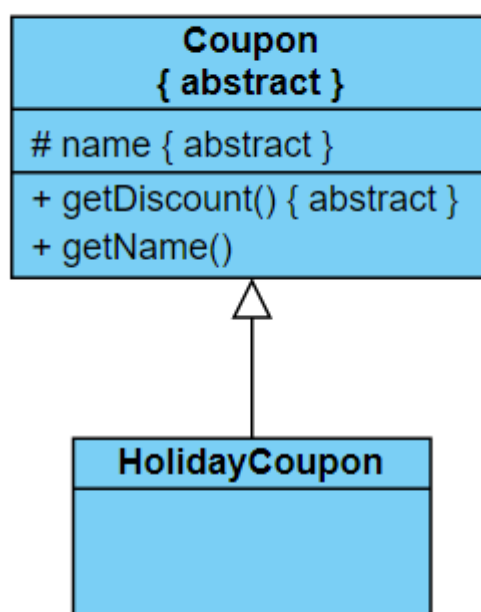
A classe OrderItem atua como uma classe associativa no relacionamento entre Order e Item.

As classes associativas mostram como uma associação específica é implementada.

No diagrama, elas aparecem como uma classe normal ligada à associação.

O suficiente para começar

Caso tenha absorvido todos os conceitos retratados nas seções anteriores, saiba que você já possui a base suficiente para começar a desenhar os seus próprios diagramas de classes quando precisar, seja em projetos pessoais ou profissionais. Assim como os outros modelos da UML, o diagrama de classes também possui vários conceitos e componentes adicionais. Entretanto, dificilmente eles são aplicados no dia a dia e é muito provável que você nunca sentirá falta de usá-los.



Fonte: Dirani,2022

Classe concreta HolidayCoupon herdando da classe abstrata Coupon.

Existem duas maneiras de retratar que uma classe ou método é abstrato na UML. Há a possibilidade de escrever o nome em itálico ou usar a propriedade abstract, representada entre chaves.

Agregação:

A classe OrderItem é um agregado da classe Order

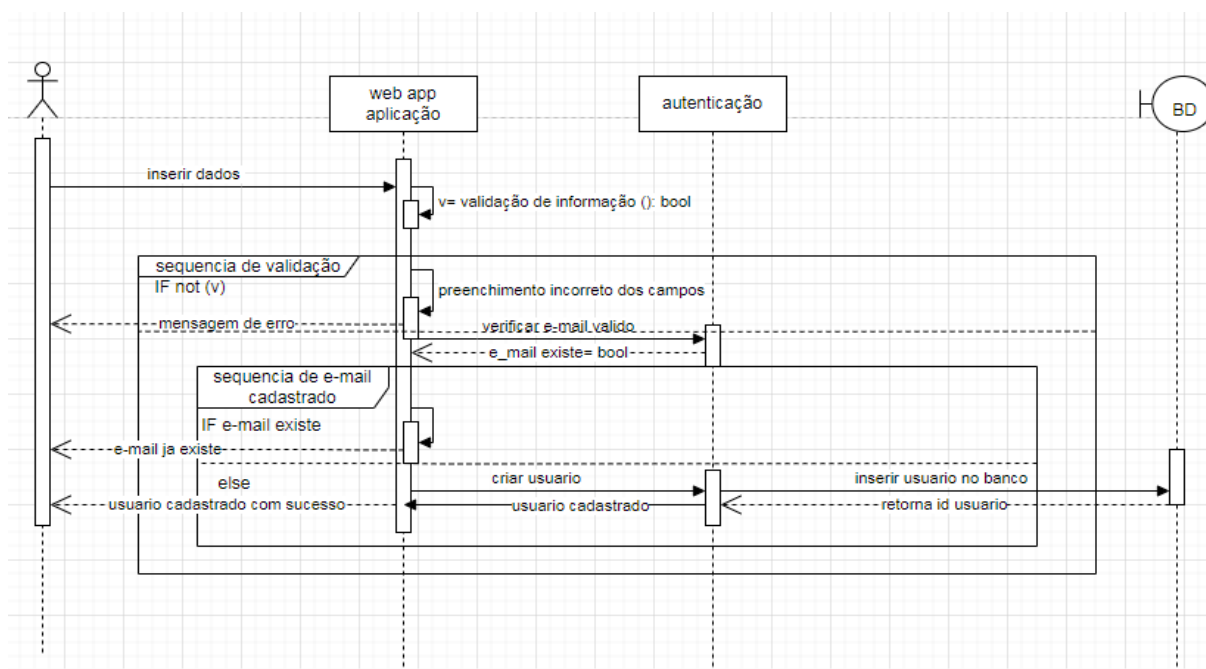
A agregação é uma forma especial de associação que significa um relacionamento todo/parte. A única regra que a UML fornece com relação às agregações é que um todo não pode ser sua própria parte. Logo, um objeto não pode ser agregado de si

mesmo.

## 5.8 Diagrama de Sequência

Um diagrama de sequência é um diagrama Unified Modeling Language (UML) que ilustra a sequência das mensagens entre objetos em uma interação. Um diagrama de sequência consiste em um grupo de objetos representados por linhas de vida e as mensagens que eles trocam durante a interação.

Um diagrama de sequência mostra a sequência de mensagens transmitidas entre objetos. Diagramas de sequência também mostram as estruturas de controle entre objetos. Por exemplo, linhas de vida em um diagrama de sequência para um cenário financeiro podem representar um cliente, um funcionário ou um gerente do banco. A comunicação entre o cliente, o funcionário e o gerente são representadas por mensagens transmitidas entre eles. O diagrama de sequência mostra os objetos e as mensagens entre os objetos.



Fonte: Becker, Souza, 2023

## 5.9 Diagrama de Atividade

Um diagrama de atividades é uma ótima ferramenta para criar um modelo de como um sistema deve se comportar ao realizar uma série de ações. Os diagramas de atividades são especialmente usados no desenvolvimento de software e podem ser empregados nas diferentes fases de um projeto. Este artigo discutirá o que é um diagrama de atividades, seus componentes e como você pode criar um para ajudar em seus projetos.

UML ou “Unified Modeling Language” é uma linguagem de modelagem de software usada para representar o design de um sistema específico. Na UML, os sistemas são modelados com base em sua estrutura ou comportamento e são classificados em diagramas UML estruturais ou comportamentais. Os diagramas de atividades UML se enquadram nos diagramas de comportamento porque modelam como um sistema se comporta quando as ações são executadas para concluir uma atividade ou processo. Embora as atividades sejam geralmente dispostas em ordem sequencial, os diagramas de atividades também podem mostrar atividades simultâneas ou que estão acontecendo ao mesmo tempo.

Um diagrama de atividades também pode ser usado para apresentar o fluxo de eventos e identificar os requisitos em um processo de negócios.

### **Para que são usados os diagramas de atividades?**

Os diagramas de atividades são usados para vários propósitos e em diferentes fases do processo de desenvolvimento. Durante a fase de planejamento, você pode criar um diagrama de atividades para descrever fluxos de trabalho e representar visualmente a sequência de atividades em um sistema. Enquanto isso, na fase de requisitos, os diagramas de atividades podem ser usados para apresentar diferentes casos de uso para um sistema. Quanto à fase de projeto, um diagrama de atividades pode modelar elementos de software como método, função e operação. Outros usos de diagramas de atividades incluem: Modelagem de processos de negócios. Ilustração dos aspectos dinâmicos de um sistema. Definição da lógica ou regras de um sistema. Identificação de casos de uso para um sistema. Simplificação de fluxos de trabalho complexos.

### **Quais são os componentes básicos de um diagrama de atividades?**

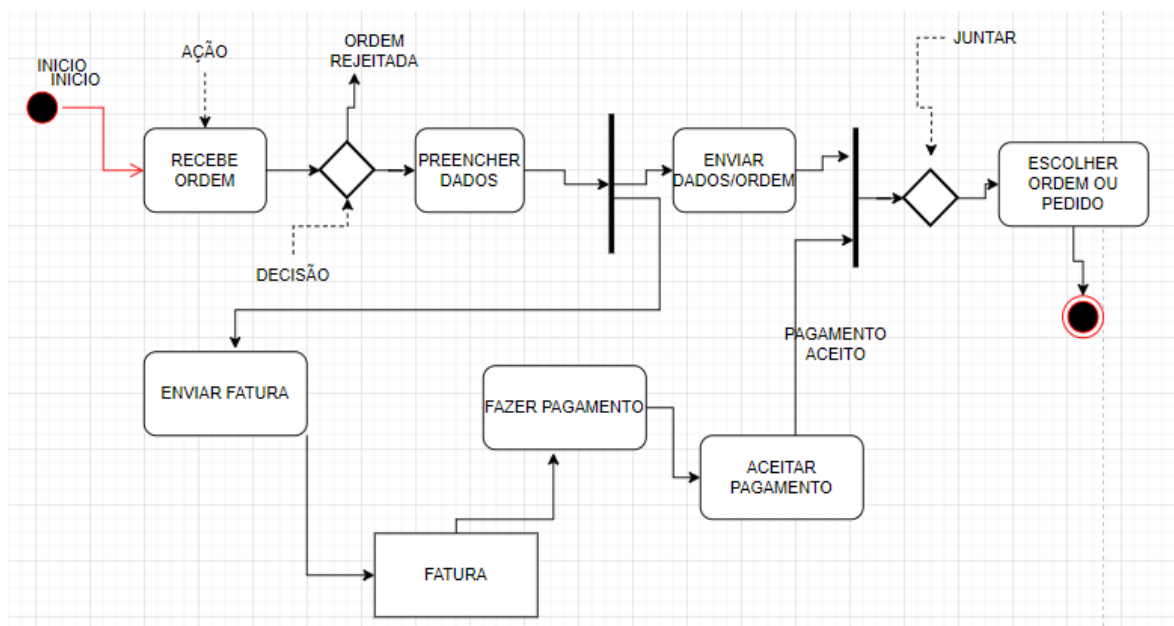
Os diagramas de atividades são frequentemente comparados aos

fluxogramas porque mostram como um processo “flui” de uma atividade para outra. No entanto, a diferença entre eles está nos elementos e símbolos que compõem um diagrama de atividades. Os componentes básicos incluem:

- **Fase inicial e fase final** – todos os diagramas de atividades têm uma fase inicial e uma fase final que marcam o início e o fim do processo.
- **Atividade ou estado de ação** – representa uma única atividade que coloca uma série de ações em movimento. Um exemplo pode ser um usuário fazendo login em sua conta em um sistema bancário para dispositivos móveis.
- **Ações** – uma ação ou etapa na atividade na qual o sistema ou o usuário executa uma tarefa. Seguindo nosso exemplo, uma ação pode ser um usuário verificando o saldo de sua conta.
- **Objetos** – estes são os materiais ou dados que são criados ou usados dentro de uma atividade.
- **Decisões** – estas são as decisões que precisam ser respondidas com ‘sim’ ou ‘não’ antes de prosseguir para outra ação ou atividade.
- **Sincronização** – consiste no nó de bifurcação que marca a criação de fluxos simultâneos e no nó de junção que mescla novamente os fluxos simultâneos em um único fluxo.
- **Sinais** – usados para indicar como ações podem ser feitas fora do sistema para modificar uma atividade. Por exemplo, para que o pagamento seja efetuado, um usuário pode receber um sinal na forma de uma OTP para autorização.
- **Swimlanes** (raias) – são colunas ou categorias criadas para agrupar atividades relacionadas que são realizadas por diferentes atores.

### O que são os símbolos do diagrama de atividades?

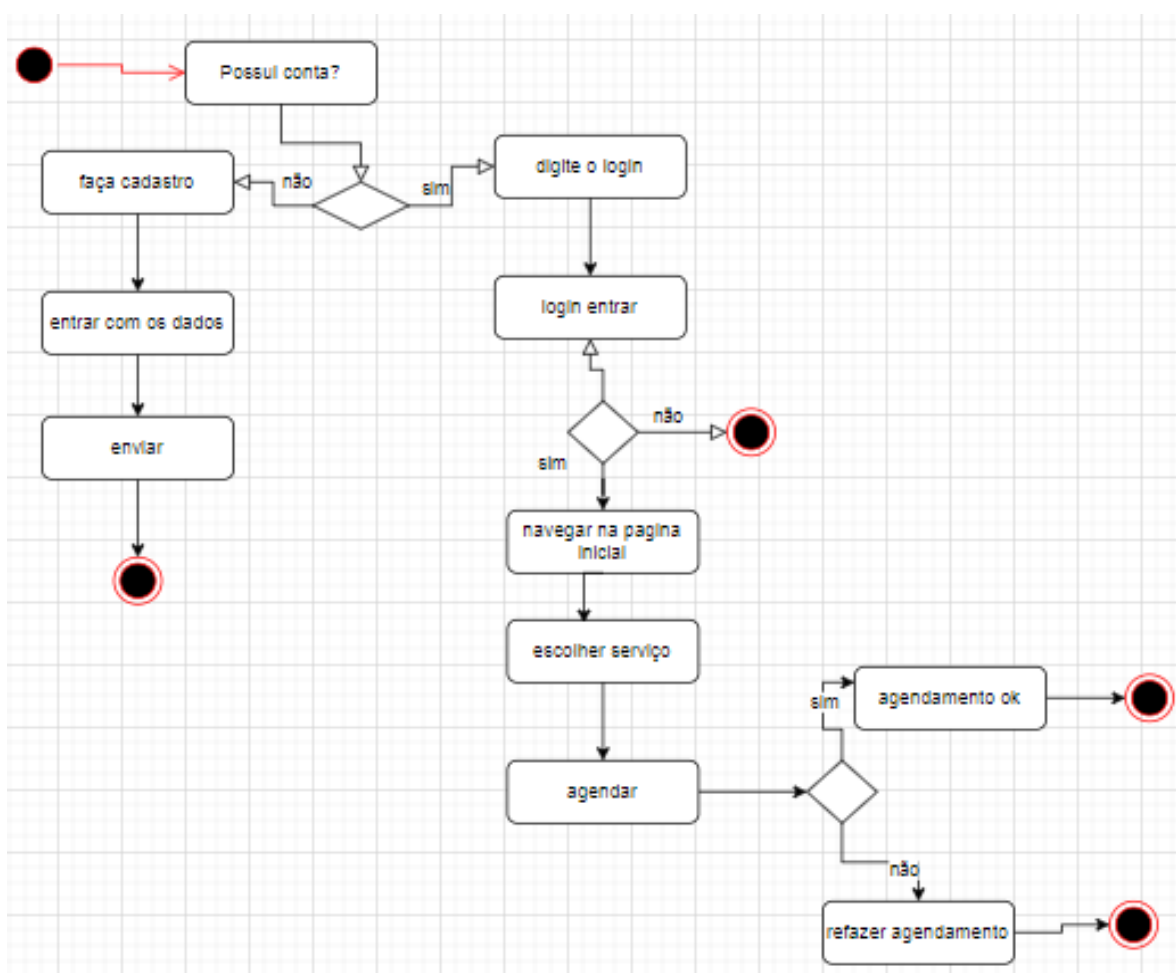
Os componentes mencionados acima são representados por várias formas e símbolos em um diagrama de atividades. Para sua referência, alguns deles estão incluídos neste exemplo:



Fonte: Ang, 2023

- **Nó inicial:** um pequeno círculo preenchido que simboliza o estado inicial ou o início da atividade.
- **Nó de atividade:** um retângulo com cantos arredondados que simboliza uma atividade ou estado de ação.
- **Nó de ação:** um símbolo em forma de estádio ou cápsula que é usado para representar uma ação.
- **Fluxo de ação:** uma seta representando a transição de uma atividade ou ação para outra. Também chamado de borda de atividade ou fluxo de controle.
- **Nó de objeto:** um retângulo que representa um objeto criado ou usado na atividade.
- **Fluxo de objetos:** um ângulo ou seta tracejada colocada após uma ação para mostrar a criação de um objeto ou antes de uma ação para mostrar que ela requer um objeto.
- **Nó de bifurcação ou nó de junção:** ambos são representados por uma linha horizontal grossa que divide uma ação em fluxos simultâneos (nó de bifurcação) ou une fluxos simultâneos em uma única ação (nó de junção).

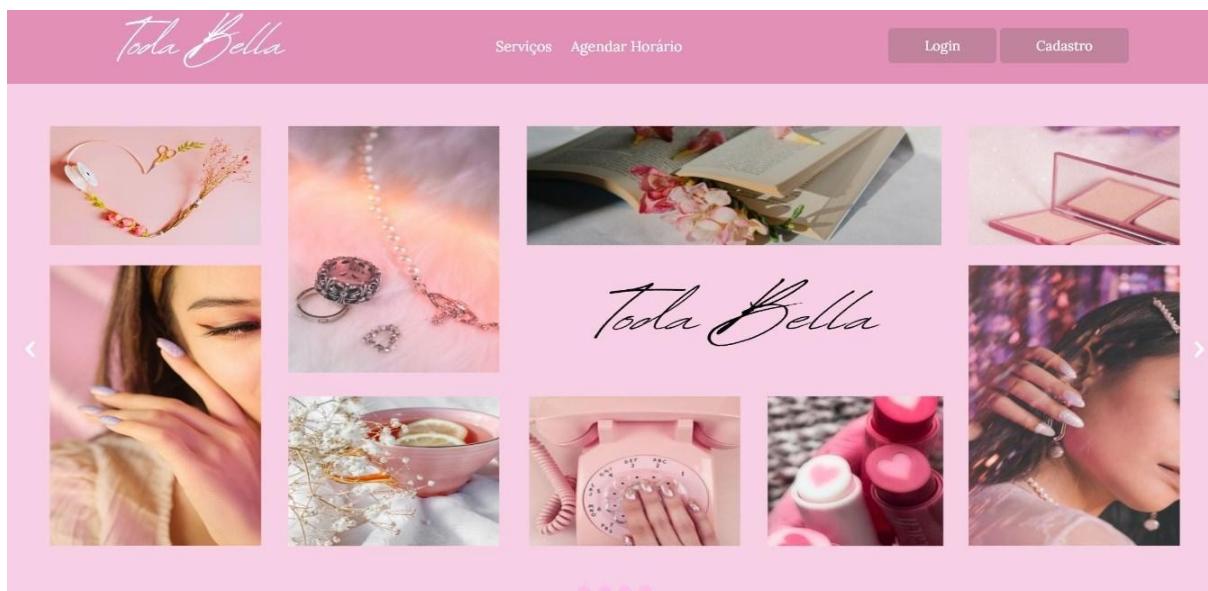
- **Evento temporal:** um símbolo de ampulheta que representa um intervalo de tempo dentro de uma atividade.
- **Sinais enviados e recebidos:** uma forma de pentágono em forma de seta (sinal enviado) indica que um sinal precisa de ser recebido para concluir uma ação, enquanto uma forma de bandeira com um rabo de andorinha (sinal recebido) indica que o sinal foi recebido.
- **Nó final:** um pequeno círculo preenchido dentro de outro círculo marcando a fase final ou o fim da atividade.





## 6 TELAS

### Homepage



**Cadastro**

## Cadastro

Digite os seus dados no campo abaixo.

<b>Nome</b>	<b>Data de Nascimento</b>
<input type="text" value="Informe o nome"/>	<input data-bbox="866 792 1209 884" type="text" value="dd/mm/aaaa"/>
<b>CPF</b>	<b>Telefone</b>
<input type="text" value="Informe seu CPF"/>	<input type="text" value="Digite seu telefone"/>
<b>E-mail</b>	<b>Senha</b>
<input type="text" value="Informe seu e-mail"/>	<input type="text" value="Informe uma senha"/>
<b>Cidade</b> <input type="text" value="Selecione"/>	

Salvar

## Login



*Toda Bella*

# Login

Digite os seus dados de acesso no campo abaixo.

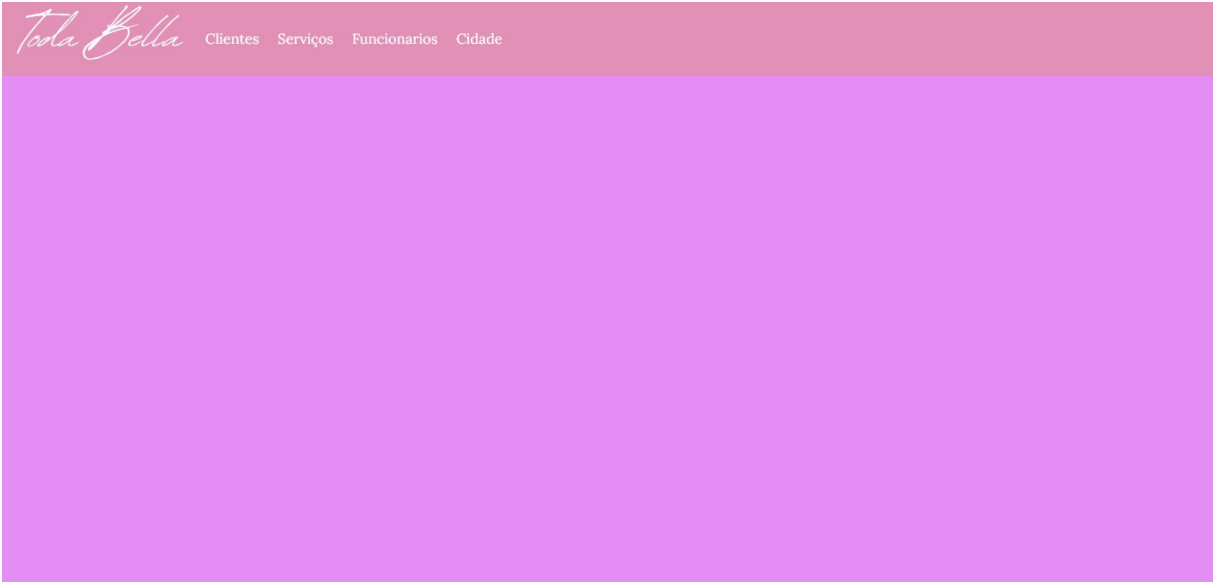
**E-mail**

**Senha**

[esqueceu a senha](#)

**Acessar**

## Area do administrador



## **7 CONCLUSÃO**

No decorrer deste trabalho, apresentamos o desenvolvimento e implementação de um software de agendamento estético, um sistema destinado a otimizar e simplificar a gestão de horários em salões de beleza. Nosso principal objetivo foi criar uma ferramenta eficaz e acessível que atendesse às necessidades tanto dos clientes quanto dos estabelecimentos de beleza, tornando o agendamento de procedimentos mais conveniente tanto para os profissionais de beleza quanto para os clientes.

O software de agendamento estético que desenvolvemos se mostrou uma solução robusta e flexível. Ele oferece recursos como agendamento online de procedimentos, gerenciamento de informações de clientes e uma interface amigável. Acreditamos que este sistema tem o potencial de melhorar significativamente a eficiência nos espaços de beleza, reduzir erros de agendamento e aprimorar a experiência do cliente.

Durante o processo de desenvolvimento, enfrentamos desafios técnicos, mas, com dedicação e trabalho em equipe, conseguimos superá-los com sucesso. A colaboração entre os membros da equipe foi fundamental para o progresso do projeto, e aprendemos muito com as dificuldades encontradas ao longo do caminho.

Gostaríamos de expressar nossos sinceros agradecimentos a todos que contribuíram para a realização deste trabalho. Em primeiro lugar, agradecemos aos nossos orientadores, pela orientação excepcional que nos proporcionaram durante todo o processo de pesquisa e desenvolvimento. Suas sugestões e insights foram inestimáveis para a conclusão deste projeto.

Agradecemos aos professores e colegas que nos apoiaram, fornecendo valiosos comentários e orientações em momentos críticos do desenvolvimento do software. Por fim, expressamos nossa gratidão às nossas famílias e amigos pelo apoio contínuo e compreensão ao longo desta jornada acadêmica. Muito obrigado a todos que fizeram parte desta nossa jornada!

## 8 REFERÊNCIAS

TECH, Equipe Pm. Diagramas de Contexto. 2021. Disponível em: <https://dicaspmp.pmtech.com.br/diagramas-de-contexto/>. Acesso em: 16 ago. 2023.

BARRO , A. C. et al. Detecção e classificação de distúrbios de tensão em sistemas de distribuição de energia elétrica usando uma rede neural ARTMAP euclidiana modificada com treinamento continuado. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE., 2013, Fortaleza. Anais... Fortaleza: [s.n.], 2013, p. 1-6.

COSTA, Psicodata Dalton (ed.). O que é um dicionário de dados. 2021. Disponível em: [https://moodle.unesp.br/pluginfile.php/24930/mod\\_resource/content/1/2-https://medium.com/psicodata/dicionario-de-dados-ac3ce726c34b.pdf](https://moodle.unesp.br/pluginfile.php/24930/mod_resource/content/1/2-https://medium.com/psicodata/dicionario-de-dados-ac3ce726c34b.pdf). Acesso em: 16 ago. 2023.

CUNHA, Fernando. Requisitos funcionais e não funcionais: o que são? 2022. Disponível em: <https://www.mestresdawe.com.br/tecnologias/requisitos-funcionais-e-nao-funcionais-o-que-sao>. Acesso em: 12 jul. 2022.

DIRANI, Lucas Ruiz. Conhecendo os diagramas de classe da UML. 2022. Disponível em: <https://medium.com/fora-de-assunto/conhecendo-os-diagramas-de-classe-da-uml-6f237f53cb45#:~:text=Os%20diagramas%20de%20classes%20nos,cont%C3%A9m%20uma%20refer%C3%Aancia%20para%20outra..> Acesso em: 11 out. 2023.

IBM. Diagramas de Seqüência. 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=uml-sequence-diagrams>. Acesso em: 18 out. 2023.

ANG, Joan. Como criar um diagrama de atividades. 2023. Disponível em: <https://pt.venngage.com/blog/diagrama-de-atividades/#:~:text=diagramas%20de%20atividades-,O%20que%20%C3%A9%20um%20diagrama%20de%20atividades%3F,s%C3%A3o%20realizadas%20em%20um%20sistema.&text=UML%20ou%20%E2%80%9CUnified%20Modeling%20Language,design%20de%20um%20sistema%20espec%C3%ADfico..> Acesso em: 18 out. 2023.