**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**

**ETEC ZONA LESTE**

**Técnico em Desenvolvimento de Sistemas**

**Fabio Evangelista Ferreira Pereira**

**Daniel Robson Alexandre Silva**

**João Lucas De Souza Conceição**

**Margareth Yoshimaro Mercer Sasaki**

**Natália Santos Lima**

**Priscila De Souza Dos Santos**

**CleanDay: Coleta Seletiva Inovadora**

**São Paulo**

**2023**

**Fabio Evangelista Ferreira Pereira**

**Daniel Robson Alexandre Silva**

**João Lucas De Souza Conceição**

**Margareth Yoshimaro Mercer Sasaki**

**Natália Santos Lima**

**Priscila De Souza Dos Santos**

**CleanDay: Coleta Seletiva Inovadora**

Trabalho de Conclusão de Curso apresentado ao Curso de Técnico em Desenvolvimento de Sistemas da Etec Zona Leste, orientado pelo Prof. Rogério Bezerra Costa, como requisito final para obtenção do título de Técnico em Desenvolvimento de Sistemas.

**São Paulo**

**2023**

**DEDICATÓRIA**

Dedicamos esse trabalho aos nossos colegas que nos inspiraram, nossa família e amigos que deram apoio emocional e motivacional e aos nossos professores que nos guiaram nas etapas do projeto.

**AGRADECIMENTOS**

Agradecemos aos colegas de equipe que trabalharam arduamente para conclusão desse projeto, ao Senhor Jesus e pelos colegas de classe que compartilharam ideias e maneiras de como levar o projeto a frente, aos nossos professores que sempre estiveram dispostos a esclarecer nossas dúvidas e a instituição Etec Zona Leste por nos proporcionar recursos para nossa formação.

**EPÍGRAFE**

“[O animal é tão ou mais sábio do que o homem: conhece a medida da sua necessidade, enquanto o homem a ignora.](http://www.oeco.org.br/blogs/frases-do-meio-ambiente/27265-frases-do-meio-ambiente-democrito-filosofo-120613/)”

Demócrito

**RESUMO**

**ABSTRACT**

**LISTA DE ILUSTRAÇÔES**

[Figura 1– Exemplo de formulário em HTML 14](#_Toc145544370)

[Figura 2 - Exemplo de tela de cadastro no navegador 15](#_Toc145544371)

[Figura 3 - Exemplo de propriedades em CSS parte I 16](#_Toc145544372)

[Figura 4 - Exemplo de propriedades em CSS parte II 16](#_Toc145544373)

[Figura 5- Exemplo de tela de cadastro no navegador estilizado em CSS 17](#_Toc145544374)

[Figura 6 - Exemplo de botão em JavaScript 17](#_Toc145544375)

[Figura 7 - Exemplo de formulário em PHP parte I 19](#_Toc145544376)

[Figura 8 - Exemplo de formulário em PHP parte II 20](#_Toc145544377)

[Figura 9 - Código em PHP 20](#_Toc145544378)

[Figura 10 - Diagrama de classe 21](#_Toc145544379)

[Figura 11- Diagrama de Caso de Uso 22](#_Toc145544380)

[Figura 12 - Diagrama de Atividade 23](#_Toc145544381)

[Figura 13 - Diagrama de sequência 24](#_Toc145544382)

[Figura 14 - Exemplo de diagrama de entidade-relacionamento 26](#_Toc145544383)

[Figura 15– Exemplo de modelo de entidade-relacionamento 27](#_Toc145544384)

[Figura 16 - Primeira forma normal. 28](#_Toc145544385)

[Figura 17 - Segunda Forma Normal 28](#_Toc145544386)

[Figura 18 - Terceira Forma Normal 28](#_Toc145544387)

[Figura 19 - Exemplo de criação de banco de dados 29](#_Toc145544388)

[Figura 20- Inserindo dados 30](#_Toc145544389)

[Figura 21 - Exemplo de seleção de dados de determinada tabela 30](#_Toc145544390)

[Figura 22 - Tabela com dados 30](#_Toc145544391)

**LISTA DE TABELAS**

**LISTA DE ABREVIATURAS E SIGLAS**

*Application Programming Interface* (API)

Associação Brasileira de Empresas de Limpeza Pública e Resíduos Especiais (Albrepe)

*Android Application Pack (*APK*)*

Banco de Dados (BD)

*Cascading Style Sheets* (CSS)

*Database Administrator* (DBA)

Diagrama de Entidade Relacionamento (DER)

Entidade Relacionamento (ER)

Forma Normal (FN)

*HyperText Markup Language* (HTML)

*Internet of Things* (IoT)

*International Solid Waste Association* (ISWA)

*Just-In-Time* (JIT)

Modelo de Entidade e Relacionamento (MER)

*Node Package Modules* (NPMS)

Organizações Não Governamentais (ONGs)

*Hypertext Preprocessor* (PHP)

**Sistema Gerenciador de Banco de Dados** (SGBD)

*Structured Query Language* (SQL)

*Unified Modeling Language* (UML)

***Universal Resource Locator*** (**URL**)

*W3Schools (*W3C)

**SUMÁRIO**

[1 - INTRODUÇÃO 12](#_Toc145538285)

[2 - REFERENCIAL TEÓRICO 13](#_Toc145538286)

[2.1 - HTML 13](#_Toc145538287)

[2.2 - CSS 15](#_Toc145538288)

[2.3 - JavaScript 17](#_Toc145538289)

[2.4 - PHP 18](#_Toc145538290)

[2.5 - UML 20](#_Toc145538291)

[2.5.1 - Diagrama de Classes 20](#_Toc145538292)

[2.5.2 - Diagrama de Caso de Uso 21](#_Toc145538293)

[2.5.3 - Diagrama de atividade 22](#_Toc145538294)

[2.6 - Banco de Dados 24](#_Toc145538295)

[2.6.1 - Propriedade ER, Entidade e Relacionamento 25](#_Toc145538296)

[2.6.2 - Atributo e MER 26](#_Toc145538297)

[2.6.3 - Normalização 27](#_Toc145538298)

[2.6.4 - MySQL 28](#_Toc145538299)

[7 - DESENVOLVIMENTO 31](#_Toc145538300)

[8 - CONCLUSÃO 32](#_Toc145538301)

[REFERÊNCIAS 33](#_Toc145538302)

# INTRODUÇÃO

No estado de São Paulo, a prefeitura da capital dispõe de um programa chamado "spRegula" que oferece diversos serviços relacionados ao descarte correto dos resíduos sólidos.

Com toda problemática que temos em relação ao não reaproveitamento dos materiais recicláveis podemos ter sérios danos tanto a curto como a logo prazo.  
Devemos conscientizar a população quanto a forma do descarte correto para amenizarmos o impacto que os resíduos causam no meio ambiente.

As consequências da má gestão dos resíduos sólidos no meio ambiente estão cada vez mais evidentes como a poluição ambiental, visual, danos à saúde, diminuição de recursos, entre outros. A cada ano que passa a produção de lixo alcança números astronômicos, de acordo com a pesquisa realizada pela Associação Brasileira de Empresas de Limpeza Pública e Resíduos Especiais (Albrepe), foi produzido em 2022 nas áreas urbanas 81,8 milhões de toneladas de lixo, o equivalente a 224 mil toneladas diárias, no entanto conforme a pesquisa realizada *pela International Solid Waste Association* (*ISWA*) somente 4% dos resíduos sólidos são processados e reciclados no Brasil, índice abaixo da média quando comparado com outros países com grau de desenvolvimento similares, dentre eles Chile, Argentina, África do Sul e Turquia, que apresentam média de 16% de reciclagem, (GANDRA, 2022).

# REFERENCIAL TEÓRICO

Neste capítulo, é apresentado o referencial teórico que sustenta a fundamentação deste trabalho. Dessa forma, exploram-se as principais linguagens de programação e abordagens relevantes para a compreensão e análise do tema, fornecendo uma base sólida para as etapas subsequentes da pesquisa.

A coleta é feita em pontos públicos decididos através de pesquisa de campo, em aspectos como segurança, acessibilidade e condições externas, onde é escolhido pela pessoa que separa/doa o material reciclável e aquela que coleta pela lista de pontos que o site disponibilizará ou através de encontros presenciais com as organizações não governamentais (ONGs) e Instituições sem fins lucrativos onde o objetivo primário é a coleta desse material.

A ponte entre as partes é o site que será programado, primariamente com recursos como HTML, estilizado por CSS e usando linguagens de programação como JavaScript, PHP e bootstrap e os dados armazenados num banco de dados, que serão abordados e explicados nesse capítulo.

## HTML

Segundo Silva (2008), *HyperText Markup Language* (HTML) é uma linguagem de marcação de hipertexto padrão, que possibilita a criação de páginas web, através de tags ou etiquetas que descrevem a estrutura de uma página e como o *browser* deve interpretá-la para exibição do conteúdo e as interligam entre si.

De acordo com Caldeira (2015), a forma mais básica para fazer programas em HTML, é a partir de editores de texto padrão como o Notepad, contudo pessoas versadas utilizam outras aplicações que podem facilitar na geração de códigos para desenvolverem as páginas web, uma dessas aplicações seria o Visual Studio Code.

As páginas Web são compostas por elementos HTML, ou tags HTML, que informam para o navegador a composição de como deve ser construída a página, por meio de imagens, parágrafos, títulos, tabelas, entre outros. São iniciados por um sinal “menor que” e fechados por um sinal “maior que”, as tags podem ser estabelecidas pelas que precisam de fechamento, definidas pela barra (/) no início e aquelas que fecham sozinhas e para que o navegador interprete o documento precisa da extensão .html. (SILVA, 2018)

**Tags, elementos e atributos HTML**

De início podemos caracterizar as tags por *block-level*, nível de bloco, e *inline*, em linha, o elemento de nível de bloco ocupa toda largura do elemento pai, já o elemento *inline* é geralmente usado para demarcação do conteúdo. (SILVA, 2018)

Os atributos são usados para caracterizar os elementos, localizados dentro da tag de abertura e seguidos de sinal de iguais e aspas, para atribuir um valor, por exemplo: <*title size*=””>. (CALDEIRA, 2015)

Lista de exemplos de elementos HTML básicos e suas definições, conforme figura 1, (W3C, 2023):

* DOCTYPE – Estabelece o tipo de documento;
* HTML – Define um documento HTML;
* Head – Engloba a metadata do documento;
* Title – Estabelece o título do documento;
* Body – Define o corpo do documento;
* P – Estabelece o parágrafo;
* Br – Insere uma quebra de linha;
* Form – Estabelece um formulário;
* Input – Especifica um campo de entrada de dados;
* Textarea – Determina um campo para inserir um texto de várias linhas;
* Button – Estabelece um botão;

Figura 1– Exemplo de formulário em HTML

Texto

Descrição gerada automaticamente

Fonte: Do próprio autor, 2023.

De acordo com a figura 2 abaixo, depois de salvo o arquivo na extensão html, aparecerá na pasta o ícone do navegador padrão, que ao abrir aparecerá a página segundo o código interpretado pelo navegador.

Figura 2 - Exemplo de tela de cadastro no navegador



Fonte: Do próprio autor, 2023.

## CSS

CSS, abreviação para *Cascading Style Sheets*, é uma linguagem para estilização de páginas Web, que descreve como o conteúdo deve ser mostrado, através dela que estiliza os elementos e o layout da página. (MEYER; WEYL, 2006)

Lista de propriedades básicas do CSS e suas definições e exemplo de uso, conforme figuras 3 e 4(W3C, 2023):

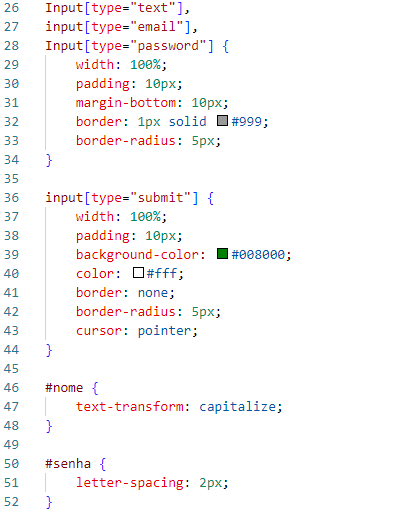
* Align-content – Determina o alinhamento entre linhas dentro de um container flexível;
* Border – Propriedade de abreviação paraborder-width, border-style e border-color;
* Color – Altera a cor;
* Font-family – Define a da fonte para o texto;
* Font-size - Muda o tamanho da fonte;
* Font-weight – Determina o peso da fonte;
* Font-Style – Define o estilo da fonte;
* Padding – Determina o espaço entre a borda e o conteúdo;

Figura 3 - Exemplo de propriedades em CSS parte I



Fonte: Do próprio autor, 2023.

Figura 4 - Exemplo de propriedades em CSS parte II



Fonte: Do próprio autor, 2023.

Após a definição das propriedades CSS para estilizar o documento html, foi adicionado a tag link para realizar a interligação entre os documentos, pois as propriedades estão definidas no arquivo externo denominado stylesheet, depois de salvo o arquivo, o navegador interpretará o código de acordo com a figura 5.

Figura 5- Exemplo de tela de cadastro no navegador estilizado em CSS

Interface gráfica do usuário, Site

Descrição gerada automaticamente

Fonte: Do próprio autor, 2023.

## JavaScript

Conforme Silva (2010), JavaScript é uma linguagem desenvolvida para rodar no lado do cliente, isto é, a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no navegador do usuário. Isso é possível porque existe um interpretador JavaScript hospedado no navegador.

Na figura 6 exemplifica como é feito e visualizado um JavaScript.

Figura 6 - Exemplo de botão em JavaScript

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: W3C, 2023.

Em concordância com Groner (2019), a linguagem JavaScript não é usada apenas no *front-end*. Ela também pode ser utilizada no *back-end*, sendo o Node.js a tecnologia responsável por isso. O número de *Node Package Modules* (NPMS), ou Módulos do Pacote Node, disponíveis em <http://www.npmjs.org>, também tem crescido exponencialmente. Além disso, o JavaScript pode ser usado em desenvolvimento para dispositivos móveis, e é um dos frameworks mais populares no Apache Cordova (<https://cordova.apache.org>), um framework híbrido para dispositivos móveis, o qual permite que os desenvolvedores programem usando *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS) e JavaScript, possibilitando construir um aplicativo e gerar um arquivo *Android Application Pack* (APK) para Android e IPA para iOS (Apple). E, claro, não vamos nos esquecer das aplicações *desktop*. Podemos escrever aplicações desktop compatíveis com Linux, Mac, OS e Windows usando um framework JavaScript chamado *Electron* (htpps://eléctron.atom.io). O JavaScript também é usado em dispositivos embarcados e *dispositivos Internet of Things* (IOT) ou Internet das Coisas.

De acordo com Morrison (2008) exemplifica, ao clicar em um hiperlink ou digitar uma *Universal Resource Locator* (URL), em seu navegador web, o navegador solicita a página em um servidor web, que então fornece a página de volta. Então, o código JavaScript funciona de comum acordo com o navegador web para responder às interações do usuário e modificar a página de acordo com a necessidade. A parte do navegador web que executa o código JavaScript é chamada de interpretador de JavaScript.

## PHP

Segundo Dall´oglio (2016), o PHP, sigla para *Hypertext Preprocessor* – Pré processador de Hipertexto, consiste em uma linguagem de programação servindo para a utilização de desenvolvimento web. A linguagem nasceu em 1994, criada pelo programador Rasmus Lerdorf que inicialmente tinha o objetivo de planejar a linguagem para projeção de páginas Web dinâmicas.

De acordo com o desenvolvedor e autor Nielsen (2018), O PHP é uma linguagem de programação poderosa e popular, especialmente adequada para o desenvolvimento web. Sua simplicidade e flexibilidade e tornam a escolha preferida de muitos desenvolvedores.

Ao longo dos anos, o PHP tem evoluído constantemente, introduzindo novos recursos e melhorias de desempenho. A versão mais recente, o PHP \*, lançada e, novembro de 2020, trouxe importantes atualizações, incluindo a introdução do *Just-in-Time*, JIT, *Compiler*, que melhora significativamente a velocidade de execução dos scripts PHP. (POWERS, 2021).

Em conformidade com o destacado pelo desenvolvedor e escritor Powers (2021), o PHP é uma linguagem de programação versátil, que permite a criação de sites e aplicações web altamente funcionais. Sua ampla adoção e comunidade ativa garantem que sempre haverá suporte e recursos disponíveis para ajudar os desenvolvedores a superarem desafios.

Em resumo, PHP é uma linguagem bem versátil, voltada para desenvolvimento web. Como uma linguagem bastante intuitiva e ampla, o PHP se tornou uma escolha popular entre os desenvolvedores de todos os níveis para criação de sites, aplicativos e sistemas. Seja um desenvolvedor iniciante, ou já experiente, o PHP oferece um ambiente enorme para ideias de criação e projeção web. A linguagem permanece como uma das principais e continua a evoluir para atender às demandas em constante mudança da tecnologia e o mundo digital. Portanto, em conclusão, PHP é uma excelente opção a ser considerada para construção de sites dinâmicos, interativos e repletos de recursos. Há várias funcionalidades para um desenvolvedor explorar e aproveitar os benefícios que o PHP tem a oferecer para aprimorar cada vez mais suas habilidades de desenvolvimento web. (CONVERSE; PARK, 2002)

Na figura 7 exemplifica um formulário simples, conectado a um documento externo welcome.php, onde é enviado as informações que o usuário inseriu e devolvidas conforme figura 8.

**Figura 7 - Exemplo de formulário em PHP parte I**

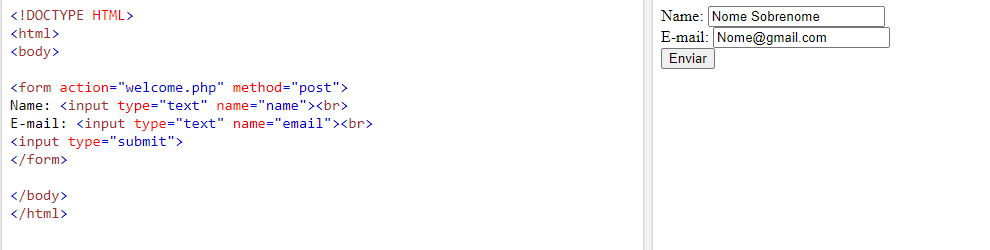
Fonte: W3C, 2023

Figura 8 - Exemplo de formulário em PHP parte II

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente com confiança média

Fonte: W3C, 2023

Na figura 9 mostra o documento externo welcome.php, que consta o código PHP, onde é realizada a ação de recebimento e retorno das informações.

Figura 9 - Código em PHP

Texto, Carta

Descrição gerada automaticamente

Fonte: W3C, 2023

## UML

Em concordância com Booch et al. (2005), *a Unified Modeling Languag*e (UML), ou linguagem de modelagem unificada, é uma linguagem visual que especifica, constrói e documenta artefatos do sistema, a UML permite uma prototipação padrão do projeto, e inclui conceitos como funções de um sistema à itens concretos como classes do projeto. A UML foi tomando forma na metade da década de 1990, quando em meio ao surgimento de métodos de modelagens e linguagens cada vez mais complexas, Grady Booch, Ivar Jacobson e James Rumbaugh começaram a juntar e adotar as ideias provenientes dos métodos orientados a objetos reconhecidos mundialmente na época.

### Diagrama de Classes

O diagrama de classes é um dos mais fundamentais da UML, pois nele está determinado as classes que constituem o sistema com os respectivos atributos e métodos e demonstra os relacionamentos entre as classes e transmissão de informações, servindo como base para a lógica do sistema e outros diagramas. As classes são compostas de atributos que armazenam as informações e os métodos são as funções que a classe pode executar. (GUEDES, 2012)

Segundo Fowler (2018), o diagrama de classe é composto pelos elementos de classes, relacionamentos, atributos e métodos.

Classes: Formadas por retângulos divididos entre 3 partes, o primeiro compartimento é o nome da classe, o segundo contém os atributos das classes, variáveis e o terceiro compartimento exibe os métodos, funções das classes.

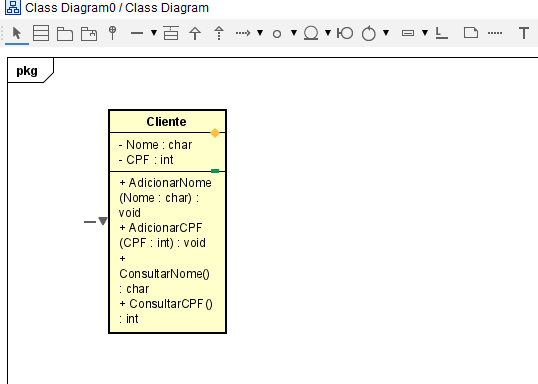
Relacionamentos: Os relacionamentos são conectados e interpretados por linhas. Os tipos de relacionamentos que existem são associação, agregação, composição, dependência.

Atributos e métodos: Todas as classes possuem atributos, que são suas características ou propriedades e os métodos são as ações que a classe executa.

Um diagrama de classe pode incluir a visibilidade dos atributos e métodos, como público, privado e protegido, além de mostrar a multiplicidade dos relacionamentos e dados dos atributos.  (FOWLER, 2018)

Na figura 10 exibe um diagrama de classe.

Figura 10 - Diagrama de classe



Fonte: Do próprio autor, 2023.

### Diagrama de Caso de Uso

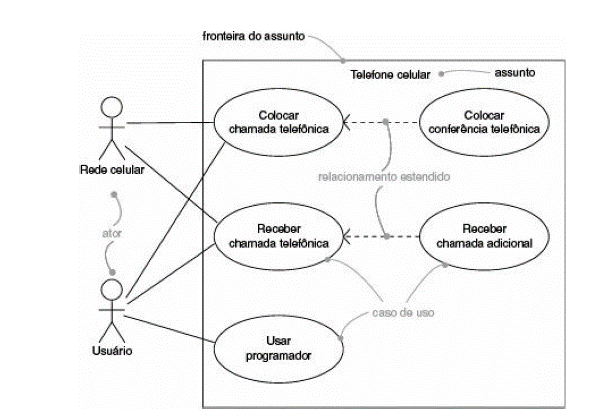
Os diagramas de caso de usos servem para verificar os requisitos funcionais do sistema, ou seja, ele simula o comportamento de um ator com o sistema. Os casos de usos são representados por elipse e dentro de si contém o texto descrevendo sua funcionalidade e interligados por linhas. (GUEDES, 2012)

Em congruência com Booch et al. (2005), o caso de uso é composto por atores, casos de uso e relacionamentos. O ator pode ser interpretado por um sistema externo, um usuário, ou pelo próprio sistema, são representados por figuras humanas. Já o caso de uso é a descrição da interação entre o sistema e o ator, representa uma função do sistema ou um recurso oferecido ao usuário, sua figura é uma elipse e o nome dentro da figura indica a ação realizada ou o objetivo pretendido.

O relacionamento entre os casos demonstra a dependência ou lógica das interações, os mais comuns são o de inclusão, include, quando o caso de uso é incorporado a outro, extensão, extend, quando o caso é estendido a outro em situações opcionais e generalização é quando um caso é uma variação de outro. (BOOCH et al. 2005)

Na figura 11 demonstra um exemplo de caso de uso.

Figura 11- Diagrama de Caso de Uso



Fonte: Booch et al., 2005.

### Diagrama de atividade

Segundo Booch et al. (2005), o diagrama de atividades serve para unir pessoas de gestão de negócios e desenvolvimento para que possam entender o mesmo processo, descrevem a atividade de caso de uso, explicam a lógica de um algoritmo e simplifica e aperfeiçoa os casos de usos mais complicados, o diagrama de atividade pode conter ações, nós de atividades, fluxos e valores de Objetos.

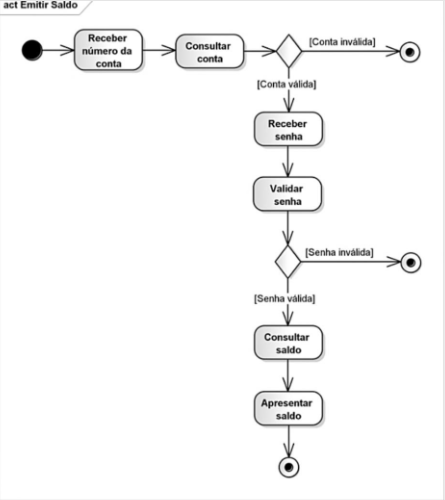
Os nós representam as atividades a serem executadas, podem ser simples, como uma ação única, ou podem conter subdivisões internas, como um diagrama de atividade aninhado. (BOOCH et al. 2005)

Sequência de execução: As setas são usadas para indicar a sequência de execução das atividades. Elas mostram a ordem em que as atividades devem ser realizadas.

Ramificações e fusões: Se o fluxo de atividade se dividir em caminhos diferentes, com base em condições ou decisões, quando acontece esses casos, são usados nós de decisão, losango, para representar as escolhas e nós de fusão, losango preenchido com uma barra horizontal, para indicar a convergência dos caminhos. (BOOCH et al. 2005)

A figura 12 mostra o exemplo de diagrama de atividade.

Figura 12 - Diagrama de Atividade



Fonte: Guedes, 2012.

**Diagrama de Sequência**

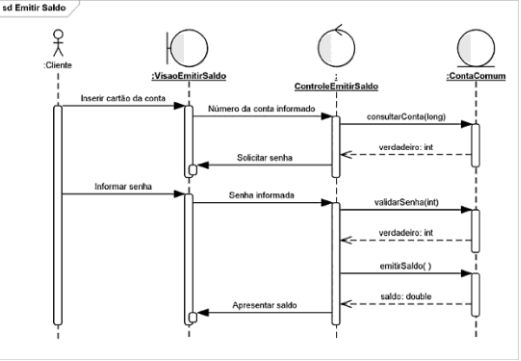
O diagrama de sequência serve como um diagrama de interação que descreve como e em qual ordem um evento deve prosseguir, é necessário para que os desenvolvedores de software possam levantar os requisitos do sistema. O diagrama de sequência pode servir para determinados cenários, como demonstra a figura 13. (GUEDES, 2012)

Objetos: Os objetos são representados por retângulos e são localizados na parte superior do diagrama. Eles interagem entre si durante a execução do cenário.

Linhas de vida, lifelines: As linhas de vida são linhas verticais que se estendem a partir dos objetos e mostram a duração da participação de um objeto no cenário. (FOWLER, 2018)

Mensagens: As mensagens são representadas por setas e indicam as interações entre os objetos. Elas podem ser mensagens síncronas, representadas por uma seta sólida, onde o remetente aguarda a conclusão da mensagem antes de continuar, ou mensagens assíncronas, representadas por uma seta tracejada, onde o remetente não aguarda uma resposta imediata. (FOWLER, 2018)

Figura 13 - Diagrama de sequência



Fonte: Guedes, 2012.

## Banco de Dados

Como afirma Date (2004), um sistema de banco de dados é basicamente um sistema computadorizado de manutenção de registros; em outras palavras, é um sistema computadorizado que a função principal é armazenar informações e permitir que os usuários busquem e atualizem informações quando as solicitarem. As informações podem ser qualquer coisa necessária para auxiliar no processo geral das atividades desse individuo ou dessa organização.

Enquanto Nunes e Moura (2018), dizem que a programação em banco de dados depende do conhecimento da Structured Query Language, ou Linguagem de Consulta Estruturada (SQL). Por isso, você precisará, inicialmente, conhecer a ferramenta MySQL e, em especial, sua versão gráfica: o MySQL Workbench. Essa poderosa ferramenta gráfica foi desenvolvida para o ambiente Windows e realiza de maneira gráfica todas as tarefas possíveis de serem realizadas na versão de linha de comando, no inglês, Command Line Client. Com as instruções SQL, você poderá investigar como é a definição da estrutura de qualquer banco de dados.

### Propriedade ER, Entidade e Relacionamento

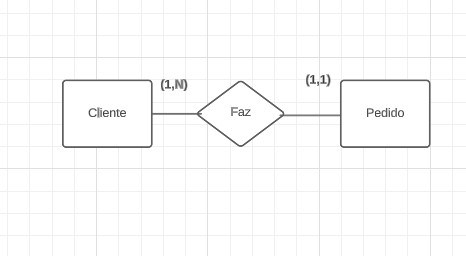
Segundo Heuser (2008), um Diagrama de entidade de Relacionamento (DER) é um modelo formal, preciso e não ambíguo usado na modelagem de bancos de dados. É essencial que todos os envolvidos na criação e uso dos diagramas de entidade de relacionamento (ER) sejam treinados para compreendê-los corretamente. Em algumas organizações, os modelos ER são subutilizados, servindo apenas para apresentação informal de ideias. Isso pode ser evitado com treinamento formal. É importante destacar que o fato de um DER ser gráfico e intuitivo pode levar a uma falsa compreensão por parte de pessoas não treinadas. Problemas podem surgir quando os usuários não entendem completamente o DER e isso afeta a implementação do banco de dados. Portanto, é necessário treinar os usuários na leitura e compreensão de diagramas ER para evitar problemas futuros.

De acordo com Heuser (2008), o conceito fundamental da abordagem ER (entidade-relacionamento) é o conceito de entidade. Uma entidade representa, no modelo conceitual, um conjunto de objetos da realidade modelada. Como o objetivo de um modelo ER é modelar de forma abstrata um banco de dados (BD), interessa somente os objetos sobre os quais deseja-se manter informações. Uma entidade pode representar tantos objetos concretos da realidade por exemplo, uma pessoa, um automóvel, quanto objetos abstratos um departamento, um endereço.

Conforme afirma Heuser (2008), além de especificar os objetos sobre os quais deseja-se manter informações, o DER deve permitir a especificação das propriedades dos objetos que serão armazenadas no BD. Uma das propriedades sobre as quais pode ser desejável manter informações é a associação entre objetos. Exemplificando, pode ser desejável saber quais pessoas estão associadas a quais departamentos em uma organização. Em um DER, um relacionamento é representado através de um losango, ligado por linhas aos retângulos representativos das entidades que participam do relacionamento.

Na figura 14 exemplifica um diagrama de entidade de relacionamento simples onde o cliente faz o pedido, representado a entidade e relacionamento entre eles.

**Figura 14 - Exemplo de diagrama de entidade-relacionamento**

****

Fonte: Do próprio autor, 2023

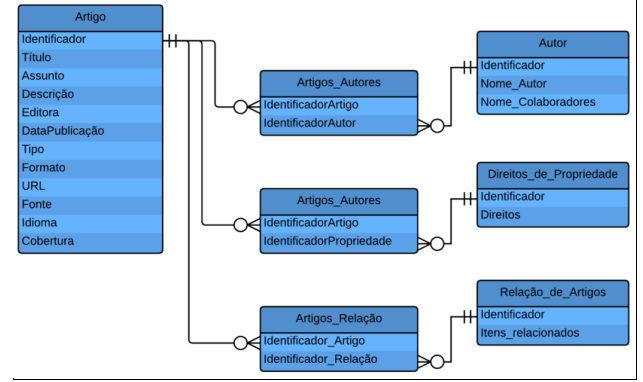
### Atributo e MER

Em concordância com Heuser (2008), para associar informações a ocorrências de entidades ou de relacionamentos usa o conceito de atributo. Na prática, atributos não são representados graficamente, para não sobrecarregar os diagramas, já que muitas vezes entidades possuem muitos atributos. Prefere-se usar uma representação textual que aparece separadamente do diagrama ER. Ao final deste capítulo, é fornecida uma possível sintaxe para uma representação textual dos atributos. No caso de ser usado um software para construção de modelos ER, o próprio software encarrega-se do armazenamento da lista de atributos de cada entidade em um dicionário de dados.

Segundo Takai et al. (2005), o MER é um modelo de dados conceitual de alto-nível, ou seja, seus conceitos foram projetados para serem compreensíveis a usuários, descartando detalhes de como os dados são armazenados. Atualmente, o MER (Modelo de entidade e relacionamento) é usado principalmente durante o processo de projeto da base de dados. Existem expectativas para que uma classe de Sistema Gerenciador de Banco de Dados (SGBD) baseados diretamente no MER esteja disponível no futuro.

A figura 15 mostra um modelo de entidade e relacionamento.

Figura 15– Exemplo de modelo de entidade-relacionamento

****

Fonte: Dantas *et al.* 2016

### Normalização

França e Celestino Júnior (2015) afirmam que a normalização procura simplificar a maneira como os dados serão armazenados no banco de dados para conseguir mais eficiência. Neste contexto a palavra “eficiência” não se refere melhorar o desempenho do banco de dados ou facilitar o processo de consulta. A eficiência procurada aqui se refere a diminuição da complexidade da estrutura lógica do banco de dados. A normalização é o processo de análise efetuado sobre esquemas relacionais para conseguir características desejáveis, tais como a minimização de redundância e, consequentemente, a redução de anomalias de inserção, atualização e exclusão.

* **Forma normal**

Em conformidade com França e Celestino Júnior (2015), uma forma normal é uma regra que deve ser seguida para que uma tabela seja bem avaliada. A forma normal sujeita o esquema de relação a uma cadeia de avaliação para garantir que ele satisfaz a forma normal. Esse processo de avaliação segue o estilo top-down, onde cada relação é avaliada sob os critérios das formas normais.

* **Primeira Forma normal (1Fn)**

Condizente com França e Celestino Júnior (2015), uma tabela está na 1FN se não possuir atributo multivalorado ou atributo composto, esse procedimento elimina tabelas aninhadas, conforme figura 16.

Figura 16 - Primeira forma normal.

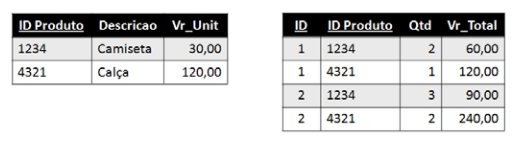
****

Fonte: Segundo França e Celestino Júnior, 2015

* **Segunda Forma normal (2Fn)**

Segundo França e Celestino Júnior (2015), uma tabela está na 2FN se estiver na 1FN e não possuir dependência funcional parcial. Uma dependência parcial ocorre quando os atributos não chave não dependem de toda chave primária composta, conforme figura 17.

Figura 17 - Segunda Forma Normal



Fonte: Segundo França e Celestino Júnior, 2015

* **Terceira Forma normal (3Fn)**

De acordo com França e Celestino Júnior (2015), uma tabela está na 3FN se estiver na 2FN e não possuir nenhuma dependência funcional transitiva. Uma dependência transitiva ocorre quando um atributo não chave depende de outro atributo não chave, conforme figura 18.

Figura 18 - Terceira Forma Normal



Fonte: Segundo França e Celestino Júnior, 2015

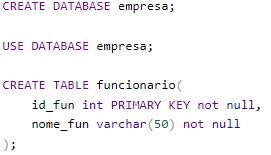
### MySQL

Takai *et al.* (2005) explica que a *Structure Query Language* (SQL) permitiu padronizar a construção e acesso a Sistema de Gerenciamento de Bancos de Dados Relacional (SGBDR) de diferentes tipos e em diferentes plataformas de software e hardware. Essa padronização impulsionou não apenas a disseminação dos SGBDR, mas também a própria SQL. Para entender a importância da SQL são mostrados nesta subseção todos os pontos que levaram os *Database Administrator (DBAs)* – Administrador de Banco de Dados a ter na SQL um aliado importante.

* CREATE DATABASE nomedobanco: Esse comando cria o banco de dados.
* DROP DATABASE: Com esse comando conseguimos deletar nosso banco de dados, excluindo todas as informações nele.
* USE DATABASE nomedobanco: Entramos dentro do banco.
* CREATE TABLE nometabela: Aqui iniciamos a tabela.
* INSERT INTO nometabela(codigo,nome) VALUES (1, ’funcionario1’): Esse comando insere dados nas colunas criadas em nossa tabela.
* SELECT \* FROM nometabela: Selecionamos uma tabela e conseguimos ver os dados inseridos nela.

Na figura 19 é possível identificar a execução do comando CREATE DATABSE que cria o banco de dados, também mostra a utilização do banco de dados com o 2comando USE DATABASE e a criação de uma tabela com o comando CREATE TABLE já com seus atributos.

Figura 19 - Exemplo de criação de banco de dados



Fonte: Do próprio autor, 2023

Na figura 20 mostra o comando que deleta o banco de dados, ao aplicar esse comando não é possível mais o acesso ao banco de dados, inibindo qualquer contato posterior ao comando.

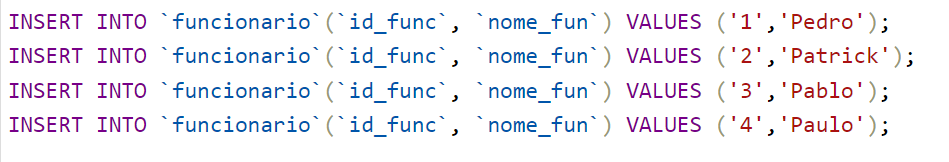
Figura 20- Exemplo de exclusão de banco de dados



Fonte: Do próprio autor, 2023

Na figura 21, mostra a inserção de dados no banco criado anteriormente, esse comando é muito importante para a criação do banco.

Figura 20- Inserindo dados



Fonte: Do próprio autor, 2023

O SELECT \* FROM serve para selecionar uma tabela com os dados já inseridos conforme figura 22.

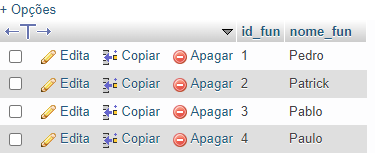
Figura 21 - Exemplo de seleção de dados de determinada tabela



Fonte: Do próprio autor, 2023

Na figura 23 demonstra a exibição da tabela criada anteriormente, já com os dados inseridos e após a execução do comando SELECT \* FROM.

Figura 22 - Tabela com dados



Fonte: Do próprio autor, 2023

# DESENVOLVIMENTO

Este capítulo contém o embasamento do desenvolvimento do trabalho e as tecnologias utilizadas na criação, o intuito do trabalho é criar um site voltado para coleta seletiva, denominado CleanDay, no qual o usuário pode pesquisar pontos de coletas mais próximos da sua localidade e informações adicionais sobre os materiais recicláveis e o meio ambiente.

As tecnologias fundamentais que serão utilizadas nesse projeto são Laravel, Hypertext Preprocessor (PHP), HyperText Markup Language (HTML), Cascading Style Sheets (CSS), bootstrap, javascript, Application Programming Interface (API) e o Structure Query Language (SQL) e MySQL.

O HTML foi utilizado para criação da estrutura do site, o CSS e o bootstrap foram aplicados na estilização do site, sendo o CSS uma linguagem de estilo para controlar a apresentação da página, permitindo a simplificação da manutenção e criação de recursos reutilizáveis. O bootstrap é um framework de código aberto que fornece estilos e componentes predefinidos, junto com um sistema de grade flexível, permitindo um desenvolvimento rápido e responsivo de páginas web.

O Laravel é um framework de desenvolvimento web utilizando como base o PHP, nele embutido ferramentas e bibliotecas que simplificam a criação de aplicativos web, pois oferece recursos roteamento, autenticação, controle de acesso, entre outros, o site foi criado utilizando esse framework.

A linguagem de programação JavaScript é executada no navegador do usuário para a manipulação de eventos como cliques de mouses e entrada de teclado para a criação de um site interativo e dinâmico com intuito de responder a ações do usuário em tempo real.

O SQL é a linguagem de gerenciamento e manipulação de dados relacionais e o MySQL é sistema de gerenciamento de banco de dados (SGDB) que foram usados para criação do banco de dados do site e manipulação dos mesmos.

A API, reúne regras e protocolos que permite sistemas de softwares diferentes interagirem entre si, definindo os métodos para permitir como os aplicativos e serviços devem interagir para trocarem informações e funcionalidades, no projeto Clean Day usaremos uma API para realizar as buscas de locais.

## Diagrama de entidade

## Modelo entidade relacionamento

## Caso de uso

# CONCLUSÃO

# REFERÊNCIAS

BOOCH, Grady et al. UML: guia do usuário. 6. ed. Rio de Janeiro: Elsevier, 2005. 454 p.

CALDEIRA, Carlos Pampulim. Introdução ao HTML. 2015. 45 f. TCC (Graduação) - Curso de Informática, Universidade de Evora, Évora, 2015.

CONVERSE, Tim; PARK, Joyce. **PHP**: a bíblia. 2. ed. São Paulo: Pearson Education, 2002.

DA REDAÇÃO. Exame. Lixo de brasileiros em 2022 equivale a 85 milhões de carros populares. 2023. Disponível em: <https://exame.com/ciencia/lixo-de-brasileiros-em-2022-equivale-a-85-milhoes-de-carros-populares/>. Acesso em: 29 maio 2023.

DALL”OGLIO, Pablo. **PHP**: programando com orientação a objetos. 3. ed. São Paulo: Novatec, 2016.

DANTAS, Célia Medeiros *et al*. ANÁLISE DA REPRESENTAÇÃO DA INFORMAÇÃO EM MODELOS ENTIDADE RELACIONAMENTO COM BASE EM METADADOS. João Pessoa: Archeion Online, 2016.

FOWLER, Martin. **UML Distilled**: a brief guide to the standard object modeling language. 3. ed. [S.I]: Martin Fowler, 2018. 208 p.

FRANÇA, Cicero Tadeu Pereira Lima; CELESTINO JÚNIOR, Joaquim. **Computação**: banco de dados. 2. ed. Fortaleza: Eduece, 2015. 120 p.

GANDRA, Alana. Índice de reciclagem no Brasil é de apenas 4%, diz Abrelpe. 2022. Disponível em: <https://agenciabrasil.ebc.com.br/geral/noticia/2022-06/indice-de-reciclagem-no-brasil-e-de-4-diz-abrelpe>. Acesso em: 30 maio 2023.

GRONER, Loiane. Estrutura de dados e algoritmos com JavaScript. São Paulo: Novatec, 2019. 2e.

GUEDES, Gilleanes Thorwald Araujo. UML 2: guia do usuário. 3. ed. São Paulo: Novatec, 2012.

HEUSER, Carlos Alberto. **Projeto de Banco de dados**. 4. ed. [S.I]: Bookman, 2008. 206 p.

LUCIDCHART. **O que é diagrama de atividades UML?** 2023. Disponível em: https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-atividades-uml. Acesso em: 15 jun. 2023.

MEYER, Eric; WEYL, Estelle. CSS: The Definitive Guide. 3. ed. [S. L.]: O'Reilly Media, 2006. 538 p.

MORRISON, Michael. Use a Cabeça! JavaScript. Rio de Janeiro: ALTA BOOKS, 2008. p 86.

NUNES, Sergio Eduardo; MOURA, Ricardo Alexandre Plati. Programação em Banco de Dados. Londrina: Editora e Distribuidora Educacional S.A., 2018.

Prefeitura de São Paulo. **O que tem sido feito em São Paulo para estimular a coleta seletiva?** 2022. Disponível em: https://www.prefeitura.sp.gov.br/cidade/secretarias/spregula/residuos\_solidos/noticias/?p=337523. Acesso em: 15 jun. 2023.

SANTOS, Marcos Vinicius Soares et al. Segurança em Banco de Dados: uma visão geral sobre segurança e suas principais deficiências. Montes Claros: Universidade Estadual de Montes Claros, 2014.

SILVA, Maurício Samy. Criando Sites com HTML: sites de alta qualidade com html e css. São Paulo: Novatec, 2008. 425 p.

SILVA, Maurício Samy. JavaScript - Guia do Programador: Guia completo das funcionalidades da linguagem JavaScript. São Paulo: Novatec, 2010. p 23.

TAKAI, Osvaldo Kotaro *et al*. **INTRODUÇÃO A BANCO DE DADOS**. São Paulo: Dcc-Ime-Usp, 2005. 124 p.

W3C. CSS Introduction. 2023. Disponível em: <https://www.w3schools.com/css/css_intro.asp>. Acesso em: 14 jun. 2023.

W3C. HTML Element Reference. 2023. Disponível em: <https://www.w3schools.com/tags/default.asp>. Acesso em: 19 maio 2023.

W3C. My First JavaScript. 2023. Disponível em: <https://www.w3schools.com/js/tryit.asp>?