**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**

**ETEC DA ZONA LESTE**

**Técnico em Desenvolvimento de Sistemas**

**Jefferson Laurindo da Silva**

**João Vitor Santos Portela**  
**Lucas Quinteiro Campos**

**Thiago Cruz da Silva**

**HASTYDEV: Plataforma Gamificada de Projetos para Desenvolvedores**

**São Paulo**

**2024**

**Jefferson Laurindo da Silva**

**João Vitor Santos Portela  
Lucas Quinteiro Campos**

**Thiago Cruz da Silva**

**HASTYDEV: Plataforma Gamificada de Projetos para Desenvolvedores**

Trabalho de Conclusão de Curso apresentado ao curso técnico em 2023 da ETEC Zona Leste orientado pelo Prof. Ediney Ciasi Barreto, como requisito parcial para obtenção do título de técnico em Desenvolvimento de sistemas.

**São Paulo**

**2024**

DEDICATÓRIA

Dedicamos este trabalho a todo o curso de Desenvolvimento de Sistemas da ETEC Zona Leste, corpo docente e discente, a quem fico lisonjeado por dele ter feito parte. Este trabalho é, também, todo dedicado aos nossos pais, pois é graças ao esforço deles que hoje podemos concluir esse curso. E, por último, agradecemos a todas as pessoas especiais da nossa vida que de alguma forma contribuíram com as nossas vidas.

AGRADECIMENTOS

Em primeiro lugar, a Deus, o que fez com que nossos objetivos fossem realizados, durante todo esse um ano e meio de estudos.

Aos nossos pais e irmãos, que cultivaram nos momentos difíceis e compreenderam a nossa ausência enquanto nos dedicamos à realização deste trabalho.

Aos amigos e Pessoas Especiais, que sempre estiveram ao nosso lado, pela amizade incondicional e pelo apoio demonstrado ao longo de todo o período em que me dediquei a este trabalho.

Aos professores, pelas correções e ensinamentos que me permitiram apresentar um melhor desempenho no meu processo de formação profissional ao longo do curso.

Aos meus colegas de curso, com quem conviveram intensamente durante esse tempo, pelo companheirismo e pela troca de experiências que me permitiram crescer não só como pessoa, mas também como formando.

EPÍGRAFE

“Hoje a programação é uma corrida entre os engenheiros de software para tentar construir maiores e melhores programas à prova de idiotas, e o Universo tentando produzir maiores e melhores idiotas. Até agora, o Universo está ganhando.”

RICH COOK

RESUMO

A *"HastyDEV"* tem como propósito facilitar a colaboração entre desenvolvedores, Microempreendedor Individual (MEIs) e Organizações não governamentais (ONGs) no mercado de desenvolvimento de software. Para desenvolvedores, a plataforma oferece a oportunidade de construir um portfólio sólido e adquirir experiência ao criar aplicações para empresas. MEIs podem encontrar desenvolvedores talentosos a preços acessíveis, enquanto ONGs podem receber soluções tecnológicas gratuitas, tudo isso por meio de um sistema gamificado. A metodologia da aplicação envolve empresas postando tarefas com especificações detalhadas, e os desenvolvedores competem para desenvolver a melhor solução. A *"HastyDEV"* representa uma abordagem inovadora para unir desenvolvedores, MEIs e ONGs, criando oportunidades de trabalho e desenvolvimento de portfólio, além de destacar projetos para ONGs por meio da gamificação.

**Palavras-chave**: Desenvolvedores; Portfólio; Gamificação.

ABSTRACT

The *"HastyDEV"* application aims to facilitate collaboration among developers, microenterprises (MEIs), and non-governmental organizations (OGs) in the software development market. For developers, the platform offers the opportunity to build a strong portfolio and gain experience by creating applications for businesses. MEIs can find talented developers at affordable prices, while NGOs can receive free technological solutions, all through a gamified system. The application's methodology involves companies posting tasks with detailed specifications, and developers compete to develop the best solution. *"HastyDEV"* represents an innovative approach to bring together developers, MEIs, and NGOs, creating job opportunities and portfolio development, while highlighting projects for NGOs through gamification.

**Keywords**: Developers; Portfolio; Gamification.

SUMÁRIO

[**1 INTRODUÇÃO 8**](#_Toc1630980044)

[**2 REFERENCIAL TEÓRICO 9**](#_Toc453887777)

[**2.1 Desafios para Profissionais Iniciantes 10**](#_Toc1480758113)

[**2.2 Visibilidade Digital para ONGs 10**](#_Toc1829531210)

[**2.3 Gamificação no Desenvolvimento Profissional 11**](#_Toc1313261293)

[**3 LINGUAGENS DE PROGRAMAÇÃO 11**](#_Toc1242130816)

[**3.1 HTML 11**](#_Toc9721105)

[**3.2 CSS 14**](#_Toc772401240)

[**3.3 Javascript 17**](#_Toc1758340568)

[**3.4 Typescript 20**](#_Toc1442862776)

[**3.5 REACTJS 21**](#_Toc1197330939)

[**3.6 NODEJS 25**](#_Toc1729008514)

# **1 INTRODUÇÃO**

O mercado de desenvolvimento de software está em constante crescimento, mas muitas empresas pequenas e ONGs não têm recursos para contratar desenvolvedores em tempo integral. Desenvolvedores novos e experientes também podem ter dificuldade em encontrar oportunidades de trabalho e construir um portfólio.

Este trabalho propõe o desenvolvimento de uma aplicação web para conectar desenvolvedores, MEIs e ONGs, criando oportunidades de trabalho e desenvolvimento de portfólio.

A hipótese é que uma aplicação que permita que empresas postem tarefas com especificações detalhadas e valores a serem pagos pode facilitar a conexão entre desenvolvedores e empresas. Além disso, um sistema gamificado pode incentivar desenvolvedores a participar da aplicação e contribuir para a plataforma.

Os objetivos do trabalho são desenvolver uma aplicação web e mobile que atenda aos objetivos propostos, analisar a viabilidade técnica e econômica da aplicação e avaliar o impacto da aplicação na comunidade de desenvolvedores, MEIs e ONGs.

A metodologia da aplicação envolve empresas postando tarefas com especificações detalhadas e os valores que estão dispostas a pagar. Os desenvolvedores interessados competirão para desenvolver a melhor solução. As empresas selecionarão a melhor aplicação e recompensarão o desenvolvedor vencedor.

A aplicação será desenvolvida utilizando as tecnologias ReactJS, Node.js, Express.js, MySQL e MongoDB.

# **2 REFERENCIAL TEÓRICO**

Este presente capítulo tem como objetivo a pesquisa para os embasamentos teóricos das razões sociais utilizadas para elaborar a plataforma *"HastyDEV"*.

## 2.1 Desafios para Profissionais Iniciantes

No panorama contemporâneo do desenvolvimento de software, onde os profissionais em início de carreira enfrentam desafios significativos para se inserirem no mercado de trabalho, emerge a plataforma inovadora denominada *"HastyDEV"*. Destinada a mitigar as barreiras tradicionais que obstruem a entrada desses programadores no cenário profissional, a *"HastyDEV” redefine* paradigmas convencionais ao amalgamar de maneira única as esferas de recrutamento, gamificação e responsabilidade social.

Conforme Samuel Ferreira (2022), A inovação é o caminho para o futuro e para os profissionais de tecnologia. Estes são os mais aptos a se adaptar às necessidades, físicas ou tecnológicas no ambiente de trabalho. O mercado da tecnologia entende que um dos maiores desafios dos desenvolvedores é de buscar soluções que facilitem os processos e tragam comodidade aos usuários. Este processo requer atenção aos detalhes e expertise nas habilidades e estratégias, cada vez mais exigidos neste ramo da profissão

A *"HastyDEV"* surge como uma resposta visionária para a problemática enfrentada pelos novos desenvolvedores, proporcionando um ambiente catalisador para a interação entre aspirantes a programadores e empresas em busca de soluções tecnológicas. A plataforma transcende as fronteiras do recrutamento convencional, abraçando o conceito inovador de gamificação para transformar o processo de aprendizado e colaboração em uma jornada envolvente e recompensadora.

À luz da dinâmica acirrada do mercado de trabalho para desenvolvedores, a *"HastyDEV” se* destaca ao oferecer não apenas oportunidades de emprego, mas também um terreno fértil para a construção ativa de portfólios profissionais. Ao permitir que empresas, inclusive as de menor porte, apresentem suas demandas específicas, a plataforma propicia um canal facilitador para que novos talentos desenvolvam suas habilidades práticas e tangíveis.

## 2.2 Visibilidade Digital para ONGs

A ”*HastyDEV"* não se limita ao escopo do mercantil, abraçando um compromisso de cunho social. Ao incorporar uma dimensão solidária, a plataforma oferece espaço e visibilidade para organizações não governamentais sem fins lucrativos. Desse modo, a *"HastyDEV” transcende* a esfera meramente comercial, alinhando-se a uma narrativa de impacto social positivo ao proporcionar aos desenvolvedores a oportunidade de contribuir para causas altruístas.

Nesse contexto, a *"HastyDEV” surge* não apenas como uma plataforma de intermediação entre demanda e oferta no setor de desenvolvimento de software, mas como uma força propulsora que redefine a dinâmica convencional, fomentando o crescimento profissional, a inovação e o comprometimento social.

## 2.3 Gamificação no Desenvolvimento Profissional

De acordo com Kapp (2012), em seu livro *The Gamification of Learning and Instruction*: “Gamificação é a utilização de mecânicas baseadas em games, estética e pensamento gamer para engajar as pessoas, motivar ações, promover o aprendizado e a solução de problemas” (KAPP, 2012, traduzido pelo autor).

Com base em seu livro "*Gamification*: princípios e estratégias", Raul Inácio Busarello (2016), define gamificação como "o uso de elementos de jogos em contextos não lúdicos, com o objetivo de motivar e engajar as pessoas em processos de aprendizagem, resolução de problemas, mudança de comportamento, entre outros". Ele também destaca que a gamificação pode ser aplicada em diferentes áreas, como corporativa, mercadológica ou instrucional.

Segundo Cristiano Tonéis (2017), A Gamificação corresponde ao uso de mecanismos de jogos aplicados em situações que não correspondem a jogos, ou seja, para solucionar problemas práticos ou ainda despertar engajamento entre um público específico pode-se utilizar elementos dos games.

# **3 LINGUAGENS DE PROGRAMAÇÃO**

## 3.1 HTML

Segundo Robertha Pereira Pedroso (2007), A linguagem HTML (*Hypertext Markup Language*) tem o objetivo de formatar textos através de marcações especiais denominadas *tags*, para que possam ser exibidos de forma conveniente pelos clientes Web, também denominados navegadores ou browsers. Além disso, está linguagem possibilita a interligação entre páginas da Web, criando assim documentos com o conceito de hipertexto.

O HTML é dividido em duas partes principais:

* Elementos: Os elementos são blocos de construção básicos do HTML. Eles são usados para definir o conteúdo e a estrutura de uma página web.
* Atributos: Os atributos são informações adicionais que podem ser adicionadas aos elementos HTML. Eles são usados para controlar a aparência e o comportamento dos elementos.

Para usar o HTML, os desenvolvedores web criam arquivos HTML. Esses arquivos contêm o código HTML que define a estrutura de uma página web. Os arquivos HTML são interpretados pelos navegadores web, que mostram as páginas web para os usuários.

Aqui estão alguns exemplos de elementos HTML comuns:

* `<html>`: Define o início do documento HTML.
* `<head>`: Define o cabeçalho do documento HTML.
* `<title>`: Define o título do documento HTML.
* `<body>`: Define o corpo do documento HTML.
* `<p>`: Define um parágrafo.
* `<img>`: Define uma imagem.
* `<a>`: Define um link.

Na Figura 1, é exibido o código HTML empregado na concepção de uma página web elementar, contemplando um formulário de contato. Por sua vez, a Figura 2 ilustra o desfecho decorrente da aplicação desse código.

**Figura 1 – Código em HTML**

Texto

Descrição gerada automaticamente com confiança baixa

Fonte: Autoria Própria (2023)

**Figura 2 – Resultado do código em HTML**

Uma imagem contendo Gráfico de cascata

Descrição gerada automaticamenteFonte: Autoria Própria (2023)

Este código define que o documento HTML contém um formulário de contato na parte principal da página, permitindo que os usuários forneçam seus nomes, e-mails e mensagens.

Então, a partir disso, entende-se que, o HTML é uma ferramenta poderosa que pode ser usada para criar página web com uma variedade de formatos e estilos. Com um pouco de prática, os desenvolvedores *WEB* podem usar o HTML para criar páginas web atraentes e informativas.

## 3.2 CSS

De acordo com Joaquim Celestino Júnior (2015), *Cascading Style Sheets* (CSS) é um acrônimo para CSS, termo em inglês que significa Folhas de Estilo em Cascatas. Trata-se de uma linguagem de estilo simples e bastante útil, utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação. Uma das características principais do CSS é sua capacidade de proporcionar a separação entre o formato e o conteúdo de um documento, facilitando a criação de documentos mais bem estruturados e fáceis de manter.

E, segundo Joaquim Celestino Júnior (2015), o CSS desempenha um papel fundamental na definição da apresentação de documentos escritos em linguagem de marcação. Seu principal benefício reside na separação entre o formato e o conteúdo de um documento. O funcionamento é simples: em vez de inserir a formatação diretamente no documento original utilizando as tags de marcação do HTML ou derivados, o desenvolvedor cria um link ou importa um arquivo de folha de estilos na página contendo as informações a serem formatadas. Em seguida, os estilos são chamados e aplicados, resultando na formatação da página conforme definido no arquivo CSS.

O autor ressalta que o CSS se destaca como uma das ferramentas mais amplamente difundidas na atualidade, sendo aplicado para diversos fins e proporcionando uma abordagem eficaz para aprimorar a apresentação e o estilo de documentos.

Aqui estão alguns dos atributos mais comumente empregados no CSS:

* *Background*: controle da cor de fundo, imagem e transparência.
* *Border*: definição da largura, estilo e cor das quatro bordas.
* *Position*: determinação da posição do elemento na página.
* *Text-align*: alinhamento do texto.
* *Width*: especificação da largura do elemento.
* *Padding*: configuração do espaço ao redor do elemento em todos os lados.
* *Margin*: estabelecimento do tamanho das quatro margens.
* *Font-family*: escolha do tipo de fonte.
* *Color*: controle da cor do texto.
* *Font-size*: especificação do tamanho da fonte.
* *Line-height*: definição da altura da linha de texto.

Um exemplo de como utilizar a linguagem CSS está nas imagens 3 e 4, onde na figura 3 temos o código foi estilizado para melhorar visualmente o formulário, aplicando fontes, cores e bordas. Destaques incluem fundo escuro no cabeçalho, contêiner centralizado e botão interativo. A figura 4 é o resultado do código.

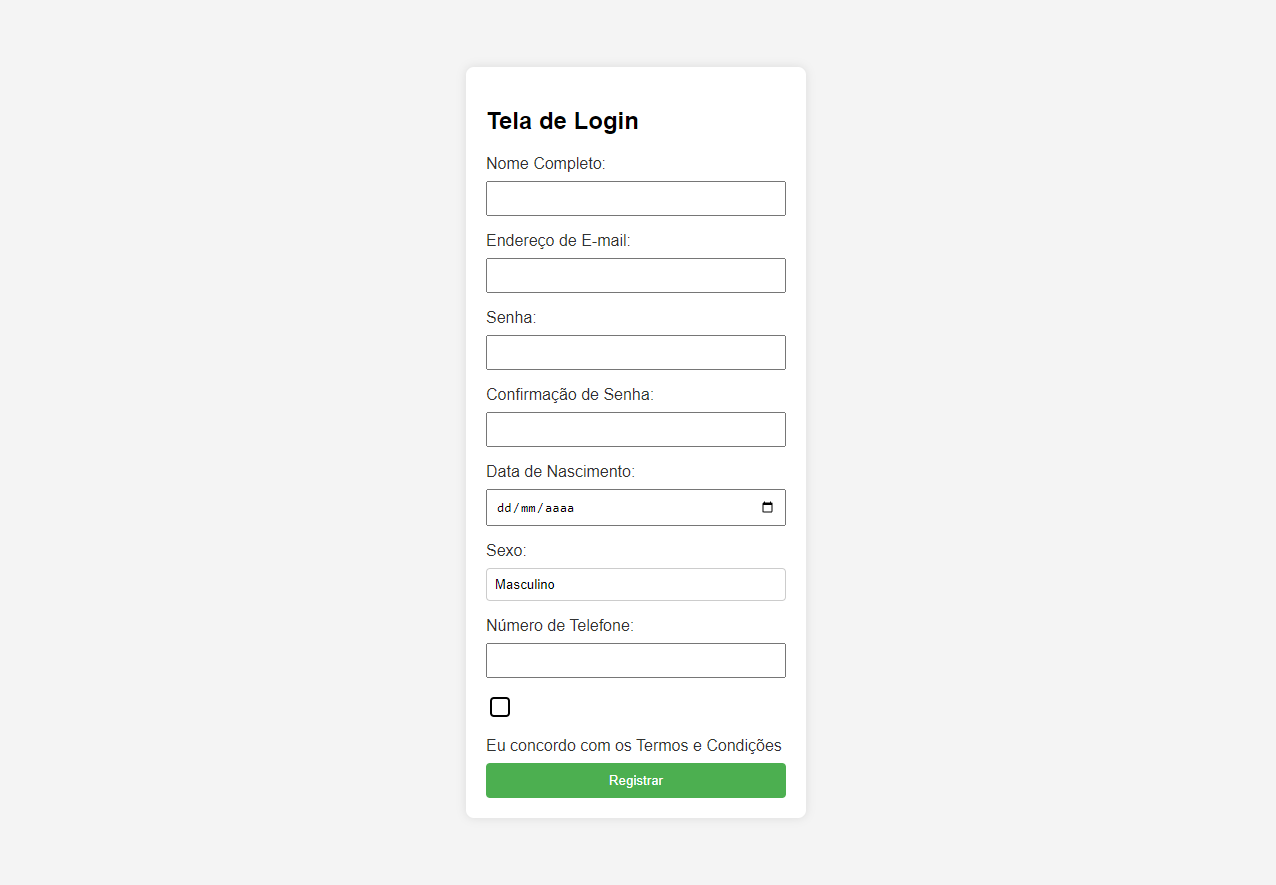
**Figura 3 – Código em CSS**

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

Fonte: Autoria Própria (2023)

**Figura 4 – Resultado do código em CSS**



Fonte: Autoria Própria (2023)

Este código define que o documento HTML que contém um formulário de contato seja estilizado a partir do CSS, adicionando cores, editando o botão, adicionando margens internas e externas.

Então, a partir disso, entende-se que, o CSS é uma ferramenta poderosa que pode ser usada estilizar e customizar páginas *WEB* por completo de modo básico até coisas avançadas. Com um pouco de prática, os desenvolvedores *WEB* podem usar o CSS para estilizar páginas *WEB* de maneira profissional.

## 3.3 Javascript

De acordo com Maurício Samy Silva (2010), Javascript (JS) 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.

Ainda de acordo com Maurício Silva (2010), Javascript é capaz de definir, alterar e controlar de forma dinâmica a apresentação de um documento HTML, como os aspectos relacionados à cor de fundo, de textos e de links, ou mesmo interferir no posicionamento dos elementos HTML de um documento. É possível manipular a folha de estilos associada ao documento criando regras CSS ou anulando regras existentes.

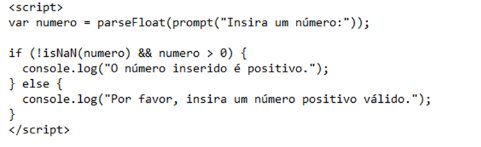
Segundo Loiane Groner (2017), A linguagem Javascript é extremamente eficaz. É a linguagem mais popular do mundo e uma das mais proeminentes na internet. Por exemplo, o GitHub (o maior host de códigos do mundo, disponível em https://github.com) hospeda mais de 400 mil repositórios de Javascript (a maioria dos projetos está em Javascript). O número de projetos em Javascript no GitHub aumenta a cada ano.

Aqui estão alguns exemplos de *tags* HTML mais utilizadas com o JS:

* <script>: é fundamental para incorporar código JavaScript em documentos HTML. Pode ser usado no corpo da página ou na seção de cabeçalho.
* <button>: é comumente usada para criar botões interativos. Eventos Javascript, como click, podem ser associados a esses botões para executar funções quando clicados.
* <input>: é frequentemente utilizada em formulários HTML. Javascript pode ser usado para manipular os valores de entrada e realizar ações com base nas interações do usuário.
* <form>: usada para criar formulários em. Javascript pode ser empregado para validar entradas ou realizar ações quando o formulário é enviado.
* <script src=”meuscript.js”>: A *tag* <script> com o atributo src é usada para incorporar arquivos Javascript externos em uma página HTML. O atributo src especifica o URL do arquivo Javascript que será importado.

A seguir, na figura 5, um exemplo simples ilustra o funcionamento do *"if"* e *"else"*, os quais são empregados para lidar com a confirmação ou negação de uma condição. Na figura 6, é apresentado um prompt que solicita ao usuário inserir dados, permitindo que o *"if"* e *"else"* realizem a verificação dos dados inseridos e a figura 7 exibe o resultado no console do navegador.

**Figura 5 – Código em Javascript**



Fonte: Autoria Própria (2023)

**Figura 6 – Solicitação do prompt**

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Fonte: Autoria Própria (2023)

**Figura 7 – Resultado do código em JS**



Fonte: Autoria Própria (2023)

Este código define um programa, na qual será inserido um número e o Javascript irá retornar se aquele número era positivo ou negativo, demonstrando um caso simples de utilização, mas que pode ser exponencialmente escalado para aplicações maiores, como a construção de um ecossistema inteiro de um *e-commerce,* por exemplo.

Então, a partir disso, entende-se que, o JS é uma ferramenta fundamental na *WEB* e pode implementar interatividade em qualquer *website,* desde coisas simples como um envio de formulário até a criação de um *gateway* de pagamentos. Com muito estudo de lógica de programação e aprendendo a sintaxe, os desenvolvedores *WEB* podem usar o JS para transformar as páginas estáticas por completo.

## 3.4 Typescript

De acordo com Ugo Roveda (2023), O Typescript é um *superset* do Javascript (JS), ou seja, um conjunto de ferramentas, foi criado com o objetivo de incluir recursos que não estão presentes no JS. Por meio dele é possível definir a tipagem estática, parâmetros e retorno de funções. Além de ser uma ferramenta orientada a objetos, fortemente tipada e que pode ser escrita em qualquer ambiente de desenvolvimento, o Typescript quando instalado via gerenciador de pacotes JS, permite checar erros e utilizar outros compiladores que suportam este mecanismo.

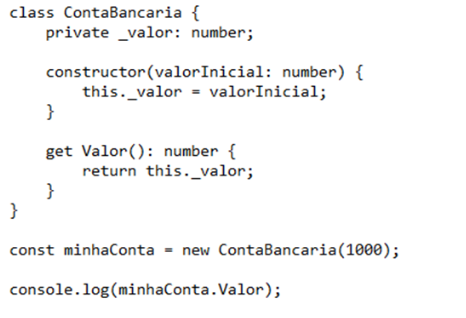
De acordo com Peter Vogel (2015), “Na verdade geralmente descreve-se as construções Typescript em termos de código Javascript resultante. Mas é igualmente útil pensar no Typescript como uma linguagem própria que compartilha recursos com o Javascript.”

Conforme Josh Goldberg (2022), As tipagens do Typescript permitem que editores como o *VS Code* recebam informações muito mais detalhadas sobre o código. Eles podem então usar essas informações para gerar sugestões inteligentes enquanto digitamos. Essas sugestões podem ser extremamente úteis para o desenvolvimento.

Nas palavras de Cairo Noleto (2020), “Ao aplicarmos esse superconjunto, podemos melhorar o suporte à programação orientada a objetos, já que turbinamos o Javascript com várias possibilidades que a linguagem pura não dispõe”. Sendo assim o Typescript amplia a capacidade do Javascript e facilita o desenvolvimento.

Veja abaixo um exemplo de código Typescript envolvendo encapsulamento, um conceito de orientação a abjetos que dá visibilidade e acessibilidade a elementos internos de uma classe. Nesse caso, a declaração de uma variável de saldo, do tipo “*number*”, é acessível pelo método “Valor()”.

**Figura 8 – Exemplo de código em TS**



Fonte: Autoria Própria (2023)

## 3.5 REACTJS

De acordo com os preceitos de David Flanagan (2012) sobre Javascript, é crucial compreender que esta linguagem desempenha um papel central no desenvolvimento *WEB*, principalmente no lado do cliente, por ser “A Linguagem de Programação da *WEB*”. No entanto, à medida que a tecnologia avança, bibliotecas e frameworks emergem para oferecer soluções mais eficientes e estruturadas. Nesse contexto, o React, uma biblioteca Javascript desenvolvida e mantida pelo Facebook, ganhou destaque como uma ferramenta robusta para a construção de interfaces de usuário interativas e dinâmicas. Segundo a pesquisa Developer Survey realizada pela plataforma Stack Overflow no ano de 2022, o React foi eleito a biblioteca mais utilizada para desenvolvimento WEB.

Ao contrário do Javascript convencional, onde as manipulações do DOM podem se tornar complexas, o React introduz um modelo de programação declarativo, facilitando a criação e manutenção de componentes reutilizáveis. Como destacado por Buna (2017), ReactJS tem ganhado destaque e se tornou uma escolha proeminente no desenvolvimento *front-end*, sendo amplamente adotado por inúmeros projetos e empresas.

Ela também destaca os principais motivos pelos quais o ReactJS tem sido cada vez mais utilizado por empresas de diferentes tamanhos e setores. Esta biblioteca, baseada em JavaScript, emergiu como uma das tecnologias mais promissoras, oferecendo desempenho superior em relação às tecnologias existentes e uma curva de aprendizado acessível. Empresas, independentemente de serem pequenas, médias ou grandes, de diversos setores têm escolhido o ReactJS para desenvolver desde sistemas internos até aplicações voltadas para usuários comuns.

Uma das características marcantes do React é a sua eficácia no gerenciamento do estado da aplicação. Por meio de um sistema de estado, os componentes React podem atualizar de forma eficiente a interface do usuário em resposta a eventos e interações do usuário, proporcionando uma experiência fluida e responsiva.

Diferentemente das *tags* HTML tradicionais mencionadas em relação ao Javascript puro, React utiliza uma sintaxe especial chamada JSX. JSX combina elementos HTML com a lógica do Javascript, permitindo a criação de componentes de maneira mais intuitiva e legível.

No universo do React, alguns elementos fundamentais merecem destaque:

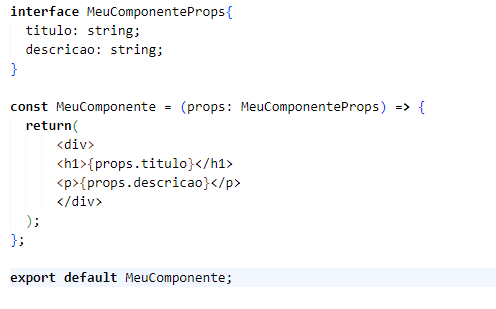
* Componentes: A base do desenvolvimento React. Componentes encapsulam a lógica e a interface do usuário, promovendo a modularidade e reutilização de código.
* Props (Propriedades): Mecanismo para passar dados entre componentes. As props facilitam a comunicação eficiente entre diferentes partes da aplicação.
* Estado (*State*): Permite que os componentes reajam a eventos e atualizem dinamicamente a interface do usuário. O estado é gerenciado internamente pelos componentes React.
* Ciclo de Vida (*Lifecycle*): Define os diferentes estágios de um componente, desde sua criação até sua remoção. Este controle granular permite a execução de ações específicas em momentos específicos.

Assim como o Javascript é incorporado ao HTML por meio da *tag* <script>, o React é integrado aos projetos por meio de ferramentas como o npm (*Node Package Manager*) e *bundlers* como o *Webpack*, permitindo uma gestão eficiente das dependências e do ciclo de desenvolvimento.

O React se destaca como uma ferramenta poderosa no desenvolvimento *front-end*, proporcionando uma abordagem estruturada e eficiente para a construção de interfaces de usuário interativas. A crescente comunidade e a vasta adoção em projetos renomados reforçam sua posição como uma tecnologia essencial no arsenal de desenvolvedores web modernos.

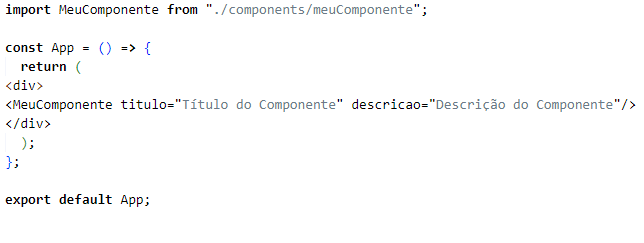
O código apresentado abaixo na figura 9 é um exemplo básico de um componente React reutilizável, utilizando do *superset* Typescript e demonstrando o conceito de props. Na figura 10, temos um exemplo de como o componente pode ser utilizado dentro de um componente pai e na figura 11 temos o resultado disso em tela.

**Figura 9 – Exemplo de código em React em typescript usando props**



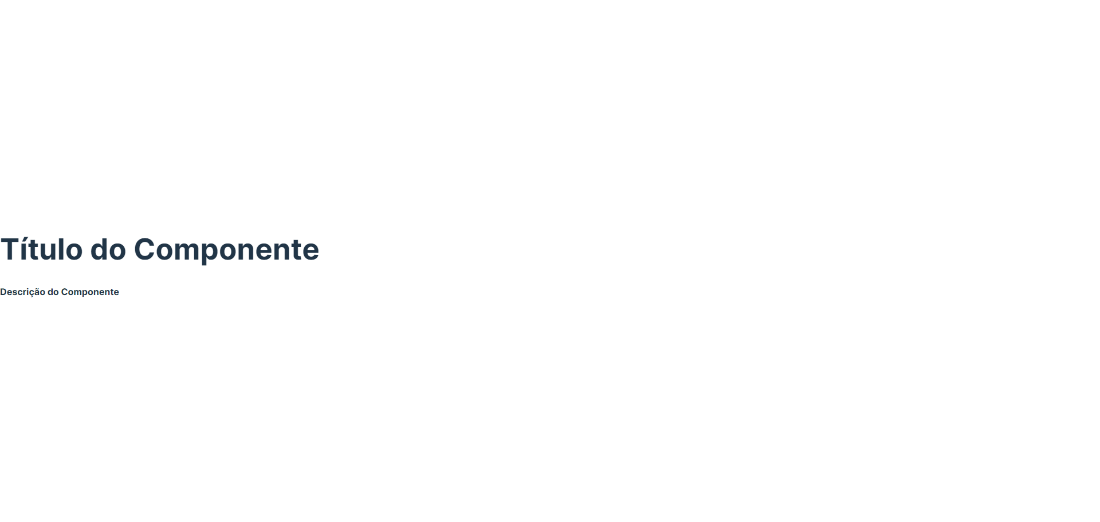
Fonte: Autoria Própria (2023)

**Figura 10 – Exemplo de utilização de um componente React com props**



Fonte: Autoria Própria (2023)

**Figura 11 – Resultado do código em React**



Fonte: Autoria Própria (2023)

## 3.6 NODEJS

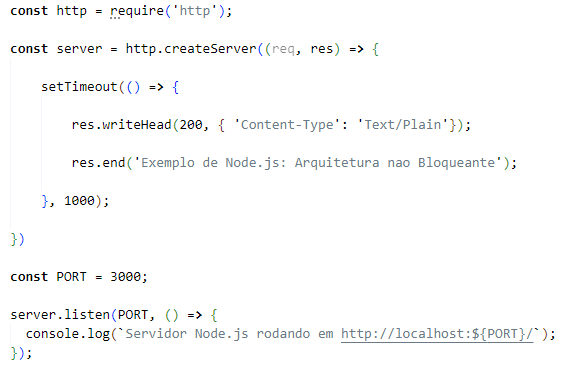
Node.js é um ambiente de execução Javascript do lado do servidor, construído sobre o motor V8 da Google. Ele permite que desenvolvedores usem Javascript para criar aplicativos escaláveis e de alta performance, unificando o desenvolvimento *WEB* ao usar a mesma linguagem tanto no lado do cliente quanto no do servidor.

O escritor Caio Ribeiro Pereira destaca a significativa contribuição de Ryan Dahl, que, no final de 2009, criou o Node.js em colaboração com 14 desenvolvedores. A inovação reside na arquitetura não bloqueante, proporcionando um desempenho otimizado em operações intensivas de I/O e processamento de arquivos, maximizando a eficiência do poder de processamento dos servidores.

Além disso, a abordagem de único *thread* por processo no Node.js, como mencionado por Caio Ribeiro Pereira, não apenas evita *dead-locks*, mas também simplifica e torna prático o desenvolvimento de sistemas. Ao explorar o potencial do Javascript no *server-side*, o Node.js emerge como uma plataforma poderosa para a criação de aplicações *WEB* escaláveis, conforme discutido pelo autor em "Aplicações *WEB real-time* com Node.js" (Pereira, 2014).

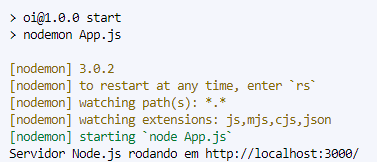
No contexto da arquitetura não bloqueante do Node.js, consideremos um exemplo prático de servidor HTTP básico (Figura 12), a resposta no console (Figura 13) e quando fazemos uma requisição (Figura 14):

### Figura 12 – Exemplo de código em Node.js



Fonte: Autoria Própria (2023)

**Figura 13 – Resultado do código no console**



Fonte: Autoria Própria (2023)

**Figura 14 - Resultado do código após requisição**



Fonte: Autoria Própria (2023)

O código define um servidor HTTP simples usando o módulo HTTP. A função de *callback* dentro de *createServer* simula uma operação demorada usando *setTimeout*. Mesmo durante essa operação demorada, o servidor pode continuar a processar outras solicitações, demonstrando a eficiência do Node.js em operações não bloqueantes

**4 UML**

De acordo com o livro "UML: Guia do Usuário", a UML (Unified Modeling Language) é descrita como uma "linguagem-padrão para a elaboração da estrutura de projetos de software, adequada para a modelagem de sistemas cuja abrangência pode incluir sistemas de informação corporativos, aplicações baseadas em Web e sistemas complexos embutidos de tempo real" (Booch, 2006, p. 13).

"A UML abrange a documentação da arquitetura do sistema e de todos os seus detalhes. A UML também proporciona uma linguagem para a expressão de requisitos e para a realização de testes. Por fim, a UML oferece uma linguagem para a modelagem das atividades de planejamento do projeto e de gerenciamento de versões. Essa abordagem holística da UML, conforme destacado por Booch (2006), não apenas facilita a compreensão e comunicação entre os membros da equipe de desenvolvimento, mas também contribui para um desenvolvimento de software mais eficiente e de alta qualidade. A capacidade da UML de integrar a documentação, expressão de requisitos, testes e modelagem de atividades faz dela uma ferramenta valiosa em todas as fases do ciclo de vida do desenvolvimento de software."