

FUNDAMENTOS DE TECNOLOGIAS WEB: TRILHA DE PROGRAMAÇÃO PARA O NOVO CURRÍCULO DO ENSINO MÉDIO DO PARANÁ

FUNDAMENTALS OF WEB TECHNOLOGIES: PROGRAMMING TRACK FOR THE NEW PARANÁ HIGHSCHOOL CURRICULUM

Fernando Barreto ¹

Dyana Grazielli Altomani Braga ²

Estefane de Lima dos Santos ³

Kauane dos Santos Vieira ⁴

Resumo: *A tecnologia computacional tem proporcionado grandes transformações na vida e relações do ser humano. No estado do Paraná, a difusão de disciplinas de programação no meio educacional implicou na necessidade de formação de professores para essas áreas tecnológicas. Este trabalho apresenta um relato de experiência de um projeto de extensão para capacitar os professores nos conceitos de linguagem de programação Web. A pesquisa é de natureza qualitativa, do tipo exploratório, envolvendo os professores do Ensino Médio da rede pública de ensino. Os resultados obtidos mostraram que os professores desenvolveram a concentração na execução das atividades propostas e interação com as ferramentas de programação. A análise do corpus de pesquisa indicou que as atividades contribuíram para a aprendizagem e enriquecimento da prática pedagógica dos professores, desenvolvendo habilidades como identificação e seleção de estratégias para construção dos algoritmos, oferecendo-lhes oportunidade de pensar, discutir, reconhecer e desenvolver a lógica de programação, tornando assim o aprendizado mais significativo.*

Palavras-chave: *Programação. Educação. Ensino Médio. Aprendizagem.*

Abstract: *Computer technology has provided major changes in human life and relationships. In the state of Paraná, the spread of programming disciplines in the educatio-*

1 Possui graduação em Ciência da Computação pela Universidade Estadual de Maringá (1999), mestrado em Ciência da Computação pela Universidade Federal de Santa Catarina (2002) e doutorado em Ciências pelo Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná. Atualmente é professor da Universidade Tecnológica Federal do Paraná. Lattes: <http://lattes.cnpq.br/1183013437890631>. ORCID: <https://orcid.org/0000-0001-8271-2658>. E-mail: fbarreto@utfpr.edu.br

2 Possui graduação em Licenciatura em Química pela Universidade Norte do Paraná - UNOPAR (2002), graduação em Química Industrial pela Universidade Norte do Paraná - UNOPAR (2002) e Licenciatura em Matemática - Claretiano Centro Universitário (2018). Especialização em Bioquímica Aplicada pela Universidade Estadual de Londrina - UEL (2004), Especialização em Gestão e Análise Ambiental pela Universidade Norte do Paraná - UNOPAR (2006), Pós-Graduação em Atendimento Educacional Especializado - Faculdade Eficaz (2011), Pós-Graduação em Tutoria de Ensino a Distância com ênfase em Química - Faculdade Eficaz (2012). Mestre em Ensino de Matemática (2024) pelo PPGMAT da UTFPR, campus Londrina/Pr. Lattes: <http://lattes.cnpq.br/8131688808163254>. ORCID: <https://orcid.org/0000-0003-1079-3537>. E-mail: dyana.altomani@escola.pr.gov.br

3 Graduanda na Universidade Tecnológica Federal do Paraná. E-mail: estefane@alunos.utfpr.edu.br

4 Graduanda de Engenharia de Computação na Universidade Tecnológica Federal do Paraná (UTFPR) - Campus Apucarana. Lattes: <http://lattes.cnpq.br/4803386757177057>. ORCID: <https://orcid.org/0000-0003-1079-3537>. E-mail: kauanevieira@alunos.utfpr.edu.br

nal environment implied the need to train teachers for these technological areas. This paper presents an experience report of an extension project to train teachers in the concepts of Web programming language. The research is qualitative and exploratory in nature, involving high school teachers from the public school system. The results obtained showed that the teachers developed concentration in the execution of the proposed activities and interaction with the programming tools. The analysis of the research **corpus** indicated that the activities contributed to the learning and enrichment of the teachers' pedagogical practice, developing skills such as identification and selection of strategies for constructing algorithms, offering them the opportunity to think, discuss, recognize and develop programming logic, thus making learning more meaningful.

Keywords: Programming. Education. High School. Learning.

Introdução

Nos dias atuais, faz-se necessário repensar as metodologias e estratégias a serem desenvolvidas em sala de aula, de forma a possibilitar aos estudantes que se tornem sujeitos ativos na construção do próprio conhecimento. Com esse entendimento, os professores precisam se apropriar das possibilidades de utilização de recursos tecnológicos em suas aulas, buscando desenvolver um ensino mais próximo da realidade de seus estudantes.

Para tanto, “com a utilização do computador na educação é possível ao professor e a escola dinamizarem o processo de ensino-aprendizagem, com aulas mais criativas, mais motivadoras e que despertem, nos estudantes, a curiosidade e o desejo de aprender” (MEC, 2007). Assim, nas diversas áreas de conhecimento, as ações educativas dos professores, segundo Bezerra e Carvalho (2011), devem estar centradas na construção de um processo educativo alicerçado na interatividade e na criatividade, que provoque discussões, dúvidas e instigue a aprendizagem dos estudantes.

Para tanto, a partir do Parecer CNE/CEB Nº 2/2022 (MEC1, 2022), a Base Nacional Comum Curricular (BNCC) teve a inserção um documento denominado “Computação - Complemento à BNCC” (MEC2, 2022) para ensino do 5º até 9º anos do ensino fundamental e ensino médio. Esse complemento pretende garantir direitos de aprendizagem relacionados ao pensamento computacional, mundo digital e cultura digital, extremamente importantes no mundo atual. Assim, no estado do Paraná, a Secretaria de Estado da Educação do Paraná (SEED-PR) realizou mudanças no currículo para abordar os direcionamentos do documento de “Computação - Complemento à BNCC”. Essas mudanças resultaram no Novo Ensino Médio Paranaense, que adiciona Itinerários Formativos (IF) em áreas como Projeto de Vida, Educação Financeira e Pensamento Computacional. No que tange Pensamento Computacional, a SEED-PR organizou ementas para contemplar conteúdos direcionados às Tecnologias Web, como Linguagem de Marcação de Hipertexto (HTML), Folhas de Estilo em Cascata (CSS) e Linguagem de Programação JavaScript (JS). No entanto, como são áreas que não contemplam os componentes curriculares da base comum, os Núcleos Regionais de Educação (NRE) apresentam carência de professores habilitados ou com competências nes-

ses conteúdos específicos necessários para a demanda atual. Diante dessas dificuldades, a SEED-PR disponibiliza, por meio do sistema Registro de Classe Online (RCO) (RCO, 2024), materiais para aos professores dos NREs, com o intuito de contornar essas dificuldades, com o acompanhamento da “Programação Paraná” (SEED-PR, 2024).

Diante do exposto, faz-se necessário proporcionar formação aos professores do NREs que possuem uma graduação na base comum, que não é direcionada para essa área, para suprir as lacunas de aprendizagem desses professores e proporcionar uma boa evolução no desenvolvimento das ementas no componente curricular de pensamento computacional junto aos estudantes do Ensino Fundamental e Ensino Médio.

A partir dessa percepção, o NRE de Apucarana, que atende os municípios de Apucarana, Arapongas, Bom Sucesso, Borrazópolis, Califórnia, Cambira, Cruzmaltina, Faxinal, Jandaia do Sul, Kaloré, Mari-lândia do Sul, Marumbi, Mauá da Serra, Novo Itacolomi, Rio Bom e Sabáudia, estabeleceu uma parceria junto à Universidade Tecnológica Federal do Paraná - Campus Apucarana (UTFPR Apucarana) por meio do curso de Engenharia de Computação, para obter uma formação aos professores do NRE nas linguagens de HTML, CSS e JavaScript.

Mediante essa parceria e pelo interesse institucional, a UTFPR Apucarana organizou junto ao NRE Apucarana um projeto de extensão de Fundamentos de Tecnologias Web, o qual foi contemplado com dois estudantes bolsistas do curso de Engenharia de Computação para organizar material didático e realizar capacitações presenciais junto aos professores.

Este relato de experiência descreve a evolução desse projeto de extensão por meio da formação e capacitação dos professores junto ao NRE Apucarana.

Metodologia

Após as reuniões iniciais entre NRE Apucarana e a UTFPR Apucarana, foi deliberado o encaminhamento do projeto como sendo: elaborar material didático de tecnologias Web, seleção de ferramentas de desenvolvimento e disponibilização desse material através de um ambiente virtual de aprendizagem Moodle (MOODLE, 2024) da UTFPR Apucarana.

O conteúdo do material de tecnologias Web foi elaborado e estruturado a partir do próprio material já disponibilizado e homologado pela SEED-PR, uma vez que os professores do NRE o utilizam para trabalhar suas aulas, e materiais abertos oficiais referente aos padrões de tecnologias Web: HTML (W3C1, 2024), CSS (W3C2, 2024) e JavaScript (ECMA, 2024).

O HTML não é uma linguagem de programação e sim linguagem de marcadores para a criação de páginas web e a estruturação de seu conteúdo. O HTML permite a estruturação em texto de documentos utilizando uma série de elementos aninhados, como cabeçalhos, parágrafos, links, indicação de imagens e outros recursos multimídia. Cada elemento é representado por tags de texto que definem seu propósito e formato dentro do documento. Com o advento do HTML5, a linguagem expandiu suas capacidades para incluir suporte nativo a áudio e vídeo, gráficos vetoriais escaláveis (SVG) e novos elementos semânticos que melhoram a acessibilidade e a otimização para motores de busca. A padronização do HTML é gerida pela World Wide Web Consortium (W3C), garantindo interoperabilidade e compatibilidade entre diferentes navegadores e dispositivos. A ampla adoção do HTML como padrão da Web moderna evidencia sua importância na facilitação da comunicação e disseminação de informação online (W3C, 2024).

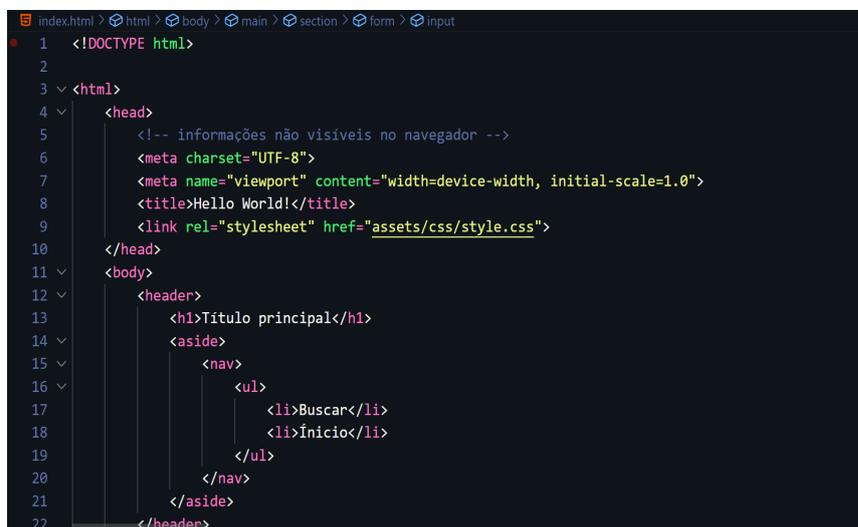
O CSS (Cascading Style Sheets) surgiu devido à necessidade de separar a apresentação visual do conteúdo estrutural das páginas Web, algo que não era possível com o HTML sozinho. Diferente do HTML, o CSS é uma linguagem de estilo utilizada para descrever a apresentação de documentos escritos em HTML ou XML. Com a separação do conteúdo da estrutura, o CSS permite controlar a aparência de elementos como cores, fontes, espaçamento, layout e animações, proporcionando uma experiência de usuário mais consistente e atraente. A cascata e a especificidade são conceitos fundamentais no CSS, per-

mitindo que estilos sejam aplicados de maneira hierárquica e precisa. As mais recentes especificações do CSS3 introduziram funcionalidades avançadas, como transições, transformações, e gradientes, elevando o potencial estético e interativo das páginas web (CSS, 2024).

Como as linguagens HTML e CSS focam na estrutura e estilo, respectivamente, o JavaScript surgiu da necessidade de adicionar dinamismo e interatividade às páginas web no navegador Web. Portanto, o JavaScript permite aos desenvolvedores web criar comportamentos dinâmicos diretamente no navegador do cliente, como validação de formulários em tempo real, manipulação do Modelo de Documento por Objeto (DOM) (W3C3, 2024), recurso para possibilitar criar dinamismo ao HTML padrão, para atualizar o conteúdo de uma página sem recarregá-la completamente, e comunicação assíncrona com servidores (AJAX), possibilitando atualizações de conteúdo em segundo plano. Dessa forma, o JavaScript é fundamental para o desenvolvimento de aplicações web modernas e interativas (ECMA, 2024).

Para editar os arquivos fonte desses padrões de tecnologia Web, optou-se por adotar uma ferramenta gratuita e que estivesse alinhada com o material homologado pela SEED-PR: Visual Studio Code (VSCode) (VSCODE, 2024). Essa ferramenta possui inúmeros recursos com suporte a várias linguagens de programação e desenvolvimento Web. Ela permite editar arquivos de código fonte HTML, também chamado apenas de hipertexto, CSS e JavaScript com realce de marcações e palavras reservadas, recurso de autocompletar, sugestões, etc que facilitam o desenvolvimento. Pode-se observar nas Figuras 1, 2 e 3 como o editor VSCode facilita a edição de HTML, CSS e JavaScript respectivamente, pois há realce do texto/tags/palavras importantes da sintaxe além de coloração diferenciada, indentação facilitada, indicador e alinhamento de início/fim de tag ou palavras reservadas, recursos de autocompletar, dentre outros que facilitam o trabalho do desenvolvedor.

Figura 1. Exemplo de código HTML no editor VSCode



```
index.html > html > body > main > section > form > input
1 <!DOCTYPE html>
2
3 <html>
4   <head>
5     <!-- informações não visíveis no navegador -->
6     <meta charset="UTF-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>Hello World!</title>
9     <link rel="stylesheet" href="assets/css/style.css">
10  </head>
11  <body>
12    <header>
13      <h1>Titulo principal</h1>
14      <aside>
15        <nav>
16          <ul>
17            <li>Buscar</li>
18            <li>Início</li>
19          </ul>
20        </nav>
21      </aside>
22    </header>
```

Fonte: Elaborado pelo autor (2024).

Figura 2. Exemplo de código CSS no editor VSCode

```
1  /*FONTE IMPORTADA*/
2  @import url('https://fonts.googleapis.com/css2?family=Noto+Sans+Buhid&family=Tac+One&display=swap');
3
4  :root { /*VARIÁVEIS*/
5      --cor-section: #e98888;
6      --font-padrao: "Noto Sans Buhid", "sans-serif";
7      --cor-botao: #72bedb;
8  }
9
10 * { /*SELETOR UNIVERSAL*/
11     margin: 0;
12     padding: 0;
13     box-sizing: border-box;
14 }
15
16 h1 { /*SELEÇÃO POR TAG HTML*/
17     color: yellow;
18 }
19
20 .subtitulo { /*SELEÇÃO POR CLASSE*/
21     color: green;
22     display: inline;
23     font-weight: bold;
24     font-style: italic;
25     text-decoration: underline;
26 }
27
28 #paragrafo { /*SELEÇÃO POR ID*/
29     width: 50%;
30     color: rgb(34, 67, 120);
31 }
32
33 h1:active { /*PSEUDO-CLASSE*/
34     color: var(--cor-section);
35 }
36
37 #paragrafo::first-line { /*PSEUDO-ELEMENTO*/
38     font-size: 2rem;
39 }
```

Fonte: Elaborado pelo autor (2024).

Figura 3. Exemplo de código JavaScript no editor VSCode

```
44 //Estruturas condicionais
45 let numero;
46 numero = prompt("Digite um número: ");
47 if((numero % 2) == 0){
48     console.log("O número é par!");
49 }else{
50     console.log("O número é impar!");
51 }
52
53 //Operadores de Comparação
54 let idade;
55 idade = prompt("Digite sua idade: ");
56 if(idade >= 18){
57     console.log("Você é maior de idade!");
58 }else{
59     console.log("Você não é maior de idade!");
60 }
61
62 //Operadores Lógicos
63 if(idade < 12){
64     console.log("Você é uma criança!");
65 }else if(idade >= 18 && idade <= 40){
66     console.log("Você é jovem!");
67 }else{
68     console.log("Você é adulto!");
69 }
70
71 let numero2 = 34;
72 if(numero2 > 20 || numero2 < 30){
73     console.log(numero2);
74 }
75
76 //valores booleanos
77 let valor = false;
78 console.log(!valor);
79
80 //Number
81 let variavel = "10";
82 console.log(Number(variavel));
83
84 //Array (vetor)
85 let array = [1, 10, 12];
86 console.log(array[0]);
87 console.log(array[1]);
88 console.log(array[2]);
89
90 //Estruturas de Repetição
91 for(let i = 0; i < 5; i++){
92     console.log(i);
93 }
94
95 let l = 0;
96 while(l < 10){
97     console.log(l);
98     l++;
99 }
100
101 //Função
102 function somaFuncao(numero1, numero2){
103     return numero1 + numero2;
104 }
```

Fonte: Elaborado pelo autor (2024)

Para poderem acompanhar como está o desenvolvimento da implementação do código fonte HTML, CSS e JavaScript, deve-se utilizar qualquer aplicativo navegador Web comumente disponível (Chrome, Firefox, Edge, etc). Os navegadores basicamente interpretam e renderizam o código HTML, CSS e JavaScript recebido dos servidores Web para exibir páginas web de maneira compreensível e interativa. Eles também conseguem abrir arquivos de código fonte HTML, CSS e JavaScript diretamente do sistema de arquivos do computador pessoal, o que seria o caso da capacitação realizada nesse projeto. Como são padrões abertos de tecnologias Web, qualquer navegador Web atual suportará esses códigos fonte, desde que estruturados corretamente conforme as normas vigentes.

A partir da sequência didática apresentada no curso e material, os participantes desenvolveram os códigos HTML, CSS e JavaScript com a ferramenta VSCode, salvando esses arquivos em uma pasta que o participante criava. A partir do navegador Web da preferência do participante, que geralmente era o Chrome, realizava-se a verificação do que foi desenvolvido até o momento ao abrir o arquivo principal, com conteúdo HTML, para interpretar e renderizar a página Web desenvolvida.

Além da organização e preparação do material com as tecnologias Web e ferramentas, optou-se

por realizar a capacitação no formato de curso presencial para facilitar a interação com os professores do NRE e sanar possíveis dúvidas diretamente.

O foco do curso estava em direcionar o uso do material didático para o desenvolvimento de um projeto de site web que pudesse aplicar todos os conceitos de fundamentos de tecnologias Web que os professores do NRE deveriam saber, segundo a SEED-PR. O tema do site escolhido foi de Calculadora de Pegada Ecológica, pois é um projeto que auxilia os professores do NRE na execução de atividades que trabalham o tema da sustentabilidade, tal como o Projeto Agrinho (FAEP, 2024).

Desenvolvimento, resultados e discussão

O desenvolvimento do projeto junto aos professores do NRE, com a temática Calculadora de Pegada Ecológica, teve a primeira capacitação de vinte horas, ocorrendo em quatro sábados com 4 horas cada, no formato presencial totalizando 16 horas, e as 4 horas restante no formato à distância para desenvolverem atividades do tema do projeto.

A primeira metade da capacitação iniciou apresentando a ferramenta VSCode para gerenciar os arquivos a serem utilizados nesse projeto. Em seguida, iniciou-se os conteúdos fundamentais de HTML e CSS, para que os participantes adquirissem competências de estruturação/formatação de uma página Web conforme as normas vigentes desses padrões. Em relação aos conteúdos de HTML, construiu-se uma estrutura básica de documento HTML, além de abordar as principais tags utilizadas em desenvolvimento web, como tags de texto, imagem e formulários. Em relação aos conteúdos de CSS, trabalhou-se a estilização da página web feita no HTML. Primeiro, iniciou-se os métodos de seleção dos elementos a serem estilizados. Em seguida, os conteúdos referentes às unidades de medida, cores, fontes, tamanho, espaçamento, alinhamento (modelo Flexbox (FLEXBOX, 2024)), pseudo-classes e pseudo-elementos foram abordados.

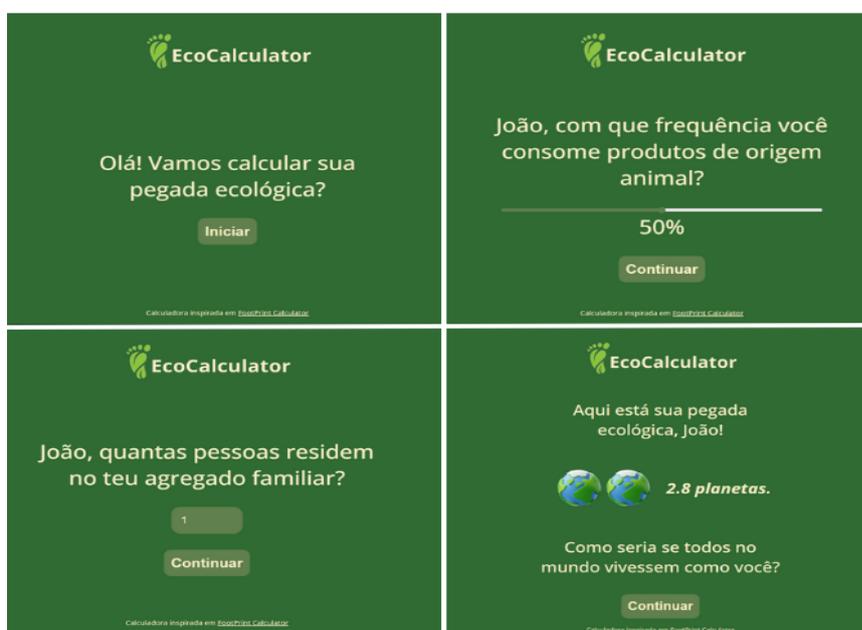
Durante essa primeira metade da capacitação, o acompanhamento dos participantes ocorreu continuamente no momento da capacitação presencial, observando a evolução dos mesmos no laboratório de informática conforme andamento da sequência didática do material. Alguns professores possuíam algum conhecimento nesses temas, uma vez que muitos já haviam feito capacitação online junto à SEED-PR no programa Programação Paraná. Mesmo assim, como são professores concursados em áreas do núcleo básico do ensino fundamental ou médio, diferentes da área de computação, as dificuldades existiam e foram sanadas conforme evolução da sequência didática.

A segunda metade da capacitação ocorreu com conteúdos de JavaScript, para que os participantes pudessem implementar recursos dinâmicos, ou responsividade, na página Web desenvolvida na primeira metade da capacitação. Esse dinamismo permite uma interação do usuário com a página Web, o que cria uma experiência idêntica ao que existe em sites Web comumente encontrados na Internet. Durante essa etapa, buscou-se destacar os principais conceitos de lógica de programação, a fim de consolidar uma boa base dos fundamentos da linguagem JavaScript, por meio do ensino de variáveis, arrays, estruturas condicionais, laços de repetição e objetos. Foram abordados também os métodos do Modelo de Documentos por Objeto (DOM), a fim de reunir todas as ferramentas de tecnologias web trabalhadas ao longo das aulas (HTML/CSS/JavaScript). Por fim, vários exemplos foram direcionados para a ideia do projeto de site Web a ser desenvolvido nessa capacitação: Calculadora de Pegada Ecológica. O projeto de site foi inspirado na página Web *Ecological Footprint Calculator* (FOOTPRINT, 2024) para que os participantes pudessem desenvolver uma aplicação Web real a partir dos conhecimentos adquiridos ao decorrer do curso.

A Figura 4 apresenta 4 capturas de tela resultado da aplicação dos conteúdos desenvolvidos ao longo do curso. Em relação ao HTML, a calculadora de pegada ecológica contempla a estrutura básica de um documento de hipertexto, além de elementos como imagens, parágrafos, inputs e botões. Em relação ao CSS, ela apresenta o posicionamento e o alinhamento dos elementos por meio de FlexBox e a estilização de cores, fontes e tamanhos com o auxílio de propriedades CSS. Por fim, em relação ao JavaS-

cript, essa linguagem é utilizada para fazer a coleta, o armazenamento e o processamento dos dados que o usuário insere nos inputs do formulário, por meio da aplicação dos principais conceitos de JavaScript: captura de elementos e eventos de uma página HTML utilizando os métodos do DOM, armazenamento de dados em variáveis, estruturas condicionais para determinar o algoritmo a ser implementado no cálculo da pegada ecológica, etc.

Figura 4. Capturas de tela da Calculadora de Pegada Ecológica



Fonte: Elaborado pelo autor (2024).

O conteúdo de JavaScript foi o que mais acarretou dificuldades entre os participantes, uma vez que há mais conceitos envolvendo sintaxe e lógica de programação inerentes da linguagem JavaScript, como notação de variáveis e variáveis compostas, estruturas condicionais, estruturas de repetição, funções, bibliotecas, objetos, etc. Dentre esses conceitos, observou-se que ao chegar em estruturas de repetição e vinculação de funções JavaScript com as marcações HTML e objetos da página Web, os participantes tiveram maiores dificuldades. Isso ocorre, pois o nível de abstração de desenvolvimento de código de programação é maior, o que dificulta a análise do que está realmente acontecendo no sistema. Essas dificuldades podem ser sanadas através da prática de desenvolvimento de código por meio de exercícios simples que evoluem em dificuldade para que o indivíduo consiga adquirir o pensamento computacional lógico para seu entendimento (IFTIKHAR, 2022). A adoção de capacitação presencial possibilitou um melhor nivelamento dos participantes, pois houve atividades, passo a passo durante as aulas, para evoluir o desenvolvimento de uma página Web dinâmica.

Considerações finais

A demanda gerada pela inclusão de tecnologias de programação nas escolas do Ensino Fundamental e Ensino Médio na rede pública de ensino do Paraná é uma tendência para estimular o pensamento computacional nas crianças e adolescentes. No entanto, há uma carência de formação dos professores nas escolas, uma vez que é um componente curricular novo, que não consta na formação dos cursos de licenciatura ou pedagogia dos atuais professores da educação básica. Este projeto de extensão em tec-

nologias Web, resultante da parceria entre a UTFPR Apucarana com NRE Apucarana, proporcionou aos professores da educação básica a formação na área de programação, enriquecendo a prática pedagógica no componente curricular de Pensamento Computacional, tendo como foco as competências da BNCC e material didático disponibilizado pela SEED no RCO, abordando a linguagem de programação HTML, CSS e JavaScript.

A escolha do projeto “Calculadora de Pegada Ecológica”, desenvolvido com os professores, foi elaborada por meio de uma sequência didática para atender uma temática direcionada pela SEED e NRE Apucarana sobre sustentabilidade, com o intuito de que os mesmos possam replicar esse aprendizado com seus estudantes nas escolas. Ao término da capacitação, os professores elogiaram a iniciativa e relataram sobre o enriquecimento em suas práticas pedagógicas no componente curricular de Pensamento Computacional em sala de aula.

Esse projeto pretende ser mantido, conforme necessidade dessa demanda, para proporcionar a novas turmas de professores das cidades abrangidas pelo NRE Apucarana a formação em tecnologias Web.

Referências

BEZERRA, M. A.; CARVALHO, A.B.G. Tutoria: Concepções e Práticas na Educação a Distância. In: SOUSA, R.P., et al. **Tecnologias digitais na educação**. Campina Grande: EDUEPB, p. 232-257, 2011.

ECMA, ECMA International. **ECMAScript 2024 Language Specification**. 2024. Disponível em: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>. Acesso em: 08 ago. 2024.

FAEP, **Federação da Agricultura do Estado do Paraná**. 2024. Disponível em: <https://www.sistemafaep.org.br/agrinho/>. Acesso em: 08 ago. 2024.

FLEXBOX. Disponível em: <https://www.w3.org/TR/css-flexbox-1/>. Acesso em: 08 ago. 2024.

FOOTPRINT, **Global Footprint Network**. Disponível em: <https://www.footprintcalculator.org/home/pt>. Acesso em: 08 ago. 2024.

IFTIKHAR, Sidra; GUERRERO-ROLDÁN, Ana-Elena; MOR, Enric. Practice Promotes Learning: Analyzing Students' Acceptance of a Learning-by-Doing Online Programming Learning Tool. **Applied Sciences**, v. 12, n. 24, 2022. Disponível em: <http://dx.doi.org/10.3390/app122412613>. Acesso em: 08 ago. 2024.

MEC. **Informática Aplicada a educação** – 2007. Disponível em: http://portal.mec.gov.br/seb/arquivos/pdf/profunc/infor_aplic_educ.pdf. Acesso em: 8 ago. 2024.

MEC1. **Normas sobre Computação na Educação Básica** – Complemento à Base Nacional Comum Curricular (BNCC). Disponível em: http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=235511-pceb002-22&category_slug=fevereiro-2022-pdf&Itemid=30192. Acesso em: 08 ago. 2024.

MEC2. **Computação**: complemento à BNCC. Disponível em: <http://portal.mec.gov.br/docman/fevereiro>

-2022-pdf/236791-anexo-ao-parecer-cneceb-n-2-2022-bncc-computacao/file. Acesso em: 08 ago. 2024.

MOODLE, **Modular Object-Oriented Dynamic Learning Environment**. Disponível em: <https://moodle.org>. Acesso em: 08 ago. 2024.

RCO. **Registro de Classe Online**. Disponível em: https://professor.escoladigital.pr.gov.br/rco_mais_aulas. Acesso em: 08 ago. 2024.

SEED-PR, Programação Paraná. Disponível em: <https://www.educacao.pr.gov.br/programacao>. Acesso em: 08 ago. 2024.

VSCODE. **Visual Studio Code**. 2024. Disponível em: <https://code.visualstudio.com/>. Acesso em: 08 ago. 2024.

W3C1, World Wide Web Consortium. 2024. **HTML Standard**. Disponível em: <https://www.w3.org/TR/html/>. Acesso em: 08 ago. 2024.

W3C2, World Wide Web Consortium. 2024. **Cascading Style Sheets Standard**. Disponível em: <https://www.w3.org/Style/CSS/>. Acesso em: 08 ago. 2024.

W3C3, World Wide Web Consortium. 2024. **Document Object Model (DOM)**. Disponível em: <https://www.w3.org/DOM/>. Acesso em: 08 ago. 2024.

Recebido em 09 de dezembro de 2024.

Aceito em 17 de janeiro de 2025.