

DESENVOLVEDOR FULL STACK

CONTEÚDO DA REVISÃO

DIA 1 LÓGICA	
Tema	Abordagem
Variáveis, constantes e tipos de dados (20 minutos)	<p>Abordar os conceitos de variáveis e constantes como forma de armazenar dados. Com uso da ferramenta JSFiddle (https://jsfiddle.net/) ou outra ferramenta online para execução de JavaScript.</p> <p>Abordar os diferentes tipos de dados que uma variável ou constante podem assumir, e suas finalidades. Demonstre a concatenação de uma frase com os nomes dos alunos e insira o tema da aula seguinte com demonstrações iniciais.</p>
Operadores aritméticos e de atribuição (20 minutos)	<p>Após isto, aborde a ordem de precedência matemática dos operadores para resolução das expressões, que é a seguinte:</p> <ul style="list-style-type: none">1º parêntesis2º potências e raízes3º multiplicação e divisão4º soma e subtração
Operadores relacionais (20 minutos)	<p>Apresentar aos alunos os conceitos dos operadores relacionais (igual, diferente, menor, maior, menor ou igual e maior ou igual) e seus usos nas expressões.</p>
Operadores lógicos (20 minutos)	<p>Abordar os operadores lógicos (E, OU e de NEGAÇÃO) e seus usos nas expressões. Aborde e realize experimentos também com os operadores de igualdade (==), desigualdade (!=) e idênticos (===).</p>
Estruturas condicionais (40 minutos)	<p>Abordar o conceito da estrutura condicional SE, ENTÃO e SENÃO (IF, THEN, ELSE) e suas estruturas aninhadas (ELSE IF).</p> <p>Abordar também, após isto, a estrutura SWITCH.</p>
Exercícios e prática (120 minutos)	<p>Praticar exercícios de diferentes níveis abordando os conteúdos do dia, iniciando com uma complexidade mais baixa para mais alta.</p> <p>https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascript/</p>

Desafio (40 minutos)	<p>Proposta de desafio: os alunos devem elaborar um experimento que implemente uma calculadora, que deve realizar as operações de soma, subtração, multiplicação ou divisão, de acordo com escolha do usuário.</p> <p>A calculadora deve considerar o resultado da operação como valor inicial da operação seguinte. A calculadora deve repetir estas instruções até que o operador (na etapa 2) seja informado com a palavra "fim":</p> <ol style="list-style-type: none"> 1. solicitar valor 2. solicitar operador 3. solicitar valor 4. informar resultado da operação 5. retornar para etapa 2. <p>O valor da etapa 1 será sempre o resultado da operação da etapa 4.</p> <p>O gabarito deste desafio está neste endereço: https://jsfiddle.net/codeforfree/dgt7asf6/ </p>

DIA 2 LÓGICA

Tema	Abordagem
Operadores condicionais e loops (60 minutos)	Reforçar o entendimento das estruturas condicionais IF, ELSE e CASE/SWITCH. Em seguida, abordar as estruturas de repetição FOR, WHILE e DO-WHILE.
Funções e parâmetros (60 minutos)	Hora de apresentar que as instruções que vimos até agora podem ser agrupadas em blocos. Abordar o conceito de funções e seu reuso. Demonstre também que as funções podem ser iniciadas com variáveis que podem ser recebidas como parâmetro, e que podem também ter um valor retornado.
Exercícios e prática (120 minutos)	Praticar exercícios de diferentes níveis abordando os conteúdos do dia, iniciando com uma complexidade mais baixa para mais alta. https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascript/
Desafio (60 minutos)	Desafio em grupo: o grupo deve criar uma estrutura de repetição que imprima os números primos que existem entre 0 e 100, utilizando uma função. Desafio individual: o aluno deve criar uma estrutura de loops FOR aninhados. A proposta é criar um experimento para exibir o fatorial dos números de 0 a 10. O fatorial de um número é calculado pela multiplicação desse número por todos os seus antecessores até chegar ao número 1. Por exemplo, 3! (lê-se: fatorial de 3) é igual a "3 x 2 x 1", que é igual a 6.

DIA 3 HTML	
Tema	Abordagem
Introdução (20 minutos)	<p>Inicie abordando o conceito da Linguagem de Marcação para mais próximo da realidade do aluno fazendo uma associação com o aplicativo de mensagens WhatsApp: para se colocar um trecho do texto em negrito ou itálico, por exemplo, é colocado um * ou um _ antes e depois do texto que se deseja "marcar", respectivamente.</p> <p>Por exemplo, se for enviado via Whatsapp o texto <i>*informação importante!*</i> , este mesmo texto será exibido no destinatário de forma destacada: informação importante!</p> <p>Esta é uma <i>linguagem de marcação bem simples</i>, criada para que o usuário do aplicativo possa enviar o texto com marcações, e estas marcações devem ser processadas na visualização pelo aplicativo do destinatário, assim como o HTML.</p> <p>No caso do Whatsapp, o marcador é igual para abertura e fechamento. Neste exemplo:</p> <p><i>*Este texto está em negrito e dentro dele tem um em _itálico_, que legal*</i></p> <p>O resultado será: Este texto está em negrito e dentro dele tem um em <i>itálico</i>, que legal</p> <p>Estes marcadores são chamados de tags e funcionam da mesma forma, a diferença é que o HTML possui uma diferença na tag de fechamento <code><tag>conteúdo</tag></code>, pois ela permite encadeamento (mesma tag dentro de uma igual).</p> <p>Complete a introdução explicando a estrutura do HTML (HTML Page Structure) do link https://www.w3schools.com/html/html_intro.asp.</p>
Principais Tags (30 minutos)	<p>Com o entendimento da estrutura e das regras estabelecidas, os seguintes tópicos devem ser abordados ao longo da aula, seguindo o W3Schools:</p> <ul style="list-style-type: none"> • Attributes • Headings • Paragraphs • Formatting

Listas (20 minutos)	<p>Abordar as tags de criação de listas ordenadas e listas desordenadas. Utilizar o link do W3Schools para exemplos e execuções: https://www.w3schools.com/html/html_lists.asp.</p>
Blocos (20 minutos)	<p>Abordar os elementos HTML div e span de forma que fique claro para o aluno seus usos como elementos HTML em nível de bloco e de linha.</p> <p>Inicialmente, utilize os links abaixo que aborda as tags sem o uso de CSS:</p> <p>https://www.w3schools.com/html/html_blocks.asp https://www.w3schools.com/tags/tag_div.asp https://www.w3schools.com/tags/tag_span.asp</p>
Formulários (30 minutos)	<p>Expor o conceito de formulário e como eles oferecem interação do usuário com uma página de internet para envio de dados a um servidor remoto.</p> <p>Como forma de exposição dos elementos básicos input e button de um formulário HTML, apresente um formulário simples de envio de contato, demonstrando para o aluno todo o processo de desenvolvimento, desde o planejamento até a codificação.</p> <p>Para exemplificar o envio de dados utilizando os métodos HTTP get e post, explore e explique aos alunos os formulários e o funcionamento do envio e retorno fornecido pelo servidor, contidos neste link: https://jsfiddle.net/codeforfree/dsuhqyz9/</p> <p>Elementos de formulário HTML que deverão ser abordados:</p> <ul style="list-style-type: none"> • input • select • option • textarea • button • datalist • output <p>Aborde todos os atributos dos elementos, com destaque para o atributo input e seus tipos (<i>type</i>). Como referência, utilize as informações do W3Schools em https://www.w3schools.com/html/html_form_input_types.asp</p>
Exercícios e prática (120 minutos)	<p>Elaborar exercícios com diferentes níveis abordando os conteúdos do dia, iniciando com uma complexidade mais baixa para mais alta.</p>

Desafio

(60 minutos)

Elaborar exercícios com diferentes níveis abordando os conteúdos do dia, iniciando com uma complexidade mais baixa para mais alta.

DIA 4 CSS	
Tema	Abordagem
Introdução (20 minutos)	<p>Aborde sobre os estilos que os elementos HTML podem ter, como cores, tamanhos, alinhamentos etc. Apresente o CSS como a linguagem de estilos e como pode ser inserido na propriedade style em qualquer elemento.</p> <p>Apresente também as outras 2 formas de uso do CSS: <i>internal</i> e <i>external</i> CSS, sendo o interno entre as tags styles e o externo como link para um arquivo ".css", ambos dentro da tag head.</p> <p>Utilize o código no link https://jsfiddle.net/codeforfree/9fx05ct3/ para mostrar aos alunos as possibilidades de uso do CSS: classes, id, elemento ou na propriedade style dos elementos.</p> <p>Aborde e explique aos alunos cada uma destas formas.</p>
Bordas, margens, padding, altura e largura (40 minutos)	<p>Demonstre e realize experimentos com os tipos de borda e espessuras, assim como margens, padding, altura e largura.</p> <p>Utilize o link https://www.w3schools.com/css/tryit.asp?filename=trycss_boxmodel e estimule os alunos a alterar seu os números e a redimensionar a janela, alterando entre pixel e percentual.</p> <p>Explique a diferença entre uso de pixel e percentual para altura, largura, margem e padding.</p> <p>Explique também o uso de border: espessura + estilo + cor, e realize experimentos com a definição de um dos 4 lados possíveis de border, margin e padding (-top, -bottom, -left e -right).</p>
Cores (20 minutos)	<p>Aborde neste momento os conceitos de cores, as cores nomeadas, o hexadecimal e RGB.</p> <p>Utilize o link a seguir como guia: https://www.w3schools.com/css/css_colors.asp.</p> <p>Demonstre aos alunos que as cores podem ser aplicadas a fontes, fundos, bordas dos elementos, inclusive do elemento body.</p>
Display, Position, float e overflow (40 minutos)	<p>Explore com os alunos a possibilidade de ocultar os elementos utilizando o display.</p> <p>Aborde com os alunos a importância deste tema. Comente sobre tableless e o papel da div no desenvolvimento de interfaces web. Comente também sobre como a capacidade de posicionamento e flutuação dos elementos possibilitaram a</p>

	criação de interfaces ricas e frameworks responsivos e dinâmicos, com efeitos de drag-and-drop, por exemplo.
Exercícios e prática (120 minutos)	Elaborar exercícios com diferentes níveis abordando os conteúdos do dia, iniciando com uma complexidade mais baixa para mais alta.
Desafio (60 minutos)	Elaborar exercícios com diferentes níveis abordando os conteúdos do dia, iniciando com uma complexidade mais baixa para mais alta.

DIA 5

UI/UX E BOOTSTRAP

Tema	Abordagem
UI/UX (20 minutos)	<p>Comente sobre identidade visual, usabilidade, responsividade e desenhos de interface.</p> <p>Recomende aos alunos assistiam ao vídeo que demonstra algumas das diferenças entre UX e UI: https://www.youtube.com/watch?v=hQ1rpJKyj68</p>
O Framework Bootstrap (10 minutos)	<p>O professor pode agora abordar o Bootstrap como um dos frameworks mais utilizados na atualidade, criado pela equipe do Twitter, e que une as tecnologias HTML, CSS e JavaScript para fornecer de forma simples uma forma mais rápida e prática de criar interfaces padronizadas, responsivas e compatíveis com todos os navegadores, se adaptando a todo tipo de tela, principalmente as de dispositivos móveis.</p>
Container (20 minutos)	<p>Iniciar a abordagem apresentando a estrutura inicial do Bootstrap e sua forma de organização orientado a classes CSS e com estrutura baseada em blocos (div e span).</p> <p>Aborde, em seguida, como primeira classe Bootstrap a sua estrutura inicial: o container, que pode organizar todo o restante do conteúdo em seu interior.</p> <p>Apresente também a sua variação, o container-fluid. O link a seguir mostra uma comparação entre estes dois tipos de container: https://jsfiddle.net/codeforfree/zfqd1n8j/.</p> <p>Estimule os alunos a acessar este link e redimensionar a janela para perceber a diferença entre eles.</p>
Grids (20 minutos)	<p>Abordar como o Bootstrap divide a tela em grids em linhas (row) e sua divisão principal com 12 colunas, que podem ser organizadas de forma simplificada.</p> <p>Abordar também que as colunas são divididas em 4 classes diferentes, que se adaptam ao tamanho da tela de acordo com a classe escolhida: xs, sm, md e lg (eXtra Small devices, SMal devices, Medium Devices e Large Devices, respectivamente).</p> <p>Utilizar o link e código em https://www.w3schools.com/bootstrap/bootstrap_grid_basic.asp e realize experimentos com os alunos, alternando entre as classes de colunas (xs, sm, md e lg) e redimensionando a largura da janela. Estimule os alunos a observar que as colunas se empilham em linhas diferentes quando o tamanho da janela alcança uma determinada largura, dependendo da classe definida.</p>

<p>Cores e alinhamento</p> <p>(30 minutos)</p>	<p>Explique que o Bootstrap possui uma identidade visual com cores padrão para que não tenhamos que nos preocupar com que código de cores estamos utilizando, e que estas cores estão divididas em contextos:</p> <ul style="list-style-type: none"> • important • success • information • warning • danger <p>Comente que estes contextos estão presentes em todo o tipo de elemento que for visto sobre o framework.</p> <p>Estimule os alunos a acessarem os links www.w3schools.com/bootstrap/tryit.asp?filename=trybs_txt_colors&stacked=h e www.w3schools.com/bootstrap/tryit.asp?filename=trybs_txt_bgcolors&stacked=h para visualizar o funcionamento das classes nas cores e fundo dos textos, respectivamente.</p> <p>Além das cores, o Bootstrap fornece também classes para tratamento do alinhamento e organização do texto, que pode ser visto neste link: www.w3schools.com/bootstrap/tryit.asp?filename=trybs_ref_text-left.</p> <p>Estimule os alunos a alterar o tamanho da janela e verificar o comportamento do texto.</p>
<p>Jumbotron, wells e alertas</p> <p>(20 minutos)</p>	<p>Abordar as classes jumbotron e wells que também auxiliam na criação de estrutura de cabeçalhos e detalhes de conteúdo, respectivamente.</p> <p>Em seguida, abordar sobre a classe alert e seu uso com a identidade visual baseada nos contextos abordados na aula anterior. Mostre também sobre o uso da classe alert acrescentando o botão para fechamento e sua simplicidade. Comente com os alunos que nenhum desenvolvimento adicional de manipulação de HTML foi necessária. Utilize o link https://www.w3schools.com/bootstrap/bootstrap_alerts.asp para demonstração deste conteúdo.</p>
<p>Tabelas</p> <p>(10 minutos)</p>	<p>Aborde como o Bootstrap influencia o componente table do HTML. Estimule os alunos alterar os códigos e a remover a classe e comparar o efeito realizado sobre a tabela e seu resultado, com as classes:</p> <ul style="list-style-type: none"> • table • table-striped • table-bordered • table-hover • table-condensed • table-responsive <p>Utilize o link w3schools.com/bootstrap/bootstrap_tables.asp para abordar este conteúdo.</p>

Formulários e botões (20 minutos)	<p>Mostre como fica a estrutura do formulário e estimule os alunos a redimensionar o tamanho da janela para visualizar sua responsividade. Utilize os seguintes links para abordar este conteúdo:</p> <ul style="list-style-type: none"> • www.w3schools.com/bootstrap_forms • www.w3schools.com/bootstrap_forms_inputs • www.w3schools.com/bootstrap_forms_inputs2 <p>Aborde também sobre o agrupamento de botões. Utilize os seguintes links para isto:</p> <ul style="list-style-type: none"> • www.w3schools.com/bootstrap_buttons • www.w3schools.com/bootstrap_button_groups
Exercícios e prática (90 minutos)	<p>Estimule os alunos a realizar experimentos com outras classes do Bootstrap. Indique que pesquisem o site do framework e estudem suas possibilidades no link: https://getbootstrap.com/docs/3.3/components/</p> <p>Exercício #1: reconstruir uma página de notícia do site do G1, copiando seu estilo de título, autor e texto de conteúdo.</p> <p>A fonte de consulta do estilo visual e os textos para sua construção podem ser no link: https://justpaste.it/desafioBootstrapEstilo</p>

DIA 6 Javascript	
Tema	Abordagem
Tag script (20 minutos)	<p>Apresente para os alunos a tag <script> e onde o JavaScript pode ser executado:</p> <ul style="list-style-type: none"> • tag script na tag head: <pre><head><script>alert ("oi");</script></head></pre> <ul style="list-style-type: none"> • tag script na tag body: <pre><body><script>alert ("oi");</script></body></pre>

	<ul style="list-style-type: none"> • tag script para arquivo externo: <pre><script src="/js/hello.js"></script></pre> <ul style="list-style-type: none"> • diretamente no evento do elemento: <pre></pre>
Funções e eventos (20 minutos)	<p>Faça uma revisão com os alunos sobre as funções, parâmetros e retorno. Em seguida, aborde sobre os eventos javascript oriundos dos elementos HTML, principalmente:</p> <ul style="list-style-type: none"> • onload • onclick • onchange • onmouseover • onmouseout • onkeydown
Funções de string (20 minutos)	<p>Aborde as funções de string e seu uso com exemplos práticos. A principais funções que devem ser apresentadas são:</p> <ul style="list-style-type: none"> • length • indexOf • slice • substring • substr • replace • toUpperCase • toLowerCase • Split
Arrays (30 minutos)	<p>Para introduzir o assunto de Arrays, inicie explorando o comando split já abordado, utilizando exemplos usuais, explicando como este comando parte uma string inicial e a transforma em uma cadeia de strings, ou array de strings.</p> <p>seguir: https://jsfiddle.net/codeforfree/kuvmb5pf/</p> <p>Neste link é demonstrado que uma string comporta-se como um “array de caracteres”, utilizando alguns dos mesmos métodos normalmente utilizados em variáveis do tipo array.</p>
Iteração de arrays (30 minutos)	<p>Aborde com os alunos os métodos de ordenação de arrays, que são os seguintes com estas particularidades:</p> <ul style="list-style-type: none"> • forEach: executa uma função para cada item do array, sem retornar um novo array; • map: executa uma função para cada item, porém retorna um novo array; • filter: igual ao map, mas só retorna os itens aprovados no critério estipulado, eliminando os reprovados;

	<ul style="list-style-type: none"> ● reduce: passa por todos os itens acumulando os resultados anteriores, retornando ao final um único resultado. <p>Estas 4 funções são a base para utilização eficiente de tecnologias web e manipulação de dados. Lembre de comentar a importância disto para a vida do profissional. Outro ponto é que o professor deve atentar que cada uma das funções requerem uma função de callback com diferentes atributos. Explore isto com os alunos sempre alertando para este detalhe.</p> <p>Estas funções introduzem um temas muito importantes para o JavaScript e qualquer outra linguagem, uma delas é o uso de funções de callback. Explore com os alunos o funcionamento da função de callback com o script do link https://jsfiddle.net/codeforfree/sm2yu8q0/ e certifique-se de que seu entendimento tenha sido minimamente absorvido.</p>
Exercícios e prática (120 minutos)	Elaborar exercícios com diferentes níveis abordando os conteúdos do dia, iniciando com uma complexidade mais baixa para mais alta.
Desafios (60 minutos)	<p>Desafio 1:</p> <p>Criar uma interface de cadastro de livros e uma saída de referência bibliográfica no padrão da ABNT.</p> <p>O formulário deve conter os campos:</p> <ul style="list-style-type: none"> ● nome do autor: Eça de Queiróz ● título: O primo Basílio ● número da edição: 25 ● local de publicação: Rio de Janeiro ● nome da editora: Ediouro ● ano de publicação: 1878 <p>Por fim, o programa deve apresentar a seguinte saída:</p> <pre>QUEIRÓZ, E. O Primo Basílio: 25. ed. Rio de Janeiro: Ediouro, 1878</pre> <p>O gabarito com uma solução deste desafio pode ser visto neste link: https://jsfiddle.net/codeforfree/zydvnrUL/</p>

Desafio 2:

O desafio consiste em reproduzir a seguinte saída:

```
array_de_numeros = 3,7,9,11,19,23,34  
soma dos valores = 209  
array_de_resultados = 3,7,9,11,19
```

Para isto, o aluno deve criar uma variável do tipo array com alguns elementos numéricos e exibir as seguintes saídas, representadas por cada uma das linhas acima, na mesma ordem:

- o conteúdo do array inicial;
- a soma dos números do array inicial;
- um novo array apenas com os números que, multiplicados por 10, sejam menores que a soma do array inicial.

Neste exemplo, os itens 23 e 34 do array, quando multiplicados por 10, são maiores que o resultado da soma de todos, que é 209. O gabarito deste desafio pode ser verificado neste link: <https://jsfiddle.net/codeforfree/aweuhty3/>

DIA 7 Javascript	
Tema	Abordagem
Funções nativas (30 minutos)	<p>Utilize esta aula para abordar algumas das funcionalidades do Javascript muito comuns para o dia-a-dia, tais como:</p> <ul style="list-style-type: none"> • Datas <ul style="list-style-type: none"> ◦ Formatos ◦ Métodos get e set • O objeto Math <ul style="list-style-type: none"> ◦ Random ◦ Abs ◦ Ceil ◦ Floor ◦ Sqrt ◦ Pow • Debug <ul style="list-style-type: none"> ◦ Uso do console do browser ◦ console.log
JSON (30 minutos)	<p>Nesta etapa deve ser apresentado o JSON, ou <i>JavaScript Object Notation</i>. Professor, evidencie a importância do JSON no universo do desenvolvimento de sistema e como foi adotado por praticamente todas as linguagens e tecnologias como padrão ou linguagem única para troca de dados. Mostre ainda que, inclusive, existem sistemas de banco de dados que são inteiramente baseados em JSON, como o MongoDB e o FireBase. E mais: que toda esta versatilidade e universalidade do JSON nasceu a partir do JavaScript.</p> <p>JSON é um objeto. Realize experimentos que demonstre isto utilizando o parsing de dados de string para JSON. Utilize o script em w3schools.com/js/tryit.asp?filename=tryjs_json_parse.</p> <p>Utilize a ferramenta online https://jsonformatter.org/ para demonstrar aos alunos a possibilidade de formatar e converter o JSON em outros formatos, como o XML e o CSV. Aproveite para abordar também estes dois outros formatos utilizados para troca de dados, apontando as semelhanças entre o JSON e o XML. Utilize o link sample da ferramenta para mostrar exemplo de dados e seus conteúdos, que podem ser, inclusive, arrays.</p>
Exercícios e prática (120 minutos)	<p>Elaborar exercícios com uso intenso de <i>arrays</i> e manipulação de <i>JSON</i>.</p>

Desafios

(120 minutos)

Desafio 1:

O desafio consiste em, partindo de uma estrutura de dados em JSON, criar uma tabela com os dados no formato específico esperado como resultado.

Os dados JSON e o formato da tabela estão no seguinte link:

<https://justpaste.it/desafioJSON>.

O gabarito para este desafio pode ser visto neste link:

<https://jsfiddle.net/codeforfree/L2uqr3pd/>

Desafio 2:

Este desafio final consiste numa expansão do desafio anterior. O JSON contém dados de atividades de projeto com prazo de conclusão de cada uma, e que novamente deve ser criado a tabela. Porém, o prazo está em dias, e deve ser exibido na tela a data atual acrescida deste prazo.

O formato da data também é específico para nosso formato usual, ou seja, dia/mês/ano. A data do prazo deve ser sinalizada com cores de acordo com uma regra específica.

Estas regras e a tabela de resultado esperada estão neste link:

<https://justpaste.it/desafioFinalJAVASCRIPT>

O gabarito deste desafio pode ser visto neste link:

<https://jsfiddle.net/codeforfree/h60tgysx/>

DIA 8 React	
Tema	Abordagem
Arquitetura (30 minutos)	<p>Apresente o aplicativo inicial do React e realize um overview com os alunos de sua estrutura utilizando o do link a seguir: https://stackblitz.com/edit/react-7y9vcj.</p> <p>Utilize sempre como guia a documentação oficial do React: https://pt-br.reactjs.org/docs/hello-world.html</p>
JSX (30 minutos)	<p>Antes de abordar sobre o JSX, apresente aos alunos sobre o XML, eXtensible Markup Language. Utilize o link https://www.w3schools.com/xml/default.asp para abordar este conteúdo.</p> <p>Com o alinhamento do entendimento sobre a estrutura do XML, apresente o JSX como uma expansão do HTML/XML somando todo o poder do JavaScript, podendo realizar expressões e acessos diretamente deste conteúdo.</p> <p>Utilize a documentação oficial para exposição deste conteúdo, através do link https://pt-br.reactjs.org/docs/introducing-jsx.html.</p>
Renderizando elementos (30 minutos)	<p>Aborde com os alunos como o ReactJS renderiza o conteúdo no navegador. Explique o conceito de DOM do HTML, utilize para isto o link https://www.w3schools.com/xml/dom_intro.asp para abordar este assunto.</p> <p>Realize experimentos para explicar o funcionamento do React e seu funcionamento para renderizar os elementos. Utilize o link da documentação oficial para apresentar este assunto: https://pt-br.reactjs.org/docs/rendering-elements.html. Utilizando os recursos da documentação, mostre que mesmo ao atualizar uma estrutura inteira de elementos, apenas o elemento alterado é renderizado no DOM.</p>
Componentes e props (45 minutos)	<p>Explore com os alunos os conceitos de componentes e como as informações podem ser passadas como parâmetros: os props. Utilize o link da documentação oficial para abordar o assunto: https://pt-br.reactjs.org/docs/components-and-props.html.</p> <p>Solicite que os alunos estudem a documentação oficial específica e detalhada sobre componentes, retornando com dúvidas e implementações realizadas como experimentos, através do endereço https://pt-br.reactjs.org/docs/react-component.html.</p>

<p>State e ciclo de vida</p> <p>(45 minutos)</p>	<p>Aborde com os alunos o conceito de state e sua forma de instânciação no construtor da classe, além de sua forma de acesso no JSX.</p> <p>Mostre para os alunos o funcionamento do ciclo de vida e sua aplicação. Utilize para esta abordagem o conteúdo e experimentos contidos na documentação oficial do ReactJS com o link https://pt-br.reactjs.org/docs/state-and-lifecycle.html.</p>
<p>Desafios</p> <p>(120 minutos)</p>	<p>Desafio 1:</p> <p>Proponha aos alunos realizar o desafio partindo <i>Olá Mundo</i> exibido anteriormente (https://stackblitz.com/edit/react-7y9vcj), onde deverá ser alterado da seguinte forma:</p> <ul style="list-style-type: none"> • na classe principal App, devem ser adicionados 2 novos métodos: getTitulo e getParagrafo. • getTitulo deve receber um parâmetro de texto e retornar um JSX com o texto envolvido pela tag h1. • getTitulo deve receber dois parâmetros: nome e texto. O parâmetro nome deve ser envolvido pela tag de negrito (b), seguido do parâmetro texto, então a função deve retornar um JSX ambos envolvidos pela tag p. <p>A imagem com o resultado esperado do desafio proposto pode ser visto no link https://justpaste.it/desafioJSX.</p> <p>Já o gabarito para este desafio pode ser visto neste link: https://stackblitz.com/edit/react-fuh1tg.</p> <p>Desafio 2:</p> <p>Como desafio, proponha aos alunos realizar uma alteração no desafio do dia anterior, modificando o projeto para utilizar componentes e props.</p> <p>O resultado esperado é exatamente o mesmo, que pode ser revisitado no link https://justpaste.it/desafioJSX.</p> <p>Os alunos devem, neste desafio, alterar o projeto e construir 2 novos componentes: <i>CursoHeader</i> e <i>CursoContent</i>. O primeiro cria o título e o segundo criando as linhas seguintes, ambos recebendo os dados por props.</p> <p>O gabarito deste desafio pode ser visto neste link: https://stackblitz.com/edit/react-tjenj2.</p>

Para melhor entendimento e explicação, os novos componentes foram mantidos no mesmo arquivo do componente principal. Porém, a boa prática requer que as classes estejam em arquivos separados, mostre esta modularização enfatizando o uso de ***import*** e ***export*** através do código deste link: <https://stackblitz.com/edit/react-awhxtr>.

DIA 9 React	
Tema	Abordagem
Manipulando eventos (45 minutos)	<p>Para abordar este tema, utilize a documentação oficial do ReactJS como guia através do link https://pt-br.reactjs.org/docs/handling-events.html.</p> <p>Aborde e realize experimentos também com os eventos sintéticos. Utilize o link da documentação oficial como guia: https://pt-br.reactjs.org/docs/events.html.</p>
Renderização condicional (45 minutos)	<p>Aborde com os alunos as formas de renderização possíveis com o ReactJS. Para isto, continue utilizando a documentação oficial usando o link https://pt-br.reactjs.org/docs/conditional-rendering.html.</p> <p>Motive os alunos a executar todos os experimentos existentes e use-os para abordar e fixar os conceitos.</p>
Exercícios e prática (120 minutos)	<p>Executar exercícios para fortalecer a prática e conhecimento dos conteúdos abordados.</p>
Desafios (90 minutos)	<p>Este desafio consiste em, partindo do projeto do desafio anterior, os alunos devem refatorar o projeto incluindo um botão no topo da página, que deve alternar seu texto entre "Português" e "English".</p> <p>Quando o texto do botão estiver em "Português", o texto no título e nas linhas 1 e 2 seguintes deverão estar também em português. Quando estiver em "English", estes textos deverão estar em inglês.</p> <p>O link a seguir mostra o resultado esperado e fornece a estrutura com os textos em ambos idiomas: https://justpaste.it/desafioReactCondicional.</p> <p>O gabarito para este desafio pode ser visto no link https://stackblitz.com/edit/react-2aefij.</p>

DIA 10 React	
Tema	Abordagem
Listas e chaves (45 minutos)	Aborde com os alunos como o ReactJS renderiza as listas de conteúdo e suas chaves. Utilize a documentação oficial e seus experimentos contidos no link https://pt-br.reactjs.org/docs/lists-and-keys.html .
Forms (45 minutos)	Aborde com os alunos como o ReactJS renderiza os elementos de formulário. Utilize a documentação oficial e seus experimentos contidos no link https://pt-br.reactjs.org/docs/forms.html .
Routes (90 minutos)	Aborde com os alunos como o ReactJS utiliza o roteamento das páginas. Utilize a documentação oficial e seus experimentos contidos no link https://pt-br.reactjs.org/docs/code-splitting.html#route-based-code-splitting
Exercícios e prática (60 minutos)	Executar exercícios para fortalecer a prática e conhecimento dos conteúdos abordados.
Desafios (60 minutos)	<p>O desafio deste dia consiste em, partindo do resultado do desafio do dia anterior, refatorar o código para colocar os textos em array, e renderizar os itens com os conceitos aprendidos neste dia.</p> <p>O formato esperado de saída e os textos adicionais podem ser encontrados neste link: https://justpaste.it/desafioReactLista.</p> <p>O gabarito deste desafio pode ser visto no link a seguir: https://stackblitz.com/edit/react-pzsihf.</p>