

ESCOLA E FACULDADE DE TECNOLOGIA SENAI ROBERTO MANGE

DESENVOLVIMENTO DE SISTEMAS

JULHO DE 2022

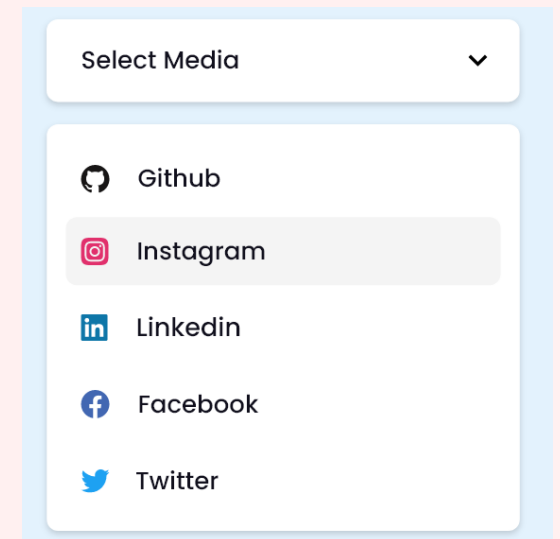


ESCOLA SENAI "ROBERTO MANGE"
UMI. REAGRIZADO DA INDÚSTRIA

FRONTEND

EX. 1 – Treinando Nuxt Components e SCSS

- *Crie um projeto em Nuxt onde você terá que desenvolver um componente que conterà um select e uma label, onde você terá que passar para este componente algumas propriedades como: nome da label, as opções existentes no select (tag option do html), e mais outras propriedades que julgar conveniente com seu componente.*
- *Não se esqueça de passar o model para quem usar o componente.*
- *Estilize o componente à sua escolha (não vale usar componente pronto do primevue)*
- *Faça o teste usando este componente em uma página e passando todas as props necessárias, imprimindo o resultado model vindo deste componente na tela.*
(Dica, veja o componente CustomInput que criamos no projeto SGE)



FRONTEND

EX. 2 – Treinando Nuxt Components, SCSS e Typescript

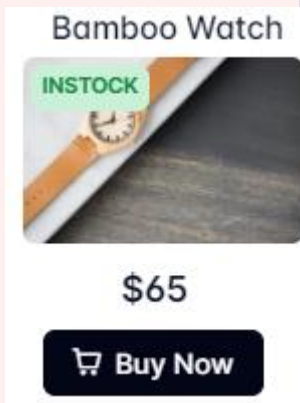
- Crie um componente que contenha dois inputs (e duas labels), e que faça a operação matemática de multiplicação, divisão, soma, subtração e raiz quadrada dos números que forem colocados nestes inputs, enviando o resultado para o model do mesmo.
- Este componente deve poder fazer qualquer uma das operações matemáticas citadas, porém deve receber de quem usa o componente qual a operação desejada via props.
- Também deverá ter uma prop que vai mudar o estilo do componente (dark ou light mode)
- Faça o teste em uma página usando este componente e crie um botão para cada operação matemática, possibilitando com que o usuário selecione qual deseja efetuar, imprimindo o resultado em tela, além também do botão dark ou light mode (o botão de dark/light mode use o InputSwitch do primevue!)

<https://primevue.org/inputswitch/>

FULL STACK

EX. 3 – Treinando Nuxt, SCSS, Typescript

- *Crie uma aplicação Nuxt que fará algumas operações básicas de uma loja virtual;*
- *Neste loja crie uma página que mostrará cards dos produtos a serem vendidos (cada card deve conter a foto do produto, nome, preço, quantidade disponível e botão de compra)*
- *Perceba que terá que fazer um card com essas informações, então crie-o como um componente.*
- *Nesta página dos produtos, como ainda não temos backend, você deve construir um Array de objetos em seu script para listar os produtos disponíveis e assim fazer um v-for da quantidade de cards a serem mostrados.*
- *Haverá um botão de carrinho de compras na parte de cima da página, e cada vez que for clicado o botão comprar de um item será adicionada na variável do carrinho*



FULL STACK

Continuação EX. 3 – Treinando Nuxt, SCSS, Typescript

- **Ao se clicar no botão de carrinho de compras vai ser direcionado para outra página onde listará em uma tabela os itens do carrinho e no fim da tabela o valor total a ser pago.**
- **No final de cada linha da tabela deve ter um botão de excluir item da lista e um input com a possibilidade de alterar a quantidade (estas operações devem atualizar automaticamente o valor total na tela)**
- **Haverá um botão de concluir compra na tela de carrinho de compras que ao ser clicado envia uma mensagem de compra concluída e imprime no console o resultado final da compra.**
- **Use o componente Toast do primevue para exibir a mensagem (se não conseguir pode usar um alert mesmo) <https://primevue.org/toast/>**

FULL STACK

EX. 4 – Treinando Django e integração full stack

- No exercício 3 não possuíamos um backend, agora crie um backend simples que contenha a lista dos produtos e as compras efetuadas após enviado o carrinho para compra;***
- Faça as alterações no frontend onde havia a lista dos produtos em variável e agora obtenha a lista dos produtos via backend;***
- Após submetida a compra no carrinho, salve no backend.***