



PROJETO EM GRUPO

Módulo 2 – The
Frontenders



Todos os direitos reservados
©2022 Resilia Educação

RESILIA |  Senac

CONTEXTO



Agora é a hora de brilhar em equipe e colocar tudo o que vocês aprenderam num único projeto! Você e sua squad deverão **desenvolver o front end** para **apresentar um novo produto** lançado pela empresa FrontEnders.

Vocês podem inventar o produto que quiserem. Pensem em algo que seria legal para ajudar no seu dia a dia com programação. Que tal um bot que vai sugerir boas práticas de CSS? Ou então um app que valida seu layout e atribui uma nota de 0 a 10? Vocês estão livres para inventar qualquer produto, por mais incomum que seja!



O QUE É PARA FAZER?

Desenvolver o front end para apresentar um novo produto lançado pela empresa FrontEnders.

Apresentação deve ser feita como se fosse uma simulação de apresentação de case em dinâmicas em grupo!

Portanto, ela deverá conter as técnicas de pitch e postura profissional voltadas para a venda de seu case em uma dinâmica!

COMO FAZER?



Detalhes do Projeto

Todas as páginas devem utilizar o mesmo cabeçalho com menu de navegação e rodapé.

⇒ Página inicial (home):

- Deve conter breve descrição do que o serviço se propõe a fazer;
- Deve conter uma chamada para a inscrição na lista de espera do serviço

⇒ Página de detalhes sobre o serviço:

- Deve detalhar o funcionamento do serviço;
- Apresentar os planos de assinatura e quais benefícios tem em cada plano.



⇒ Página de cadastro na lista de espera:

■ Deve conter formulário com:

- ☐ Nome;
- ☐ E-mail;
- ☐ Senha;
- ☐ Confirmação de senha;
- ☐ RG;
- ☐ CEP (que completa os demais campos do endereço com consulta à API e trata erros);
- ☐ Campos de endereço (estado, cidade, bairro, rua, número, complemento).

COMO FAZER?



Detalhes do Projeto

⇒ Página de contato:

- Deve conter contato dos alunos envolvidos no projeto;
- Algumas páginas serão criadas também para simular algumas funções, elas não vão realizar por completo suas funções (login por exemplo apenas vai mostrar sucesso e validar sem realmente realizar o login do usuário).

⇒ Página de log in:

- Deve conter formulário com:
 - Usuário (e-mail);
 - Senha;
 - Deve validar o campo de entrada como e-mail;
- Deve conter link para página de recuperação de senha (esqueci minha senha).

⇒ Página de recuperação de senha:

- Deve conter formulário com:
 - email de cadastro;
 - Deve validar o campo de entrada como e-mail;
- Deve exibir mensagem após submissão de formulário de e-mail enviado.

COMO FAZER?



Requisitos

- ⇒ O framework CSS Bootstrap deve ser utilizado na elaboração do projeto (versão 4 ou superior);
- ⇒ A API ViaCEP deve ser utilizada para obter dados referentes aos CEPs utilizados no cadastro;
- ⇒ Utilizar repositório público no github, com todos os integrantes do grupo como colaboradores e disponibilização de uma GithubPage com o projeto;

EXTRAS



Requisitos Extras

- ⇒ Utilização de classes e POO na construção do projeto
- ⇒ Utilização de jQuery para manipulações de DOM e requisições HTTP

COMO FAZER?



Documentação e referências

↪ API ViaCEP - <<https://viacep.com.br/>>





F.A.Q.

Vamos ver o conteúdo de Bootstrap no curso?

Não! A ideia é que vocês pesquisem a documentação e comecem a praticar ativamente o processo de consumir a documentação de uma tecnologia para aprender como utilizá-la.

Posso usar outro framework de CSS? Bulma por exemplo?

Sim, desde que os demais requisitos não sejam afetados por conta disso.

Posso utilizar outras APIs além da API do ViaCEP?

Sim, desde que os demais requisitos não sejam afetados por conta disso.

A página de cadastro deve salvar os dados em algum banco de dados?

Não, vamos chegar nessa parte ainda nos módulos 3 e 4, até lá vamos criar páginas que simulam esse fluxo, mas não vamos cadastrar de verdade os usuários.

Todos devem trabalhar no código do projeto?

Sim, vamos checar no Github os envios e quem participou escrevendo código para o projeto, se organizem para dividirem as tarefas e garantir que todos estão atuando em algum ponto do código.



MÃO NA MASSA



Momento 1 - Início

Fazer um *brainstorm* (toró de ideias) para decidir o produto a ser criado!

Dica: crie arquivos padrão contendo de código html/css com coisas em comum como o cabeçalho e o rodapé, que serão compartilhadas entre todas as páginas;

Dividir para conquistar!
Verifique as páginas que precisam ser desenvolvidas e divida entre as pessoas integrantes do grupo;

Crie o repositório do Github na conta de um dos integrantes e faça o convite para as demais pessoas participem do projeto;

Façam o clone do repositório na máquina local para iniciar o desenvolvimento.

Escolha quem será a pessoa responsável por administrar o merge das branches com a main/master.



MÃO NA MASSA



Momento 2 - Desenvolvimento

Começando o desenvolvimento!
Certifique-se da divisão de tarefas entre os integrantes!

Dica: use o comando "git pull" para manter a branch main/master local atualizada! É bom ter uma segurança no repositório local caso algo dê errado com o remoto;

Cada uma deve criar sua branch, de acordo com sua tarefa e iniciar;

Comece pelo que tem mais facilidade e não se esqueça de pensar bem no fluxo de informação entre as camadas antes de começar a escrever o código;

Não se esqueça do fluxo de trabalho com o Github! Abra a PR da sua branch sempre que for fazer o merge;

Analise com cuidado as alterações e sempre peça ajuda sempre que sentir necessidade quando for resolver conflitos dentro da Merge.



MÃO NA MASSA



Momento 3 - Aplicação

Continuar com o desenvolvimento do projeto!

Verifique se as demais pessoas do seu grupo estão precisando de alguma ajuda;

Não se esqueça de testar cada parte do código;

É bom já começar a pensar em como será feita a apresentação.



MÃO NA MASSA



Momento 4 - Últimos ajustes

Hora de fazer as verificações!

Toques finais! Verifique se seu projeto está atendendo todos requisitos necessários.

Sobrou tempo? Vale ver se tem algum extra que pode ser aplicado. Cuidado para não quebrar o que já está funcionando.

Não esqueça de subir seu código para o Github e ativar o Github Pages. Vale conferir se ele está funcionando bem lá também.



MÃO NA MASSA



Momento 5 - Finalização

Momento dos últimos testes!

Organizar a *squad* para que, enquanto alguns integrantes testam o *site*, os demais montam a apresentação final.

Dicas para arrasar na apresentação:

- Montar uma apresentação com base em slides organizados, bonitos e objetivos, com pouco texto.
 - ◆ Conferir dicas [aqui](#).
 - ◆ Opção fácil e gratuita para criar apresentações: [Canva](#).
- A apresentação precisa ter começo, meio e fim!
 - ◆ Iniciar com a apresentação da *squad*;
 - ◆ Mostrar como foi o processo de organização e andamento do projeto;
 - ◆ Indicar o que usaram para fazer o *site*;
 - ◆ Finalmente: mostrar o site na prática!



RUBRICA



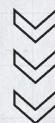
Conteúdo	Habilidades
Organização de código	<ol style="list-style-type: none">1. O projeto está organizado em pastas2. O conteúdo de HTML/CSS/JS está separado em arquivos diferentes..3. O projeto está estruturado seguindo uma lógica de MVC4. O código não apresenta problema de sintaxe.5. Nenhum código CSS/JS está inline em arquivos HTML.
Javascript	<ol style="list-style-type: none">1. O código executa corretamente.2. A(s) consulta(s) da(s) API(s) funcionam e eventuais falhas foram tratadas com try/catch.3. O projeto não apresenta erros ao ser executado no navegador.4. Foram aplicados conceitos de POO e de reutilização/reaproveitamento de código.5. Javascript foi desenvolvido seguindo a lógica de MVC.
CSS	<ol style="list-style-type: none">1. CSS foi desenvolvido no arquivo externo2. Seletores avançados no CSS e código CSS otimizado.3. O projeto apresenta customização/personalização além das classes do Bootstrap.4. O projeto está responsivo.5. Foi utilizado o Bootstrap.



RUBRICA



Conteúdo	Habilidades
Estrutura das páginas	<ol style="list-style-type: none">1. O projeto foi entregue.2. Todas as páginas solicitadas foram desenvolvidas.3. Todas as funcionalidades das páginas solicitadas foram desenvolvidas.4. Além das funcionalidades solicitadas nos requisitos do projeto foram desenvolvidas funcionalidades extras.5. Existe uma consistência entre elementos em comum das páginas (cabeçalho e rodapé).
Git/Github	<ol style="list-style-type: none">1. Entregou o link do repositório no Github.2. O código está completo/funcionando no Github.3. Além do código, foi colocado um arquivo README explicando do que se trata e como pode ser executado o projeto.4. O código foi enviado em commits por etapas.5. As descrições dos commits/PRs estão bem redigidas e apresentam bem as mudanças realizadas.





**Até a próxima e
#confianoprocesso**



Todos os direitos reservados
©2022 Resilia Educação