DIGITALHOUSE

Certified Tech Developer

The Ultimate Degree

Especialização em **Front End II**

Especialização em Front End II

Fundamentação

O desenvolvimento front-end engloba o conjunto de tecnologias utilizadas para desenvolver os componentes de uma aplicação web com os quais os usuários interagem. É por isso que muitas vezes se diz que eles estão do lado do cliente.

A Especialização Front End II dá continuidade e complementa o aprendizado que foi construído no decorrer da trilha Front End (I, II e III) e durante os dois primeiros meses desta especialização. Aprofunda e complementa alguns conceitos básicos e apresenta ferramentas e funcionalidades avançadas usadas no desenvolvimento de aplicativos web modernos e escaláveis.

Dentro desta segunda disciplina da especialização, vamos nos concentrar nos conceitos relacionados à **arquitetura de software.** Em particular, aprofundaremos alguns **padrões de projeto** que usamos até agora, ao mesmo tempo em que introduzimos novos padrões que completarão nosso conhecimento sobre o tema. Da mesma forma, reforçaremos os principais conceitos relacionados às **boas práticas** que devem ser considerados no desenvolvimento de qualquer projeto.

Durante a segunda metade do curso, apresentaremos novas ferramentas para o desenvolvimento de aplicações web modernas e escaláveis, como **Styled Components** e **Apollo GraphQL**. A primeira delas permitirá criar componentes com estilos já incorporados, enquanto a segunda nos dará a oportunidade de simplificar e tornar mais eficiente a conexão de nossa aplicação com uma API. Fecharemos a matéria reforçando conceitos sobre **testes**, apresentando recursos e ferramentas-chave para testar aplicações feitas com React.

Ao final desta Especialização, cada estudante terá adquirido novas tecnologias e ferramentas necessárias para poder atuar como desenvolvedor front-end em sua vida profissional.

Objetivos de aprendizagem

- Identificar diferentes padrões de design para o desenvolvimento de aplicações web, reconhecendo os pontos fortes e fracos de cada uma, e construir critérios para sua aplicação em diferentes projetos.
- Reforçar conceitos relacionados ao uso de boas práticas na escrita de código e sua aplicação no desenvolvimento de projetos futuros.
- Entender a importância da acessibilidade em nosso aplicativo, seguindo os princípios padronizados das WCAG.
- Introduzir o GraphQL como linguagem e o Apollo como ferramenta alternativa ao uso de APIs REST, entendendo vantagens e desvantagens do uso.
- Fortalecer as habilidades no uso de estilos para componentes. Introduzir os componentes Styled, identificando vantagens e desvantagens de uso.
- Reforçar conceitos sobre testes com foco em React, introduzindo novas ferramentas para este fim. Promover a metodologia TDD (Test Driven Development) como uma prática recomendada ao escrever código.

Metodologia de ensino-aprendizagem

Na Digital House, propomos um modelo educacional que inclui ambientes de aprendizagem síncrona e assíncrona com uma abordagem que une teoria e prática, por meio da aprendizagem ativa e colaborativa. Nossa proposta inclui aulas ao vivo com seu grupo de estudantes e docentes, que você pode participar de onde estiver. Além disso, temos um campus virtual customizado, no qual você encontra aulas virtuais com atividades, vídeos, apresentações e recursos interativos, para fazer no seu próprio ritmo antes de cada aula ao vivo. Ao longo da sua experiência de aprendizagem na Digital House, você poderá desenvolver competências técnicas e sociais, como trabalho em equipe, criatividade, responsabilidade, compromisso, comunicação eficaz e autonomia.

Na Digital House, utilizamos a metodologia "Sala de Aula Invertida". O que isso quer dizer? Toda semana, vamos pedir para você se preparar para a próxima, lendo textos, assistindo vídeos, fazendo atividades, entre outros recursos. Dessa forma, ao chegar ao encontro ao vivo, você terá mais preparo para abordar o tema de maneira mais rica.

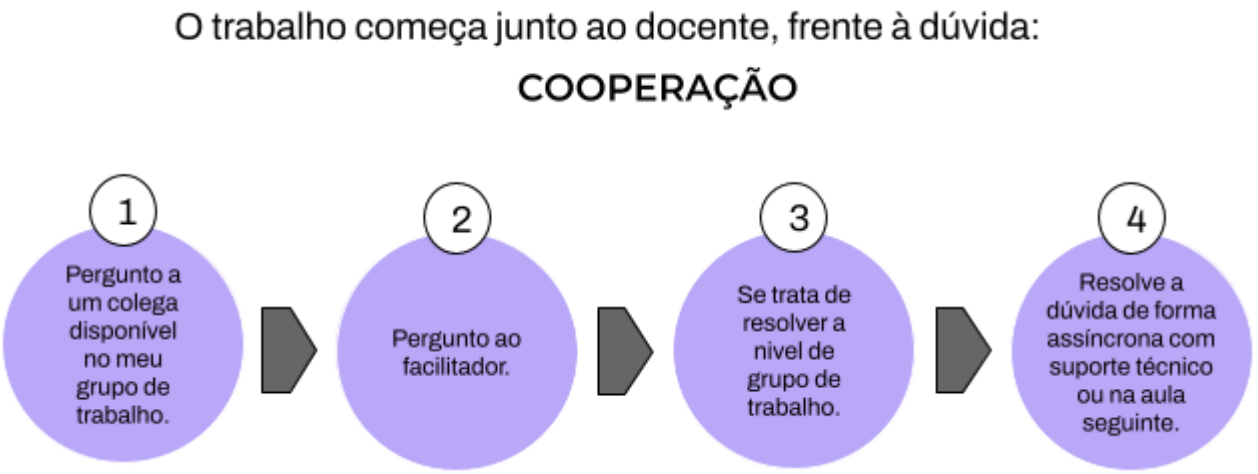
Utilizamos atividades e estratégias baseadas em métodos participativos e ativos para colocar você em movimento, porque só você sabe o que está fazendo por si. Por isso, organizamos as aulas para que você realmente trabalhe nela e possa colocar em prática as diferentes ferramentas, linguagens e habilidades que compõem a formação de um programador. Concebemos a aula como um espaço de trabalho.

Uma das questões centrais da nossa metodologia de ensino é aprender fazendo. Por esta razão, os exercícios estarão muito presentes ao longo do curso, ou seja, a prática de atividades de vários tipos e níveis de complexidade que lhe permitirão consolidar a aprendizagem e verificar se a assimilou corretamente. Desta forma, é possível alcançar uma aprendizagem mais significativa e profunda, assimilar o conhecimento de maneira mais eficaz e duradoura, relacionar o que foi aprendido com a realidade dos programadores web, promover autonomia e autoconhecimento, melhorar a análise, relação e compreensão de conceitos, que ajuda a exercitar uma infinidade de habilidades.

A aprendizagem entre pares é um dos elementos centrais da nossa metodologia, por esta razão, em cada aula propomos que trabalhe em mesas de trabalho com seus colegas, ao longo do curso iremos variar a composição dos grupos para potencializar a cooperação. O que se propõe é uma mudança de perspectiva sobre o curso em questão, onde o aluno não é mais visto percorrendo seu percurso acadêmico individualmente, mas como parte de uma equipe que resulta da soma das potencialidades de cada um. A distribuição em grupos de trabalho estimula a diversidade e o aproveitamento do potencial de cada integrante para melhorar o desempenho da equipe. A explicação recíproca como eixo do trabalho diário não só facilita o aprendizado dos colegas, mas sobretudo potencializa a consolidação do conhecimento por parte de quem explica. Responsabilidade, autonomia e proatividade são promovidas, tudo no âmbito da cooperação, levando a ressignificar a experiência de aprendizagem relacionada a emoções positivas.

O trabalho cooperativo permite estabelecer relações responsáveis e duradouras, aumenta a motivação e o comprometimento, além de promover um bom desenvolvimento cognitivo e social. A cooperação surge diante da dúvida. Se alguém tiver uma pergunta, perguntará a algum integrante do grupo designado que esteja disponível. Se a dúvida persistir, o facilitador é acionado. Se não resolverem, o facilitador pedirá a todos que parem para cooperar em equipe na resolução do conflito que levantou a dúvida. É assim que todos os integrantes da mesa vão debater em busca da solução.

Caso ainda não consigam resolver, anotarão a questão que será abordada de forma assíncrona pelo suporte técnico ou de forma síncrona, na próxima aula pelo professor.



Todos os dias, ao final do dia, estudantes reconhecerão um dos integrantes do grupo com quem compartilharam aquele dia. O critério para esse reconhecimento é a cooperação.

Cada grupo terá um facilitador que será escolhido a partir dos reconhecimentos e gerando um sistema de rotação, onde qualquer pessoa poderá desempenhar esse papel. O facilitador não é uma figura estática, mas cumpre um papel dinâmico e versátil. O facilitador é um estudante que mobiliza a realização dos objetivos comuns da equipe colocando em jogo a cooperação. É quem compartilha seu potencial com a mesa em favor do restante da equipe e que, portanto, promove a cooperação.

Informações sobre a matéria

- Modalidade 100% remota
- Número total de semanas: 9
- Número de aulas ao vivo semanais: 3
- Número de aulas virtuais no Playground: 27
- Número total de aulas ao vivo: 27

Requisitos e correlações

Para fazer esta seção da Especialização Front-End, é necessário ter aprovado na primeira disciplina da Especialização. Por sua vez, a aprovação desta matéria é requisito para o estudo da disciplina Especialização em Front End III.

Modalidade de trabalho

Nossa proposta educacional é especialmente pensada para esta modalidade 100% a distância, por meio da aprendizagem ativa e colaborativa seguindo nosso pilar de "aprender fazendo". Os ambientes de aprendizagem são síncronos e assíncronos, com uma abordagem que une teoria e prática, para que ambas estejam presentes em todos os momentos.

Temos um Campus Virtual próprio, onde encontraremos atividades, vídeos, apresentações e recursos interativos com instâncias de trabalho individual e em equipe para aprofundar cada um dos conceitos. Além disso, realizaremos encontros online e ao vivo com o grupo de estudantes e docentes, aos quais poderemos participar de onde estivermos por meio de uma plataforma de videoconferência com nossa câmera e microfone para gerar uma experiência próxima.

Metodologia de avaliação

A avaliação formativa é um processo contínuo que gera informações sobre a formação de nossos alunos e de nós como educadores.

Ao mesmo tempo, gera-se um conhecimento de caráter feedback, ou seja, tem uma função de conhecimento, pois nos permite conhecer os processos de ensino e aprendizagem. Também tem uma função de melhoria contínua porque nos permite saber onde estamos no processo, para validar se continuamos no caminho planejado ou se precisamos tomar novas decisões para cumprir os objetivos propostos.

Finalmente, a avaliação desempenha um papel importante em termos de promoção do desenvolvimento de competências muito valiosas.

Nosso objetivo é fugir da avaliação tradicional, onde muitas vezes é um momento difícil, entediante e tenso. Para isso, vamos utilizar a gamificação, que é uma técnica em que elementos do jogo são aplicados para tornar o conteúdo mais atraente, os participantes se sentem motivados e imersos no processo, utilizam os conteúdos de aprendizagem como desafios que realmente desejam superar e aprendem com o erro. Por sua vez, para o registro da referida formação, utiliza-se um conjunto de instrumentos, para os quais é imprescindível utilizar a mais ampla variedade e técnicas de análise possíveis.

Critérios de Aprovação

- Realização de atividades de Playground (80% concluídas).
- Presença nas reuniões síncronas (90% de presença).
- Obter nota igual ou superior a 7 na avaliação parcial.
- Obter pontuação igual ou superior a 7 na nota final da disciplina.

Conteúdo

Módulo 1 - Padrões de Projeto Avançados e Boas Práticas

Os padrões de projeto são apresentados como soluções testadas e padronizadas para problemas comuns.

Aula 1 - Padrões de Projeto

Introdução aos padrões de design e revisão do que aprendemos na trilha Front End.

- Composição vs. herança.
- Revisão de provider pattern e render props.
- Componentes funcionais: arquitetura com custom hooks.
- Apresentação do projeto de trabalho.

Aula 2 - Padrões de Projeto (continuação)

Continuamos nos aprofundando no mundo dos padrões de design e suas possibilidades de desenvolvimento Front End.

- Compound components.
- High Order Components (HOC) em componentes funcionais.

Aula 3 - Integração

Aula prática de integração: trabalhamos no caso!

Módulo 2 - Clean code e Acessibilidade

As boas práticas são aprofundadas sobre o Clean Code. Conceitos de acessibilidade em aplicações web são explorados.

Aula 4 - Clean code

Vamos um pouco mais longe nas boas práticas, com os conceitos de Clean code e ferramentas de formato automáticas.

- Boas práticas e Clean code.
- Regras de formato e análise de código com ESLint.
- Nomenclatura: nome das variáveis e funções.
- Tratamento de erros.

Aula 5 - Acessibilidade

Introduzimos a acessibilidade com seus diversos critérios de aplicação.

- O que entendemos por acessibilidade?
- Web Content Accessibility Guidelines (WCAG).
- Critérios relacionados com a percepção.
- Critérios relacionados com a operacionalidade.
- Critérios relacionados com a compreensão.
- Critérios relacionados à robustez.

Aula 6 - Integração

Aula prática de integração: trabalhamos no caso!

Módulo 3 - Princípios SOLID

Este módulo abrange princípios de programação que facilitam o desenvolvimento, tornando o código fácil de manter e estender.

Aula 7 - Princípios SOLID

Nesta aula, falamos sobre os princípios relacionados à responsabilidade, extensão e substituição de código em nossas aplicações.

- Single Responsibility Principle.
- Open-Closed Principle.
- Liskov Substitution Principle.

Aula 8 - Princípios SOLID (continuação)

Levamos os princípios um pouco mais adiante, abordando a implementação de interfaces e métodos segregados e a dependência de abstrações.

- Interface Segregation Principle.
- Dependency Inversion Principle.

Aula 9 - Integração

Aula prática de integração: trabalhamos no caso!

Aula 10: Vamos para a revisão!

Os conteúdos propostos durante a primeira metade da disciplina são revistos em preparação para a instância de exame.

Aula 11: Workshop integrador

Uma instância de trabalho conjunto é proposta a partir dos exercícios e aplicações que foram construídos nas aulas anteriores.

Aula 12: Primeira parcial

Avaliação dos tópicos apresentados durante a primeira parte da disciplina.

Módulo 4 - Estilo

Os conceitos relacionados à aplicação de estilos em componentes React são revistos. Novos conceitos e ferramentas são introduzidos para criar componentes estilizados.

Aula 13 - Estilo

Revisamos como aplicar estilos CSS usando Javascript. Analisamos quais são as principais vantagens e desvantagens do uso dessa técnica. Além disso, vemos como construir estilos dinâmicos usando Javascript.

- CSS em JS, o que é?
- Vantagens e desvantagens.
- Estilos dinâmicos.

Aula 14 - Estilo (continuação)

Apresentamos a biblioteca Styled Components, que nos permite criar componentes estilizados. Analisamos quais são as principais vantagens e desvantagens do uso desta biblioteca. Vemos como aplicar estilos dinâmicos com base nas propriedades de cada componente.

- O que são Styled Components?
- Vantagens e Desvantagens.
- Estilos baseados em prop utilizando Styled Components.

Aula 15 - Integração

Aula prática de integração: trabalhamos no caso!

Módulo 5 - Testes

Apresentamos as práticas de testing juntamente à metodologia Test Driven Development. Colocamos esses conceitos em prática através das ferramentas Jest e Testing-Library.

Aula 16 - Jest

Introdução aos conceitos de testing e à metodologia de trabalho Test Driven Development.

- O que é Jest?
- Asserts.
- Mocks.

- Coverage.
- Test Driven Development.

Aula 17 - React Testing-library

Aprofundamos o conhecimento de testing aplicando-o a componentes React.

- Introdução. Fundamentos e sintaxe básica.
- Testing components.
- Testing hooks.

Aula 18: Integração

Aula prática de integração: trabalhamos no caso!

Módulo 6 - Testing assíncrono

Aula 19 - Testing assíncrono

Avançamos um pouco mais nos conceitos de testing e introduzimos o teste assíncrono.

- Testing assíncrono.

Aula 20 - Testing assíncrono com React Query

Aprofundamos nosso entendimento sobre testes assíncronos aplicando-os aos componentes do React Query.

- Testes Assíncronos com React Query.

Aula 21: Integração

Aula prática de integração: trabalhamos no caso!

Aula 22: Vamos para a revisão!

Os conteúdos propostos durante a primeira metade da disciplina são revistos em preparação para a instância de exame.

Aula 23: Workshop integrador

Uma instância de trabalho conjunto é proposta a partir dos exercícios e aplicações que foram construídos nas aulas anteriores.

Aula 24: Exame Final

Avaliação dos tópicos apresentados no desenvolvimento da disciplina.

Módulo 7: Aulas Especiais e Encerramento

Aula 25: GraphQL. Introdução ao Apollo

Introdução à linguagem GraphQL, juntamente à biblioteca Apollo como uma alternativa ao REST.

- O que é GraphQL?
- Vantagens e desvantagens.
- Casos de uso.
- O que é Apolo?
- Conexão com Apollo Client.
- Queries. useQuery.
- Uso de cache.

Aula 26: Continuamos com o Apollo

Aprofundamos nosso conhecimento sobre a biblioteca Apollo para interação com o servidor.

- Mutations.
- useMutation.
- Paginação.

Aula 27: Encerramento da matéria

Revisamos todos os tópicos que aprendemos e revisamos os critérios que devemos considerar para escolher um recurso em detrimento de outro de acordo com a implementação.

Material de referência

- Componentes funcionais: arquitetura com custom hooks.
    - https://www.youtube.com/watch?v=axL59Dc5rZA
    - https://reactpatterns.com/#layout-component
- Composição vs herança:
    - https://es.reactjs.org/docs/composition-vs-inheritance.html

- Compound components: https://kentcdodds.com/blog/compound-components-with-react-hooks
    - Clean code: https://dev.to/thawkin3/react-clean-code-simple-ways-to-write-better-and-cleaner-code-2loa
    - https://www.youtube.com/watch?v= qcIhUoBZaHg&ab_channel=Base do foguete
- SOLID:
    - https://www.youtube.com/watch?v=6Sfr03D4dHM
    - https://www.youtube.com/watch?v=9PgcXvqIaos
    - https://www.youtube.com/watch?v=GVo7Kz02agg&ab_channel=ErickWendel
- GraphQL