algaworks

Conteúdo do treinamento

ESPECIALISTA REACT

Atualizado em 28/06/2021

Fundamentos sólidos do React

O React.js é a biblioteca fundamental do ecossistema, e você aprenderá tópicos dos mais simples aos mais avançados sobre ela.

Você vai entender quais os tipos de componentes que o React permite com que você trabalhe (componentes funcionais e componentes baseados em classes).

Vai aprender o que é estado e como ele influencia na renderização dos componentes, ciclos de vida, observadores, gerenciamento de efeitos colaterais e tudo que envolve gerenciamento de dados dentro de React.

Você também aprenderá o que são SPAs (Single Page Applications) como elas conseguem entregar uma ótima experiência para o usuário.

Estilização avançada

Você vai aprender como estilizar seus componentes de diversas formas:

Criar estilos utilizando arquivos CSS e como importá-los dentro de seus componentes React, estilizar componentes utilizando expressões JavaScript e, por fim, trabalhar com Styled Components, que é a ferramenta de estilização mais completa para React.

O Styled Components muda a forma como você enxerga a estilização, permitindo com que você utilize JavaScript dentro do CSS, com uma alta produtividade e performance.

Criação de componentes seguindo um Design System

Você aprenderá como identificar os pontos importantes de um design system e como criar os componentes seguindo exatamente o layout projetado pela equipe de UI (que anda lado a lado com o time de desenvolvimento front-end).

Além de componentes lindões, você também vai criar componentes com uma API de fácil utilização, para que seus colegas de equipe (ou até mesmo você) tenham uma boa *developer experience*.

E de quebra, você vai aprender a manipular eventos do DOM de forma correta com React.js.

Do Figma para o código

Figma é um editor para criação de interfaces gráficas muito usado por times de UI/UX no mercado e é uma ferramenta que você, como desenvolvedor front-end, possivelmente terá contato diário.

Você vai aprender a transformar protótipos de alta e baixa fidelidade em componentes e páginas em aplicações React.js.

Vai aprender como identificar tamanhos, espaçamentos e exportar assets utilizando o Figma.

Ant Design

Você vai aprender a utilizar uma das bibliotecas de componentes mais completas e customizáveis, que é amplamente utilizada no mercado atualmente.

Com o Ant Design, você poderá criar grandes sistemas empresariais com uma alta produtividade, customização e qualidade.

Service Layer Pattern

Você vai aprender a usar o Service Layer Pattern para ajudar a abstrair a lógica de requisições HTTP, deixando o projeto mais organizado e facilitando futuras manutenções no código.

Hooks avançados

Os hooks marcam presença na vida do desenvolvedor React nos dias de hoje.

Durante o desenvolvimento do projeto do curso, você vai aprender a utilizar vários hooks que o próprio React.js provê, além de aprender a criar seus próprios Hooks personalizados que abstraem as lógicas de dados.

Ao criar os seus próprios hooks, você vai perceber a melhora significativa na manutenção do seu código, fazendo com que sua aplicação se torne mais simples de entender e com mais camadas de abstração, diminuindo muito o nível de complexidade de vários componentes.

OpenAPI e SDK

Você vai aprender a visualizar e interpretar a documentação de uma REST API a partir de um documento OpenAPI para que você possa implementar as requisições HTTP pelas aplicações front-end.

Além disso, vai aprender a criar um SDK da REST API e publicar ele no NPM, para reutilizar os serviços entre as aplicações de forma mais sólida e abstraindo lógicas de requisições HTTP.

Gerenciamento inteligente de rotas

Você vai aprender como funcionam as rotas dentro das SPAs (que possuem apenas um único arquivo HTML, mas várias páginas) e como configurar parâmetros no path e recuperar parâmetros na query string de forma muito simples.

Code splitting & Lazy loading

O fato da aplicação ser uma SPA, não nos impede de carregar as rotas e funcionalidades por demanda.

Você vai aprender a segregar os componentes em chunks (blocos de código) que só serão carregados quando houver necessidade, diminuindo assim o tempo de carregamento inicial da sua aplicação (e consequentemente, o tráfego de rede).

Arquitetura Flux

A arquitetura mais utilizada em aplicações front-end, atualmente, para gerenciamento de dados também será ensinada neste curso.

Você vai aprender o que é o Flux de um jeito descomplicado, com exemplos lógicos e simples de entender.

Redux Toolkit

Essa história de que Redux é difícil já ficou para trás. Após entender o que é a arquitetura Flux, você aplicará ela nos projetos de forma simples com o Redux Toolkit, uma ferramenta que facilita muito o uso do Redux em projetos React.

Com o Redux Toolkit, você escreve menos código e produz mais.

Fundamentos da comunicação HTTP

Você aprenderá como exatamente uma aplicação front-end que está em determinado domínio se comunica com uma API em outro domínio seguindo o estilo arquitetural REST.

Vai aprender também o que são e para que servem os verbos HTTP mais utilizados, qual o significado de cada faixa de código de status das respostas, qual é a diferença entre requisições de recuperação de informação em relação a requisições de envio de informação e muito mais.

Cacheando requisições HTTP

Você vai aprender a evitar requisições repetitivas para a REST API, configurando um tempo de cache local para o Axios (pacote que utilizaremos para gerenciar as requisições HTTP) de forma super simples, que poderá armazenar o cache tanto em memória, quanto no Local Storage.

Tratativa de erros com Error Boundaries

Você vai aprender a tratar erros de forma individual, para cada tipo de situação, prevenindo que componentes quebrem a aplicação por inteiro e provendo uma interface alternativa para componentes que não conseguem comunicação com a API por algum erro ou instabilidade.

Também vai aprender um método que captura todos os erros que não foram tratados, para não deixar o usuário sem receber a informação, caso algum erro inesperado aconteça.

Você aprenderá a identificar os erros da API e lançar erros personalizados, para ficar mais fácil identificá-los pela aplicação sobre o que se trata cada erro.

Erros de validação são diferentes de erros de comunicação, que são diferentes de erros de regra de negócio.

Para cada tipo de erro, teremos uma classe personalizada, com um tipo de informação, que poderá ser identificada de forma fácil.

Boas práticas de desenvolvimento

Todo o conteúdo do curso seguirá as boas práticas de desenvolvimento, pensando sempre no mercado de trabalho e nas futuras manutenções do código.

Durante o curso, você será estimulado a repensar o nome das variáveis e dos métodos, de forma com que possamos evitar ao máximo a necessidade de adicionar comentários no código.

Você também será estimulado a programar utilizando inglês, porque muitas empresas padronizam o uso desse idioma no código-fonte.

As interfaces das aplicações serão inteiramente em português. Apenas o código-fonte que será implementado em inglês.

Documentação de componentes com Storybook

Você vai aprender a documentar os componentes que você mesmo criou do zero em uma interface visual e testável fornecida pelo Storybook.

Com o Storybook, você pode criar os seus componentes e documentar exatamente cada propriedade, podendo testar os elementos visualmente e testar como a interface se comporta com base nos valores das propriedades que são passadas.

Validação de formulários client-side e server-side

Você vai aprender a trabalhar com validação de formulários no cliente e no servidor, facilitando a experiência do usuário que receberá feedbacks de validação no ato da digitação, e, caso a validação do cliente não seja igual a validação da REST API, o usuário também receberá informação visual mostrando onde exatamente houve o erro.

Upload de arquivos

Você vai aprender a implementar upload de arquivos para um serviço de storage terceirizado, seguindo o padrão de URL assinada (Signed URLs), que é utilizado nos

maiores e mais conhecidos serviços de cloud do mercado (como Amazon S3 e Google Cloud Storage).

Vai implementar todo o fluxo do zero, sem a ajuda de uma biblioteca externa.

Você também vai entender o motivo pelo qual esse padrão de URLs assinadas é tão importante e como ele ajuda a evitar altas latências e tráfego de dados duplicado.

Crop de imagens

Além do upload da imagem possuir redimensionamento (pequeno, médio e grande) pela API, você aprenderá a fazer a edição de uma imagem antes de seu upload, usando uma biblioteca que casa perfeitamente com o Ant Design, que permite com que você recorte a imagem e rotacione ela, para se ajustar melhor ao propósito.

Impressão de relatórios com CSS e JavaScript

Você aprenderá a preparar uma página web para impressão utilizando Media Queries do CSS e ativar a impressão por meio de JavaScript.

Além disso, aprenderá a adicionar elementos na página somente no ato da impressão, deixando a página mais padronizada e com um aspecto mais profissional.

Downgrade e upgrade de pacotes

Com o passar do tempo, os componentes e pacotes vão se atualizando, e pode ser que uma versão recente de um pacote pare de funcionar no seu projeto.

Você será estimulado a identificar estes casos e instalar versões anteriores (e até mesmo mais utilizadas) de determinados pacotes.

A mesma lógica serve para atualizar pacotes que estão desatualizados e, por este motivo, não encaixam no ecossistema da aplicação.

CORS

Você aprenderá o que é para que serve o CORS, o que é uma requisição de preflight e por quê ela é tão importante para a segurança de aplicações front-end.

Deploy e Continuous Delivery

Você aprenderá o que é Continuous Delivery e por que esse conceito vem se tornando popular no dia a dia do desenvolvedor, independente da sua stack.

Você também vai aprender a fazer o deploy das aplicações front-end no Netlify e Vercel, que são serviços muito conhecidos e utilizados, que possuem planos gratuitos e permitem continuous delivery sem cobrança extra.

Domínio personalizado e DNS

Você vai aprender a registrar um domínio na internet e configurar subdomínios personalizados, tanto para a REST API, quanto para as aplicações front-end.

No final, teremos todas as aplicações do projeto do curso implantado na internet em subdomínios diferentes, por exemplo: api.alganews.com.br, admin.alganews.com.br, cms.alganews.com.br e www.alganews.com.br.

Segurança com OAuth2, JWT e Refresh Token

Você aprenderá a implementar a segurança do zero com OAuth2 e JWT (JSON Web Tokens) em aplicações front-end utilizando o Authorization Code Flow com Proof Key for Code Exchange (PKCE), que é o fluxo mais seguro e recomendado para SPAs.

Você também vai aprender a manter a sessão do usuário ativa, mesmo quando ele fechar a aplicação (armazenar os tokens em um local seguro que não é perdido ao sair da aplicação).

Além disso, vai aprender a lidar com Refresh Tokens de uma forma simples, onde todas as suas requisições HTTP serão validadas antes mesmo de serem enviadas, e, caso haja a necessidade de atualizar o Access Token, isso será feito de forma automática, sem que o usuário perceba.

Server-Side Rendering (SSR)

Você vai aprender a criar aplicações que são renderizadas do lado do servidor, que diferente das SPAs, já vêm renderizadas para o cliente na primeira requisição.

Além disso, aprenderá algumas técnicas básicas de SEO para ranquear melhor a sua página nos resultados de motores de busca como o Google, que só são possíveis graças ao SSR.

Além disso, você vai entender conceitos como Back-end x Back-end e módulos CSS.

Docker e Docker Compose

Por vezes, desenvolvedores front-end precisam usar o Docker para subir um ambiente de desenvolvimento local.

No curso você contará com um módulo exclusivo de Docker e Docker Compose, onde você vai aprender os principais fundamentos da tecnologia e ficar preparado para subir ambientes inteiros na sua máquina com um simples comando.

Depuração (debug) facilitada

A depuração de aplicações React pode se tornar um assunto bem extenso, visto que existe uma solução para cada situação.

O curso conta com um módulo exclusivo sobre depuração, que falará especificamente sobre extensões, ferramentas e técnicas de depuração.

Testes automatizados

Você aprenderá, em um módulo exclusivo, como criar testes unitários e E2E (end-to-end) com as tecnologias mais recentes e utilizadas no mercado.

Você vai conseguir garantir que as funcionalidades das suas aplicações funcionem corretamente e vai entender o conceito de Code Coverage.

Conteúdo programático

1. Introdução ao curso
1.1. Bem-vindo ao curso
1.2. Por que React?
1.3. Metodologia de componentes com React
1.4. Hands on com Codepen
1.5. Obtendo o melhor do suporte
1.6. Preparando o ambiente de desenvolvimento
1.7. Prévia do projeto final do curso

2. Introdução ao React
2.1. Introdução ao React
2.2. Entendendo mais sobre SPAs
2.3. Criando uma aplicação React com CRA
2.4. Entendendo o código gerado pelo CRA
2.5. Habilitando o ESLint no VSCode
2.6. Criando o componente de Hello World
2.7. Entendendo um pouco mais sobre o JSX
2.8. Incorporando expressões dentro do JSX
2.9. JSX também é uma expressão
2.10. Atributos com JSX
2.11. A segurança do JSX
2.12. Como o JSX é Renderizado no DOM
2.13. Atualizando um componente em tempo real
2.14. Os estados dentro de componentes React (com o hook useState)
2.15. Como funcionam as props dentro do React
2.16. Encapsulando componentes com props.children
2.17. Introdução à estilização de componentes (atributo style)
2.18. Estilizando componentes com CSS (de forma dinâmica)
2.19. Estilizando componentes com Styled Components
2.20. A importância histórica dos componentes baseados em classe
2.21. Criando um Class Component (Componente baseado em classe)
2.22. Acessando props dentro de um Class Component
2.23. Estados em Class Components
2.24. Introdução aos ciclos de vida com componentDidMount
2.25. Recuperando atualizações do estado com componentDidUpdate
2.26. Limpando efeitos colaterais com componentWillUnmount
2.27. Renderização condicional de várias formas
2.28. Manipulação de eventos
2.29. Interligação com formulários
2.30. Dica preciosa para lidar com formulários
2.31. Renderizando listas

3. Hooks avançados
3.1. Introdução aos Hooks

3.2. Os hooks que você já conhece
3.3. Criando um hook personalizado
3.4. O estado local dos hooks
3.5. Passando parâmetros para os hooks
3.6. Como realmente funciona o hook useRef
3.7. Gerenciamento de estado complexo com useReducer
3.8. Gerenciando funções imperativas do DOM com useImperativeHandle
3.9. O hook useCallback
3.10. Exhaustive Dependencies

4. Componentes documentados com StoryBook
4.1. Briefing do CMS (Projeto 1)
4.2. Introdução ao Figma
4.3. Entendendo melhor o Figma
4.4. Configurando o StoryBook no projeto
4.5. Entendendo melhor o StoryBook
4.6. Criando o boilerplate do Botão
4.7. Criando as variantes do botão
4.8. Recebendo todas as props de um button dentro do componente personalizado
4.9. Finalizando o componente de botão com o Polished
4.10. Criando a API do Input
4.11. Estilizando o componente Input
4.12. Adicionando estilização global (fontes) no Storybook
4.13. Criando o componente do ValueDescriptor
4.14. Uma nova forma de renderizar componentes de forma condicional
4.15. Criando componente de parágrafo com estilização condicional
4.16. Desafio - FieldDescriptor
4.17. Utilizando Icones dentro do React
4.18. Desafio - Componente Confirm
4.19. Apresentando o React Table
4.20. Criando a estrutura da Tabela
4.21. Estilizando a tabela
4.22. Alinhando as colunas da tabela
4.23. Reutilizando o componente de Tabela
4.24. Recuperando dados além do accessor
4.25. Alinhando os Headers
4.26. Criando componente NoData
4.27. Aplicando o NoData dentro da tabela
4.28. Apresentando e instalando o ChartJS
4.29. Configurando a altura do Chart
4.30. Downgrade do ChartJS
4.31. Configurando as Legendas no Chart
4.32. Configurando a tensão da linha no Chart
4.33. Labels do ChartJS
4.34. Entendendo como estruturar os dados no Chart
4.35. Estilizando os datasets
4.36. Entendendo um pouco mais sobre os eixos do Chart
4.37. Removendo gridlines do Chart

4.38. Finalizando o visual do Chart
4.39. Propificando o componente de Chart
4.40. Desafio - Progress Bar
4.41. Iniciando o componente CircleChart
4.42. Documentando o CircleChart
4.43. Setup do CircleChart
4.44. Criando os componente estilizados do CircleChart
4.45. Finalizando (e apreciando) o CircleChart
4.46. Componente Image Upload
4.47. Desafio profile
4.48. Desafio - SessionController
4.49. Criando o contador de palavras
4.50. Introdução ao TagInput
4.51. Estilizando o TagInput
4.52. Exemplo funcional no StoryBook (com o TagInput)
4.53. Desafio - ErrorDisplay
4.54. Editor de texto em Markdown

5. Como funcionam as rotas
5.1. O que esperar do módulo
5.2. Dica valiosa para seguir com o módulo
5.3. Sobre conflitos
5.4. Instalando e Configurando o React Router DOM
5.5. Quem é o BrowserRouter
5.6. Quem é o Switch
5.7. Route e a propriedade exact
5.8. As views são só componentes
5.9. A nomenclatura dos arquivos
5.10. Recuperando parâmetros na rota
5.11. Recuperando parâmetros da Query String
5.12. Manipulando o histórico com useHistory
5.13. A facilidade do componente Link
5.14. Alterando o título da página conforme as rotas
5.15. Code Splitting

6. Páginas e funcionalidades do CMS
6.1. Introdução ao módulo
6.2. Instalando as fontes no projeto
6.3. Aplicando estilos globais com StyledComponents
6.4. Criando o layout base do sistema
6.5. Aplicando os componentes no layout
6.6. Estilizando o menu lateral com base na página atual
6.7. Criando as primeiras Features
6.8. Feature UserTopTags
6.9. Feature UserEarnings
6.10. Adicionando espaçamento no final da página
6.11. Criando a rota de lista de editores
6.12. Feature EditorsList

6.13. Criando o formulário de post
6.14. Fazendo o contador de palavras funcional
6.15. Página 404
6.16. Criando componente de confirmação
6.17. Replicado a lógica do confirm no Info
6.18. Transformando um método em uma macrotask (event loop)
6.19. Estilizando um componente de uma biblioteca externa
6.20. Desafio - Grid do perfil do editor
6.21. Escondendo dados sensíveis do usuário
6.22. Blindando a view com ErrorBoundary

7. O Docker que o front-end precisa saber
7.1. Introdução ao módulo
7.2. Introdução ao Docker
7.3. Instalando o Docker no Windows
7.4. Instalando o Docker no Linux
7.5. O que são Imagens
7.6. Criando um Container
7.7. Criando um Dockerfile
7.8. Como transformar uma API REST em um Container
7.9. O que são Volumes
7.10. Rede no Docker
7.11. O que é o Docker Compose
7.12. Removendo dados não utilizados pelo Docker

8. Interligando a SPA com a API
8.1. Introdução ao módulo
8.2. Front-end vs Back-end (e requisições HTTP)
8.3. Requisições HTTP (verbos, uri, headers e body)
8.4. O retorno das requisições HTTP (status code)
8.5. Subindo a infra com Docker
8.6. O que é OpenAPI
8.7. Enviando uma requisição HTTP com fetch
8.8. Alterando propriedades da requisição
8.9. Como o fetch lida com status code
8.10. Por que o Axios é melhor que o fetch
8.11. Services Layer Pattern
8.12. Gerando interfaces TypeScript automaticamente com OpenAPI
8.13. Criando a classe de Serviço
8.14. Criando o primeiro serviço
8.15. Enviando dados (método POST) com axios
8.16. Lidando com parâmetros da API
8.17. A Developer Experience do SDK
8.18. Upload de arquivos em cloud
8.19. Criando FileService
8.20. Enviando o arquivo para o Storage
8.21. Abstraindo a lógica do upload
8.22. Implementando o cadastro de posts

8.23. Fazendo a lista de editores funcionar
8.24. O primeiro contato com manipulação de datas
8.25. Perfil de editor
8.26. Top 3 Tags
8.27. Ganhos do usuário
8.28. Transformando dados para o ChartJs
8.29. Trocando de usuário na API
8.30. Buscando dados paginados da API
8.31. Resolvendo pequenos erros de depreciação
8.32. Aplicando error boundaries
8.33. Limitações do error boundary
8.34. Criando um HOC de Boundary
8.35. Adicionando Skeleton na Interface
8.36. Criando um Loading
8.37. Fazendo o Loading funcionar
8.38. Desafio - Loading animado com blur
8.39. Prevenindo erros de passarem em branco com onUnhandledRejection
8.40. Paginaçao Server-side vs Client-side
8.41. Aplicando paginação no react-table
8.42. Paginando no servidor
8.43. React Paginate
8.44. Desafio - Criar modal de Preview de Post
8.45. Renderizando markdown
8.46. Desafio - Layout do Post
8.47. Recuperando post da API
8.48. Abrindo links em uma nova aba
8.49. Estilizando o título do post
8.50. Upload de imagem no post
8.51. Desabilitando botões por status do post
8.52. Publicando um post
8.53. Melhorando a experiência do cadastro de post
8.54 - O problema do BrowserRouter
8.55 - Edição do post
8.56 - Criando erros personalizados
8.57 - Identificando o tipo de erro

9. Criando e Publicando um SDK
9.1. Introdução ao módulo
9.2. O que é um SDK?
9.3. Inicializando o SDK
9.4. Versionamento Semântico
9.5. Publicando o SDK
9.6. Instalando o SDK no CMS
9.7. A importância do prepublish
9.8. Implementando os primeiros serviços
9.9. Desafio - Serviço de Pagamentos
9.10. Desafio - Serviço de Fluxo de Caixa
9.11. Desafio - Finalizar o serviço de Usuários

9.12. Desafio - Finaliza o serviço de Posts
9.13. Content-Type personalizado
9.14. Publicando a versão final do SDK com Types

10. Gerenciamento de Estado Global
10.1. Introdução ao módulo
10.2. O que é Estado Global?
10.3. Entendendo melhor a arquitetura Flux
10.4. O que é Redux?
10.5. O que é Redux Toolkit?
10 6. Instalando o Redux Toolkit no projeto
10.7. Criando e acessando o primeiro Store
10.8. Funções Reducers
10.9. O que são actions?
10.10. Disparando uma action dentro do componente
10.11. O que são thunks?
10.12. Criando a primeira Thunk
10.13. Criando uma Thunk
10.14. Disparando uma thunk dentro do componente
10.15. Identificando estados de uma Thunk
10.16. Gerenciando vários stores
10.17. Instalando e configurando o Redux Devtools
10.18. Armazenando dados locais com Redux Persist
10.19. O Redux na sua forma mais pura (sem Toolkit)
10.20. Criando hooks conectados ao Redux
10.21. Criando modais com Redux

11. Server-Side Rendering com Next
11.1. Introdução ao módulo
11.2. O que é Server-Side Rendering?
11.3. O que é Next.js?
11.4. Iniciando o projeto com Next.js e TypeScript
11.5. Como funcionam as páginas e rotas
11.6. Static Generation vs. Server-Side Rendering
11.7. Recebendo dados na página
11.8. Estilização com CSS Modules
11.9. Instalando o Styled-Components
11.10. Criando o Layout da HomePage
11.11. Criando o componente de Post
11.12. Criando o componente de Post Destacado
11.12. Adicionando paginação
11.13. Redirecionando para página inicial caso a página seja inválida
11.14. Página de erro 404
11.15. Página de erro 500
11.16. Configurando o Favicon
11.17. Adicionando comentários com Disqus
11.18. O que é SEO?
11.19. Adicionando tags no Head

11.19. Adicionando tags de OpenGraph

12. Conhecendo o Ant.Design
12.1. Introdução ao módulo
12.2. Introdução ao Ant.Design
12.3. Instalando e configurando o Ant.Design no projeto
12.4. Criando a primeira Tela
12.5. O jeito de pensar do Ant.Design
12.6. Sistema de Grid
12.7. Responsividade
12.8. Tipografia
12.9. Tabela de dados
12.10. Modais
12.11. Diálogos de confirmação
12.12. Configurando a interface para pt_BR
12.13. Formulários e validação

13. As funcionalidades do Admin
13.1. Introdução ao módulo
13.2. Configurando o Redux no projeto
13.3. Estruturando o projeto
13.4. Criando a listagem de usuários
13.5. Adicionando filtro na coluna de email
13.6. Desafio - Adicionar filtro na coluna de nome
13.7. Adicionar ação de atualizar dados
13.8. Limitando o tamanho da coluna de email
13.9. Aplicando responsividade no layout
13.10. Aplicando responsividade na tabela de usuários
13.11. Criando o formulário de usuário
13.12. Adicionando máscara nos campos
13.13. Criando um componente monetário
13.14. Upload de imagem com Crop
13.15. Segregando os Dados bancários
13.16. Identificando problemas na aba dados bancários
13.17. Cadastrando usuário
13.18. Configurando timezone e linguagem do Moment
13.19. Adicionando onUnhandledRejection
13.20. Criando o perfil do usuário
13.21. Adicionando os posts ao perfil do usuário
13.22. Desabilitando e Habilitando posts no perfil do usuário
13.23. Criando tela de detalhamento do post
13.24. Adicionando atalhos para ver post no blog
13.25. Desafio - Criando a listagem de pagamentos
13.26. Adicionando paginação server-side
13.27. Adicionando filtro fora da tabela
13.28. Selecionando vários registros para ações em batch
13.29. Criando a tela de cadastro de pagamentos
13.30. Adicionando campos dinamicamente ao formulário

13.31. Pré-visualização do pagamento
13.32. Desabilitando datas no DatePicker
13.33. Um resumo sobre o Fluxo de Caixa
13.34. Criando o cadastro de catetorias
13.35. Criando o cadastro de despesas
13.36. Desafio - Listagem de pagamentos
13.37. Filtrando pagamentos por mês
13.38. Desafio - Excluir pagamentos em batch
13.39. Editar pagamento
13.40. Desafio - Excluir pagamento individual
13.41. Duplicando o módulo de receitas para despesas
13.42. Menu lateral acompanhando as rotas
13.43. Adicionando atalho para o perfil no Header
13.44. Configurando as cores do Antd com less
13.45. Desafio - Criar a Homepage completa

14. Blindando a aplicação (Segurança)
14.1. Introdução ao módulo de segurança
14.2. Introdução ao OAuth
14.3. Habilitando a segurança da API
14.4. Entendendo o fluxo Authorization Code com PKCE
14.5. Criando o serviço de autenticação
14.6. Criando o hook de autenticação
14.7. Validando o login do usuário antes de carregar a aplicação
14.8. Logout programático
14.9. Criando bloqueadores de renderização com base em perfis
14.10. Bloqueando acesso do usuário a determinadas rotas
14.11. Bloqueando ações do usuário com base no perfil

15. Deploy do projeto em produção
15.1. Introdução ao módulo de Deploy
15.2. O que são variáveis de ambiente?
15.3. Fazendo build de um projeto na máquina local
15.4. Analisando o bundle final de um projeto e suas dependências
15.5. Substituindo valores fixos por variáveis de ambiente
15.6. Fazendo deploy de uma SPA no Netlify
15.7. Desafio - Fazer deploy do projeto Admin no Netlify
15.8. Como configurar domínios personalizados no Netlify
15.9. Fazendo deploy de uma app com SSR no Vercel
15.10. Configurando domínios personalizados no Vercel
15.11. O que é CORS?

16. Testes automatizados
16.1. Introdução ao módulo
16.2. O que são testes automatizados?
16.3. O que é o Jest?
16.4. Enzyme vs. RTL
16.5. Garantindo funcionamento de componentes

16.6. Testes unitários vs. Testes de integração
16.7. O que são testes E2E (end-to-end)?
16.8. Criando um teste E2E com Cypress
16.9. Rodando testes antes de commits ou pushs
16.10. Conclusão e próximos passos

** Todas as aulas até o módulo 8 serão entregues imediatamente (mais de 200 aulas). O restante será entregue até dia 30/07/2021. Os nomes e quantidade de aulas dos últimos módulos podem mudar durante as gravações, mas sem alteração de escopo do que será ensinado.*