

CHECKLIST DE HABILIDADES DE UM DESENVOLVEDOR FRONT-END



UDACITY



BEM-VINDO À UDACITY

A Udacity é uma plataforma online e global de aprendizado contínuo que conecta educação e mercado, oferecendo aos estudantes as habilidades que precisam para se preparar para as profissões do futuro, hoje.

Nossos programas Nanodegree proporcionam credenciais reconhecidas pela indústria em cursos online sobre diversos assuntos, de veículos autônomos à inteligência artificial, data science e marketing digital.

Estes cursos são construídos em parceria com as principais empresas globais de tecnologia e negócios, incluindo Google, Facebook, Amazon, IBM Watson e Nvidia, na intenção de fechar as lacunas de talentos do mercado.

Com sede no Vale do Silício, Estados Unidos, também operamos no Brasil, Europa, China, Índia, Egito e nos Emirados Árabes Unidos. São mais de 40 mil estudantes ativos estudando com a Udacity no mundo todo.

O Brasil já representa hoje o segundo maior mercado para a Udacity, atrás apenas dos norte-americanos. Além de mais de 40 profissionais na equipe sediada em São Paulo, temos um time de 250 revisores para cerca de 5,7 mil projetos entregues pelos estudantes brasileiros a cada mês.

ALUNOS EM MAIS DE 160 PAÍSES



CURSOS CRIADOS COM QUEM FAZ O FUTURO



Mercedes-Benz



INTRODUÇÃO

Seja bem-vindo(a) ao checklist de habilidades de um desenvolvedor front-end. Você está prestes a embarcar em uma jornada profissional cheia de oportunidades que possibilitarão melhorar a vida das pessoas e expandir sua criatividade. Construímos este checklist juntamente a potenciais empregadores para que as demandas atuais do mercado estivessem destacadas – e tudo isso foi desenvolvido para você.

Na Udacity, há uma trilha de cursos em front-end feita para que você aprenda as principais habilidades citadas neste guia. Construídos em parceria com as gigantes desta indústria, como Google, GitHub e Hack Reactor, nossos programas Nanodegree são constantemente atualizados para refletir o que há de mais moderno na área.

Os cursos também abrangem diferentes níveis de expertise. Desde aprender os principais conceitos de HTML e CSS, passando por conhecimentos avançados em JavaScript até a especialização em React, uma biblioteca muito popular atualmente e que possibilita desenvolvimento de aplicativos multi-plataforma.

Estamos felizes por você ter dado o primeiro passo para uma carreira em desenvolvimento web front-end! Leia a seguir o checklist e nossas recomendações.





CHECKLIST DE HABILIDADES DE UM DESENVOLVEDOR WEB FRONT-END

Estas são as habilidades e competências importantes para ser um desenvolvedor front-end completo. Reserve um tempo para analisar esta lista com calma. Quantas das opções descritas você domina? Quais você ainda precisa dominar?

CLIQUE EM CADA UM DOS TÓPICOS PARA EXPLORAR MAIS DETALHES.

Habilidades gerais	05
HTML	07
CSS	08
Web design responsivo	10
Framework CSS	11
JavaScript	12
Framework JavaScript	13
Desempenho web	14
Ferramentas de desenvolvimento de navegadores	15
Ferramentas de construção e automação	16
Testes	18
Habilidades pessoais	19
Possibilidades de carreira: freelancer vs. full time	20
Recursos de aprendizado	22
Programas Nanodegree	23
Cursos abertos	25

Em cada item das habilidades, você poderá observar cores diferentes nas caixas de seleção. Essas cores indicam qual o curso da Udacity que ensinam tais habilidades. São quatro possibilidades:

[Nanodegree Fundamentos Web Front-End](#)

[Nanodegree Web Front-End Avançado](#)

[Nanodegree Desenvolvedor React](#)

Todos os programas Nanodegree acima



HABILIDADES GERAIS



Aqui estão as competências necessárias para qualquer desenvolvedor web, seja ele front-end, back-end ou full stack.

Aprender a pesquisar: esta é uma parte essencial para qualquer pessoa que trabalhe com resolução de problemas. Hoje, não é possível desenvolver algo sem fazer uma pesquisa independente e aprender de maneira autodidata. Portanto, saber como extrair os melhores resultados para suas perguntas no Google (ou qualquer outro buscador) e consultar documentações para te ajudar a encontrar soluções para problemas, novas tecnologias, etc., é fundamental. O [Stack Overflow](#) também entra junto neste campo.

HTTP, HTTPS e HTTP/2: HTTP é o protocolo que rege toda a Web. Entender o seu funcionamento é fundamental para qualquer desenvolvedor web.

Estruturas de dados e algoritmos: são a base de qualquer sistema computacional. Saber a diferença entre pilha, fila, lista, árvore, array e em quais momentos utilizar cada uma dessas estruturas é importante. Compreender diferentes algoritmos, determinar o melhor para cada momento e saber como medir sua performance também. Essas competências você adquire com a prática do dia-a-dia de desenvolvimento web, encontrando problemas e desafios que exigem o estudo de novas estruturas de dados e algoritmos. O importante é saber disso para evoluir na carreira todos os dias: você precisará ter um domínio cada vez maior de estruturas de dados e algoritmos para resolver problemas cada vez mais complexos.

Terminal e shell script

Dominar o uso do terminal e criação de shell scripts que interajam com o sistema operacional é fundamental para que você seja capaz de configurar seu ambiente de desenvolvimento local, gerar controle de versão do seu código com o Git e efetuar configurações em servidores, pois grande parte da web roda em servidores Linux (que só podem ser configurados através do terminal e de shell scripts).



Comandos do shell: shell ("casca") é uma interface usada para interagir com o sistema operacional de um computador. A interface pode ser por linha de comando (via terminal) ou por interface gráfica (o que o usuário comum mais utiliza). Neste caso, os comandos shell abaixo são executados por linha de comando.

```
echo  
ls / cd  
pwd  
mkdir  
mv  
curl  
cat / less  
rm / rmdir  
grep / wc  
variáveis de ambiente
```

Controle de versão

Como você mantém seu código em projetos complexos? Softwares de controle de versão, como o Git, auxiliam os desenvolvedores a salvarem e manterem seus códigos mesmo quando o projeto cresce a ponto de ter centenas de outros desenvolvedores envolvidos e dezenas de outros subprojetos relacionados.

Git: o Git é um sistema de controle de versão distribuído.

```
clone  
add  
commit  
push  
pull  
branch  
merge  
rebase  
log
```

GitHub: o GitHub é um serviço de hospedagem baseado na web para repositórios Git que oferece uma série de recursos que dão suporte à colaboração entre os desenvolvedores.

```
forking  
pull requests
```

HTML



HTML é a primeira das três linguagens necessárias para o desenvolvimento de um site (as outras são CSS e JavaScript) e não é uma linguagem de programação. Ele descreve como os elementos em um site devem ser definidos e fornece aos navegadores uma lista de todos os outros arquivos de que os sites precisam. Para facilitar o entendimento, considere o HTML como o projeto técnico de uma casa. É o que determina o tamanho dos ambientes e como cada um será composto – mas não retrata a aparência desses espaços.

Elementos semânticos: um elemento semântico descreve claramente seu significado tanto para o navegador como para o desenvolvedor. Exemplos: artigo e seção, em vez de sempre utilizar `<div>`.

Elementos de bloco: um elemento de nível de bloco ocupa todo o espaço de seu elemento pai.

Elementos inline: um elemento inline ocupa apenas o espaço delimitado pelas tags que o definem.

Formulários: um formulário representa a seção de documento que contém controles interativos capazes de enviar informações a um servidor web.

Inputs: um elemento de input é utilizado para criar controles interativos para formulários na web de forma a aceitar dados do usuário.

CSS



Se o HTML descreve a disposição dos espaços de uma casa, CSS é a representação de como ficará essa casa. O CSS (Cascading Style Sheets, ou folhas estilo cascata) é responsável pela aparência do site e controla cores, fontes e animações de maneira geral. Esta também não é uma linguagem de programação, mas de estilização. É usada para criar e definir o layout e o design de um site.

Tipos de valores de display: a propriedade de display permite que você determine como elementos gráficos irão se comportar na tela. Seus valores mais conhecidos são none, inline, block, inline-block, flexbox e grid.

Box model: o box model define o tamanho da caixa retangular que representa um elemento em um determinado documento.

Posicionamento básico: a propriedade da posição escolhe regras alternativas para elementos de posicionamento.

Static: o posicionamento static permite que o elemento use seu comportamento normal.

Absolut: não deixa espaço para o elemento. Em vez disso, está em uma posição específica em relação a seu antecessor direto ou ao bloco em que está contido.

Fixed: não deixa espaço para o elemento. Em vez disso, a posição é especificada em relação à exibição da tela.

Flexbox: um modo de layout que faz com que a disposição de elementos em uma página se comporte de maneira previsível quando é necessário acomodar diferentes tamanhos de tela e dispositivos de exibição.

Float: especifica que um elemento deve ser retirado do fluxo normal e posicionado à esquerda ou à direita de seu contêiner.

Estilo de fontes e fontes web: os estilos de fontes permitem a alteração da aparência do texto. Já as fontes web possibilitam o carregamento de arquivos de fontes que podem ser utilizados por todos os clientes.

Backgrounds (fundos): possibilitam a definição de uma cor ou imagem para ser usada como fundo de um contêiner.

Pseudos-classes: permitem a seleção de elementos hipotéticos que orbitam outros elementos estabelecidos dentro de seu HTML.

Animações e transições: viabilizam a animação ou definem a transição entre dois estados de um elemento.

Pré-processadores de CSS

Os pré-processadores nos ajudam a otimizar e facilitar a manutenção do trabalho de criação de interfaces com CSS. Eles nos permitem adicionar funcionalidades que não estão disponíveis no CSS puro para criar estruturas de CSS mais legíveis e fáceis de manter. Os principais:

- Sass**
- PostCSS**
- Less**
- Stylus**

Arquitetura de CSS

Há diversas maneiras de estruturar melhor o seu CSS e existem várias arquiteturas (ou guias) de CSS que auxiliam na criação de estilos de forma modular e escalável. Conforme o seu desenvolvimento avança, é importante saber que arquiteturas são essas e quais são as diferenças entre elas. Algumas das principais:

- BEM**
- OOCSS**
- SMACSS**
- SUITCSS**
- Atomic**



WEB DESIGN RESPONSIVO



Abra um site qualquer e diminua o tamanho do navegador. A página ajustou automaticamente o seu conteúdo, de forma que o layout fique proporcional à diminuição? Isso é design responsivo. As pessoas querem que os sites funcionem perfeitamente em celulares, tablets e laptops. É por meio do estudo dos fundamentos do design responsivo que você aprenderá a criar sites que permitem uma experiência linear de navegação, independentemente do dispositivo utilizado.

media queries: permitem que a apresentação do conteúdo seja adaptada – sem que haja necessidade de alteração – a tipos específicos de dispositivos.

Unidades de dimensionamento: o CSS oferece muitas outras unidades de medida além dos pixels (px). Exemplos: em, rem, vw, vh e vmin.

Viewport: é a área visível para usuário de uma página web, ou seja, a janela que torna visível o conteúdo no navegador. Sua tag meta é utilizada para garantir o correto funcionamento responsivo do seu website.

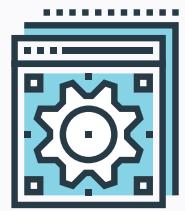
CSS Flexbox: permite projetar a estrutura de um layout responsivo de maneira flexível sem precisar utilizar float ou posicionamento de elementos.

CSS Grid: oferece um sistema de layout baseado em grade, com linhas e colunas, tornando fácil o design de páginas web sem a necessidade de uso de floats e posicionamento de elementos.

Imagens responsivas: permite a definição de imagens que variam de acordo com os diferentes tipos de tela, garantindo a otimização de uso de imagens de diferentes tamanhos de acordo com o tamanho da tela do dispositivo.



FRAMEWORKS CSS



Os frameworks – Bootstrap é um bom exemplo disso – facilitam a estruturação e a criação de sites. Eles fornecem classes CSS customizadas que simplificam a disposição e, independentemente do dispositivo, garantem a qualidade do conteúdo. Os frameworks podem ajudar você a seguir as boas práticas da indústria e os mais modernos fundamentos de design.

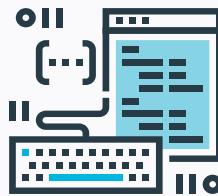
Bootstrap: é uma estrutura CSS originalmente desenvolvida pelo Twitter que simplifica a criação de projetos responsivos.

Materialize CSS: framework de front-end responsivo baseado na linguagem de design Material Design desenvolvida pela Google.

Foundation: tipo de framework CSS desenvolvido pela Zurb, que também serve como facilitador de criação de design responsivo.

Bulma: framework CSS de código aberto baseado no Flexbox

JAVASCRIPT



Das três principais linguagens web, JavaScript é a única linguagem de programação e serve para controlar as interações com o site. É uma ferramenta multitarefas capaz de executar diversas funções em uma página. Para um site simples e estático, não é necessário demandar muito do JavaScript. Mas, para um aplicativo web mais dinâmico, é preciso dominar os recursos oferecidos pela ferramenta.

Sintaxe: as regras que definem como a linguagem deve ser representada.

Tipos de dados: as diferentes variáveis suportadas pela linguagem (por exemplo: strings e números inteiros).

Funções: um bloco de código projetado para executar uma tarefa específica.

Objetos literais: tudo em JavaScript é um objeto, mas escrever seus próprios objetos literais pode tornar os códigos mais simples.

jQuery: jQuery é uma biblioteca extremamente popular que executa cross-browser DOM traversal, manipulação de eventos e AJAX de maneira muito mais simples.

Programação orientada a objetos: o JavaScript oferece várias maneiras de implementar a programação orientada a objetos. Dentre elas: funcional, protótipo e pseudo-classes.

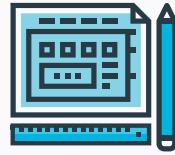
Padrões de design: é uma solução recorrente para um problema comum.

AJAX: o AJAX oferece a capacidade de solicitar dados assíncronos de um servidor web sem a necessidade de recarregar a página.

ECMAScript 6 (ou ES6): nova versão do JavaScript com novas palavras-chave, arrow functions, sintaxe de classe, built-ins, promises, polyfills e outras atualizações.



FRAMEWORKS E BIBLIOTECAS DE JAVASCRIPT



Frameworks e bibliotecas de JavaScript facilitam a criação de seus aplicativos na web ao estipular um padrão organizacional para os vários arquivos com os quais você estiver trabalhando. Também abordam a maioria dos problemas de compatibilidade entre navegadores e incluem várias otimizações de desempenho. Exemplos: Angular, Ember, React e Vue.js.

React: permite a criação de aplicações web de larga escala que utilizam dados que podem mudar com o tempo, sem causar reload de página. Provê velocidade, simplicidade e escalabilidade. É uma biblioteca declarativa, baseada em componentes e de fluxo unidirecional. Rápida curva de aprendizado.

Vue.js: seu foco é deixar as ideias de desenvolvimento de UI (componentes, UI declarativa, hot-reloading, time-travel debugging) mais acessíveis. De fácil aprendizado.

Angular: utiliza data bind bidirecional e permite ampliar o vocabulário HTML para criar aplicativos web front-end.

Ember: elimina a necessidade de código boilerplate por meio do uso de arquivo strict e convenções de nomenclatura.



DESEMPENHO WEB



O que garante que seu site será rápido? O domínio de alguns princípios básicos da renderização de navegadores garantirá que os usuários de seu site tenham uma experiência de navegação satisfatória.

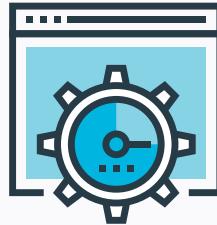
O caminho crítico de renderização: é o processo em que os navegadores transformam seu HTML, CSS e JavaScript nos pixels que são projetados na tela do usuário.

Otimização de imagem: nome dado ao processo de usar tipos de imagens apropriadas ao conteúdo, bem como editar os metadados de um arquivo de imagem.

Minificação de JavaScript: é o processo que remove caracteres desnecessários dos arquivos de JavaScript para torná-los mais leves.



FERRAMENTAS DE DESENVOLVIMENTO DE NAVEGADORES



No desenvolvimento dos aplicativos web, você enfrentará erros na codificação, problemas de desempenho e questões relacionadas à renderização de páginas no navegador – e resolver tudo isso passa pela compreensão da forma que o navegador interpreta os códigos. As ferramentas de desenvolvimento de navegadores são os instrumentos de teste, mensuração e iteração de seus códigos.

Inspeção de elementos: o painel de elementos permite que você veja tudo em uma árvore DOM, além de inspecionar e editar elementos do DOM.

Rede: o painel de rede registra informações sobre cada operação de rede em seu aplicativo.

Timeline: permite gravar e analisar toda a atividade de seu aplicativo à medida em que é executado.

Perfil de aplicativos: o painel de perfil permite observar o uso de memória de seu aplicativo à medida em que é executado.

Recursos: o painel de recursos inspeciona tudo que é carregado no seu aplicativo.



FERRAMENTAS DE DESENVOLVIMENTO E AUTOMAÇÃO



Criar um aplicativo web vai muito além da codificação. É preciso executar conjuntos de testes, otimizar imagens, aderir aos guias de estilo de código da sua empresa e ainda preparar a implantação de tudo em um servidor de produção. É um trabalho extremamente repetitivo. Ferramentas como o Grunt e o Gulp podem cuidar de todas essas tarefas enquanto você se concentra na criação do que realmente importa.

Task runners:

Npm scripts: npm possui um comando de execução que permite rodar scripts definidos na propriedade scripts do arquivo package.json. Largamente utilizados, os scripts do npm permitem automatizar uma grande variedade de tarefas em seu ambiente de desenvolvimento que consomem tempo desnecessário.

Gulp: é um kit de ferramentas para automatizar tarefas que consomem muito tempo em seu fluxo de trabalho como desenvolvedor, permitindo que não gaste seu tempo com não essenciais e com isso possa focar na construção da aplicação.

Grunt: é uma ferramenta de automação permitindo performar tarefas repetitivas como minificação, compilação, testes unitários, linting, etc.

ESLint: é uma ferramenta de utilitários que faz o lint de JavaScript. Linting de código é um tipo de análise usada para encontrar padrões problemáticos ou código que não adere ao guia de estilo determinado. Ela é muito útil para garantir um código legível e padronizado.

Bundlers

Webpack: é um bundler de módulos. Seu propósito principal é fazer o bundle de arquivos JavaScript para uso em um browser. Ele ainda é capaz de transformar, fazer o bundle, ou empacotar qualquer recurso ou asset além de JavaScript.

Browserify: permite que você organize códigos JavaScript seguindo o padrão CommonJS, implementando o lado do cliente em JavaScript como se fosse código Node.js (no caso, utilizado no lado do servidor).

Rollup: é um budler de módulos para JavaScript que compila pequenos pedaços de código em algo maior e mais complexo, como um biblioteca ou aplicação. Ele usa um formato padronizado de módulos de código no padrão ES6.



Parcel: empacota aplicações web de maneira rápida e sem configuração. Utiliza processos que habilitam a compilação multicore e faz cache do sistema de arquivos para reconstruir mesmo após um reinício.

Gerenciador de pacotes:

Npm: é o gerenciador de pacotes para o Node.js, que por sua vez é a plataforma em que a maioria das ferramentas de automação e desenvolvimento foram escritas.

Bower: É um gerenciador de pacotes para bibliotecas HTML, CSS e JavaScript que permite que você defina a versão e recupere suas dependências.

TESTES



À medida que seu aplicativo se torna mais complexo, fica mais fácil ter bugs ou a quebra de uma funcionalidade já existente. Unidade, integração e teste de comportamento são excelentes maneiras de garantir que não haja problemas quando você estiver adicionando novos recursos. Mocha e Jasmine são excelentes exemplos de ferramentas de teste.

Jest: é um framework de testes do Facebook feito para testar todo o código JavaScript, incluindo aplicações React. Uma das filosofias do Jest é prover uma experiência integrada de “zero-configuração”. Observa-se que, quando engenheiros recebem ferramentas prontas para uso, eles acabam escrevendo mais testes, o que resulta em uma base de código mais estável e saudável.

Enzyme: é um pacote de utilidades de testes para React criada pelo Airbnb que torna mais fácil declarar, manipular e atravessar os output de componentes React. Comumente, Enzyme é muito utilizado junto com Jest.

Jasmine: é um framework de testes do tipo BDD (Behavior-Driven Development) para JavaScript. Ele é independente de qualquer framework de JavaScript e não requer o DOM. Possui uma sintaxe simples, limpa que permite que você escreva testes de maneira fácil.

Mocha: o Mocha é uma estrutura de testes assíncronos do JavaScript executada no Node.js e no navegador.



HABILIDADES PESSOAIS



Ter as habilidades técnicas necessárias é importante, mas são as competências comportamentais que permitem que você tenha um desempenho de alto nível, seja trabalhando em equipe ou por conta própria.

Cultivar sua capacidade de comunicar, ensinar e inspirar quem está ao seu redor e ver as coisas de maneira mais holística são exemplos de habilidades pessoais essenciais para avançar no carreira.

Capacidade de comunicação: os trabalhos mais cobiçados envolvem transformar o que é complexo em algo digerível, seja para colegas (pense em um cientista de dados apresentando seus insights para a empresa) ou para clientes (pense num pitch de vendas de uma tecnologia complexa para as massas). Saiba transmitir os objetivos, os progressos alcançados e as questões mais importantes para todos os envolvidos no processo.

Agilidade na tomada de decisões: desconstrua os grandes problemas em pequenas etapas e torne-os mais facilmente solucionáveis.

Paixão pelo trabalho e adaptabilidade: mantenha-se atualizado sobre tudo o que acontece dentro do segmento e fique atento aos avanços tecnológicos. Consiga tempo para ler relatórios e notícias, participar de discussões online e estar presente em eventos importantes.

Automotivação: arrisque-se, explore e experimente o novo. Mas, sobretudo, mantenha a persistência perante as eventuais dificuldades.

Flexibilidade: horários flexíveis e trabalho remoto não são mais exceções: não faltam exemplos de empresas de todo porte e indústria que adotaram esse modelo, de grandes escritórios de advocacia às startups de realidade aumentada do Vale do Silício. O que essa nova realidade exige do candidato, no entanto, é um alto grau de flexibilidade, que vem envolvida em autoliderança, automotivação e autonomia para focar no trabalho e priorizar (por conta própria) o que precisa ser feito.



POSSIBILIDADES DE CARREIRA: FREELANCER VS. FULL TIME



Desenvolvedores especializados em front-end são novidade no mercado: até poucos anos atrás, a categoria se resumia simplesmente ao desenvolvimento web, termo que incluía tanto front-end quanto back-end.

Com o avanço das tecnologias e a demanda crescente por aplicações cada vez mais sofisticadas, tornou-se necessário ter pessoas especializadas em criar experiências cada vez mais interativas, eficientes e interessantes.

Hoje, as oportunidades de trabalho podem ser tanto para freelancers (aqueles que trabalham de maneira independente) quanto para full time (quem trabalha dentro de uma empresa em tempo integral). Saber quais são os prós e contras de cada um é essencial para descobrir qual combina melhor com seu momento.

DESENVOLVEDORES FRONT-END FREELANCERS

Freelancers frequentemente trabalham com pequenos negócios e indivíduos que precisam criar novos sites ou atualizar sites que já existem. Para encontrar esse tipo de trabalho e se conectar com clientes, há plataformas como [gun.io](#) e [upwork.com](#) (nos EUA) e [trampos.co](#) e [99freelans.com.br](#) (no Brasil).

Com sua independência como freelancer, você terá a oportunidade de usar as tecnologias e ferramentas que quiser. É uma chance de testar as novidades e expandir seu conjunto de habilidades em front-end na prática.

O outro lado da moeda é que um freelancer precisa ser seu próprio gestor, assistente, vendedor e contador. Além de fazer o desenvolvimento em si, você precisará investir tempo gerindo seus relacionamentos com clientes e buscando novas oportunidades.

Para um desenvolvedor front-end freelancer, é essencial manter um portfólio detalhado. Não há maneira melhor de explicar para um cliente em potencial que você é perfeito para o trabalho do que mostrar do que você é capaz.

Quer ser freelancer? Construir um belo portfólio cheio de sites que lhe dão orgulho é o primeiro passo.



DESENVOLVEDORES FRONT-END FULL TIME

Desenvolvedores front-end que estão dentro de empresas em tempo integral constroem sites em equipe. Para ter sucesso, os membros dessa equipe precisam de ótimas habilidades de comunicação e organização.

Como parte de uma empresa, você construirá produtos que a ajudam a conquistar seus objetivos. Encontrar uma companhia que tenha uma missão que você apoia tornará esse trabalho mais gratificante e divertido. Além disso, conforme você delineia o futuro do seu produto, também ganhará conhecimentos profundos sobre ele.

Fazer parte de uma equipe grande significa que você precisa especializar suas habilidades de desenvolvimento web. Se está trabalhando em uma empresa que desenvolve web apps, por exemplo, vai encontrar desenvolvedores front-end que trabalham principalmente com JavaScript. Já outros se especializam em HTML e CSS porque passam seu tempo focados em diagramar e customizar websites.

Em comum com os freelancers, desenvolvedores front-end que querem um emprego fixo numa empresa precisam construir a mesma coisa: um ótimo portfólio para demonstrar suas habilidades.

COMO AUMENTAR SUAS CHANCES DE SUCESSO

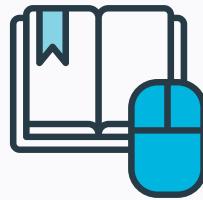
Seja qual for seu caminho de escolha, conseguir trabalho como desenvolvedor front-end significa estar sempre atualizado com as novas tecnologias e experimentar toda e qualquer novidade. Leia documentações, faça experimentos e entenda o que funciona bem em seus projetos para continuar à frente desta área em evolução contínua.

RECURSOS DE APRENDIZADO





RECURSOS DE APRENDIZADO



Parabéns! Você chegou ao fim da lista.

Se você conseguiu identificar habilidades que já possui ou se vai começar a explorar a lista desde o primeiro item, orgulhe-se: você já está progredindo. E, como mencionamos no início, estamos aqui para guiá-lo nessa jornada rumo ao aprendizado.

PROGRAMAS NANODEGREE

A Udacity preparou um trilha de cursos online para que você se torne um desenvolvedor web front-end completo. Durante os programas Nanodegree, você realizará projetos colocando em prática tudo o que aprendeu e receberá feedback de especialistas para acelerar sua curva de aprendizado. Ao final dos cursos, você terá um portfólio completo para mostrar suas habilidades ao mercado.



[Nanodegree Fundamentos Web Front-End](#) construído em parceria com Google, GitHub e Hack Reactor.

Você vai aprender a criar e executar interfaces web responsivas, escaláveis e de ótima performance com linguagens modernas.

- ✓ Domine HTML e CSS para desenvolver sites responsivos e otimizados para todos os tipos de tela;
- ✓ Aprenda os fundamentos de JavaScript, sintaxe do ES6 e aprenda a manipular a DOM com jQuery;
- ✓ Controle as versões de código do seu projeto com Git e crie um repositório no GitHub;
- ✓ Construa seu site de portfólio durante o curso para evidenciar suas novas habilidades ao mercado.



Nanodegree Web Front-End Avançado construído em parceria com Google e GitHub.

Você vai aprender as melhores práticas em front-end e as habilidades mais requisitadas neste mercado.

- ✓ Otimize seu código utilizando ferramentas web e automações
- ✓ Descubra como realizar testes estruturados no seu código JavaScript;
- ✓ Utilize APIs para fazer requisições avançadas e manipular os dados de suas aplicações;
- ✓ Construa aplicações web que funcionam offline;
- ✓ Tenha seu primeiro contato com React - poderosa biblioteca JavaScript de UI criada pelo Facebook.



Nanodegree Desenvolvedor React.

React está transformando o desenvolvimento front-end. Domine essa incrível biblioteca de UI do Facebook e torne-se um desenvolvedor web e mobile.

- ✓ Aprenda os fundamentos desta biblioteca de UI, criada pelo Facebook
- ✓ Organize sua interface por componentes reutilizáveis, o que aumentará sua organização e produtividade
- ✓ Use o React Router para adicionar diferentes rotas aos aplicativos - livrando-se da preocupação de fluxos de links
- ✓ Saiba quando usar Redux - especialmente para gerenciar estados de aplicações complexas



- ✓ Aumente sua versatilidade com React Native, preparando aplicativos também para Google Play Store e Apple App Store

CURSOS ABERTOS

Se estiver em busca de um plano de aprendizado com uma demanda menor de tempo ou algo que preencha uma lacuna específica dentro de suas habilidades atuais, você pode acessar nossos cursos abertos:

- ✓ **Introdução a HTML e CSS** - Aprenda a converter modelos de design em páginas web estáticas e crie um site de portfólio responsivo.
- ✓ **Introdução a JavaScript** - Aprenda a sintaxe do JavaScript e as convenções de codificação que os desenvolvedores web utilizam para criar sites interativos e dinâmicos. Isso ajudará a enriquecer seu currículo e seu portfólio.
- ✓ **Introdução a jQuery** - Aprenda a acessar e modificar o DOM usando jQuery! Esse curso o ensinará a utilizar os recursos essenciais como seleção de elementos DOM, travessia e manipulação.
- ✓ **JavaScript Orientado a Objetos** - Saiba como utilizar os vários recursos de programação orientada a objeto no JavaScript e, principalmente, como escrever bibliotecas reutilizáveis e sustentáveis que irão tornar a sua vida mais fácil.
- ✓ **HTML5 Canvas** - Saiba como usar HTML5 Canvas para criar e modificar imagens e animações interativas.
- ✓ **Otimização de Performance de Website** - Saiba como os navegadores convertem HTML, CSS e JavaScript em sites enquanto você mede e otimiza a velocidade usando as ferramentas de desenvolvedor do Google Chrome.
- ✓ **Introdução ao AJAX** - Aprenda a fazer requisições assíncronas com JavaScript usando a funcionalidade AJAX do jQuery. Realize também requisições ao criar um aplicativo web com dados do Google Street View, do jornal The New York Times e da Wikipédia.



UDACITY

CHECKLIST DE HABILIDADES
DE UM DESENVOLVEDOR
FRONT-END