

iOS

TechGuide - Alura, FIAP e PM3

Nível 1

Swift - Fundamentos:

- Swift é uma linguagem de programação de código aberto desenvolvida pela Apple Inc. e pela comunidade de desenvolvedores. É projetada para ser segura, rápida e intuitiva. Swift é a linguagem preferida para o desenvolvimento de aplicativos iOS, macOS, watchOS e tvOS. Veja as principais habilidades que você precisa desenvolver:
- Conhecer as palavras chaves do Swift;
- Entender como funciona o controle de fluxo e as condicionais;
- Construir aplicações de terminal usando Swift;
- Conhecer os protocolos de Encodable e Decodable;
- Compreender e aprender a usar coleções no Swift.

Conteúdos

- **Site** Começando no Swift (inglês) (<https://www.swift.org/getting-started/>)
- **YouTube** Playlist do Youtube: Fundamentos do Swift | Michel Lopes (<https://www.youtube.com/watch?v=e6elPukdnRk&list=PLs5gUvDGMKI8DF0rOLFMZk9SYqX4AN2fi>)
- **YouTube** Swift (a linguagem do iOS) // Dicionário do Programador | Código Fonte TV (<https://youtu.be/ELa-PgWIYDI>)
- **YouTube** Diferenças e Curiosidades de Class e Struct no Swift 🇧🇷 | Michel Lopes (<https://youtu.be/NsMJb43X7jE>)
- **YouTube** [SWIFT] STRUCT OU CLASS, PRA QUE SERVE? | DevPoli (https://youtu.be/tKSNjg9Cb_g)
- **YouTube** (Playlist) Curso de Swift do Zero | Tiago Aguiar (<https://www.youtube.com/watch?v=MnL3-a3eKNQ&list=PLJ0AcghBBWShgIH122uw7H9T9-NlaFpP-&index=1>)
- **Artigo** Coleções Sets, explorando seu poder e peculiaridades. | Medium (<https://medium.com/@msmatheusmatos/cole%C3%A7%C3%B5es-sets-explorando-seu-poder-e-peculiaridades-526ee361384a>)

Conteúdos Alura:

- **Artigo** O que é iOS: o sistema operacional dos dispositivos móveis da Apple (<https://www.alura.com.br/artigos/o-que-e-ios>)
- **Artigo** Closures no Swift: o que são e como utilizá-las (<https://www.alura.com.br/artigos/closures-swift-como-utiliza-las>)
- **Curso** Domine a linguagem Swift (<https://cursos.alura.com.br/formacao-domine-linguagem-swift>)

iOS - UIKit - Fundamentos:

- O UIKit é a biblioteca de código padrão para aplicativos iOS. Ele fornece uma ampla gama de recursos para criar interfaces de usuário, incluindo views, controles e animações. O UIKit é uma

ótima maneira de criar aplicativos iOS que sejam visualmente atraentes e fáceis de usar. Veja as principais habilidades que você precisa desenvolver:

- Conhecer Swift ou Objective-C que são as linguagens para desenvolver apps iOS;
- Entender como o SDK do iOS funciona (Core Foundation e Cocoa Touch);
- Conhecer os componentes de interface do UIKit para criação de layouts;
- Dominar a criação de interfaces utilizando o UIKit, tanto através de código (ViewCode) quanto pelo uso de Storyboards;
- Explorar os princípios de Auto Layout e como criar layouts flexíveis e responsivos para diferentes tamanhos de tela;
- Implementar a navegação entre telas e a passagem de dados utilizando o sistema de view controllers do UIKit.

Conteúdos

- **YouTube** Introdução à interface programática - UIKit | Xcode 14 (inglês)
(https://youtu.be/U6_I58Cv4E)
- **YouTube** Desenvolvimento iOS UIKit | freeCodeCamp.org (inglês)
(<https://youtu.be/KCgYDCKqato>)
- **Site** Documentação oficial Apple - UIKit (inglês)
(<https://developer.apple.com/documentation/uikit>)
- **Site** Documentação oficial Apple - Cocoa Touch (inglês)
(<https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/CocoaFundamentals/Introducti>)
- **Site** Documentação oficial Apple - Core Foundation (inglês)
(<https://developer.apple.com/documentation/corefoundation>)

Conteúdos Alura:

- **Artigo** iOS e Swift: Diferenças na construção de layouts com Storyboard, XIB e View Code
(<https://www.alura.com.br/artigos/ios-swift-diferencas-construcao-layouts-storyboard-xib-view-code>)
- **YouTube** Desenvolvimento iOS pra iniciantes com Giovanna Moeller | #HipstersPontoTube
(<https://youtu.be/MGIs7DSQ-0A>)
- **YouTube** Desenvolvimento iOS com Giovanna Moeller | #HipstersPontoTube
(https://youtu.be/4AH9H_dx_7g)
- **Curso** iOS: construindo seu primeiro aplicativo (<https://cursos.alura.com.br/course/ios-construindo-primeiro-aplicativo>)

Conceitos de Orientação a Objetos:

- A Programação Orientada a Objetos é um paradigma de programação de software baseado na composição e interação entre diversas unidades chamadas de 'objetos' e as classes, que contêm uma identidade, propriedades e métodos). Ela é baseada em quatro componentes da programação: abstração digital, encapsulamento, herança e polimorfismo.
- Como funcionam objetos
- Criar e utilizar construtores
- O que são classes
- Criar e utilizar métodos
- Como funciona encapsulamento
- O que é herança
- O que é polimorfismo

- Como funcionam interfaces
- O que são abstrações

Conteúdos

- **Artigo** Princípios básicos da Programação Orientação a Objetos (POO) (<https://yanborowski.medium.com/princ%C3%ADpios-b%C3%A1sicos-da-programa%C3%A7%C3%A3o-orienta%C3%A7%C3%A3o-a-objetos-poo-62da3998b7ce>)
- **YouTube** FernandaDev: O que é Programação Orientada a Objetos?(POO) (<https://www.youtube.com/watch?v=QJjY2TNyl-8>)
- **YouTube** Giuliana Bezerra: Desmistificando os tipos de referência - Ep. 8 (<https://youtu.be/D9hANYbPYCc>)

Conteúdos Alura:

- **Podcast** Hipsters.tech: TechGuide - Orientação a Objetos – Hipsters Ponto Tech #350 (<https://www.hipsters.tech/techguide-orientacao-a-objetos-hipsters-ponto-tech-350/>)
- **Artigo** POO: o que é programação orientada a objetos? (<https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos>)
- **Site** Apostila: Java e Orientação a Objetos (<https://www.alura.com.br/apostila-java-orientacao-objetos>)
- **Site** Livro: Orientação a Objetos (<https://www.casadocodigo.com.br/products/livro-oo-conceitos>)
- **YouTube** Alura: Orientação a objetos (com Roberta Arcoverde) (<https://www.youtube.com/watch?v=jpuJ1grluoU>)
- **YouTube** Alura: Imersão Java: Orientação a Objetos, APIs e além (e live coding!) (<https://www.youtube.com/watch?v=WdT90ffB-0Q>)
- **Curso** Formação Java e Orientação a Objetos (<https://cursos.alura.com.br/formacao-java>)
- **Curso** Formação C# e Orientação a Objetos (<https://cursos.alura.com.br/formacao-c-sharp-orientacao-objetos>)
- **Curso** Formação Python e orientação a objetos (<https://cursos.alura.com.br/formacao-Python-linguagem>)
- **Curso** Curso Orientação a Objetos com PHP: Classes, métodos e atributos (<https://www.alura.com.br/curso-online-php-oo-classes-metodos-atributos>)
- **Curso** Curso JavaScript: programação orientada a objetos (<https://www.alura.com.br/curso-online-javascript-passos-programacao-orientada-objetos>)

iOS - Storyboard:

- Um storyboard no iOS é um arquivo de XML que contém um diagrama visual de uma interface do usuário. Os storyboards são usados para criar aplicativos iOS que são visualmente atraentes e fáceis de navegar. Os storyboards contêm uma hierarquia de views, que são os elementos visuais que compõem a interface do usuário. As views podem ser adicionadas aos storyboards arrastando-as da paleta de views para o diagrama. As views podem ser personalizadas alterando suas propriedades, como tamanho, cor e fonte. Veja as principais habilidades que você precisa desenvolver:
- Conhecer o conceito de Storyboards e como eles são usados para criar aplicativos iOS;
- Criar e usar Storyboards para criar layouts de interface do usuário;
- Implementar a navegação entre telas em aplicativos iOS;
- Aprofundar os princípios de Auto Layout e como criar layouts flexíveis e responsivos para diferentes tamanhos de tela.

Conteúdos

- **YouTube** Introdução à interface programática - UIKit | Xcode 14 (inglês) (https://youtu.be/U6_I58Cv4E)
- **Site** Documentação oficial Apple - Storyboard (inglês) (<https://developer.apple.com/library/archive/documentation/General/Conceptual/Devpedia-CocoaApp/Storyboard.html>)
- **YouTube** iOS Developer Legacy: A linha do tempo com Storyboard e UIKit | Tiago Aguiar (<https://youtu.be/5WSqsnWw80Q>)

Conteúdos Alura:

- **Artigo** iOS e Swift: Diferenças na construção de layouts com Storyboard, XIB e View Code (<https://www.alura.com.br/artigos/ios-swift-diferencas-construcao-layouts-storyboard-xib-view-code>)
- **Curso** iOS: auto layout com Constraints (<https://cursos.alura.com.br/course/ios-auto-layout-constraints>)
- **Curso** iOS layout: mais recursos para criar telas responsivas (<https://cursos.alura.com.br/course/ios-layout-parte-2-recursos-telas-responsivas>)

iOS - ViewCode:

- O método de viewcode no UIKit é uma abordagem para a criação de interfaces de usuário que envolve a codificação da interface de usuário diretamente em Swift, em vez de usar um storyboard ou nib. O método de viewcode pode ser uma ótima maneira de criar interfaces de usuário mais personalizadas e eficientes, mas é importante notar que ele pode ser mais complexo do que usar um storyboard ou nib. Veja as principais habilidades que você precisa desenvolver:
- Conhecer o conceito de ViewCode e como ele é usado para criar aplicativos iOS;
- Criar e usar ViewCode para criar layouts de interface do usuário;
- Explorar os princípios de Auto Layout com ViewCode.

Conteúdos

- **Artigo** View Code em Swift (<https://medium.com/mackmobile/view-code-em-swift-6026f42bf780>)
- **Artigo** Manual do View Code para desenvolvedores iOS (<https://betterprogramming.pub/ios-view-codes-handbook-1a08e28b0420>)
- **YouTube** COMO criar uma TELA usando ViewCode/iOS/SWIFT, Criando ViewControllers | Swift para todos (<https://youtu.be/iFzTBIRVH6s>)
- **YouTube** Como utilizar STACK/UISTACKVIEW, tutorial em SWIFT utilizando VIEWCODE - Desenvolvimento Mobile iOS | Swift para todos (<https://youtu.be/KyZZ5QthY4A>)
- **YouTube** COMO criar COMPONENTES em SWIFT utilizando VIEWCODE, Criando seu aplicativo do zero com SWIFT | Swift para todos (<https://youtu.be/zSzxFillumts>)

Conteúdos Alura:

- **Curso** Formação: Desenvolva aplicativos iOS utilizando view code (<https://cursos.alura.com.br/formacao-aplicativos-ios-view-code>)

iOS - Imagens e Ícones:

- Os recursos do app em iOS são os ativos visuais que são usados para representar o seu app. Estes recursos incluem ícones, imagens, fontes e outros tipos de arquivos. Os recursos do app são importantes porque eles ajudam os usuários a identificar e se conectar com o seu app. Eles

também podem ser usados para melhorar a estética do seu app e torná-lo mais atraente para os usuários. Veja as principais habilidades que você precisa desenvolver.

- Conhecer os diferentes tipos de recursos de aplicativos iOS, como imagens, ícones e outros recursos;
- Adicionar e usar imagens, ícones e outros recursos em aplicativos iOS;
- Personalizar recursos de aplicativos iOS para diferentes dispositivos e resoluções de tela.

Conteúdos

- **YouTube** Adicionando UIImage na UIView programaticamente usando swift (inglês) | Anil Kumar Giri (<https://youtu.be/Qufbjtda4iQ>)
- **YouTube** Como carregar uma imagem por uma URL no UIImageView (inglês) | Paul Hudson (<https://youtu.be/8mBexDIgkI4>)
- **YouTube** SFSymbols - Why use them? (inglês) | CodeWithChris (<https://youtu.be/HjX4cAcvAWc>)

iOS - Persistência de dados:

- A persistência de dados em iOS é o processo de armazenar e recuperar dados em um dispositivo iOS. Isso pode ser feito usando uma variedade de métodos, incluindo arquivos, Core Data e Realm. Veja as principais habilidades que você precisa desenvolver:
- Conhecer os diferentes tipos de persistência de dados no iOS, como arquivos, Core Data, SwiftData e Realm;
- Armazenar e recuperar dados em um aplicativo iOS usando arquivos, Core Data, SwiftData e Realm;
- Personalizar a persistência de dados em um aplicativo iOS para diferentes necessidades.

Conteúdos

- **Site** Documentação oficial Realm Database (inglês) (<https://www.mongodb.com/docs/realm/introduction/>)
- **Site** Documentação oficial Apple - Core Data (inglês) (<https://developer.apple.com/documentation/coredata>)
- **Site** Documentação oficial Apple - Conheça SwiftData (inglês) (<https://developer.apple.com/videos/play/wwdc2023/10187/>)
- **YouTube** Aprenda a usar o Core Data no iOS com Swift | Mobilizando (<https://youtu.be/bKCpT-mAgZY>)
- **YouTube** SwiftData Relationships & UINavigationController | Build a Full SwiftData App (inglês) | Flo writes Code (<https://youtu.be/al9dJ8SMZzY>)

Conteúdos Alura:

- **Curso** iOS: usando recursos nativos (Core Data e Biblioteca de fotos) (<https://cursos.alura.com.br/course/ios-usando-recursos-nativos>)

iOS - SwiftUI:

- SwiftUI é uma biblioteca de interface de usuário declarativa para criar aplicativos iOS, iPadOS, macOS, watchOS e tvOS. Ele é construído sobre o UIKit e o AppKit, mas fornece uma maneira mais declarativa de criar interfaces de usuário. Isso torna mais fácil criar interfaces de usuário bonitas e envolventes. Veja as principais habilidades que você precisa desenvolver:
- Conhecer SwiftUI e como usá-lo para criar interfaces de usuário para aplicativos iOS;
- Usar combinações de componentes para criar layouts;
- Usar os métodos de navegação para adicionar mais versatilidade para seu app.

Conteúdos

- **Site** Documentação oficial Apple - SwiftUI (inglês) (<https://developer.apple.com/documentation/swiftui>)
- **YouTube** Introdução ao SwiftUI - Construindo aplicativos iOS | Girl Coding (<https://youtu.be/4eC49rYeugs>)
- **Site** Construindo Listas e navegações - Swift Tutorials (inglês) (<https://developer.apple.com/tutorials/swiftui/building-lists-and-navigation>)
- **YouTube** SwiftUI Tutorial Básico (inglês) | Sean Allen (<https://youtu.be/HXoVSbwWUIk>)
- **YouTube** (Playlist) Como Criar Um Aplicativo iOS completo (e simples) com SwiftUI | Tiago Aguiar (<https://www.youtube.com/watch?v=Gt75k60tMjc&list=PLJ0AcghBBWSgknzUhjYpPkLijK4Ny3WF>)

Conteúdos Alura:

- **Artigo** Vale a pena aprender SwiftUI hoje? (<https://www.alura.com.br/artigos/vale-a-pena-aprender-swift-ui-hoje>)
- **Curso** Formação: Construa aplicativos iOS com SwiftUI (<https://cursos.alura.com.br/formacao-ios-swiftui>)
- **Curso** Formação Evolua Apps em SwiftUI: CRUD, MVVM e Autenticação (<https://www.alura.com.br/formacao-apps-swiftui-crud-mvvm-autenticacao>)

Xcode:

- O Xcode é a ferramenta que os desenvolvedores usam para criar apps para o ecossistema da Apple – MacOS, iOS e todos os produtos da Apple.
- Conhecer a ferramenta Xcode;
- Criar um novo projeto;
- Usar o emulador;
- Debugar seu código;
- Debugar é o processo de encontrar e corrigir erros em um aplicativo. É uma parte essencial do desenvolvimento de software e é importante saber como fazer debugging de forma eficaz. O Xcode é o IDE oficial da Apple para desenvolvedores iOS. Ele inclui uma série de ferramentas integradas que podem ser usadas para debugging de aplicativos, incluindo um depurador, um analisador de memória e um analisador de desempenho. Veja as principais habilidades que você precisa desenvolver:
- Conhecer as ferramentas e técnicas mais comuns para debugging de aplicativos iOS;
- Usar as ferramentas e técnicas mais comuns para debugging de aplicativos iOS para encontrar e corrigir erros.

Conteúdos

- **Site** Apple: Kit do Xcode (<https://www.apple.com/br/education/docs/xcode-guide.pdf>)
- **Artigo** Como baixar e instalar o Xcode no seu Mac e como atualizá-lo para o desenvolvimento em iOS (<https://www.freecodecamp.org/portuguese/news/como-baixar-e-instalar-o-xcode-no-seu-mac-e-como-atualiza-lo-para-o-desenvolvimento-em-ios/>)
- **Artigo** Não te ensinaram a usar o debugger | Medium (<https://medium.com/@dinhani/debugando-com-o-debugger-aba48a1de694>)
- **YouTube** Tiago Aguiar: Xcode - Como criar um aplicativo iOS do zero (https://www.youtube.com/watch?v=t8JI3Ur51zY&ab_channel=TiagoAguiar)
- **YouTube** Dicas para debug no Xcode 13 (inglês) | iOS Academy (<https://youtu.be/ZAqnJQn7xp4>)

- **YouTube** XCode Breakpoints - Debug básico (inglês) | Sean Allen (<https://youtu.be/qPWfOkHcKdU>)
- **YouTube** Improve Debugging Skills | iOS | Swift/Obj C | Xcode (inglês) | iCode (<https://youtu.be/ipuxycB1ewY>)
- **Artigo** 11 atalhos que todo desenvolvedor iOS deve saber para ser mais produtivo no Xcode | Tiago Aguiar (<https://tiagoaguiar.co/8-xcode-shortcuts-teclas-de-atalho>)

Estruturas de Dados:

- No contexto dos computadores, uma estrutura de dados é uma forma específica de armazenar e organizar os dados na memória do computador para que esses dados possam ser facilmente recuperados e utilizados de forma eficiente quando necessário posteriormente.
- Conhecer as principais estruturas de dados
- Implementar as principais estruturas de dados

Conteúdos

- **Artigo** Estrutura de dados (<https://medium.com/php-word/estrutura-de-dados-6afb386a6fc1>)
- **Artigo** Estrutura de Dados e Algoritmo - parte 1 (<https://medium.com/@PaoloProdossimoLopes/01-estrutura-de-dados-e-algoritmo-9cb6a9b14aa>)
- **Artigo** Estrutura de Dados e Algoritmo - parte 2 (<https://medium.com/@PaoloProdossimoLopes/02-estrutura-de-dados-e-algoritmo-deba0b384a0f>)
- **YouTube** Mario Souto - Dev Soutinho: Estruturas de dados com JavaScript (<https://www.youtube.com/watch?v=MweeZn1rR8s>)
- **YouTube** Fabio Akita: Árvores (<https://www.youtube.com/watch?v=9GdesxWtOgs>)
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Lista Encadeada (https://www.youtube.com/watch?v=RW0oD2L_tSg)
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Filas (https://www.youtube.com/watch?v=RdoLTgJSV_c)
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Pilhas (<https://www.youtube.com/watch?v=ZdU4wMyiTSs>)

Conteúdos Alura:

- **Site** Apostila Python: Estruturas de dados (<https://www.alura.com.br/apostila-python-orientacao-a-objetos/estrutura-de-dados>)
- **Artigo** Estruturas de dados: uma introdução (<https://www.alura.com.br/artigos/estruturas-de-dados-introducao>)
- **Artigo** Curso Estrutura de Dados: computação na prática com Java (<https://www.alura.com.br/artigos/estrutura-dados-computacao-na-pratica-com-java>)
- **Podcast** Hipsters.tech: Algoritmos e estrutura de dados - Hipsters 186 (<https://www.alura.com.br/podcast/hipsterstech-algoritmos-e-estrutura-de-dados-hipsters-186-a375>)
- **Curso** Curso PHP: dominando as Collections (<https://cursos.alura.com.br/course/php-dominando-collections>)
- **Curso** Curso Python: crie a sua primeira aplicação (<https://cursos.alura.com.br/course/python-crie-sua-primeira-aplicacao>)
- **Desafio** 7 Days of Code: Estruturas de Dados (<https://7daysofcode.io/matricula/estruturas-de-dados>)

Nível 2

iOS - Assíncrono:

- Assincronicidade é uma técnica de programação que permite que as tarefas sejam executadas em segundo plano, enquanto a interface do usuário continua a ser executada normalmente. Isso pode melhorar o desempenho dos aplicativos, pois evita que eles travem ou fiquem lentos enquanto aguardam a conclusão de tarefas demoradas. Uma das maneiras mais comuns de usar assincronicidade em iOS é usar os métodos `async` e `await`. Os métodos `async` e `await` permitem que você execute código em segundo plano e retorne para o código principal antes que a tarefa seja concluída. Isso pode ser útil para tarefas que demoram muito para serem concluídas, como baixar um arquivo grande ou processar uma imagem grande. Veja as principais habilidades que você precisa desenvolver:
- Conhecer os fundamentos da assincronicidade em iOS;
- Usar as classes `OperationQueue` e `async` e `await` para executar tarefas em segundo plano.

Conteúdos

- [Site](https://developer.apple.com/documentation/foundation/operationqueue) Documentação oficial Apple - Classe `OperationQueue` (<https://developer.apple.com/documentation/foundation/operationqueue>)
- [YouTube](https://youtu.be/zLfXPSeCk8) Programação assíncrona vs Programação paralela - Entenda a diferença | Programando com Vinicius Dias (<https://youtu.be/zLfXPSeCk8>)
- [YouTube](https://youtu.be/uWqy5KZXsIA) Async/Await Concorrência com Swift (inglês) | Swift Heroes (<https://youtu.be/uWqy5KZXsIA>)
- [YouTube](https://www.youtube.com/watch?v=X9H2M7xMi9E&list=PLSbpzz0GJp5RTjum9gWTqPhM4L3Kop0S&pp=iAQB) Playlist Concorrência em Swift (inglês) | iCode (<https://www.youtube.com/watch?v=X9H2M7xMi9E&list=PLSbpzz0GJp5RTjum9gWTqPhM4L3Kop0S&pp=iAQB>)

iOS - Comunicação com APIs:

- As comunicações API em iOS são o processo de comunicação entre um aplicativo iOS e um serviço externo. Isso pode ser feito usando uma variedade de métodos, incluindo HTTP, HTTPS e Sockets. As comunicações API são importantes porque elas permitem que os aplicativos iOS acessem dados e recursos que não estão disponíveis no dispositivo local. Elas também podem ser usadas para compartilhar dados e recursos entre aplicativos iOS e serviços externos. Veja as principais habilidades que você precisa desenvolver:
- Conhecer os diferentes tipos de APIs que podem ser usadas em aplicativos iOS;
- Conhecer e utilizar bibliotecas/ferramentas para facilitar a comunicação a APIs, como o Alamofire, `URLSession` e Codable;
- Escolher e usar APIs que sejam adequadas para as necessidades do seu aplicativo;
- Testar comunicações API em diferentes condições para garantir que elas funcionem corretamente.

Conteúdos

- [YouTube](https://youtu.be/5hcHbhlWleI) iOS Alamofire Tutorial (inglês) | Code Pro (<https://youtu.be/5hcHbhlWleI>)
- [Site](https://alamofire.github.io/Alamofire/) Documentação oficial Alamofire (inglês) (<https://alamofire.github.io/Alamofire/>)
- [YouTube](https://youtu.be/MBCX1atOvdA) Enviando e recebendo dados com `URLSession` e protocolo Codable (inglês) | Paul Hudson (<https://youtu.be/MBCX1atOvdA>)
- [Artigo](https://medium.com/codex/alamofire-networking-made-easy-108e8f55b279) Alamofire - Network facilitado (inglês) (<https://medium.com/codex/alamofire-networking-made-easy-108e8f55b279>)

Conteúdos Alura:

- **Curso** iOS com SwiftUI: fazendo requisições HTTP e conexão com API (<https://www.alura.com.br/curso-online-ios-swiftui-requisicoes-http-conexao-api>)
- **Curso** Formação: Evolua Apps em SwiftUI: CRUD, MVVM e Autenticação (<https://cursos.alura.com.br/formacao-apps-swiftui-crud-mvvm-autenticacao>)

iOS - Gerenciamento de Estado:

- Gerenciamento de estado é o processo de rastrear e gerenciar a condição de um aplicativo. Isso é importante para aplicativos que são complexos ou que têm muitos estados diferentes. No UIKit, o gerenciamento de estado é feito manualmente pelo desenvolvedor. Isso significa que o desenvolvedor é responsável por rastrear o estado do aplicativo e atualizar a interface do usuário conforme o estado muda. Existem várias maneiras diferentes de gerenciar o estado no UIKit. Uma maneira comum é usar uma estrutura de dados para armazenar o estado. No SwiftUI, o gerenciamento de estado é feito automaticamente pelo framework. Isso significa que o desenvolvedor não precisa se preocupar em rastrear o estado do aplicativo ou atualizar a interface do usuário conforme o estado muda. O SwiftUI usa um sistema de gerenciamento de estado baseado em propriedade para gerenciar o estado dos aplicativos. Isso significa que o desenvolvedor pode definir propriedades que armazenam o estado do aplicativo e o framework atualizará automaticamente a interface do usuário conforme o estado muda. Veja as principais habilidades que você precisa desenvolver:
- Conhecer os fundamentos do gerenciamento de estado em iOS;
- Conhecer programação reativa para frameworks que usam o padrão declarativo (UIKit);
- Usar estruturas de dados e bibliotecas de gerenciamento de estado para armazenar e gerenciar o estado de um aplicativo.

Conteúdos

- **YouTube** iOS Swift Tutorial: RxSwift - Uma introdução prática (Inglês) | Brian Advent (<https://youtu.be/2m92mRI8IU>)
- **YouTube** Gerenciando Estados Visuais no SwiftUI - @State & Binding [Ponta-a-Ponta] | Michel Lopes | Irmandade Swift (https://youtu.be/vL_UUCtRAQo)
- **YouTube** Dividindo o fluxo de dados e estados no SwiftUI (inglês) | tundsdev (https://youtu.be/0Emh_7Vn3gM)
- **YouTube** @StateObject vs @ObservedObject vs @EnvironmentObject - Diferenças no SwiftUI | Tiago Aguiar (<https://youtu.be/lyx0IfgM0Uw>)

Conteúdos Alura:

- **Curso** iOS com SwiftUI: implementando navegação e gerenciamento de estados (<https://cursos.alura.com.br/course/ios-swiftui-navegacao-gerenciamento-estados>)

iOS - Recursos do sistema:

- O iOS fornece acesso a uma variedade de recursos do sistema, incluindo localização, câmera e sensores. Esses recursos podem ser usados para criar aplicativos que são mais envolventes e informativos. Por exemplo, um aplicativo de jogos pode usar o recurso de localização para rastrear a localização do usuário e adaptar o jogo de acordo. Um aplicativo de fotografia pode usar o recurso da câmera para tirar fotos e vídeos. E um aplicativo de saúde pode usar o recurso de sensores para rastrear a atividade física do usuário. Veja as principais habilidades que você precisa desenvolver:
- Conhecer os recursos do sistema em iOS;
- Acessar a localização, a câmera e os sensores;

- Criar aplicativos que usam recursos do sistema.

Conteúdos

- **YouTube** Tutorial de Swift: Como conseguir acessar a localização no iOS (Inglês) | Brian Advent (<https://youtu.be/Cu2QvRt4zM>)
- **Site** Documentação oficial Apple - Core Location (inglês) (<https://developer.apple.com/documentation/corelocation>)
- **Artigo** Acessando a Câmera e a Galeria do iOS com Swift 5 | Medium (<https://medium.com/mackmobile/acessando-cc6f7d5753c9>)
- **YouTube** Idiomas: Domine a internacionalização de aplicativos | Tiago Aguiar (<https://youtu.be/7ISYbEH17Qo>)
- **YouTube** PencilKit - Criando Desenhos com SwiftUI | Tiago Aguiar (<https://youtu.be/4SUHMGHlm-8>)

Conteúdos Alura:

- **Curso** iOS: autenticação, GPS e mais recursos nativos (<https://cursos.alura.com.br/course/ios-autenticacao-gps-recursos-nativos>)
- **Curso** iOS: usando recursos nativos (<https://cursos.alura.com.br/course/ios-usando-recursos-nativos?preRequirementFrom=ios-autenticacao-gps-recursos-nativos>)

iOS - Testes:

- Os testes são uma parte importante do desenvolvimento de software. Eles ajudam a garantir que o software esteja funcionando corretamente e que atenda aos requisitos do usuário. Existem vários tipos de testes que podem ser usados em iOS, incluindo testes de unidade, testes de integração e testes de sistema. Os testes de unidade são usados para testar unidades individuais de código, os testes de integração são usados para testar como diferentes unidades de código interagem umas com as outras e os testes de sistema são usados para testar todo o aplicativo como um todo. O teste é uma parte importante do processo de desenvolvimento de software e pode ajudar a garantir que os aplicativos iOS sejam de alta qualidade e confiáveis. Veja as principais habilidades que você precisa desenvolver:
- Conhecer as ferramentas e técnicas mais comuns para debugging de aplicativos iOS;
- Usar as ferramentas e técnicas mais comuns para debugging de aplicativos iOS para encontrar e corrigir erros.

Conteúdos

- **Artigo** iOS Teste Unitário e Teste de UI — Tutorial | Medium (<https://hoffsilva.medium.com/ios-teste-unit%C3%A1rio-e-teste-de-ui-tutorial-1360bb66ff73>)
- **YouTube** Fundamentos de Teste com iOS e Swift | Mobilizando (https://youtu.be/iQUo4_8AOX0)

Conteúdos Alura:

- **Curso** iOS: melhorando o app com testes de unidade e TDD (<https://cursos.alura.com.br/course/ios-melhorando-app-testes-unidade-dd>)
- **Curso** iOS: aplicando testes de unidade com mocks, stubs e outros frameworks (<https://cursos.alura.com.br/course/ios-testes-unidade-mocks-stubs-outros-frameworks>)

iOS - Vazamento de Memória:

- Uma memory leak é um problema que ocorre quando um aplicativo não libera a memória que não está mais usando. Isso pode levar a uma série de problemas, como o aplicativo ficar lento, travar ou até mesmo ser fechado pelo sistema operacional. Para evitar memory leaks, é importante entender como o gerenciamento de memória funciona no iOS e usar as ferramentas e

técnicas adequadas para gerenciar a memória do seu aplicativo. Veja as principais habilidades que você precisa desenvolver:

- Conhecer as causas de memory leaks em iOS;
- Identificar os sintomas de memory leaks e resolvê-los.

Conteúdos

- **Artigo** 3 Different Techniques to Find Memory Leaks in iOS | Medium (<https://medium.com/bitso-engineering/finding-memory-leaks-in-ios-like-a-pro-83a0212309b6>)
- **YouTube** Memory Leaks no iOS: Encontrar, diagnosticar e resolver (inglês) | iOS Academy (<https://youtu.be/b2AgibUg47k>)

iOS - Bibliotecas de terceiros:

- As bibliotecas de terceiros mais utilizadas em iOS são aquelas que fornecem funcionalidades adicionais ao SDK da Apple. Essas bibliotecas podem ser usadas para acessar dados da web, trabalhar com imagens e vídeos, desenvolver jogos, criar animações e muito mais. Veja as principais habilidades que você precisa desenvolver:
- Conhecer os diferentes tipos de bibliotecas que podem ser usadas em aplicativos iOS;
- Escolher e usar bibliotecas que sejam adequadas para as necessidades do seu aplicativo.

Conteúdos

- **Site** Kingfisher - Download e cache de imagens da internet (inglês) (<https://swiftpackageindex.com/onevc/Kingfisher>)
- **Site** Animações com LottieFiles (inglês) (<https://lottiefiles.com/blog/working-with-lottie/how-to-add-lottie-animation-ios-app-swift>)
- **Site** Alamofire - Documentação (inglês) (<https://alamofire.github.io/Alamofire/>)
- **YouTube** iOS Alamofire Tutorial (inglês) | Code Pro (<https://youtu.be/5hcHbhIWiel>)
- **YouTube** Realm Database tutorial (inglês) | CodeWithChris (<https://youtu.be/PmsJW59rNY8>)
- **YouTube** Integração com SDK Google Maps no SwiftUI | Tiago Aguiar (<https://youtu.be/lcpQLFicVNw>)
- **YouTube** Cocoapods: Como usar o gerenciador de dependências | Tiago Aguiar (https://youtu.be/_NHqZkap6Ao)

Nível 3

iOS - Arquitetura:

- As arquiteturas MVVM, MVC e VIPER são maneiras populares de estruturar aplicativos iOS. Ambas as arquiteturas se baseiam no conceito de separar a lógica de negócios da interface do usuário. Isso torna os aplicativos mais fáceis de manter e estender. Veja as principais habilidades que você precisa desenvolver:
- Conhecer os conceitos básicos de arquiteturas;
- Entender as diferenças entre as arquiteturas e suas vantagens;
- Implementar alguma delas em um aplicativo iOS.

Conteúdos

- **Artigo** MVVM para iOS (inglês) | Medium (<https://medium.com/technology-nineleaps/mvvm-architecture-for-ios-6d78794e13d8>)

- **Artigo** Arquitetura/Padrão de projeto VIPER para iOS | Medium (<https://medium.com/@smalam119/viper-design-pattern-for-ios-application-development-7a9703902af6>)
- **YouTube** MVVM (Model View ViewModel - A Arquitetura de Apps Mobile) // Dicionário do Programador | Código Fonte TV (<https://youtu.be/B2pJWtSyVFA>)
- **YouTube** MVC // Dicionário do Programador | Código Fonte TV (<https://youtu.be/jyTNhT67ZyY>)
- **YouTube** iOS MVVM Pattern - Binding | Swift Academy (<https://youtu.be/iI0LabCYZJo>)

Conteúdos Alura:

- **Curso** Formação evolua Apps em SwiftUI: CRUD, MVVM e Autenticação (<https://cursos.alura.com.br/formacao-apps-swiftui-crud-mvvm-autenticacao>)
- **Curso** iOS com View Code: cursos para aprender clean code e arquitetura de software (<https://cursos.alura.com.br/formacao-ios-view-code-clean-code-arquiteturas-software>)

iOS - Design System:

- A Apple oferece ferramentas e recursos para que os aplicativos sigam as melhores práticas de design e interação no iOS.
- Entender as Human Interface Guidelines (HIG) da Apple;
- Criar componentes reutilizáveis e escaláveis para iOS;
- Aplicar o Design System em diferentes dispositivos iOS.

Conteúdos

- **Artigo** Human Interface Guidelines | Apple Developer (<https://developer.apple.com/design/human-interface-guidelines/>)
- **Artigo** Designing for iOS | Apple Developer (<https://developer.apple.com/design/>)
- **Artigo** SwiftUI and Design Systems | Apple Developer (<https://developer.apple.com/xcode/swiftui/>)

Conteúdos Alura:

- **Curso** Formação iOS com SwiftUI: crie design system para projetos (<https://cursos.alura.com.br/formacao-ios-swiftui-crie-design-system-projetos>)

iOS - Injeção de dependências:

- A injeção de dependências (DI) é um padrão de projeto que permite que os componentes de um aplicativo recebam as dependências de que precisam em tempo de execução. Isso torna os componentes mais testáveis e flexíveis. Existem várias maneiras de implementar a DI em iOS. Uma maneira comum é usar um contêiner de injeção de dependências. Um contêiner de injeção de dependências é um objeto que armazena as dependências de um aplicativo e as fornece aos componentes conforme necessário. A DI é uma ferramenta poderosa que pode ajudar os desenvolvedores a criar aplicativos mais testáveis e flexíveis. Ao aprender como usar a DI, você pode criar aplicativos que são mais fáceis de manter e estender. Veja as principais habilidades que você precisa desenvolver:
- Conhecer e utilizar o padrão de projeto de injeção de dependências.

Conteúdos

- **YouTube** Injeção de dependências com Swift - O que é e quais os benefícios (Inglês) | Sean Allen (<https://youtu.be/I0QehVWz2i0>)
- **YouTube** Injeção de Dependências (O Guia Mais Prático do Youtube) // Dicionário do Programador | Código Fonte TV (<https://youtu.be/evhskJG1kvY>)

- **YouTube** [Devpass Talks] Injeção de Dependência com Swift (Eduardo Bocato) | Devpass (<https://youtu.be/0DDeOw5Jflc>)

iOS - Animações:

- As animações são uma parte importante do desenvolvimento de aplicativos iOS. Elas podem ser usadas para melhorar a experiência do usuário, tornando os aplicativos mais atraentes e envolventes. Existem várias maneiras de adicionar animações aos aplicativos iOS. Uma maneira é usar o framework UIKit, que fornece uma variedade de métodos para animar componentes da interface do usuário. Outra maneira é usar um framework de animação de terceiros, como o Lottie ou o Anima. Veja as principais habilidades que você precisa desenvolver:
- Entender como usar o framework UIKit para animar componentes da interface do usuário;
- Criar animações com o SwiftUI, como mostrar transições entre telas, atualizar a interface do usuário em resposta a eventos e criar efeitos visuais interessantes.

Conteúdos

- **Artigo** Primeiros passos com animações UIKit | Medium (<https://medium.com/my-ios-studies/primeiros-passos-com-anima%C3%A7%C3%B5es-uikit-d1172c04e3cd>)
- **YouTube** Adding Animations in SwiftUI | Bootcamp #25 (inglês) | SwiftUI Thinking (https://youtu.be/0WY-wrW2_bs)
- **Site** Documentação Oficial Apple (SwiftUI) - Animando Views e aplicando transições (inglês) (<https://developer.apple.com/tutorials/swiftui/animating-views-and-transitions>)

Conteúdos Alura:

- **Curso** iOS com SwiftUI: trabalhando com animações e DragGesture (<https://cursos.alura.com.br/course/ios-swiftui-trabalhando-animacoes-draggesture>)

iOS - Implantação (Deployment):

- O deployment ou implantação é o processo de disponibilizar um aplicativo iOS para os usuários. Existem várias maneiras de fazer isso, um deles é o uso da App Store. A App Store é a maneira mais comum de distribuir aplicativos iOS. Ela permite que os desenvolvedores distribuam seus aplicativos para um público global. O TestFlight é uma ferramenta beta privada que permite que os desenvolvedores compartilhem seus aplicativos com um grupo limitado de usuários para feedback. O Distribute é uma ferramenta que permite que os desenvolvedores distribuam seus aplicativos para seus próprios servidores. Veja as principais habilidades que você precisa desenvolver:
- Conhecer os conceitos básicos de deployment de aplicativos iOS;
- Criar uma conta de Apple Developer;
- Usar a App Store para distribuir aplicativos iOS para um público global.

Conteúdos

- **Artigo** Publicando um aplicativo na AppStore (inglês) | Medium (<https://medium.com/swlh/publishing-an-ios-app-to-app-store-d4f88c3686de>)
- **YouTube** App Store: Guia para lançar um app iOS (inglês) | App Radar (<https://youtu.be/1D-ZhTSpx1E>)
- **YouTube** Como publicar um app - XCode para AppStore | AC Soluções (<https://youtu.be/EAFr2qp6sso>)

Entrega e integração contínuas (CI/CD):

- CI/CD é a abreviação de Continuous Integration/Continuous Delivery, traduzindo para o português "entrega e integração contínuas". Trata-se de uma prática de desenvolvimento de software que visa tornar a integração de código mais eficiente por meio de builds e testes automatizados.
- Automatizar a integração de código entre varias partes da equipe se tornou cada vez mais importante, ja que assim é possível acelerar o desenvolvimento e diminuir o tempo de entrega de software.
- Executar testes automatizados da aplicação para verificar seu funcionamento.
- Realizar a entrega de atualizações de forma automatica e com segurança.
- Realizar testes de conexão e testes de carga para evitar que a aplicação apresente problemas ao ser atualizada.

Conteúdos

- **Artigo** O que é CI/CD? (<https://www.redhat.com/pt-br/topics/devops/what-is-ci-cd>)
- **Artigo** Sobre integração contínua (<https://docs.github.com/pt/actions/automating-builds-and-tests/about-continuous-integration>)
- **YouTube** Código Fonte TV: Pipeline (CD/CI) | Dicionário do Programador (<https://www.youtube.com/watch?v=AZtTd3pFVTY>)

Conteúdos Alura:

- **Artigo** Entrega e integração contínua de aplicações Vue (<https://www.alura.com.br/artigos/entrega-e-integracao-continua-de-aplicacoes-vue>)
- **Artigo** GitOps: o que é? (<https://www.alura.com.br/artigos/o-que-e-git-ops>)
- **Artigo** Tipos de testes: quais os principais e por que utilizá-los? (<https://www.alura.com.br/artigos/tipos-de-testes-principais-por-que-utiliza-los>)
- **Podcast** Integração Contínua, Deploy Contínuo e Github Actions (<https://cursos.alura.com.br/extra/hipsterstech/integracao-continua-deploy-continuo-e-github-actions-hipsters-213-a335>)
- **Podcast** Testes Automatizados (<https://cursos.alura.com.br/extra/hipsterstech/testes-automatizados-hipsters-51-a535>)
- **Curso** Curso Integração Contínua: mais qualidade e menos risco no desenvolvimento (<https://cursos.alura.com.br/course/desenvolvimento-software-integracao-continua>)
- **Curso** Curso Entrega Contínua: confiabilidade e qualidade na implantação de software (<https://cursos.alura.com.br/course/entrega-continua-confiabilidade-qualidade>)
- **Curso** Formação: Integração Contínua e Entrega Contínua (<https://cursos.alura.com.br/formacao-integracao-continua-entrega-continua>)

iOS - Modularização:

- A modularização é o processo de dividir um grande aplicativo em módulos menores, mais gerenciáveis. Isso pode tornar o código mais fácil de entender e manter, e também pode facilitar a reutilização de código. Existem várias maneiras de modularizar um aplicativo iOS. Uma maneira é usar frameworks. Os frameworks são bibliotecas de código que podem ser usadas para adicionar funcionalidade a um aplicativo. Outra maneira de modularizar um aplicativo é usar módulos de projeto. Os módulos de projeto são coleções de arquivos de código que podem ser usados para agrupar o código relacionado. Veja as principais habilidades que você precisa desenvolver:
- Conhecer os conceitos básicos de modularização de código.

Conteúdos

- **YouTube** Gerenciador de pacotes Swift & Modularização (inglês) | iOS Conf SG (<https://youtu.be/QmBZ9wJguS4>)
- **YouTube** NSBrazil19 - Modularizando o seu aplicativo | CocoaHeads Brasil (<https://youtu.be/dqE4IG6Gngw>)
- **YouTube** Como criar Build Variants no iOS | Tiago Aguiar (<https://youtu.be/0oWTZJL4kBQ>)
- **YouTube** Como criar um framework no Xcode com Swift | Tiago Aguiar (<https://youtu.be/pg65ceyAefQ>)

Conteúdos Alura:

- **Artigo** Deeplinks em SwiftUI: o que são e como utilizar (<https://www.alura.com.br/artigos/deeplinks-swiftui-o-que-sao-como-utilizar>)
- **Curso** Formação: Escalabilidade em aplicativos iOS (<https://cursos.alura.com.br/formacao-escalabilidade-aplicativos-ios>)

Objective-C - Fundamentos:

- Objective-C é uma linguagem de programação orientada a objeto que foi criada em 1984 por Brad Cox e Tom Love. Ela é usada para desenvolver aplicativos para o macOS, iOS e tvOS. Objective-C é uma linguagem de programação poderosa e flexível, e é usada por muitos desenvolvedores experientes. Veja as principais habilidades que você precisa desenvolver:
- Conhecer os tipos primitivos;
- Declarar e usar variáveis e constantes;
- Usar estruturas condicionais (if, else);
- Usar estruturas de repetição e laços (while, for);
- Usar funções, passando parâmetros e argumentos;
- Implementando métodos e reutilizando eles;
- Exceptions e Throwables;
- Convenções de código;
- Compreensão de Orientação a objetos.

Conteúdos

- **YouTube** Lesson 1: Introduction to Objective-C Programming | Medium (<https://medium.com/ios-objective-creation/lesson-1-introduction-to-objective-c-programming-22f5fe71172>)
- **YouTube** Lesson 2: Creating Custom Classes in Objective-C | Medium (<https://medium.com/ios-objective-creation/lesson-2-creating-custom-classes-in-objective-c-17f760ce9732>)
- **YouTube** O que é Objective-C - História da linguagem e funcionalidades importantes (inglês) | Your IT Class (<https://youtu.be/VmrpCi1XCvg>)
- **YouTube** Tutorial Objective-C Tutorial (inglês) | Derek Banas (<https://youtu.be/5esQqZIJ83g>)

iOS - Jogos:

- A plataforma iOS é uma ótima opção para o desenvolvimento de jogos. Ela possui uma grande base de usuários, com bilhões de dispositivos ativos, e uma série de recursos que tornam o desenvolvimento de jogos fácil e acessível. Para desenvolver jogos para iOS, você pode usar uma variedade de linguagens e ferramentas, incluindo Swift, Objective-C e Unity. Você também pode usar uma variedade de frameworks, como SpriteKit e SceneKit. O desenvolvimento de jogos para iOS é uma ótima maneira de criar experiências envolventes e divertidas para seus usuários. Se você está interessado em desenvolver jogos para iOS, há uma série de recursos

disponíveis para ajudá-lo a começar, incluindo tutoriais, cursos e comunidades online. Veja as principais habilidades que você precisa desenvolver:

- Conhecer as linguagens e ferramentas usadas para desenvolver jogos para iOS, como por exemplo C# e Unity;
- Explorar recursos de desenvolvimento de jogos, como o uso de gráficos 3D e animação.

Conteúdos

- **YouTube** Criando um jogo mobile na UNITY | Crie seus jogos (<https://youtu.be/ta-sFLHxJ4I>)

Habilidade Auxiliar: Infraestrutura e boas práticas

Git e GitHub - Fundamentos:

- Git é um sistema de controle de versão distribuído gratuito e de código aberto projetado para lidar com tudo, desde projetos pequenos a muito grandes com velocidade e eficiência.
- GitHub é um serviço de hospedagem para desenvolvimento de software e controle de versão usando Git.
- Criar um repositório
- Clonar um repositório
- Fazer commit, push e pull de e para o repositório
- Reverter um commit
- Criar branches e pull requests
- Lidar com merge e conflitos

Conteúdos

- **Site** GitHub Documentação (<https://docs.github.com/pt>)
- **Site** GitHub Pages Documentação (<https://docs.github.com/pt/pages/getting-started-with-github-pages/about-github-pages>)
- **Site** Git School - Visualizing Git (<https://git-school.github.io/visualizing-git/>)
- **Site** Dangit, Git!?! (<https://dangitgit.com/>)
- **YouTube** Rafaella Ballerini: O que é Git e GitHub? - definição e conceitos importantes 1/2 (<https://www.youtube.com/watch?v=DqTITcMq68k>)
- **YouTube** Rafaella Ballerini: Como usar Git e GitHub na prática! - desde o primeiro commit até o pull request! 2/2 (<https://www.youtube.com/watch?v=UBAX-13g8OM>)
- **YouTube** Mario Souto - Dev Soutinho: Git: Entendendo de vez como funciona do melhor e mais visual jeito possível (<https://www.youtube.com/watch?v=4-tfJ-ZyAOQ>)
- **YouTube** Mario Souto - Dev Soutinho: Como colocar seu projeto no ar DE GRAÇA via GitHub! | Hospedagem com GitHub Pages (https://www.youtube.com/watch?v=BU-w2_Aae54)
- **YouTube** CodandoTV(Rods) - 5 coisas que você precisa saber sobre Git (https://youtu.be/MqogPzjQyCY?si=VI_mT8EowuemptmU)
- **YouTube** CodandoTV(Rods) - README de Sucesso: Transforme seu Projeto ou seu Perfil em Destaque no GitHub (https://youtu.be/v9ZM2PVzctM?si=49ah_HRPlwpSEX4A)

Conteúdos Alura:

- **Artigo** Git e Github: O que são, Como Configurar e Primeiros Passos (<https://www.alura.com.br/artigos/o-que-e-git-github>)
- **Artigo** Mais git com o hub: a linha de comando do Github (<https://www.alura.com.br/artigos/github-na-linha-de-comando>)

- **Podcast** Hipsters 184: Guia do Iniciante em Github (<https://cursos.alura.com.br/extra/hipsterstech/guia-do-iniciante-em-github-hipsters-184-a378>)
- **Site** GitHub: diferentes maneiras de compartilhar seu projeto (<https://cursos.alura.com.br/extra/alura-mais/github-diferentes-maneiras-de-compartilhar-seu-projeto-c2002>)
- **Site** Websérie: Git e Github para Sobrevivência (<https://www.alura.com.br/webseries/git-e-github-para-sobrevivencia>)
- **Podcast** Hipsters 109: Git e Github (<https://www.alura.com.br/podcast/hipsterstech-git-e-github-hipsters-109-a474>)
- **YouTube** Alura: Git e Github para Sobrevivência 01: Como o Git funciona? (<https://www.youtube.com/watch?v=BAvmvmaKQkIQ>)
- **Curso** Curso Git e GitHub: compartilhando e colaborando em projetos (<https://cursos.alura.com.br/course/git-github-compartilhando-colaborando-projetos>)
- **Curso** Curso Git e GitHub: dominando controle de versão de código (<https://cursos.alura.com.br/course/git-github-dominando-controle-versao-codigo>)
- **Desafio** 7 Days of Code: GitHub (<https://7daysofcode.io/matricula/github>)

HTTP - Fundamentos:

- HTTP significa Hyper Text Transfer Protocol. A comunicação entre computadores cliente e servidores web é feita enviando solicitações HTTP e recebendo respostas HTTP.
- Entender a diferença dos verbos HTTP
- Testar os requests e ver os status codes no navegador
- Saber fazer uma requisição HTTP na linha de comando com WGET
- Baixar uma imagem com WGET
- Fazer um post

Conteúdos

- **Site** MDN Web Docs: Uma visão geral do HTTP (<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>)
- **Site** MDN Web Docs: Métodos de requisição HTTP (<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>)
- **Site** HTTP Cats (<https://http.cat/>)
- **Site** HTTP Dogs (<https://http.dog/>)
- **YouTube** Fabiano Gabardo Lemos: Requisições HTTP - GET, POST, PUT, PATCH DELETE (<https://www.youtube.com/watch?v=kncOJZrnkTg>)
- **YouTube** Programador a Bordo: Protocolo HTTP e TCP/IP - Introdução (<https://www.youtube.com/watch?v=V4XZ81vRGtM>)

Conteúdos Alura:

- **Artigo** HTTP: Desmistificando o protocolo da Web (<https://www.alura.com.br/artigos/desmistificando-o-protocolo-http-parte-1>)
- **Artigo** Métodos de requisição do HTTP (<https://www.alura.com.br/artigos/metodos-de-requisicao-do-http>)
- **Artigo** Qual é a diferença entre HTTP e HTTPS? (<https://www.alura.com.br/artigos/qual-e-diferenca-entre-http-e-https>)
- **Podcast** Hipsters.tech: HTTP/2: magia com o novo protocolo - Hipsters 13 (<https://www.alura.com.br/podcast/http-2-magia-com-o-novo-protocolo-hipsters-13-a573>)

- **Curso** Curso HTTP: Entendendo a web por baixo dos panos (<https://www.alura.com.br/curso-online-http-entendendo-web-por-baixo-dos-panos>)

JSON:

- JSON significa JavaScript Object Notation (notação de objeto JavaScript). É um formato de texto para armazenar e transmitir dados.
- Criar um objeto
- Transformar um objeto em uma string
- Transformar uma string em objeto
- Manipular um objeto

Conteúdos

- **Site** MDN Web Docs: JSON (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/JSON)
- **Site** MDN Web Docs: Trabalhando com JSON (<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/JSON>)
- **Artigo** Introdução ao JSON (<https://juliocarneiro.medium.com/introdu%C3%A7%C3%A3o-ao-json-7825b1a550ff>)
- **YouTube** Marco Bruno: O que é JSON e como criar um objeto (<https://www.youtube.com/watch?v=oCY5YEEjlwE>)

Conteúdos Alura:

- **Artigo** O que é JSON? (<https://www.alura.com.br/artigos/o-que-e-json>)
- **Curso** Curso MySQL e JSON: persistindo JSON de maneira eficiente (<https://www.alura.com.br/curso-online-mysql-json-persistencia>)

Linha de comando - Fundamentos:

- CLI é um programa de linha de comando que aceita entradas de texto para executar funções do sistema operacional.
- Conhecer os principais comandos

Conteúdos

- **Site** Microsoft Docs: Como usar argumentos de linha de comando para terminal do Windows (<https://docs.microsoft.com/pt-br/windows/terminal/command-line-arguments?tabs=windows>)
- **YouTube** Estevan Maito: Linha de comando básica - 8 comandos mais frequentes - WINDOWS e LINUX (<https://www.youtube.com/watch?v=rKBqXJZocYk>)
- **YouTube** Ninja do Linux: Comandos básicos da linha de comando do Linux (https://www.youtube.com/watch?v=rs_yshFGu8E)

Conteúdos Alura:

- **Artigo** CMD: dicas para trabalhar no prompt do Windows (<https://www.alura.com.br/artigos/cmd-dicas-para-trabalhar-no-prompt-do-windows>)
- **Artigo** Como configurar variáveis de ambiente no Windows, Linux e macOS (<https://www.alura.com.br/artigos/configurar-variaveis-ambiente-windows-linux-macos>)
- **Site** Primeiras aulas do Curso Windows Prompt: Trabalhando na linha de comando (<https://www.alura.com.br/conteudo/windows-prompt-utilizando-cmd>)
- **Curso** Curso Windows Prompt: Trabalhando na linha de comando (<https://www.alura.com.br/curso-online-prompt>)

- **Curso** Curso Terminal: aprenda comandos para executar tarefas (<https://cursos.alura.com.br/course/terminal-comandos-executar-tarefas>)

Cloud - Fundamentos:

- Cloud, ou computação em nuvem é a distribuição de serviços de computação pela Internet usando um modelo de preço pago conforme o uso. Uma nuvem é composta de vários recursos de computação, que abrangem desde os próprios computadores (ou instâncias, na terminologia de nuvem) até redes, armazenamento, bancos de dados e o que estiver em torno deles. Ou seja, tudo o que normalmente é necessário para montar o equivalente a uma sala de servidores, ou mesmo um data center completo, estará pronto para ser utilizado, configurado e executado.
- Conhecer a diferença entre IaaS, PaaS e SaaS
- Conhecer os maiores provedores de cloud
- Especializar-se em algum provedor

Conteúdos

- **Artigo** Computação em nuvem (<https://medium.com/sysadminas/computa%C3%A7%C3%A3o-em-nuvem-515930304cf9>)
- **Artigo** O que é cloud? (<https://gabriel-faraday.medium.com/o-que-%C3%A9-cloud-991109e708c6>)
- **YouTube** Gabs Ferreira: Por que investir e estudar cloud? (<https://www.youtube.com/watch?v=Z45BTNeZ1lQ>)
- **YouTube** Andre Iacono: O que é MICROSOFT AZURE? Qual Certificação começar em 2022? (<https://www.youtube.com/watch?v=f-oVzkvMwnE>)
- **YouTube** AWS: O que é a AWS? (<https://www.youtube.com/watch?v=8Jl9wQ8sUdQ>)
- **YouTube** O que é Google Cloud e por que aprender? (<https://www.youtube.com/shorts/Lzq3f1DHWcl>)
- **Artigo** AWS vs Google Cloud vs Azure: o que cada um tem de melhor? (<https://medium.com/data-hackers/aws-vs-google-cloud-vs-azure-o-que-cada-um-tem-de-melhor-52107174f7b7>)
- **YouTube** Código Fonte TV: Azure (A plataforma Cloud da Microsoft) (<https://www.youtube.com/watch?v=YgE-sZaCuJ0>)
- **YouTube** Mundo da Cloud: AWS do Zero ao Expert (<https://www.youtube.com/watch?v=HiBCv9DolxI&list=PLtL97Owd1gkQ0dfqGW8OtJ-155Gs67Ecz>)

Conteúdos Alura:

- **Podcast** Hipsters.tech: TechGuide - Fundamentos Cloud – Hipsters Ponto Tech #348 (<https://www.hipsters.tech/techguide-fundamentos-cloud-hipsters-ponto-tech-348/>)
- **Artigo** Cloud: o que é, História e Guia da computação em nuvem (<https://www.alura.com.br/artigos/cloud>)
- **Podcast** Hipsters.tech: Uma jornada Para o Cloud - Hipsters Deep Dive 005 (<https://www.alura.com.br/podcast/uma-jornada-para-o-cloud-hipsters-deep-dive-005-a1100>)
- **Podcast** Hipsters.tech: Histórias do Cloud – Hipsters 04 (<https://www.alura.com.br/podcast/historias-do-cloud-hipsters-04-a582>)
- **Artigo** Heroku, Vercel e outras opções de cloud como plataforma (<https://www.alura.com.br/artigos/heroku-vercel-outras-opcoes-cloud-plataforma>)
- **Artigo** AWS: Guia sobre o que é Amazon Web Services, seus Serviços e Certificações (<https://www.alura.com.br/artigos/aws>)

- **Artigo** Terraform: criando máquinas na Azure (<https://www.alura.com.br/artigos/terraform-maquinas-na-azure>)
- **YouTube** Alura: O que é cloud? (<https://www.YOUTUBE.com/watch?v=wev9fMrg-TU>)
- **YouTube** Alura: AWS, Google Cloud e Azure: Por onde começar? | Hipsters.Talks (<https://www.YOUTUBE.com/watch?v=z9k6rsdmWc0&t=300s>)
- **YouTube** Alura: Certificação em Cloud: Azure, AWS, Google | Hipsters.Talks (https://www.YOUTUBE.com/watch?v=W4K82n_WK5g&t=290s)
- **Curso** Formação Começando em Cloud Computing (<https://cursos.alura.com.br/formacao-cloud-computing>)
- **Curso** Formação Amazon Web Services (<https://cursos.alura.com.br/formacao-amazon-web-services>)
- **Curso** Formação Google Certified Associate Cloud Engineer (<https://cursos.alura.com.br/formacao-google-certified-associate-cloud-engineer>)
- **Curso** Formação Certificação AWS Certified Cloud Practitioner (<https://cursos.alura.com.br/formacao-aws-certified-cloud-practitioner>)
- **Curso** Formação Containers com AWS ECS e EKS (<https://cursos.alura.com.br/formacao-containers-aws>)
- **Curso** Formação Google Cloud Platform (<https://cursos.alura.com.br/formacao-google-cloud>)
- **Curso** Formação Certificação Google Certified Associate Cloud Engineer (<https://cursos.alura.com.br/formacao-google-certified-associate-cloud-engineer>)
- **Curso** Formação Azure (<https://cursos.alura.com.br/formacao-conhecendo-azure>)
- **Curso** Formação Certificação AZ-900: Microsoft Azure Fundamentals (<https://cursos.alura.com.br/formacao-certificacao-az-900-microsoft-azure-fundamentals>)

SOLID:

- O Solid possui cinco princípios considerados como boas práticas no desenvolvimento de software que ajudam os programadores a escrever os códigos mais limpos, separando as responsabilidades, diminuindo acoplamentos, facilitando na refatoração e estimulando o reaproveitamento do código.

Conteúdos

- **Artigo** Princípios de S.O.L.I.D em C# — Guia prático (<https://medium.com/beelabacademy/princ%C3%ADpios-de-s-o-l-i-d-em-c-guia-pr%C3%A1tico-cbb1e6584284>)
- **YouTube** Filipe Deschamps: SOLID fica fácil com Essas Ilustrações (<https://www.youtube.com/watch?v=6SfrQ3D4dHM>)
- **YouTube** Central dotNET: Princípios de SOLID com C# (<https://www.youtube.com/watch?v=iU4BMUcjg8g>)
- **YouTube** Felipe Pinheiro - Tech: Princípios SOLID, o que é DIP? (<https://www.youtube.com/watch?v=q0BGgQJcp7w>)
- **YouTube** Dev Eficiente: SOLID - Uma reflexão sobre o Princípio da responsabilidade única (https://www.youtube.com/watch?v=GGe0o_v5vjM)

Conteúdos Alura:

- **Artigo** SOLID: o que é e quais os 5 princípios da Programação Orientada a Objetos (POO) (<https://www.alura.com.br/artigos/solid>)
- **Podcast** SOLID: Código bom e bonito - Hipsters Ponto Tech 219 (<https://www.alura.com.br/podcast/hipsterstech-solid-codigo-bom-e-bonito-hipsters-ponto->

[tech-219-a649](#))

- **YouTube** Alura: Clean Code e Solid (https://www.youtube.com/watch?v=XV9B4LX_re8)
- **Curso** Curso C#: aplique princípios SOLID (<https://cursos.alura.com.br/course/csharp-aplique-principios-solid>)
- **Curso** Curso Boas práticas de programação: melhore o código de uma API Java (<https://cursos.alura.com.br/course/boas-praticas-programacao-melhora-codigo-api-java>)
- **Curso** Curso SOLID com PHP: princípios da programação orientada a objetos (<https://cursos.alura.com.br/course/solid-php-principios-orientacao-a-objetos>)
- **Curso** Curso iOS: escrevendo código de qualidade com SOLID em Swift (<https://www.alura.com.br/curso-online-ios-escrevendo-codigo-qualidade-solid-swift>)
- **Curso** SOLID com TypeScript: aplicando boas práticas em orientação a objetos (<https://www.alura.com.br/curso-online-solid-typescript-boas-praticas-orientacao-objetos>)
- **Curso** Formação Boas práticas em C# (<https://cursos.alura.com.br/formacao-boas-praticas-c-sharp>)
- **Curso** Formação Boas práticas em PHP (<https://cursos.alura.com.br/formacao-boas-praticas-php>)
- **Curso** Formação Boas práticas em Java (<https://cursos.alura.com.br/formacao-boas-praticas-java>)
- **Site** Livro Casa do Código: Desbravando SOLID - Práticas avançadas para códigos de qualidade em Java moderno (<https://www.casadocodigo.com.br/products/livro-desbravando-solid>)
- **YouTube** CodandoTV(Rods): SOLID e o 'D' do Princípio da Inversão de Dependência usando a Injeção de Dependência para ajudar (<https://youtu.be/xpcCDCH5k4Y>)

Clean Code:

- Aplicar técnicas simples que visam facilitar a escrita e leitura de um código
- Refatorar seu código para que fique mais claro

Conteúdos

- **Artigo** Clean Code: O que é? Porque usar? (<https://medium.com/desenvolvendo-com-paixao/1-clean-code-o-que-%C3%A9-porque-usar-1e4f9f4454c6>)
- **YouTube** Filipe Deschamps: CLEAN CODE 1 - Introdução (<https://www.youtube.com/watch?v=9w3o9NHXqu0>)
- **YouTube** Filipe Deschamps: CLEAN CODE 2 - O que é código limpo? (https://www.youtube.com/watch?v=ZVtGbO_CnfA)
- **Livro** Código limpo: habilidades práticas do Agile software (<https://www.amazon.com.br/C%C3%B3digo-limpo-Robert-C-Martin/dp/8576082675>)

Conteúdos Alura:

- **Artigo** Clean Code: O que é, Casos de Uso, Exemplo de Código Limpo (<https://www.alura.com.br/artigos/o-que-e-clean-code>)
- **YouTube** Alura: Clean Code e Solid (https://www.youtube.com/watch?v=XV9B4LX_re8)
- **Livro** Orientação a Objetos e SOLID para Ninjas (https://www.casadocodigo.com.br/products/livro-oo-solid?_pos=2&_sid=fe732c657&_ss=r)

Clean Architecture:

- A Clean Architecture (Arquitetura Limpa) é uma forma de desenvolver software, de tal forma que apenas olhando para o código fonte de um programa, você deve ser capaz de dizer o que o

programa faz.

Conteúdos

- **Artigo** Descomplicando a Clean Architecture (<https://medium.com/luizalabs/descomplicando-a-clean-architecture-cf4dfc4a1ac6>)
- **YouTube** Como DEV ser!: Entenda CLEAN ARCHITECTURE de uma vez por todas! (<https://www.youtube.com/watch?v=HynTfTii4mw>)
- **YouTube** Full Cycle: O que é Clean Architecture (<https://www.youtube.com/watch?v=dQys-mnOtQg>)
- **YouTube** CodandoTV(Rods) - Simplificando o Clean Architecture +MVVM na sua aplicação mobile - Guia Completo (<https://youtu.be/8ehlZfyN1S0?si=l7-l5l4zsnYLIjD>)

Conteúdos Alura:

- **Site** Clean Architecture (Arquitetura Limpa) - O que é? (<https://cursos.alura.com.br/extra/alura-mais/clean-architecture-arquitetura-limpa-o-que-e--c204>)
- **Podcast** Hipsters.tech: Clean Architecture - Hipsters Ponto Tech 254 (<https://www.hipsters.tech/clean-architecture-hipsters-ponto-tech-254/>)
- **YouTube** Alura: Arquitetura de sistemas (<https://www.youtube.com/watch?v=oedWxgAZc2A>)
- **Site** Primeiras aulas do curso Java e Clean Architecture: descomplicando arquitetura de software (<https://www.alura.com.br/conteudo/java-clean-architecture>)
- **Curso** Curso Java e Clean Architecture: descomplicando arquitetura de software (<https://www.alura.com.br/curso-online-java-clean-architecture>)
- **Curso** Curso PHP e Clean Architecture: descomplicando arquitetura de software (<https://cursos.alura.com.br/course/php-introducao-clean-achitecture>)

Firebase:

- O Firebase é uma plataforma de desenvolvimento de aplicativos Backend-as-a-Service (BaaS) que fornece serviços de backend hospedados, tais como banco de dados em tempo real, armazenamento em nuvem, autenticação, relatórios de falhas, aprendizado de máquina, configuração remota e hospedagem para seus arquivos estáticos.
- Entender como Instalar o Firebase
- Conhecer a documentação do Firebase
- Conhecer as ferramentas do Firebase disponíveis

Conteúdos

- **Site** Produtos do Firebase | Firebase (<https://firebase.google.com/products-build?hl=pt-br>)
- **Site** Casos de Uso do Firebase | Firebase (<https://firebase.google.com/use-cases?hl=pt-br>)
- **Site** Adicionando o Firebase ao App Flutter | Firebase Docs (<https://firebase.google.com/docs/flutter/setup?platform=android>)
- **Site** Adicionando o Firebase ao seu projeto Android | Firebase Docs (<https://firebase.google.com/docs/android/setup>)
- **YouTube** Firebase: Começando no Firebase com Flutter - Firecasts (inglês) (https://www.youtube.com/watch?v=EXp0gg9kGxI&ab_channel=Firebase)
- **YouTube** Prof. Diego Antunes: Instalação do Firebase no Flutter em 2022 (FlutterFire e Realtime Database) (https://www.youtube.com/watch?v=QjdGSoDntZQ&ab_channel=Prof.DiegoAntunes)
- **Site** Precificação do Firebase | Firebase (<https://firebase.google.com/pricing?hl=pt-br>)

Conteúdos Alura:

- **Artigo** Integrando App Android com o Firebase Cloud Messaging (<https://www.alura.com.br/artigos/integrando-app-android-com-o-firebase-cloud-messaging>)
- **Artigo** Tratando notificações recebidas do Firebase no Android (<https://www.alura.com.br/artigos/tratando-notificacoes-recebidas-do-firebase-no-android>)
- **Artigo** Autenticando com a conta Google no Android utilizando o Firebase Authentication (<https://www.alura.com.br/artigos/autenticando-no-google-com-firebase-authentication>)
- **Artigo** Flutter: Tratamento de exceções com Firebase Crashlytics (<https://www.alura.com.br/artigos/tratamento-de-execucoes-com-firebase-crashlytics>)
- **YouTube** Alura: Primeiros passos em Cloud Firestore no Flutter (Parte 1) | #AluraMais (<https://youtu.be/6jsR5CXhHOs>)
- **YouTube** Alura: Primeiros passos em Cloud Firestore no Flutter (Parte 2) | #AluraMais (<https://youtu.be/KWWTTY6gSw4>)
- **YouTube** Alura: Registro de exceções no Firebase Crashlytics | #AluraMais (https://www.youtube.com/watch?v=LOLjsG_eNyY&ab_channel=AluraCursosOnline)
- **Curso** Curso Flutter: Push Notifications com Firebase Cloud Messaging (<https://www.alura.com.br/curso-online-flutter-push-notifications-firebase-cloud-messaging>)
- **Curso** Formação Firebase com Android (<https://www.alura.com.br/formacao-firebase-android>)

Habilidade Auxiliar: UX & Design

iOS - Diretrizes de Interface Humana (HIG):

- As Diretrizes de Interface Humana contêm orientações e práticas recomendadas que podem ajudá-lo a criar uma ótima experiência para qualquer plataforma Apple. Veja as principais habilidades que você precisa desenvolver:
- Aprender os fundamentos das Diretrizes de Interface Humana (HIG) para iOS;
- Usar cores, fontes, layouts, ícones e animação de acordo com as diretrizes HIG.

Conteúdos

- **Site** Documentação oficial Apple (inglês) - Human Interface Guidelines (<https://developer.apple.com/design/human-interface-guidelines>)
- **YouTube** Acessibilidade no iOS - Gabriel Oliva | InfoQ Brasil (https://youtu.be/6Z_E2_Dx6-s)
- **Site** Documentação Oficial Apple (inglês) - Padrões para ações do usuário (<https://developer.apple.com/design/human-interface-guidelines/patterns>)
- **Site** Documentação Oficial Apple (inglês) - Padrões para inserção de dados (<https://developer.apple.com/design/human-interface-guidelines/inputs>)

Design System:

- Um Design System (sistema de design) é uma coleção de componentes reutilizáveis, guiados por padrões claros, que podem ser colocados juntos para construir aplicações.
- Criar e manter bibliotecas que serão consumidas e usadas como padrão para a construção de um projeto
- Design tokens
- Estilos fundamentais
- Construção de componentes
- Microinterações

- Documentação

Conteúdos

- **Artigo** Design System: o que é e quais os benefícios? (<https://medium.com/ipnet-growth-partner/design-system-o-que-e-438773dd811>)
- **Artigo** Afinal, o que é Design System? (<https://brasil.uxdesign.cc/afinal-o-que-%C3%A9-design-system-448c257b0021>)
- **Artigo** O que são Design Tokens (<https://medium.com/pretux/design-tokens-112b2ee11ddf>)
- **YouTube** Caio Gonzalez: O que é Design System? / Guia para começar o seu próprio Design System (<https://www.youtube.com/watch?v=ajgbXpFVaAw>)

Conteúdos Alura:

- **Artigo** O que é Design System? (<https://www.alura.com.br/artigos/o-que-e-design-system>)
- **Artigo** Design Systems: exemplos práticos (<https://www.alura.com.br/artigos/design-systems-exemplos-praticos>)
- **YouTube** Alura: Design System & Style Guide (<https://www.youtube.com/watch?v=rSLvpOqC5wQ>)
- **YouTube** Alura: Design Systems (com Charles Assunção) (<https://www.youtube.com/watch?v=MLGFctEtmYQ>)
- **Podcast** Hipsters.tech: Design Systems - Hipsters 170 (<https://www.alura.com.br/podcast/hipsterstech-design-systems-hipsters-170-a399>)
- **Podcast** Layers.tech: Design System – Layers ponto tech 36 (<https://www.alura.com.br/podcast/layerstech-design-system-layers-ponto-tech-36-a1056>)
- **Curso** Design System: definindo estilos e tokens (<https://cursos.alura.com.br/course/design-system-definindo-estilos-tokens>)
- **Curso** Design System: projetando e construindo componentes (<https://cursos.alura.com.br/course/design-system-projetando-construindo-componentes>)
- **Curso** Design System: documentando um design system (<https://cursos.alura.com.br/course/design-system-documentando-design-system>)
- **Desafio** Experimente o que é ser um(a) profissional de Ux (<https://www.alura.com.br/challenges/ux-4>)

Sistemas de cores:

- Definir uma paleta de cores que faça sentido para determinada interface

Conteúdos

- **Artigo** Cores em UI: Um Guia Rápido Para Usar em Seus Projetos (<https://medium.com/aela/cores-em-ui-um-guia-r%C3%A1pido-para-usar-em-seus-projetos-31ccffe3e16b>)
- **Artigo** A Psicologia das cores e sua relação com o UX Design (<https://brasil.uxdesign.cc/a-psicologia-das-cores-e-sua-rela%C3%A7%C3%A3o-com-o-ux-design-af02460639cd>)
- **YouTube** kacio.design: Aplicação de Cores - Princípios do UI (https://www.youtube.com/watch?v=C_VhzEyqT6U)

Conteúdos Alura:

- **Curso** Curso Cores para Designers: escolhendo e trabalhando com cores em um projeto (<https://www.alura.com.br/curso-online-cores-para-seu-projeto>)
- **Site** Primeiras aulas do curso Cores: sistemas básicos e paletas (<https://www.alura.com.br/conteudo/fundamentos-da-cor>)

- **Curso** Cores: sistemas básicos e paletas (<https://www.alura.com.br/curso-online-fundamentos-da-cor>)
- **Curso** Curso Design editorial: criação de materiais gráficos (<https://www.alura.com.br/curso-online-design-editorial>)

Como usar fontes:

- Escolher a fonte mais adequada para determinado projeto

Conteúdos

- **Artigo** Você sabe usar tipografia em UI Design? (<https://medium.com/ui-lab-school/voc%C3%AA-sabe-usar-tipografia-em-ui-design-9ce4ccdbab43>)
- **Artigo** As 20 fontes mais usadas pelos designers gráficos (<https://apenasumchico.medium.com/as-20-fontes-mais-populares-de-todos-os-tempos-38a6c121dfb>)
- **Artigo** Usando ícones SVG como fontes com o IcoMoon (<https://dev.to/sucodelarangela/usando-icone-sv-g-como-fontes-com-o-icomoon-563e>)
- **YouTube** Nadine Fronza: Como escolher a melhor fonte para seu projeto (<https://www.youtube.com/watch?v=MSKPUtldadI>)
- **YouTube** Thiago Abreu: Como Escolher Fontes para um Trabalho de Design (<https://www.youtube.com/watch?v=I-TrNbZi-aU>)

Conteúdos Alura:

- **Artigo** 10 tipografias gratuitas para um design incrível (<https://www.alura.com.br/artigos/10-tipografias-gratuitas-design-incriveI>)
- **Site** Tipografia para Web (<https://tipografiaparaweb.netlify.app/>)
- **Site** Primeiras aulas do curso Tipografia: conhecendo o que há por trás dos tipos (<https://www.alura.com.br/conteudo/tipografia-conceito>)
- **Curso** Curso Tipografia: conhecendo o que há por trás dos tipos (<https://www.alura.com.br/curso-online-tipografia-conceito>)
- **Curso** Curso Design editorial: criação de materiais gráficos (<https://www.alura.com.br/curso-online-design-editorial>)

Design Responsivo:

- Ajustar suas páginas para o tamanho da tela do usuário
- Media queries
- Conhecer o conceito de Mobile first

Conteúdos

- **Artigo** Design responsivo para leigos (<https://medium.com/neworder/design-responsivo-para-leigos-888c9d0dfdaf>)
- **Artigo** Entendendo as diferenças entre design responsivo, adaptativo e mobile-first (<https://medium.com/@fnandaleite/entendendo-as-diferen%C3%A7as-entre-design-responsivo-adaptativo-e-mobile-first-ea3c61fc9181>)
- **Artigo** Web Design Responsivo — O Que É e Como Usá-lo (<https://medium.com/@carlosverza/web-design-responsivo-o-que-%C3%A9-e-como-us%C3%A1-lo-12477b168fc7>)
- **YouTube** MobGeek: O que é Design Responsivo? (<https://www.youtube.com/watch?v=WXfYtneJZzl>)

- **YouTube** Ralf Lima: Mobile First e Desktop First (https://www.youtube.com/watch?v=oqggJTC3_vA)

Conteúdos Alura:

- **Artigo** Como fazer Grids e a Responsividade na Web (<https://www.alura.com.br/artigos/como-fazer-grids-e-a-responsividade-na-web>)
- **Curso** Curso Layouts Responsivos: trabalhando com layouts mobile (<https://www.alura.com.br/curso-online-mobile-first-layouts-responsivos>)
- **Curso** HTML e CSS: responsividade com mobile-first (<https://www.alura.com.br/curso-online-html-css-responsividade-mobile-first>)
- **Desafio** 7 Days of Code: Responsividade (<https://7daysofcode.io/matricula/responsividade>)

iOS - Acessibilidade:

- A acessibilidade é a capacidade de todos os usuários de acessar e usar um aplicativo, independentemente de suas capacidades. Isso inclui pessoas com deficiências visuais, auditivas, motoras e cognitivas. Existem muitas maneiras de tornar um aplicativo acessível em iOS. Uma maneira é usar os recursos de acessibilidade do iOS, como VoiceOver, Zoom e Switch Control. Outra maneira é projetar o aplicativo de forma que seja fácil de usar por pessoas com deficiências. É importante tornar seus aplicativos acessíveis para que todos possam usá-los. Isso é bom para os negócios, pois pode aumentar sua base de clientes. Também é bom para a sociedade, pois pode ajudar a promover a inclusão e a igualdade. Veja as principais habilidades que você precisa desenvolver:
- Conhecer e utilizar os recursos de acessibilidade do iOS.

Conteúdos

- **Artigo** Tornando o seu app acessível por todos no iOS | Medium (<https://medium.com/@GusSevero/tornando-o-seu-app-acess%C3%ADvel-por-todos-e93c1aa2ad9d>)
- **YouTube** Acessibilidade em apps iOS na prática - Ashton Williams (inglês) | A11y Bytes (<https://youtu.be/aFeVVQHktaU>)

Figma - Fundamentos:

- Figma é uma aplicação web colaborativa para design de interfaces. O conjunto de recursos do Figma se concentra na interface do usuário e no design da experiência do usuário, com ênfase na colaboração em tempo real, utilizando uma variedade de editores de gráficos vetoriais e ferramentas de prototipagem.
- Criar layouts de páginas e componentes

Conteúdos

- **Site** Figma Básico - Primeiros passos com Figma (<https://www.figma.com/community/file/1190593276621265988>)
- **Artigo** O que é o Figma e por que usar ele? (<https://medium.com/nerdzaio/o-que-%C3%A9-o-figma-e-por-que-usar-ele-a71fbf1dbdd8>)
- **Artigo** Entenda o que é e como usar o Figma para criar designs (<https://blog.b2bstack.com.br/figma/>)
- **YouTube** Marco Bruno: Como usar o Figma! (<https://www.youtube.com/watch?v=qoE-2YFeW-Q>)
- **YouTube** 4 Exemplos de Componentes interativos no Figma (<https://www.youtube.com/watch?v=mGL3jrY-OBc>)

Conteúdos Alura:

- **Artigo** Figma: o que é a ferramenta, Design e uso (<https://www.alura.com.br/artigos/figma>)
- **Site** Como Front-End utiliza o Figma - Alura+ (<https://cursos.alura.com.br/extra/alura-mais/como-front-end-utiliza-o-figma-c858>)
- **Artigo** Top 5 plugins no Figma para trabalhar com Design System (<https://www.alura.com.br/artigos/top-5-plugins-figma-trabalhar-com-design-system>)
- **Artigo** 5 plugins essenciais que você precisa ter no seu Figma (<https://www.alura.com.br/artigos/5-plugins-essenciais-do-figma>)
- **Artigo** 10 truques incríveis e pouco conhecidos no Figma (<https://www.alura.com.br/artigos/10-truques-incriveis-pouco-conhecidos-figma>)
- **Podcast** Layers.tech: Figma do Design ao Código 03 (<https://www.alura.com.br/podcast/layerstech-figma-do-design-ao-codigo-layers-ponto-tech-03-a726>)
- **YouTube** Como fazer estilos locais mais rapidamente no Figma? (https://www.youtube.com/watch?v=fCHd2_lAcQQ)
- **Curso** Formação Figma (<https://www.alura.com.br/formacao-figma>)