

Flutter

TechGuide - Alura, FIAP e PM3

Nível 1

Dart - Fundamentos:

- Instalar e configurar o Dart
- Conhecer as palavras chaves do Dart
- Construir aplicações de terminal usando Dart
- Entender como funciona o dinamismo no Dart
- Compreender e aprender a usar o Null Safety no Dart

Conteúdos

- **Site** Começando no Dart (inglês) (<https://dart.dev/get-dart>)
- **Site** Passeio pela Linguagem Dart (inglês) (<https://dart.dev/guides/language/language-tour>)
- **Artigo** Sistemas de Tipos no Dart (inglês) (<https://dart.dev/guides/language/type-system>)
- **Artigo** Entendendo Null Safety (inglês) (<https://dart.dev/null-safety/understanding-null-safety>)
- **YouTube** Código Fonte TV: Dart (a linguagem do Flutter) // Dicionário do Programador (<https://youtu.be/i7IzIVImHEc>)
- **YouTube** Dotcode: Dart ou Flutter - Por onde começar? (<https://youtu.be/s7VNmqw6wLM>)

Conteúdos Alura:

- **Artigo** Dart: O que é? Como começar a estudar? Para quê serve? (<https://www.alura.com.br/artigos/dart>)
- **Artigo** Boas práticas do Dart para organizar um projeto (<https://www.alura.com.br/artigos/boas-praticas-dart>)
- **YouTube** Alura: Entendendo Input e Output no Dart (<https://www.youtube.com/watch?v=LLmBWjF6F8M>)
- **YouTube** Alura: Os 5 métodos da lista do Dart | #AluraMais (<https://youtu.be/pcnfjJG3jY4>)
- **YouTube** Alura: Const e Final no Dart | #Aluramais (<https://youtu.be/bEz0VbZ3C7A>)
- **YouTube** Alura: O que você precisa saber sobre Null Safety no Dart | #AluraMais (<https://youtu.be/OkjjhHXIRI0>)
- **Curso** Dart: entendendo a Orientação a Objetos (<https://cursos.alura.com.br/course/dart-entendendo-orientacao-objetos>)
- **Curso** Dart: trabalhando com a sintaxe e configuração de projeto (<https://cursos.alura.com.br/course/dart-trabalhando-sintaxe-configuracao-projeto>)
- **Curso** Dart: cursos para aprender a linguagem utilizada no Flutter (<https://cursos.alura.com.br/formacao-dart-cursos-aprender-linguagem-utilizada-flutter>)

Flutter - Fundamentos:

- Flutter é um kit de desenvolvimento de software de IU de código aberto criado pelo Google. Ele é usado para desenvolver aplicativos multiplataforma para Android, iOS, Linux, macOS, Windows, Google Fuchsia e para a web a partir de um único código.
- Os widgets Flutter são construídos utilizando uma estrutura moderna que se inspira na React. A idéia central é que você construa sua interface de usuário a partir de widgets. Os widgets descrevem como deve ser sua visualização, dada sua configuração e estado atuais.

- Instalar e configurar o Flutter
- Conhecer e saber usar os Widgets básicos do Flutter, como Column, Row, Scaffold, Text, Image, Container, AppBar, ElevatedButton, entre outros
- Entender as diferenças entre widgets Stateless e Stateful
- Desenvolver a primeira aplicação em Flutter

Conteúdos

- **Site** Instalando o Flutter (inglês) (<https://docs.flutter.dev/get-started/install>)
- **Site** Introdução aos Widgets (inglês) (<https://docs.flutter.dev/development/ui/widgets-intro>)
- **Site** Catálogo de Widgets (inglês) (<https://docs.flutter.dev/development/ui/widgets>)
- **YouTube** Filipe Deschamps: A Primeira Aula de Flutter Que Todo Mundo Deveria Ter (<https://youtu.be/J4BVaXkwmM8>)
- **YouTube** Flutter: Widget of the Week (Playlist em Inglês) (<https://www.youtube.com/watch?v=XawP1i314WM&list=PLjxrf2q8roU23XGwz3Km7sQZFTdB996iG>)

Conteúdos Alura:

- **Artigo** Flutter: o que é e tudo sobre o framework (<https://www.alura.com.br/artigos/flutter>)
- **Artigo** Material Design e Cupertino no Flutter: o que são e como usar (<https://www.alura.com.br/artigos/material-design-e-cupertino>)
- **YouTube** Alura: O que é Flutter? #HipstersPontoTube (<https://youtu.be/So5C-XSfGW0>)
- **YouTube** Alura: Live Coding | Desenvolva um aplicativo em Flutter! (https://youtu.be/j8Swzym_EEc)
- **Curso** Flutter: Widgets, Stateless, Stateful, Imagens e Animações (<https://cursos.alura.com.br/course/flutter-widgets-stateless-stateful-imagens-animacoes>)
- **Curso** Flutter: Controller, navegação e estados (<https://cursos.alura.com.br/course/flutter-controller-navigacao-estados>)

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=jpuJ1qrluoU>)
- **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>)

Flutter - Persistência de Dados:

- O conceito de "persistência de dados" refere-se a garantir que as informações inseridas na aplicação serão armazenadas em um meio em que possam ser recuperadas de forma consistente. Ou seja, são registros permanentes e que não são perdidos quando há o encerramento da sessão.
- Saber a armazenar dados usando o SharedPreferences
- Aprender a usar o SQLite no Flutter
- Aprender como persistir dados com Leitura e Escrita de arquivos

Conteúdos

- **Site** Persistência com SQLite - Cookbook (inglês)
(<https://docs.flutter.dev/cookbook/persistence/sqlite>)
- **Site** Persistência com SharedPreferences - Cookbook (inglês)
(<https://docs.flutter.dev/cookbook/persistence/key-value>)
- **Site** Persistência com Arquivos - Cookbook (inglês)
(<https://docs.flutter.dev/cookbook/persistence/reading-writing-files>)
- **Site** SQL: O que é e como usar os principais comandos básicos SQL
(<https://blog.betrybe.com/sql/>)
- **Artigo** SQLite: Principais comandos, dicas, como instalar... (Parte 1)
(<https://medium.com/@mattvianna/sqlite-principais-comandos-dicas-como-instalar-parte-1-bb94e0134935>)
- **YouTube** Código Fonte TV: SQLite (O Banco de Dados de Bolso) // Dicionário do Programador (https://www.youtube.com/watch?v=xOODmm-NdUc&ab_channel=C%C3%B3digoFonteTV)

Conteúdos Alura:

- **Artigo** SQLite: da instalação até sua primeira tabela
(<https://www.alura.com.br/artigos/sqlite-da-instalacao-ate-primeira-tabela>)
- **Artigo** Persistência de Dados no Flutter: o que é? Qual ferramenta usar?
(<https://www.alura.com.br/artigos/alternativas-de-persistencia-de-dados-com-flutter>)
- **Curso** Flutter: aplicando persistência de dados
(<https://cursos.alura.com.br/course/flutter-aplicando-persistencia-dados>)
- **Curso** Formação: domine aplicações offline-first com Flutter
(<https://cursos.alura.com.br/formacao-domine-aplicacoes-offline-first-Flutter>)

Dart Assíncrono:

- Na programação assíncrona as funções não são executadas em ordem. Com o assincronismo, podemos interromper do código para conseguirmos alguma outra informação necessária para a continuar a execução. Isso

significa que o código espera por uma outra parte do código e, enquanto espera, executa as demais partes.

- Entender o que é assincronismo
- Saber usar as expressões `async`, `await` e `future` no Dart

Conteúdos

- **Site** Suporte a Assincronismo | Dart Language Tour (inglês)
(<https://dart.dev/guides/language/language-tour#asynchrony-support>)
- **Artigo** Programação Assíncrona: Async, Await e Future | Dart Codelabs (inglês) (<https://dart.dev/codelabs/async-await>)
- **Artigo** Programação Assíncrona: Streams | Dart Codelabs (inglês)
(<https://dart.dev/tutorials/language/streams>)
- **Artigo** Simultaneidade no Dart | Dart Codelabs (inglês)
(<https://dart.dev/guides/language/concurrency>)
- **Artigo** Como funciona o async-await e Future no Flutter
(<https://www.treinaweb.com.br/blog/como-funciona-o-async-await-e-future-no-flutter>)
- **Artigo** Funções Assíncronas no Dart
(<https://medium.com/@matheusrmribeiro/flutter-descomplica-funcoes-assincronas-ab5380d276dc>)
- **YouTube** Flutter: Playlist Dart Avançado (inglês)
(https://www.youtube.com/playlist?list=PLjxrf2q8roU0Net_g1NT5_vOO3s_FR02J)

Conteúdos Alura:

- **Curso** Curso Persistência com Flutter: crie um app com armazenamento interno (<https://cursos.alura.com.br/course/dart-lidando-exceptions-null-safety>)

Dart - Erros e Exceções:

- O tratamento de erros refere-se aos procedimentos de resposta e recuperação de condições de erro presentes em um aplicativo de software.

Em outras palavras, é o processo de antecipação, detecção e resolução de erros de aplicação, de programação ou de comunicação.

- Entender os métodos `then` e `catchError` do Future no Dart
- Entender o que são erros e exceções no Dart
- Conhecer e saber usar a estrutura `try-on-catch-finally`

Conteúdos

- **Site** Acervo Lima: Exceções no Dart (<https://acervolima.com/tratamento-de-excecoes-no-dart/>)
- **Site** Null safety (inglês) (<https://dart.dev/null-safety>)
- **Site** Entendendo null safety (inglês) (<https://dart.dev/null-safety/understanding-null-safety>)
- **YouTube** Washington Developer: Dart - Tratamento de Erros try-catch, stack-trace, on, finally, rethrow (<https://www.youtube.com/watch?v=O4A3hHvBQ5E>)
- **YouTube** Deivid Willyan | Flutter: Dart - Tratamento de Erros - Exceptions (<https://www.youtube.com/watch?v=MMxJacOOxol>)

Conteúdos Alura:

- **Site** Primeiras aulas do curso Dart: lidando com Exceptions e Null Safety (<https://www.alura.com.br/conteudo/dart-lidando-exceptions-null-safety>)
- **Curso** Dart: lidando com Exceptions e Null Safety (<https://cursos.alura.com.br/course/dart-lidando-exceptions-null-safety>)

Dart - Comunicação com APIs:

- Uma API é uma interface que desenvolvedores de software utilizam para programar a interação com componentes ou recursos de software fora de seu próprio código. Uma definição ainda mais simples é que uma API é a parte de um componente de software que é acessível a outros componentes.
- Entender o que é uma API REST
- Conhecer os comandos básicos de comunicação HTTP

- Saber usar as ferramentas do pacote `http` do Dart
- Saber fazer requisições autenticadas para Web APIs

Conteúdos

- **Site** O que é API REST? (<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>)
- **Site** JSON e Serialização | Flutter Docs (inglês) (<https://docs.flutter.dev/development/data-and-backend/json>)
- **Site** Página do Pacote http - pub.dev (inglês) (<https://pub.dev/packages/http>)
- **Site** Obtendo dados da internet | Flutter Cookbook (inglês) (<https://docs.flutter.dev/cookbook/networking/fetch-data>)
- **Site** Fazendo requisições autenticadas | Flutter Cookbook (inglês) (<https://docs.flutter.dev/cookbook/networking/authenticated-requests>)
- **Site** Enviando dados para internet | Flutter Cookbook (inglês) (<https://docs.flutter.dev/cookbook/networking/send-data>)
- **Site** Removendo dados na internet | Flutter Cookbook (inglês) (<https://docs.flutter.dev/cookbook/networking/delete-data>)
- **YouTube** Flutterando: Consumindo API externa (http) (https://youtu.be/vjeK_oc5cXQ)
- **YouTube** Julio de Lima: Como usar o Postman para Testar APIs Rest (Aula 10) (<https://youtu.be/5S9mMBEtJK4>)

Conteúdos Alura:

- **Artigo** REST: Conceito e fundamentos (<https://www.alura.com.br/artigos/rest-conceito-e-fundamentos>)
- **Site** Primeiras aulas do curso Flutter: entendendo comunicação HTTP e utilizando Web API (<https://www.alura.com.br/conteudo/flutter-comunicacao-http>)
- **Curso** Flutter: entendendo comunicação HTTP e utilizando Web API (<https://cursos.alura.com.br/course/flutter-web-api-integrando-aplicacao>)

- **Curso** Flutter com Web API: evoluindo na integração da aplicação (<https://cursos.alura.com.br/course/flutter-web-api-integracao-aplicacao>)

Nível 2

Flutter - Pacotes Essenciais:

- O Flutter suporta o uso de pacotes compartilhados contribuídos por outros desenvolvedores para os ecossistemas Flutter e Dart. Isto permite construir rapidamente um aplicativo sem ter que desenvolver tudo do zero.
- Conhecer e saber utilizar a plataforma pub.dev
- Conhecer os principais pacotes do Flutter

Conteúdos

- **Site** Usando pacotes e plugins | Flutter Docs (inglês) (<https://docs.flutter.dev/development/packages-and-plugins/using-packages>)
- **Site** Repositório de Pacotes Dart e Flutter | pub.dev (inglês) (<https://pub.dev/>)
- **Site** Pontuação dos Pacotes | pub.dev (inglês) (<https://pub.dev/help/scoring>)
- **Site** Favoritos do Flutter | pub.dev (inglês) (<https://pub.dev/packages?q=is%3Aflutter-favorite>)
- **Site** Pacotes Populares | pub.dev (inglês) (<https://pub.dev/packages?sort=popularity>)
- **YouTube** Dotcode: TOP 5 PACOTES FLUTTER QUE VOCÊ PRECISA CONHECER (https://youtu.be/Fc8D_6TtiKw)
- **YouTube** Flutter: (Playlist) Package of the Week | Flutter (https://www.youtube.com/playlist?list=PLjxrf2q8roU1quF6ny8oFHJ2gBdrYN_AK)
- **Artigo** 12 Bibliotecas úteis para auxiliar no desenvolvimento utilizando Flutter (<https://medium.com/flutter-comunidade-br/12-bibliotecas->

<https://www.alura.com.br/artigos/para-auxiliar-no-desenvolvimento-utilizando-flutter-5982fdf01c80>)

Conteúdos Alura:

- **Curso** Formação: Utilize os serviços do Firebase com Flutter
(<https://cursos.alura.com.br/formacao-flutter-firebase>)
- **Curso** Flutter: implementando persistência de dados com Drift e Hive
(<https://cursos.alura.com.br/course/flutter-implementando-persistencia-dados-drift-hive>)
- **Curso** Flutter: utilizando o Dio para realizar a comunicação com Web API
(<https://cursos.alura.com.br/course/flutter-dio-comunicacao-web-api>)
- **Artigo** Conhecendo pacotes no Flutter
(<https://www.alura.com.br/artigos/pacotes-no-flutter>)
- **Artigo** Utilizando o Dio para comunicação com APIs
(<https://www.alura.com.br/artigos/dio-para-comunicacao-com-api>)

Flutter - Gerenciamento de Estados:

- Em Flutter, 'estado' (state) são os dados que você precisa para reconstruir sua UI a qualquer momento. Quando estes dados mudarem, isso desencadeará um redesenho da interface do usuário.
- Entender o que é Gerenciamento de Estados
- Conhecer e saber usar o Provider
- Conhecer as demais opções de Gerenciamentos de Estados no Flutter

Conteúdos

- **Site** Gerenciamento de Estados | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/data-and-backend/state-mgmt/intro>)
- **Artigo** Gerenciamento de Estado em Flutter e algumas soluções
(<https://ateliware.com/blog/gerenciamento-de-estado-em-flutter>)
- **Artigo** Flutter: Provider, BLoC e State Management
(<https://balta.io/blog/flutter-bloc-provider-state-management>)

- **Artigo** O que aprendi utilizando as gerências de estado do Flutter
(<https://medium.com/flutter-comunidade-br/o-que-aprendi-utilizando-as-ger%C3%A2ncias-de-estado-do-flutter-8f8f2bc7911c>)
- **Site** Opções de Ferramentas para Gerenciamento de Estados | Flutter Docs (inglês) (<https://docs.flutter.dev/development/data-and-backend/state-mgmt/options>)
- **YouTube** Flutterando: Aulão sobre Provider e estrutura de projetos
(https://www.youtube.com/watch?v=VhsqMahAmOk&ab_channel=Flutterando)
- **YouTube** Flutter: Pragmatic State Management Using Provider
(https://www.youtube.com/watch?v=HrBiNHEqSYU&ab_channel=Flutter)

Conteúdos Alura:

- **Artigo** O que é gerenciamento de estados no Flutter e principais ferramentas (<https://www.alura.com.br/artigos/gerenciamento-de-estados-flutter-principais-ferramentas>)
- **Artigo** Como gerenciar estados com Flutter Provider
(<https://www.alura.com.br/artigos/como-gerenciar-estados-com-flutter-provider>)
- **Artigo** Bloc ou Cubit no Flutter: qual escolher?
(<https://www.alura.com.br/artigos/bloco-c-cubit>)
- **Artigo** GetX: o canivete suíço do Flutter
(<https://www.alura.com.br/artigos/getx-canivete-suico>)
- **Curso** Flutter: gerenciamento de estados com Provider
(<https://www.alura.com.br/curso-online-flutter-gerenciamento-de-estados-provider>)
- **Curso** Formação: Gerenciamento de estados com Flutter
(<https://cursos.alura.com.br/formacao-gerenciamento-estados-flutter>)

Flutter - Testes:

- O teste de software é o processo de avaliação e verificação de que um software realmente faz o que deveria fazer. Os benefícios dos testes

incluem a prevenção de bugs, a redução dos custos de desenvolvimento e a melhoria do desempenho.

- Conhecer Testes Unitários no Flutter
- Conhecer Testes de Widgets no Flutter
- Conhecer Testes de Integração no Flutter

Conteúdos

- **Site** Testando Apps Flutter | Flutter Docs (inglês)
(<https://docs.flutter.dev/testing>)
- **Site** Uma introdução aos Testes Unitários no Flutter | Flutter Cookbook (inglês) (<https://docs.flutter.dev/cookbook/testing/unit/introduction>)
- **Site** Uma introdução aos Testes de Widget no Flutter | Flutter Cookbook (inglês) (<https://docs.flutter.dev/cookbook/testing/widget/introduction>)
- **Site** Testes de Integração no Flutter | Flutter Docs (inglês)
(<https://docs.flutter.dev/testing/integration-tests>)
- **YouTube** Flutterando: Semana do Flutter - Testes de Unidade #1 | O que é unidade? (<https://youtu.be/zlYQe-9QMug>)
- **YouTube** Prof. Diego Antunes: Flutter - Teste de Integração (Integration Test ou E2E Flutter) (<https://youtu.be/GEvNj7uogYE>)
- **YouTube** Flutter: Testando Aplicativos Flutter - Garantindo que o código vai funcionar (inglês) (<https://youtu.be/bj-oMYyLZEY>)
- **YouTube** Flutter: Dicas de Performance com Testes no Flutter (inglês)
(<https://youtu.be/PKGguGUwSYE>)
- **Site** Como testar um app Flutter | Google Codelabs (inglês)
(<https://codelabs.developers.google.com/codelabs/flutter-app-testing?hl=pt-br#0>)

Conteúdos Alura:

- **Site** Primeiras aulas do curso Flutter: aplicando testes de unidade, de Widget e Mocks (<https://www.alura.com.br/conteudo/flutter-testes-unidade-widget-mocks>)

- **Curso** Flutter: aplicando testes de unidade, de Widget e Mocks (<https://www.alura.com.br/curso-online-flutter-testes-unidade-widget-mocks>)
- **Curso** Flutter: criando testes automatizados de unidade, widgets e dublês (<https://cursos.alura.com.br/course/flutter-criando-testes-automatizados-unidade-widgets-dubles>)
- **Curso** Flutter: dominando testes de integração (<https://cursos.alura.com.br/course/flutter-dominando-testes-integracao>)
- **Curso** TDD em Flutter: desenvolvimento orientado a testes (<https://cursos.alura.com.br/course/tdd-flutter-desenvolvimento-orientado-testes>)
- **Curso** Testes no CI com Flutter: utilizando Bitrise, GitHub Actions e Codemagic (<https://cursos.alura.com.br/course/testes-ci-flutter-utilizando-bitrise-github-actions-codemagic>)
- **Artigo** Testes no Flutter: O que é? Quando usar? Como começar a aprender? (<https://www.alura.com.br/artigos/testes-flutter>)

Flutter - Depuração:

- Depuração (em inglês "debugging" ou "debug") é o processo de encontrar e reduzir defeitos num aplicativo de software ou mesmo em hardware. Erros de software incluem aqueles que previnem o programa de ser executado e aqueles que produzem um resultado inesperado.
- Conhecer as técnicas e ferramentas de depuração no Flutter
- Saber depurar códigos Dart e Flutter
- Conhecer o Dart DevTools

Conteúdos

- **YouTube** Flutterando: Depuração (Breakpoint) (<https://youtu.be/cUMsu0Kxo7k>)
- **Site** Depurando Aplicativos Flutter | Flutter Docs (inglês) (<https://docs.flutter.dev/testing/debugging>)

- **Site** Depurando Códigos de forma Programática | Flutter Docs (inglês) (<https://docs.flutter.dev/testing/code-debugging>)
- **Site** Usando um depurador OEM | Flutter Docs (inglês) (<https://docs.flutter.dev/testing/oem-debuggers>)
- **YouTube** Flutter: Propriedades de Depuração | Decoding Flutter (inglês) (<https://youtu.be/DnC7eT-vh1k>)
- **YouTube** Flutter: Depure e crie o perfil do seu aplicativo com o Flutter DevTools | Demo (inglês) (<https://youtu.be/J7s5bu49k3U>)
- **Site** Conhecendo o Dart DevTools | Dart Docs (inglês) (<https://dart.dev/tools/dart-devtools>)
- **YouTube** Flutter: Dart DevTools | Decoding Flutter (inglês) (<https://youtu.be/nq43mP7hjAE>)

Flutter - Jogos:

- Aprender a criar jogos com Flutter
- Conhecer o Kit de Desenvolvimento de Jogos Casuais do Flutter
- Conhecer o pacote Flame como Engine de Jogos Flutter
- Conhecer o pacote Bonfire como Engine de Jogos para Flutter

Conteúdos

- **Site** Flutter: Jogos! (inglês) (<https://flutter.dev/games>)
- **Artigo** Flutter: Kit de Desenvolvimento de Jogos Casuais (inglês) (<https://docs.flutter.dev/resources/games-toolkit>)
- **YouTube** Flutter: Fazendo sue primeiro jogo com Flutter (inglês) | Flutter Focus (<https://youtu.be/zGgeBNiRy-8>)
- **YouTube** Meu primeiro jogo com Flutter e Flame (inglês) (<https://youtu.be/3GGcVF9p8bY>)
- **YouTube** Flutter Mapp: Criando um jogo do estilo RPG com Flutter usando Bonfire (<https://youtu.be/Qv4ma-YHIHw>)
- **YouTube** Flame Engine: O básico do Flutter Flame | Vikings 2022 (inglês) (<https://youtu.be/NqjaLM0TWm0>)

- **YouTube** Craig Oda: Velocidade, Gravidade e Movimento no Flutter (inglês) (<https://youtu.be/hwQpBuZoV9s>)
- **YouTube** Flutterando: Criando jogos de RPG usando o Flutter Bonfire (<https://youtu.be/vrhJTdwiXBg>)

Nível 3

Flutter - Nativo:

- Conhecer as especificidades do desenvolvimento nativo para Android
- Conhecer as especificidades do desenvolvimento nativo para iOS

Conteúdos

- **Site** Android: Hospedando Views nativas do Android em seu aplicativo Flutter com visualizações de plataforma - Integração Android | Flutter Docs (inglês) (<https://docs.flutter.dev/development/platform-integration/android/platform-views>)
- **Site** Android: API de Migração - Integração Android | Flutter Docs (inglês) (<https://docs.flutter.dev/development/platform-integration/android/plugin-api-migration>)
- **Site** Android: Migração para Android X - Integração Android | Flutter Docs (inglês) (<https://docs.flutter.dev/development/platform-integration/android/androidx-migration>)
- **Site** Android: Migração da Splash Screen - Integração Android | Flutter Docs (inglês) (<https://docs.flutter.dev/development/platform-integration/android/splash-screen-migration>)
- **Site** iOS: Usando o Cupertino | Flutter Docs (inglês) (<https://docs.flutter.dev/development/ui/widgets/cupertino>)
- **Site** iOS: Veggie Seasons - iOS com Flutter (inglês) (<https://github.com/flutter/samples/tree/main/veggiesseasons>)
- **YouTube** iOS: Criando para iOS com Flutter (Google I/O'19) (inglês) (https://www.youtube.com/watch?v=ZBJa-xjZl3w&ab_channel=Flutter)

- **Site** iOS: iOS App Clip | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/platform-integration/ios/ios-app-clip>)
- **Site** iOS: Hospedando Views nativas do iOS em seu aplicativo Flutter com visualizações de plataforma | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/platform-integration/ios/platform-views>)
- **Site** Escrevendo código para uma plataforma específica | Flutter Docs (inglês) (<https://docs.flutter.dev/development/platform-integration/platform-channels>)
- **Site** Interoperabilidade do C | Dart Guides (inglês)
(<https://dart.dev/guides/libraries/c-interop>)
- **Site** Interoperabilidade do JavaScript | Dart Guides (inglês)
(<https://dart.dev/web/js-interop>)
- **Site** Interoperabilidade do C para Android | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/platform-integration/android/c-interop>)
- **Site** Interoperabilidade do C para iOS | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/platform-integration/ios/c-interop>)

Conteúdos Alura:

- **Curso** Formação Flutter: integrando com Recursos Nativos
(<https://cursos.alura.com.br/formacao-flutter-recursos-nativos>)

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>)

Flutter - Animações:

- Conhecer a diferença entre Animações Implícitas e Explícitas no Flutter
- Conhecer e saber usar os Widgets para Animações Implícitas
- Entender os conceitos de Duração (Duration) e Curvas (Curves)
- Conhecer a técnica de Animações Explícitas no Flutter

Conteúdos

- **Site** Introdução à Animações no Flutter | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/ui/animations>)
- **Site** Animações Implícitas | Flutter Docs (inglês)
(<https://docs.flutter.dev/codelabs/implicit-animations>)
- **YouTube** Flutter: Animação Básica com Animações Implícitas | Flutter on Focus (inglês) (<https://youtu.be/IVTjpW3W33s>)
- **Site** Lista de Widgets de Animação do Flutter | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/ui/widgets/animation>)
- **YouTube** Dotcode: Destrinchando o AnimatedContainer
(<https://youtu.be/8INWgsJOZUw>)
- **YouTube** Prof. Diego Antunes: 5 animações em 5 minutos
(https://www.youtube.com/watch?v=7t_xZZh_K4Y&ab_channel=Prof.DiegoAntunes)
- **Artigo** Animações com Flutter (<https://ateliware.com/blog/o-mundo-das-animacoes-em-flutter>)
- **Site** Simples animação de Opacidade | Flutter Cookbook (inglês)
(<https://docs.flutter.dev/cookbook/animation/opacity-animation>)

- **Site** Animando propriedades de um Container | Flutter Cookbook (inglês) (<https://docs.flutter.dev/cookbook/animation/animated-container>)
- **Site** Animando transições de tela | Flutter Cookbook (inglês) (<https://docs.flutter.dev/cookbook/animation/page-route-animation>)
- **Site** Animando um Widget com Simulação de Física | Flutter Cookbook (inglês) (<https://docs.flutter.dev/cookbook/animation/physics-simulation>)
- **Site** Animações Explícitas no Flutter | Flutter Docs (inglês) (<https://docs.flutter.dev/codelabs/explicit-animations>)
- **Site** Animações com Hero | Flutter Docs (inglês) (<https://docs.flutter.dev/development/ui/animations/hero-animations>)
- **Site** Animações Escalonadas | Flutter Docs (inglês) (<https://docs.flutter.dev/development/ui/animations/staggered-animations>)
- **YouTube** Flutter: Fazendo animações do jeito certo | Flutter Europe (inglês) (<https://youtu.be/wnARLByOtKA>)

Conteúdos Alura:

- **Curso** Flutter ainda mais atraente com animações (<https://cursos.alura.com.br/formacao-flutter-animacoes>)

Flutter - Multiplataforma Desktop:

- Conhecer e saber usar as opções de criação em multiplataforma do Flutter

Conteúdos

- **Site** Flutter no Desktop (inglês) (<https://flutter.dev/multi-platform/desktop>)
- **Site** Suporte para Desktop do Flutter (inglês) (<https://docs.flutter.dev/development/platform-integration/desktop>)
- **Site** Criando aplicativos para Linux com Flutter (inglês) (<https://docs.flutter.dev/development/platform-integration/linux/building>)
- **Site** Criando aplicativos para Windows com Flutter (inglês) (<https://docs.flutter.dev/development/platform-integration/windows/building>)

- **Site** Criando aplicativos para macOS com Flutter (inglês)
(<https://docs.flutter.dev/development/platform-integration/macos/building>)

Flutter - Multiplataforma Web:

- Conhecer as especificidades da criação de apps Flutter para Web

Conteúdos

- **Site** Perguntas Frequentes - Flutter Web | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/platform-integration/web/faq>)
- **Site** Renderizadores - Flutter Web | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/platform-integration/web/renderers>)
- **Site** Inicialização - Flutter Web | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/platform-integration/web/initialization>)
- **Site** Uso de imagens - Flutter Web | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/platform-integration/web/web-images>)
- **Artigo** PWA em Flutter (inglês) (<https://blog.codemagic.io/pwa-in-flutter/>)

Flutter - Implantação (Deployment):

- Implantação é a fase do ciclo de vida de um software (programa computacional, documentação e dados), no contexto de um Sistema de Informação, que corresponde textualmente à passagem do software para a produção.
- Conhecer as técnicas de ofuscação de código Dart
- Conhecer e saber implantar aplicativos Flutter em várias plataformas
- Entendendo Flavors em Flutter
- Conhecer Entrega Contínua em Flutter

Conteúdos

- **Site** Ofuscando Código Dart | Flutter Docs (inglês)
(<https://docs.flutter.dev/deployment/obfuscate>)
- **Site** Implantando para Android e lançando na PlayStore | Flutter Docs (inglês) (<https://docs.flutter.dev/deployment/android>)

- **Site** Implantando para iOS e lançando na AppStore | Flutter Docs (inglês) (<https://docs.flutter.dev/deployment/ios>)
- **Site** Implantando para macOS e lançando na AppStore | Flutter Docs (inglês) (<https://docs.flutter.dev/deployment/macos>)
- **Site** Implantando para Linux e lançando na SnapStore | Flutter Docs (inglês) (<https://docs.flutter.dev/deployment/linux>)
- **Site** Implantando para Windows | Flutter Docs (inglês) (<https://docs.flutter.dev/deployment/windows>)
- **Site** Implantando para Web | Flutter Docs (inglês) (<https://docs.flutter.dev/deployment/web>)
- **Site** Criando Flavors para Flutter | Flutter Docs (inglês) (<https://docs.flutter.dev/deployment/flavors>)
- **Site** Entrega Contínua com Flutter | Flutter Docs (inglês) (<https://docs.flutter.dev/deployment/cd>)
- **YouTube** MJSD Coding: Build and Release Flutter App to App Store 2022 (inglês) (<https://youtu.be/akFF1uJWZck>)
- **YouTube** penseapp: Como fazer o deploy do seu aplicativo flutter na Playstore! (<https://youtu.be/PM-wILFbPbg>)
- **Artigo** Publicando App Flutter na PlayStore (inglês) (<https://medium.flutterdevs.com/publishing-flutter-app-to-playstore-fa7543b61a7b>)

Conteúdos Alura:

- **Curso** Formação: Flutter CI/CD: automatize seu trabalho (<https://cursos.alura.com.br/formacao-flutter-ci-cd>)
- **Artigo** Flutter CI/CD: o que é? Como funciona e como usar? (<https://www.alura.com.br/artigos/flutter-ci-cd>)

Flutter - Arquitetura:

- Em programação, existem vários padrões de arquitetura para arquivos e pastas de projetos, que fornecem aos desenvolvedores facilidade para encontrar, pesquisar e entender o código escrito.

- Conhecer as principais Arquiteturas usadas pela comunidade Flutter, como MVC, MVVM, MVP, etc

Conteúdos

- **Artigo** Uma introdução à Arquitetura Clean com Flutter (inglês)
(<https://medium.com/ruangguru/an-introduction-to-flutter-clean-architecture-ae00154001b0>)
- **Artigo** Flutter: Arquitetura MVVM (<https://medium.com/flutterworld/flutter-mvvm-architecture-f8bed2521958>)
- **Artigo** Padrões de Projeto com Flutter: MVC
(<https://medium.flutterdevs.com/design-patterns-in-flutter-part-1-c32a3ddb00e2>)
- **Artigo** Estrutura de Projeto em Flutter: Feature-first ou Layer-first? (inglês)
(<https://codewithandrea.com/articles/flutter-project-structure/>)
- **YouTube** Deivid Willyan | Flutter: Flutter NV1 - MVC Pattern
(https://www.youtube.com/watch?v=KDDU4ZT_fXU)
- **YouTube** Flutterando: MVVM no Flutter - Estrutura para iniciantes
(<https://www.youtube.com/watch?v=WgadnZcujuC>)

Conteúdos Alura:

- **Curso** Formação: Flutter: desvendando arquiteturas
(<https://cursos.alura.com.br/formacao-flutter-desvendando-arquiteturas>)

Flutter - Pacotes e Plugins:

- Conhecer o desenvolvimento de plugins para Flutter

Conteúdos

- **YouTube** Pacotes ou Plugins | Decoding Flutter (inglês)
(<https://youtu.be/Y9WifT8aN6o>)
- **YouTube** Flutter Plugin Development (inglês)
(<https://youtu.be/PLPhmsDovwQ>)
- **Site** Desenvolvendo pacotes | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/packages-and-plugins/developing->

[packages](#))

- **Site** Pluguins Federados | Flutter Docs (inglês)
(<https://docs.flutter.dev/development/packages-and-plugins/developing-packages#federated-plugins>)
- **Artigo** Escrevendo um bom plugin Flutter (inglês)
(<https://medium.com/flutter/writing-a-good-flutter-plugin-1a561b986c9c>)
- **Site** Contribuindo para a comunidade (inglês)
(<https://github.com/flutter/flutter/blob/master/CONTRIBUTING.md>)

Habilidade Auxiliar: Infraestrutura e Back-end

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-ZyA0Q>)
- **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=VL_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=BAmvmaKQklQ>)
- **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>)

Design Patterns:

- Na engenharia de software, um "padrão de projeto" (Design Pattern em inglês) é uma solução geral e reutilizável para um problema que ocorre normalmente dentro de um determinado contexto de projeto de software.
- Conhecer e aplicar os principais Design Patterns

Conteúdos

- **Artigo** Design Patterns — O que são e quais os benefícios?
(<https://djesusnet.medium.com/design-patterns-gof-o-que-s%C3%A3o-e-quais-os-benef%C3%ADcios-9cd0cfdd6ebf>)
- **Artigo** Design Patterns (<https://medium.com/xp-inc/design-patterns-727494af001d>)
- **YouTube** Matheus Castiglioni: O que são design patterns (padrões de projetos) e para que servem? (<https://www.YOUTUBE.com/watch?v=KNWd1aWtsMw>)
- **YouTube** RinaldoDev: Design Patterns - o que são? Por que aprender a utilizar? (<https://www.YOUTUBE.com/watch?v=AWOf6Wo6gtg>)

Conteúdos Alura:

- **Podcast** Hipsters.tech: TechGuide - Design Patterns – Hipsters Ponto Tech #347 (<https://www.hipsters.tech/techguide-design-patterns-hipsters-ponto-tech-347/>)

- **Artigo** Design patterns: Breve introdução aos padrões de projeto (<https://www.alura.com.br/artigos/design-patterns-introducao-padroes-projeto>)
- **Podcast** Hipsters.tech: Design Patterns - Hipsters 206 (<https://www.alura.com.br/podcast/design-patterns-hipsters-206-a345>)
- **Curso** Curso Design Patterns em Java I: boas práticas de programação (<https://www.alura.com.br/curso-online-introducao-design-patterns-java>)
- **Curso** Curso Design Patterns C# I: boas práticas de programação (<https://www.alura.com.br/curso-online-design-patterns-dotnet>)
- **Curso** Curso Design Patterns em PHP: padrões comportamental (<https://www.alura.com.br/curso-online-php-design-pattern-comportamental>)
- **Curso** Curso Design Patterns em PHP: padrões estruturais (<https://www.alura.com.br/curso-online-php-design-pattern-estrutural>)
- **Curso** Curso Design Patterns em PHP: padrões criacionais (<https://www.alura.com.br/curso-online-php-design-pattern-criacional>)

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=Z45BTNeZ1I0>)
- **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=8JI9wQ8sUdQ>)
- **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=6SfrO3D4dHM>)
- **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-melhore-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 Principio da Inversão de Dependência usando a Injeção de Dependência para ajudar (<https://youtu.be/xpcCDCH5k4Y>)

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=HynTfTli4mw>)
- **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/8ehIZfyN1S0?si=I7-I5l4zsnYLIjD>)

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=EXp0gg9kGxl&ab_channel=Firebase)
- **YouTube** Prof. Diego Antunes: Instalação do Firebase no Flutter em 2022 (FlutterFire e Realtime Database) (https://www.youtube.com/watch?v=OjdGSoDntZQ&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=L0LjsG_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 e Design

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-112b2ee1ddf>)
- **YouTube** Caio Gonzalez: O que é Design System? / Guia para começar o seu próprio Design System (<https://www.youtube.com/watch?v=ajqbXpFVaAw>)

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>)

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_IAcQQ)
- **Curso** Formação Figma (<https://www.alura.com.br/formacao-figma>)

Componentes de design:

- Conhecer os componentes descrevem um layout ou interface

Conteúdos

- **Artigo** Componentes Principais UI
(<https://medium.com/@prikuer/componentes-principais-ui-8b538a3aaaa5>)
- **Artigo** Compartilhando CSS em projetos do Nx
(<https://www.alura.com.br/artigos/compartilhando-css-projetos-nx>)
- **YouTube** danielmacedoco: UI Design e Design System - Visão geral e prática sobre alguns componentes (<https://www.youtube.com/watch?v=VpgQNeSltpA>)
- **YouTube** 4 Exemplos de Componentes interativos no Figma
(<https://www.youtube.com/watch?v=mGL3jrY-OBc>)

Conteúdos Alura:

- **Curso** Curso Design System: criando componentes e documentando (<https://www.alura.com.br/curso-online-design-system-componentes-documentando>)
- **Curso** Curso Figma: conhecendo componentes da interface (<https://www.alura.com.br/curso-online-figma-conhecendo-componentes-interface>)

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-svg-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>)

Acessibilidade em Flutter:

- Acessibilidade Digital é a eliminação de barreiras na Web. O conceito pressupõe que os sites e aplicativos sejam projetados de modo que todas as pessoas possam perceber, entender, navegar e interagir de maneira efetiva

com as páginas, incluindo pessoas com necessidades de acessibilidades, como pessoas com problemas de visão, daltonismo, dificuldades auditivas, comprometimento motor, deficiência cognitivas e muitos outros tipos de deficiência.

- Escrever código com acessibilidade em mente
- Aumentar a visibilidade do texto
- Usar controles grandes e simples
- Descrever cada elemento de IU

Conteúdos

- **Site** Flutter - Documentação - Acessibilidade (em inglês) (<https://docs.flutter.dev/development/accessibility-and-localization/accessibility>)
- **Artigo** Flutter: dicas para acessibilidade no seu aplicativo (<https://medium.com/tableless/flutter-dicas-para-acessibilidade-no-seu-aplicati-16c8fee46342>)
- **Artigo** Melhorando acessibilidade de linhas e colunas em Flutter (<https://blog.cisne.dev/melhorando-acessibilidade-de-linhas-e-colunas-em-flutter/>)
- **YouTube** Flutter: Construindo acessibilidade com Flutter (em inglês) (<https://www.youtube.com/watch?v=bWbBgbmAdQs>)
- **YouTube** monterail: Funcionalidades de acessibilidade e como implementá-las em Flutter (em inglês) (<https://www.youtube.com/watch?v=sO97YsPxwsW>)

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>)