

# Guia Ágil

---

TechGuide - Alura

---

## Flutter

---

### 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

#### ☐ 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

#### ☐ **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

#### ☐ **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

#### ☐ **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

#### ☐ **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`

#### ☐ **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

## **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

#### ☐ **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

#### ☐ **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

#### ☐ **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

#### ☐ **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

## Nível 3

### ☐ Flutter - Nativo:

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

### ☐ 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

### ☐ 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

### ☐ Flutter - Multiplataforma Desktop:

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

### ☐ Flutter - Multiplataforma Web:

- Conhecer as especificidades da criação de apps Flutter para Web

### ☐ 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

#### ☐ **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

#### ☐ **Flutter - Pacotes e Plugins:**

- Conhecer o desenvolvimento de plugins para Flutter

## **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

## ☐ 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

## ☐ 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

## ☐ 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

## ☐ 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

## ☐ 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

#### ☐ **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.

#### ☐ **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.

#### ☐ **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



# 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

## ☐ 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

## ☐ Componentes de design:

- Conhecer os componentes descrevem um layout ou interface

## ☐ Sistemas de cores:

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

## ☐ Como usar fontes:

- Escolher a fonte mais adequada para determinado projeto

## ☐ 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

#### ☐ **Design Responsivo:**

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