Guia Ágil

TechGuide - Alura

Flutter

Nivel 1

- [ ] **Dart - Fundamentals**:
  - Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.
  - Installing and configuring Dart
  - Learning the Dart keywords
  - Building terminal applications using Dart
  - Understanding how dynamism works in Dart
  - Understanding and learning how to use Null Safety in Dart
- [ ] **Flutter - Fundamentals**:
  - Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase.
  - Flutter widgets are built using a modern framework that takes inspiration from React. The central idea is that you build your UI out of widgets. Widgets describe what their view should look like given their current configuration and state.

- Installing and configuring Flutter
- Learning how to use basic Flutter widgets, such as Column, Row, Scaffold, Text, Image, Container, AppBar, ElevatedButton, among other
- Understanding the differences between Stateless and Stateful widgets
- Developing your first Flutter application

- [ ] **Object-oriented Programming Concepts**:

- Object-oriented programming (OOP) is a programming paradigm based on the concept of 'objects', which can contain data and code: data in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). A common feature of objects is that procedures (or methods) are attached to them and can access and modify the object's data fields. Some of the main concepts are classes and instances, inheritance, and encapsulation.
- How objects work
- Creating and using constructors
- What classes are
- Creating and using Methods
- How encapsulation works
- What inheritance is
- What polymorphism is
- How interfaces work
- What abstractions are

- [ ] **Flutter - Persistence**:

- Persistence in computer science refers to saving data for future access. It also involves the actions of reading, updating and deleting data.
- Understanding what data persistence is
- Learning how to store data using SharedPreferences
- Learning how to use SQLite in Flutter

- Learning how to persist data reading and writing files

- [ ] **Dart - Asynchronous**:

- In asynchronous programming, the functions are not executed in order. We can interrupt the code to get some other information needed to continue execution. This means that the code waits for another part of the code, and while it waits, it can execute the other parts.
- Understanding what asynchronism is
- Learning how to use the `async`, `await` and `future` expressions in Dart
- Understanding Future's `then` and `catchError` methods in Dart

- [ ] **Dart - Errors and Exceptions**:

- Error handling refers to the procedures for responding to and recovering from error conditions present in a software application. In other words, it is the process of anticipating, detecting, and resolving application, programming, or communication errors.
- Understanding what errors and exceptions are in Dart
- Learning how to use the `try-on-catch-finally` structure

- [ ] **Dart - Communication with APIs**:

- An API is an interface that software developers use to programmatically interact with software components or resources outside of their own code. An even simpler definition is that an API is the part of a software component that is accessible to other components.
- Understanding REST APIs
- Learning the basic HTTP communication commands
- Learning how to use the Dart `http` package tools
- Learning how to make authenticated requests to Web APIs

Nivel 2

- [ ] **Flutter - Essential Packages**:

- Flutter supports using shared packages contributed by other developers to the Flutter and Dart ecosystems. This allows quickly building an app without having to develop everything from scratch.
- Learning how to use the pub.dev platform
- Learning about the main Flutter packages

- [ ] **Flutter - State Management**:

- In Flutter, 'state' is whatever data you need in order to rebuild your UI at any moment in time. When this data change, it will trigger a redraw of the user interface.
- Understanding State Management
- Learning how to use the Provider
- Learning about other State Management options in Flutter

- [ ] **Flutter - Testing**:

- Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.
- Understanding unit tests in Flutter
- Understanding widget tests in Flutter
- Understanding integration tests in Flutter

- [ ] **Flutter - Debugging**:

- Debugging is the process of finding and resolving bugs (defects or problems that prevent correct operation) within computer programs, software, or systems.
- Learning debugging techniques and tools in Flutter
- Learning how to debug Dart and Flutter code
- Learning about Dart DevTools

- [ ] **Flutter - Games**:

- Leaning how to create games with Flutter
- Getting to know Flutter's Casual Game Development Kit
- Learning about the Flame package as a Flutter Game Engine
- Learning about the Bonfire package as a Flutter Game Engine

Nivel 3

- [ ] **Flutter - Native**:
  - Getting to know the specifics of native development for Android
  - Getting to know the specifics of native development for iOS
- [ ] **Data Structures**:
  - In the context of computers, the data structure is a specific way of storing and organizing data in the computer's memory so that these data can be easily retrieved and efficiently used when needed later.
  - Knowing the main data structures (linked list, stack, queue, tree, etc)
  - Implementing the main data structures
- [ ] **Flutter - Animations**:
  - Learning the difference between Implicit and Explicit Animations in Flutter
  - Learning about and using Widgets for Implicit Animations
  - Understanding the concepts of Duration and Curves
  - Learning about the technique of Explicit Animations in Flutter
- [ ] **Flutter - Multiplatform - Desktop**:
  - Learning about and using Flutter's multiplatform creation options
- [ ] **Flutter - Multiplatform - Web**:
  - Getting to know the specifics of creating Flutter apps for the Web
- [ ] **Flutter - Deployment**:

- Software deployment includes all of the steps, processes, and activities that are required to make a software system or update available to its intended users.
- Learning about Dart code obfuscation techniques
- Learning know how to deploy Flutter applications on various platforms
- Understanding Flavors in Flutter
- Learning about Continuous Delivery in Flutter

- [ ] **Flutter - Architecture**:

- In programming, there are several architectural patterns for project files and folders, that provide developers with ease in finding, researching, and understanding the written code.
- Getting to know the main architecture patterns used by the Flutter community, such as MVC, MVVM, MVP etc

- [ ] **Flutter - Packages and Plugins**:

- Learning how to develop plugins for Flutter

Habilidade Auxiliar: Infrastructure and Back-end

- [ ] **Git & GitHub - Fundamentals**:

- Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
- GitHub is a hosting service for software development and version control using Git.
- Creating a repository
- Cloning a repository
- Committing, pushing and pulling to and from the repository
- Reversing a commit
- Creating branches and pull requests

- Handling merge and conflicts

- [ ] **HTTP - Fundamentals**:

- HTTP stands for Hyper Text Transfer Protocol. Communication between client computers and web servers is done by sending HTTP Requests and receiving HTTP Responses.
- Understanding the difference between HTTP verbs
- Testing requests and checking the status codes in the browser
- Learning how to make a HTTP request on the command line with WGET
- Downloading an image with WGET
- Performing a POST

- [ ] **JSON**:

- JSON stands for JavaScript Object Notation. It is a text format for storing and transporting data.
- Creating an object
- Transforming an object into a string
- Transforming a string into an object
- Manipulating an object

- [ ] **Design Patterns**:

- In software engineering, a Design Pattern is a general, reusable solution to a commonly occurring problem within a given context in software design. It is a description or template for how to solve a problem that can be used in many different situations. Design Patterns are formalized best practices that the programmer can use to solve common problems when designing an application or system.
- Getting familiarized with and applying the main Design Patterns

- [ ] **Command Line - Fundamentals**:

- CLI is a command line program that accepts text input to execute operating system functions.

- Knowing the most important commands

- [ ] **Cloud - Fundamentals**:

- Cloud, or cloud computing, is the distribution of computing services over the Internet using a pay-as-you-go pricing model. A cloud is composed of various computing resources, ranging from the computers themselves (or instances, in cloud terminology) to networks, storage, databases, and everything around them. In other words, everything that is normally needed to set up the equivalent of a server room, or even a complete data center, will be ready to use, configured, and run.
- Knowing the difference between IaaS, PaaS and SaaS
- Knowing the largest cloud providers
- Specializing in a specific provider of your choice

- [ ] **SOLID**:

- SOLID has five principles that are considered best practices in software development that help programmers write cleaner code by separating responsibilities, reducing docking, easing refactoring, and encouraging code reuse.

- [ ] **Clean Architecture**:

- Clean architecture is a way of developing software, such that just by looking at the source code of a program, you should be able to tell what the program does.

- [ ] **Firebase**:

- Firebase is a Backend-as-a-Service (BaaS) app development platform that provides hosted backend services such as a realtime database, cloud storage, authentication, crash reporting, machine learning, remote configuration, and hosting for your static files.
- Understanding how to install Firebase
- Getting acquainted with Firebase documentation
- Learning about the Firebase tools available

Habilidade Auxiliar: UX and Design

- [ ] **Design Systems**:
  - A design system is a collection of reusable components, guided by clear standards, that can be assembled together to build applications.
  - Creating and maintaining libraries that will be consumed and used as a standard for building a project
- [ ] **Figma - Fundamentals**:
  - Figma is a collaborative web application for interface design. The feature set of Figma focuses on user interface and user experience design, with an emphasis on real-time collaboration, utilising a variety of vector graphics editor and prototyping tools.
  - Creating page layouts and components
- [ ] **Design components**:
  - Knowing the components that describe a layout or interface
- [ ] **Color systems**:
  - Defining a color palette that makes sense for a given interface
- [ ] **How to use Fonts**:
  - Choosing the most appropriate font for a given project
- [ ] **Flutter - Accessibility**:
  - Web accessibility is the inclusive practice of ensuring there are no barriers that prevent interaction with, or access to, websites by people with physical disabilities, situational disabilities, and socio-economic restrictions on bandwidth and speed.
  - Writing code with accessibility in mind
- [ ] **Responsive Design**:

- Responsive web design (RWD) or responsive design is an approach to web design that aims to make web pages render well on a variety of devices and window or screen sizes from minimum to maximum display size to ensure usability and satisfaction.
- Adjusting your pages to the user's screen size
- Learning about Media queries
- Knowing the concept of Mobile first
