Android

TechGuide - Alura, FIAP e PM3

Nivel 1

Kotlin - Fundamentos:

- Kotlin é uma linguagem de programação multiplataforma, orientada a objetos, funcional e estaticamente tipada. Ela compila para a máquina virtual Java e também pode ser traduzida para a linguagem JavaScript e compilada para código nativo (via LLVM). É a linguagem oficial do sistema Android da Google.
- Entender a sua sintaxe
- 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
- Null Safety (Elimine o perigo de referências nulas)
- Exceptions e Throwables
- Convenções de código
- Manipular Coleções, arrays e listas
- Orientação a Objetos com Kotlin (Properties, Data class, Companion Objects, Delegation)
- Receber dados de uma API

Conteúdos

- `Site` Documentação Kotlin: começando com Kotlin (inglês) (https://kotlinlang.org/docs/getting-started.html)
- `Site` Documentação Kotlin: Tipos Básicos e Kotlin (inglês) (https://kotlinlang.org/docs/basic-types.html)
- `Site` Documentação Kotlin: Sintaxe básica do Kotlin(inglês) (https://kotlinlang.org/docs/basic-syntax.html)
- `Site` Documentação Kotlin: Convenções de código do Kotlin (inglês) (https://kotlinlang.org/docs/coding-conventions.html)
- `Site` Explorando a documentação: Números em Kotlin (https://dev.to/kotlinautas/explorando-a-documentacao-numeros-em-kotlin-15go)
- `YouTube` Kotlin // Dicionário do Programador (https://youtu.be/BfjRYBN7Ur8)

Conteúdos Alura:

- `Artigo` Linguagem Kotlin: o que é, para que serve e um Guia para aprender (https://www.alura.com.br/artigos/kotlin)

- `Podcast` Case Contabilizei: Kotlin - Hipsters Ponto Tech (https://www.hipsters.tech/case-contabilizei-kotlin-hipsters-ponto-tech-310/)
- `Curso` Formação Kotlin (https://cursos.alura.com.br/formacao-kotlin)
- `Desafio` 7 Days of Code: Kotlin (https://7daysofcode.io/matricula/kotlin)

Android - Fundamentos:

- Conhecer Kotlin, Java ou C++ que são as linguagens para desenvolver Apps Android.
- Entender como o SDK do Android empacota o código e recursos do App em um APK (Android Package - Pacote Android) para rodar no SO do Android
- Saber os componentes de entrada de um App - Activity, Service, Broadcast Receiver e Content Provider
- Compreender os ciclos de vida de componentes do Android e como funcionam - O ciclo de vida de uma Activity
- Ativar componentes de entrada do App com Intents
- Conhecer o arquivo de manifesto e os principais itens de configuração
- Entender quais são os recursos de um projeto Android - código fonte, recursos estáticos, drawables, layout, mipmap, values etc
- Criar um projeto Android com o Android Studio e rodar em um dispositivo físico ou virtual
- Conhecer bibliotecas do jetpack para garantir a compatibilidade entre as versões do Android

Conteúdos

- `Site` Documentação Android: Fundamentos de aplicativos (https://developer.android.com/guide/components/fundamentals?hl=pt-br)
- `Site` Documentação Android: Componentes de aplicativo (https://developer.android.com/guide/components/fundamentals?hl=pt-br#Components)
- `Site` Documentação Android: Ativação de componentes (https://developer.android.com/guide/components/fundamentals?hl=pt-br#ActivatingComponents#Components)
- `Site` Documentação Android: Visão geral do manifesto do aplicativo (https://developer.android.com/guide/topics/manifest/manifest-intro?hl=pt-br)
- `Site` Documentação Android: Visão geral dos recursos do aplicativo (https://developer.android.com/guide/topics/resources/providing-resources?hl=pt-br)
- `Site` Documentação Android: Criar seu primeiro App Android (https://developer.android.com/training/basics/firstapp?hl=pt-br)
- `Site` Documentação Android: Acessar as bibliotecas do Jetpack por tipo (https://developer.android.com/jetpack/androidx/explorer)

Conteúdos Alura:

- `Artigo` Criando o primeiro app Android (https://www.alura.com.br/artigos/criando-o-primeiro-app-android)
- `Artigo` Activity Lifecycle: por que você deve conhecer sobre o ciclo de vida da Activity (https://www.alura.com.br/artigos/activity-lifecycle-por-que-conhecer-ciclo-de-vida-activity)
- `Curso` Android com Kotlin criando um app (https://cursos.alura.com.br/course/fundamentos-android-kotlin)

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=QJjY2TNyI-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)

Android - Sistema de View:

- A classe View representa o bloco básico de construção dos componentes de interface do usuário em Android.
- Conhecer a referência View e ViewGroup para definir um Layout
- Saber as principais ViewGroups para construir layouts - LinearLayout, RelativeLayout, FrameLayout e ConstraintLayout
- Usar o editor visual de Layout para personalizar a tela, ou então, utilizar o arquivo XML para a edição
- Conhecer o Preview do editor visual e usá-lo para preparar as telas e componentes do App
- Utilizar as Views do SDK do Android - TextView, Button, EditText, CheckBox, DatePicker etc
- Criar Views personalizadas para atender demandas específicas
- Personalizar as Views com propriedades do Android ou da própria View - Ajustes de altura, largura, visibilidade, espaçamento
- Implementar Layouts de conteúdos dinâmico com adaptadores - AdapterView e RecyclerView
- Integrar código de negócio com o layout da tela - Chamar e manipular as Views com o View Binding
- Reagir a eventos das Views a partir de listeners como cliques, cliques longos, rolagem, arrastar e soltar

Conteúdos

- `Site` Documentação Android: Layouts (https://developer.android.com/guide/topics/ui/declaring-layout?hl=pt-br)
- `Site` Documentação Android: Layouts (ViewGroups) comuns (https://developer.android.com/guide/topics/ui/declaring-layout?hl=pt-br#CommonLayouts)
- `Site` Documentação Android: Criar uma IU com o Layout Editor (https://developer.android.com/studio/write/layout-editor?hl=pt-br)
- `Site` Documentação Android: Criar uma IU responsiva com o ConstraintLayout (https://developer.android.com/training/constraint-layout?hl=pt-br)
- `Artigo` Android View: Prólogo (https://medium.com/android-dev-br/android-view-pr%C3%B3logo-d44a89c14fd1)
- `Artigo` Listas com RecyclerView (https://medium.com/android-dev-br/listas-com-recyclerview-d3f41e0d653c)
- `Site` Documentação Android: Visão geral dos eventos de entrada (https://developer.android.com/guide/topics/ui/ui-events?hl=pt-br)
- `YouTube` Vinícius Thiengo: View - Entendendo os Componentes Visuais no Android (https://www.youtube.com/watch?v=2b18zJmkKKU)

Conteúdos Alura:

- `Artigo` View Binding Android (https://www.alura.com.br/artigos/view-binding-android)
- `Curso` Android com Kotlin: criando um app (https://cursos.alura.com.br/course/fundamentos-android-kotlin)
- `Curso` Android com Kotlin: personalize o seu App (https://cursos.alura.com.br/course/android-kotlin-personalize-app)
- `Curso` RecyclerView: listas flexíveis e performáticas (https://www.alura.com.br/curso-online-recyclerview-listas-flexiveis-e-performaticas)

- `Curso` RecyclerView: Listeners, animações e boas práticas (https://www.alura.com.br/curso-online-recyclerview-listeners-animacoes)
- `Curso` RecyclerView: criando telas com Constraint Layouts (https://www.alura.com.br/curso-online-layout-android-1)

Android - Fragments:

- Um Fragment representa o comportamento ou uma parte da interface do usuário em uma Activity hospedeira. É possível combinar vários fragmentos em uma única Activity para criar uma IU de vários painéis e reutilizar um fragmento em diversas atividades. Você pode imaginar um fragment como uma seção modular de uma Activity, que tem o próprio ciclo de vida, recebe os próprios eventos de entrada e que pode ser adicionada ou removida durante a execução da Activity.
- Entender o que é um Fragment
- Como usar e reutilizar Fragments na mesma Activity
- Entender os motivos para utilizar Fragments em projetos Android
- Implementar layouts com múltiplos painéis
- Migrar projetos Android que utilizam apenas Activities para utilizar fragments
- Lidar com transações do gerenciador de fragments

Conteúdos

- `Site` Documentação Android: Fragmentos (https://developer.android.com/guide/components/fragments?hl=pt-br)
- `Site` Documentação Android: Ciclo de Vida dos Fragments (inglês) (https://developer.android.com/guide/fragments/lifecycle?hl=pt-br)
- `Site` Documentação Android: Filosofia de projeto com Fragments (https://developer.android.com/guide/components/fragments?hl=pt-br#Design)
- `Artigo` Aprenda a Usar os Fragments em 4 Passos (https://www.androidpro.com.br/blog/desenvolvimento-android/fragments/)
- `YouTube` Pedro Henrique de Oliveira e Silva: Android - Utilizando Fragments (https://www.youtube.com/watch?v=Lo2qofVfA-I&ab_channel=PedroHenriquedeOliveiraeSilva)

Conteúdos Alura:

- `Curso` Android Fragments: Reutilizando componentes visuais (https://cursos.alura.com.br/course/android-fragments)

Android - Navegação no App:

- A navegação de telas em Apps é fundamental para a experiência de uso de Apps Android. Para isso é essencial conhecer os princípios de navegação do Android.
- Aprender o que é a biblioteca Navigation e como é possível implementá-la, seja para o sistema de Views ou para o Jetpack Compose
- Integrar a navegação com componentes do App, seja em navegação de abas, ou na visualização de componentes visuais dependendo da tela
- Saber o que é uma task de back stack do Android
- Conhecer os App links - Deep Link, Web Links e Android App Links

Conteúdos

- `Site` Documentação Android: Princípios de navegação (https://developer.android.com/guide/navigation/navigation-principles?hl=pt-br)

- `Site` Documentação Android: Navegação (https://developer.android.com/guide/navigation?hl=pt-br)
- `Site` Documentação Android: Como navegar com o Compose (https://developer.android.com/jetpack/compose/navigation?hl=pt-br)
- `Site` Documentação Android: Entenda as tarefas e a pilha de retorno (https://developer.android.com/guide/components/activities/tasks-and-back-stack?hl=pt-br)
- `Artigo` Usando Navigation Component — Android Jetpack (https://medium.com/android-dev-br/usando-navigation-component-android-jetpack-e1356921b31d)
- `Artigo` Introdução ao Compose Navigation (https://nglauber.medium.com/introdu%C3%A7%C3%A3o-ao-compose-navigation-574d1d9572a0)
- `YouTube` Stack Mobile: Android Para Iniciantes - Navegação Entre Telas (https://www.youtube.com/watch?v=629qVxiWonM)
- `YouTube` CodandoTV: Aprenda a usar o Navigation do Compose! (https://youtu.be/aZRS-pfbyZU?si=tejc83Raer4wsWvy)

Conteúdos Alura:

- `Curso` Navigation: transição de telas no Android (https://www.alura.com.br/curso-online-android-navigation)
- `Curso` Navigation: novas features e reutilização de código (https://www.alura.com.br/curso-online-android-navigation-features)
- `Curso` Jetpack Compose: navegando entre telas com o Navigation (https://www.alura.com.br/curso-online-jetpack-compose-navegando-telas-navigation)

Jetpack Compose - Fundamentos:

- O Jetpack Compose é uma ferramenta que trás a proposta de criar interfaces nativas Android com menos código, mais rápido e deixa seus apps mais bonitos, ele faz isso através da abordagem declarativa.
- Criar um app Android do zero utilizando o Jetpack Compose
- Construção de Layouts apartir de composables
- Pré-visualizações de Composables
- Gerenciamento de estados, eventos, composição e recomposição
- Configurar o Compose em um projeto já existente e aplicar a interoperabilidade
- Trabalhar com formulários
- Entender a diferença do XML para o Compose
- Manter estados utilizando o padrão MVVM com ViewModel e StateFlow, entendendo o ciclo de vida

Conteúdos

- `Site` Documentação Android: Tutorial do Android Compose (https://developer.android.com/jetpack/compose/tutorial?hl=pt-br)
- `Artigo` Novas pessoas desenvolvedoras android deveriam aprender Compose ou XML? (inglês) (https://dev.to/aldok/should-new-android-developer-learn-compose-or-xml-3oah)
- `YouTube` Utilizando o JETPACK COMPOSE para criar UIs modernas para Apps Android (https://www.youtube.com/watch?v=6qIjSVVzIt0&ab_channel=InstitutodeInform%C3%A1ticaUFG)

- `Site` Documentação Android: Estado e Jetpack Compose (https://developer.android.com/jetpack/compose/state?hl=pt-br)
- `Site` Documentação Android: Ciclo de vida dos composables (https://developer.android.com/jetpack/compose/lifecycle?hl=pt-br#:~:text=Quando%20o%20Jetpack%20Compose%20executa,Jetpack%20Compose%20programar%C3%A
- `YouTube` CodandoTV: Link da playList com vários vídeos ensinando como fazer telas/implementações/animações e muito mais (https://www.youtube.com/playlist?list=PL-7tME9TKyA79CO-soQsOZ1q8itjTLCcg)
- `YouTube` Alex Felipe - Playlist: Aprednendo Jetpack Compose (https://www.youtube.com/watch?v=yWWOuxeqxwE&list=PLYaQbIm_3oZ9rHMFnm7si0LTqEDcYtG2t&pp=iAQB)

Conteúdos Alura:

- `YouTube` O que é Jetpack Compose? com Alex Felipe | #HipstersPontoTube (https://youtu.be/LR5LUhTZPCE)
- `YouTube` Jetpack Compose: Começando com Compose | #AluraMais (https://youtu.be/L4nWoZTKjKw)
- `Artigo` Vale a pena aprender Jetpack Compose agora? (https://www.alura.com.br/artigos/vale-a-pena-aprender-jetpack-compose)
- `Curso` Formação: Jetpack Compose - Criando telas e gerenciando estados (https://cursos.alura.com.br/formacao-jetpack-compose-criando-telas-gerenciando-estados)

Android - Persistência:

- 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.
- No Android podemos armazenar de maneira persistente arquivos específicos ou para o App ou compartilhado, informações de tipo primitivo com preferências e dados estruturados com banco de dados.

Conteúdos

- `Site` Documentação Android: Visão geral do armazenamento de dados e arquivos (https://developer.android.com/training/data-storage?hl=pt-br)
- `Site` Documentação Android: Acessar arquivos específicos do app (https://developer.android.com/training/data-storage/app-specific?hl=pt-br)
- `Site` Documentação Android: Visão geral do armazenamento compartilhado (https://developer.android.com/training/data-storage/shared?hl=pt-br)
- `Site` Documentação Android: Salvar dados de chave-valor (https://developer.android.com/training/data-storage/shared-preferences?hl=pt-br)
- `Site` Documentação Android: Salvar dados em um banco de dados local usando o Room (https://developer.android.com/training/data-storage/room?hl=pt-br)
- `YouTube` Kaique Ocanha - Aulas de Android: LIVE #010 - Room Database - Introdução a Persistência de Dados com Kotlin localmente (https://www.youtube.com/watch?v=JQ_isBkRWzc)
- `YouTube` Alex Felipe - Aprenda a salvar dados no banco de dados do Android | Room | Coroutines | Flow | ViewModel | Koin (https://youtu.be/34P8CncWQiI?si=VGfkmV8tWsKtTFfZ)

Conteúdos Alura:

- `Artigo` Salvando informações com o Shared Preferences (https://www.alura.com.br/artigos/salvando-informacoes-com-o-shared-preferences)

- `Curso` Android com Kotlin: persistência de dados com o Room (https://www.alura.com.br/curso-online-android-kotlin-persistencia-dados-room)
- `Curso` Android com Kotlin: Migrations e relacionamento com o Room (https://www.alura.com.br/curso-online-android-kotlin-migrations-relacionamento-room)
- `Curso` Jetpack Compose: armazenamento de dados internos (https://www.alura.com.br/curso-online-jetpack-compose-armazenamento-dados-internos)

Android - Gradle:

- O Gradle e o plug-in do Android para Gradle(AGP) oferecem uma maneira flexível de compilar, criar, gerenciar e empacotar seu app ou biblioteca Android.
- O que é uma build tool
- Aprender a estrutura de um projeto Android como um projeto multi módulo do Gradle
- Para que serve o Gradle e como usá-lo
- O que são dependências e como utilizá-las

Conteúdos

- `Site` Documentação Android: Configure seu Build (https://developer.android.com/studio/build?hl=pt-br)
- `Site` Documentação Android: Dicas e receitas para o Gradle (https://developer.android.com/studio/build/gradle-tips?hl=pt-br)
- `Site` Documentação Android: Adicionando dependências do build - Gradle (https://developer.android.com/studio/build/dependencies?hl=pt-br)
- `Site` Documentação Android: Guia para Modularização de app (https://developer.android.com/topic/modularization?hl=pt-br)
- `Site` AndroidPro: Gradle para Android - Entendendo o processo de build (https://www.androidpro.com.br/blog/android-studio/gradle/)
- `Artigo` Gradle Android para Iniciantes (https://www.linkedin.com/pulse/gradle-android-para-iniciantes-roger-silva/?originalSubdomain=pt)
- `Site` Slides sobre Desenvolvimento Android - by Moro (https://github.com/gabrielbmoro/slides-about-android-development/blob/main/README_pt-br.md#gradle-)

Conteúdos Alura:

- `YouTube` Conhecendo as build tools no mundo Java (https://www.youtube.com/watch?v=6llRWbdnmuk)
- `YouTube` Começando com o Gradle: Tasks e comandos básicos (https://www.youtube.com/watch?v=uX6Ezf73OEY)

Android - Carregamento de imagens:

- As imagens vem em tamanhos e formas diferentes. Na maioria dos casos, elas são maiores do que o necessário para a Interface de usuário (UI) de um app. Dado que estamos trabalhando com memória limitada, no Android é usada uma técnica de decodificação de bitmaps, para evitar todo o trabalho de utilizar isso, temos diversas bibliotecas que facilitam esse processo.
- O que é Bitmap
- Conhecer as bibliotecas de carregamento de imagens (Glide, Picasso, Coil, entre oturas) e usá-las
- Vantagens e desvantagens das bibliotecas de carregamento de imagens

Conteúdos

- `Artigo` Bitmap: o que é, para que serve e como funciona (https://www.tecnoveste.com.br/bitmap-o-que-e-para-que-serve-e-como-funciona/)
- `YouTube` Paulo Salvatore: Carregando imagens no Android com Kotlin | Glide (https://www.youtube.com/watch?v=5FJLyFbTsZQ&ab_channel=PauloSalvatore)
- `YouTube` Stevdza-San: Coil - Biblioteca moderna de carregamento de imagem | Android Studio Tutorial (inglês) (https://www.youtube.com/watch?v=IBaUjzn2Rgo)
- `YouTube` cedr!c: Como carregar uma imagem remota em Android - PICASSO & GLIDE (https://www.youtube.com/watch?v=BM6SnH3CDE0)
- `Site` Coil - Site oficial (inglês) (https://coil-kt.github.io/coil/)
- `Site` Glide v4 - Site oficial (inglês) (https://bumptech.github.io/glide/)
- `Site` Picasso - Site oficial (inglês) (https://square.github.io/picasso/)
- `YouTube` CodandoTV: Carregar Imagens em uma lista usando Compose+Coil [StreamPlayerApp] (https://youtu.be/9ga19aK3VGU?si=0IcMIf-Ri-Y_yTVi)

Conteúdos Alura:

- `Curso` Android com Kotlin: Personalize o seu app (https://cursos.alura.com.br/course/android-kotlin-personalize-app)
- `Curso` Jetpack Compose: Criando um app android (https://cursos.alura.com.br/course/jetpack-compose-app-android)

Nivel 2

Android - Permissões:

- As permissões do app ajudam a apoiar a privacidade do usuário protegendo dados restritos, como estados do sistema e os dados de contatos dos usuários, há também as ações restritas, como a conexão a um dispositivo pareado e a gravação de áudio ou uso de câmera.
- Entender o que são permissões
- Aprender sobre os diferentes tipos de permissões
- Permissões especiais
- Práticas recomendadas

Conteúdos

- `Site` Documentação Android: Permissões no Android (https://developer.android.com/guide/topics/permissions/overview?hl=pt-br)
- `Site` Documentação Android: Uso de permissões - Android Manifest (https://developer.android.com/guide/topics/manifest/uses-permission-element?hl=pt-br)
- `Site` Documentação Android: Solicitando permissões do app (https://developer.android.com/training/permissions/requesting?hl=pt-br)
- `Artigo` Entendendo as permissões do app (inglês) (https://guides.codepath.com/android/Understanding-App-Permissions)
- `YouTube` Tiago Aguiar: Permissões Android - Tutorial Como Usar (https://www.youtube.com/watch?v=Grr5N44Ha64)

Kotlin - 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.

- Aprender as possibilidades para rodar o código de maneira assíncrona no Android
- Conhecer o pacote java.concurrent e as suas soluções
- Utilizar Coroutines como uma solução de código assíncrono no Kotlin
- Entender e usar estruturas reativas como LiveData, Flow, StateFlow, RX, etc

Conteúdos

- `Site` Documentação Android: Corrotinas do Kotlin no Android (https://developer.android.com/kotlin/coroutines?hl=pt-br)
- `Site` Documentação Android: Fluxos em Kotlin no Android (https://developer.android.com/kotlin/flow?hl=pt-br)
- `Site` Documentação Android: StateFlow e SharedFlow (https://developer.android.com/kotlin/flow/stateflow-and-sharedflow?hl=pt-br)
- `Site` Documentação Android: Visão geral do LiveData (https://developer.android.com/topic/libraries/architecture/livedata?hl=pt-br)
- `Artigo` Introdução ao RX Java com Kotlin (https://nglauber.medium.com/introdu%C3%A7%C3%A3o-ao-rx-java-com-kotlin-90c58d184c6b)
- `YouTube` Kotlin Coroutines: Criando um Projeto Android do Zero Com a Arquitetura MVVM e LiveData (https://www.youtube.com/watch?v=M9nmW9BSKUA)
- `YouTube` CodandoTV(Rods): Configurar o Networking na sua aplicação Retrofit+Okhttp+Interceptores+Coroutines (https://youtu.be/D-wckaYH5Do)

Conteúdos Alura:

- `Site` Operações Assíncronas com Coroutines no Kotlin (https://www.alura.com.br/videos/operacoes-assincronas-com-coroutines-do-kotlin-no-android-c58)
- `Artigo` Retrofit com Coroutines e LiveData no Android (https://www.alura.com.br/artigos/retrofit-com-coroutines-e-livedata-no-android)

Kotlin - 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.
- Realizar requisições HTTP para se comunicar com serviços online, como REST API é fundamental na maioria dos Apps Android. Sendo assim, é importante conhecer as principais ferramentas e técnicas necessárias para esse tipo de funcionalidade.
- Conhecer bibliotecas famosas no mundo Android para realizar requisições - Retrofit, Ktor ou Volley
- Configurar as requisições para executararem de maneira assíncrona
- Converter objetos para JSON e vice-versa

Conteúdos

- `Site` Documentação Android: Conectar-se à rede (https://developer.android.com/training/basics/network-ops/connecting?hl=pt-br)

- `Artigo` Comunicação HTTP eficiente no Android com Volley (https://nglauber.medium.com/comunica%C3%A7%C3%A3o-http-eficiente-no-android-com-volley-9772b4057a5c)
- `Artigo` Consumindo API REST no Android com Retrofit em Kotlin (https://medium.com/collabcode/consumindo-api-rest-no-android-com-retrofit-em-kotlin-parte-1-5e752ab8a877)
- `YouTube` Coder: Como fazer chamadas API usando a biblioteca Android Volley | Kotlin (https://www.youtube.com/watch?v=pfqc7ajaLNE)
- `YouTube` Tiago Aguiar: Retrofit - O Guia Definitivo para Cliente HTTP no Android (https://www.youtube.com/watch?v=NPmzpvEQn6I)

Conteúdos Alura:

- `Artigo` Retrofit com Coroutines e LiveData no Android (https://www.alura.com.br/artigos/retrofit-com-coroutines-e-livedata-no-android)
- `Curso` Android com Kotlin: comunicação com Web API (https://www.alura.com.br/curso-online-android-kotlin-comunicacao-web-api)

Android - Recursos do sistema:

- Os smartphones que estão sob a plataforma Android, na sua grande maioria possuem diversos recursos específicos do sistema, tais como câmeras, sensores, gps, entre outros.
- Aprenda quais são e como utilizar esses recursos

Conteúdos

- `Site` Documentação Android: Visão geral do CameraX - Biblioteca para utilizar a camêra no Android (https://developer.android.com/training/camerax?hl=pt-br)
- `Site` Documentação Android: Sensores - Sensores de Movimento, Posição e Ambiente (https://developer.android.com/guide/topics/sensors?hl=pt-br)
- `Site` Documentação Android: Visão Geral da Conectividade (Bluetooth, Wi-Fi, operações de rede) (https://developer.android.com/guide/topics/connectivity?hl=pt-br)

Conteúdos Alura:

- `YouTube` Alura: Utilizando o Google Maps em Apps Android (https://www.youtube.com/watch?v=rpqi7Y1NQZY)
- `YouTube` Alura: Tirando foto no app com a câmera do Android (https://www.youtube.com/watch?v=jLcrcTMoB88)

Android (IA):

- Um Client-SDK para IA é uma ferramenta intermediária que simplifica a comunicação entre o seu aplicativo e serviços de IA remotos. Imagine como uma ponte que liga o seu aplicativo (que está local no seu dispositivo) a um servidor poderoso de IA (que geralmente fica na nuvem);
- IA On-Device, ou inteligência artificial no dispositivo, refere-se à execução de tarefas de IA diretamente no seu dispositivo pessoal, como smartphone, tablet ou smartwatch, em vez de depender de servidores remotos na nuvem;
- Aprender a traduzir textos automaticamente, reconhecer imagens, gerar respostas inteligentes e mais utilizando IA no seu app;
- Como acessar a API Gemini/GPT diretamente do app Android usando o SDK do cliente para Android;
- Como baixar e utilizar ferramentas de machine learning localmente nos apps;

Conteúdos

- `Site` Documentação Google AI: Utilizar Gemini API (https://ai.google.dev/tutorials/get_started_android?hl=pt-br)
- `Site` Documentação ML Kit (https://developers.google.com/ml-kit?hl=pt-br)
- `Site` Documentação do TensorFlow Lite (https://www.tensorflow.org/lite?hl=pt-br)
- `Site` Documentação do Gemini Nano SDK (https://ai.google.dev/tutorials/android_aicore)
- `YouTube` TensorFlow - Modelos de linguagem no dispositivo com Keras e Android (https://youtu.be/pNWNMPi0Mvk)

Conteúdos Alura:

- `Alura+` Alura+: Crie seu primeiro aplicativo de IA generativa no Android Studio (https://cursos.alura.com.br/extra/alura-mais/crie-seu-primeiro-aplicativo-de-ia-generativa-no-android-studio-c9218)
- `Artigo` Gemini no Android Studio: a Github Copilot da Google (https://www.alura.com.br/artigos/gemini-no-android-studio)
- `Curso` Formação - Android com IA: criando apps mais inteligentes com o Google ML Kit (https://cursos.alura.com.br/formacao-android-ia-google-ml-kit)
- `Curso` Formação Android: criando apps de forma inovadora com CoPilot, GPT e Gemini (https://cursos.alura.com.br/formacao-android-ia-apps-inovadores-copilot-gpt-gemini)

Kotlin - Injeção de Dependências:

- Injeção de Dependências é um padrão de projeto no qual uma classe solicita dependências de fontes externas ao invés de criá-las.
- Ao escrever códigos em projetos Android, é muito que uma funcionalidade utilize códigos de bibliotecas, como o Room para salvar dados ou o Retrofit para fazer requisições para REST API. Essas bibliteocas são conhecidas como dependências dos nossos códigos, justamente pela necessidade para realizar a ação esperada. Usar essas bibliotecas com facilidade em qualquer parte do app pode ter os seus desafios, como oferecer instâncias únicas e realizar toda a configuração necessária para o funcionamento correto. Para isso, utilizamos ferramentas de injeção de dependência que facilitam o nosso trabalho.
- Aprender a usar a técnica de injeção de dependências e alguma das ferramentas comuns no Android para tal - Hilt, Dagger ou Koin

Conteúdos

- `Site` Documentação Android: Injeção de dependência no Android (https://developer.android.com/training/dependency-injection?hl=pt-br)
- `Artigo` Injeção de dependência no Kotlin com Koin (https://medium.com/collabcode/inje%C3%A7%C3%A3o-de-depend%C3%AAncia-no-kotlin-com-koin-4d093f80cb63)
- `YouTube` Chico Rasia: Viewmodel com injeção de dependência via Factory no Android com Kotlin (https://youtu.be/biFrt-Bjx-g)
- `YouTube` Douglas Motta: Injeção de Dependência com Koin no Android (https://www.youtube.com/watch?v=Mh_NK5G0WjY)
- `YouTube` CodandoTV(Rods): Configurar Koin de maneira completa (https://youtu.be/7_UEnCZD5DI)

Conteúdos Alura:

- `Artigo` Injeção de dependência no Android com o Hilt (https://www.alura.com.br/artigos/injecao-de-dependencia-do-android-com-o-hilt)

Android - 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.
- Usar testes unitários
- Usar testes de integração
- Usar testes instrumentados
- Usar mocks para facilitar a implementação de testes com dependências
- Implementar testes com ferramentas de Injeção de Dependências como o Hilt

Conteúdos

- `Site` Documentação Android: Conceitos básicos de testes (https://developer.android.com/training/testing/fundamentals?hl=pt-br)
- `Site` Documentação Android: O que testar no Android (Inglês) (https://developer.android.com/training/testing/fundamentals/what-to-test)
- `Site` Documentação Android: Como testar o layout do Compose (https://developer.android.com/jetpack/compose/testing?hl=pt-br)
- `Site` Documentação Android: Criar testes de unidade locais (https://developer.android.com/training/testing/unit-testing/local-unit-tests?hl=pt-br)
- `Artigo` Testes Instrumentados: tipos de teste e como são executados. (https://medium.com/android-dev-br/testes-instrumentados-tipos-de-teste-e-como-s%C3%A3o-executados-1f13a8cfe4c4)
- `YouTube` Dev Vai Longe: 02 Testes unitários no Android com mockito, jUnit e kotlin. (https://youtu.be/iKaPscjFa-4)

Conteúdos Alura:

- `Curso` Android com Kotlin: testes de unidade e Mocks (https://www.alura.com.br/curso-online-android-kotlin-testes-unidade-mocks)
- `Curso` Android com Kotlin: testes instrumentados (https://www.alura.com.br/curso-online-android-kotlin-testes-instrumentados)

Nivel 3

Android - Arquitetura:

- O guia de arquitetura Android aborda práticas e a arquitetura recomendada para a criação de apps robustos com alta qualidade de produção. Organizar sua base de código em partes contidas e acopladas com flexibilidade (Modularização).
- Aprender o que é e para que serve a arquitetura MVVM (Model-View-ViewModel)
- Camadas de IU, dados e domínios
- Fluxo de dados dentro do app
- Como utilizar o ViewModel
- Recomendações da arquitetura, padrão repositório, offline-first

- Gerenciamento de estado
- Injeção de dependências
- Melhorar a experiência dos usuários do app com o ViewModel
- Aprender o que é modularização
- Benefícios da modularização

Conteúdos

- `Site` Documentação Android: Guia para a arquitetura do app (https://developer.android.com/topic/architecture?hl=pt-br)
- `Site` Documentação Android: Visão geral do ViewModel (https://developer.android.com/topic/libraries/architecture/viewmodel?hl=pt-br)
- `Site` Documentação Android: Camada de IU (https://developer.android.com/topic/architecture/ui-layer?hl=pt-br)
- `Site` Documentação Android: Camada de Domínio (https://developer.android.com/topic/architecture/domain-layer?hl=pt-br)
- `Site` Documentação Android: Camada de dados (https://developer.android.com/topic/architecture/data-layer?hl=pt-br)
- `Site` Documentação Android: Camada de dados - offline first (https://developer.android.com/topic/architecture/data-layer/offline-first?hl=pt-br)
- `Site` Resumo da série 'MAD Skills' de arquitetura (inglês) (https://android-developers.googleblog.com/2022/04/architecture-mad-skills-series-wrap-up.html)
- `YouTube` Android pra valer: Como usar o ViewModel + LiveData no Android (https://www.youtube.com/watch?v=pIlcZqdrYUM&ab_channel=Androidpravaler)
- `Site` Documentação Android: Guia para a modularização de apps Android (https://developer.android.com/topic/modularization?hl=pt-br)
- `Site` Documentação Android: Recomendações para arquiteturas do Android (https://developer.android.com/topic/architecture/recommendations?hl=pt-br)
- `YouTube` CodandoTV(Rods) - Simplificando o Clean Architecture +MVVM na sua aplicação mobile - Guia Completo (https://youtu.be/8ehlZfyN1S0?si=Py9a7JbY-6XJj1BX)

Conteúdos Alura:

- `Curso` Jetpack Compose: mantendo estados com ViewModel (https://cursos.alura.com.br/course/jetpack-compose-estados-viewmodel)
- `Curso` Architecture Components: ViewModel, LiveData e Room (https://cursos.alura.com.br/course/android-archtecture-components)

Android - Critérios de qualidade do App:

- Ao escrever um App Android, existe uma série de requisitos para garantir uma qualidade esperada, como por exemplo, a compatibilidade do App com a navegação esperada pelo sistema Android, ou então, pelos gestos, etc.
- Os requisitos são classificados como - Experiência visual, Funcionalidade, Desempenho e estabilidade, Privacidade e segurança, Google Play e Procedimento de teste
- Cumprir todos os requisitos, ou quase todos, significa que o App possui uma maior qualidade para os usuários.

Conteúdos

- `Site` Documentação Android: Principais critérios de qualidade do app (https://developer.android.com/docs/quality-guidelines/core-app-quality?hl=pt-br)

- `Site` Documentação Android: Deliver high performing user experiences (inglês) (https://developer.android.com/quality)
- `Site` Documentação Android: Primeiros passos com telas grandes (https://developer.android.com/guide/topics/ui/responsive-layout-overview?hl=pt-br)
- `Site` Documentação Android: Criar para bilhões (https://developer.android.com/docs/quality-guidelines/build-for-billions?hl=pt-br)

Android - 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.
- Ao implementar novas funcionalidades do App, precisamos garantir que todas as entregas irão funcionar corretamente. Para isso, podemos utilizar técnicas de integração e entrega contínua, dessa forma, agilizamos a evolução do App e tentamos garantir os comportamentos esperados ao mesmo tempo.
- Conhecer alguma das ferramentas para fazer a entrega contínua, como Firebase Test Lab, Jenkins, GitHub Actions, etc

Conteúdos

- `Site` Comece a testar com sistemas de integração contínua (CI) (https://firebase.google.com/docs/test-lab/android/continuous?hl=pt-br)
- `Artigo` Continuous Deployment no Android: como usar a Publishing API para automatizar seu release (https://medium.com/android-dev-br/continuous-deployment-no-android-f42b96ece80d)
- `Artigo` Entrega Contínua Na Visão De Um Desenvolvedor Android (https://www.thiengo.com.br/entrega-continua-na-visao-de-um-desenvolvedor-android)
- `YouTube` Philipp Lackner: Aprenda a automatizar tarefas para Android com CI/CD (inglês) (https://www.youtube.com/watch?v=QLsgkxH-O2I)

Android - Otimização do app:

- Ao gerar um App, existem alguns detalhes de otimização para torná-lo rápido, menor e otimizado, como por exemplo, a remoção de código desnecessário, a ofuscação que reduz o nome dos códigos e a otimização que aplica estratégias mais agressivas para reduzir mais ainda o app.
- Aprender a ativar cada otimização no plugin do Android para o Gradle
- Utilizar o proguard para realizar uma otimização mais agressiva
- Ativar o multidex para que o app seja capaz de obter mais de 64000 métodos e conseguir utilizar as técnicas de otimização
- Conhecer e utilizar ferramentas auxiliarem para aumentar o desempenho do App

Conteúdos

- `Site` Documentação Android: Reduzir, ofuscar e otimizar o app (https://developer.android.com/studio/build/shrink-code?hl=pt-br)
- `Site` Documentação Android: Ativar multidex para apps com mais de 64.000 métodos (https://developer.android.com/studio/build/multidex?hl=pt-br)
- `Site` Documentação Android: Guia de desempenho do app (https://developer.android.com/topic/performance?hl=pt-br)

- YouTube Philipp Lackner: Reduzir, otimizar and tornar seu app mais seguro com R8 & ProGuard (inglês) (https://www.youtube.com/watch?v=bgpyuuzMlo0)

Java - Fundamentos:

- Java é uma linguagem de programação amplamente usada para codificar aplicações Web. Java é uma linguagem multiplataforma, orientada a objetos e centrada em rede que pode ser usada como uma plataforma em si. É uma linguagem de programação rápida, segura e confiável para codificar tudo, desde aplicações móveis e software empresarial até aplicações de big data e tecnologias do servidor.
- Conhecer os tipos primitivos
- Declarar variáveis, considerando os diferentes tipos
- Usar estruturas condicionais ('if', 'else')
- Conhecer os operadores de atribuição e comparação
- Usar estruturas de repetição e laços ('while', 'for')
- Usar funções, passando parâmetros e argumentos
- Manipular métodos
- Manipular arrays e listas
- Obter dados de uma API
- Criar construtores

Conteúdos

- Site O que é Java? (https://aws.amazon.com/pt/what-is/java/)
- Artigo Aprenda Comigo: Java — Parte 1 (https://medium.com/clebertech/aprenda-comigo-java-parte-1-42ef8fddd8b6)
- YouTube FernandaDev: Java (Aula 1 - Classes, Objetos, Atributos e Métodos) (https://www.youtube.com/watch?v=ohmHbdUhAGc)
- YouTube Loiane Groner: Primeiro Programa em Java (https://youtu.be/mu2ti43cgwc)

Conteúdos Alura:

- Artigo Java: o que é, linguagem e um Guia para iniciar na tecnologia (https://www.alura.com.br/artigos/java)
- Podcast Hipster 313 - Ecossistema Java revisitado (https://www.hipsters.tech/ecossistema-java-revisitado-hipsters-ponto-tech-313/)
- Artigo Como começar a desenvolver em Java? (https://www.alura.com.br/artigos/comecando-com-o-desenvolvimento-java)
- Artigo Meu primeiro programa em Java (https://www.alura.com.br/artigos/meu-primeiro-programa-em-java)
- Artigo Java: Conheça o método main (https://www.alura.com.br/artigos/metodo-main-em-java)
- Artigo Desenvolvendo aplicações Java com o VS Code (https://www.alura.com.br/artigos/desenvolvendo-aplicacoes-java-vs-code)
- Artigo Diferença entre int e Integer em Java (https://www.alura.com.br/artigos/diferenca-entre-int-e-integer-em-java)
- Artigo Como converter String para Date em Java (https://www.alura.com.br/artigos/como-converter-string-para-date-em-java)
- Artigo Iterando uma lista em Java (https://www.alura.com.br/artigos/iterando-uma-lista-em-java)

- `Artigo` Importando classes no Java (https://www.alura.com.br/artigos/importando-classes-em-java)
- `Site` Apostila: Java e Orientação a Objetos (https://www.alura.com.br/apostila-java-orientacao-objetos)
- `Site` Apostila: Java para Desenvolvimento Web (https://www.alura.com.br/apostila-java-web)
- `YouTube` Alura: O que é o Java? (https://www.youtube.com/watch?v=90NcVNsKGik)
- `Curso` Formação Java e Orientação a Objetos (https://cursos.alura.com.br/formacao-java)
- `Desafio` 7 Days of Code: Java (https://7daysofcode.io/matricula/java)

Android - Depuração:

- Durante o desenvolvimento de um app, é bastante comum a presença de bugs inesperados, como também, a falta de compreensão do motivo do bug. Para agilizar a análise e investigação de problemas ou comportamentos inesperados no app, precisamos aprender a depurar ou debuggar um app Android.
- Saber como executar um app em depuração no Android Studio
- Aprender a ativar a depuração em dispositivos físicos
- Utilizar Logs para identificar eventos
- Analisar stack track
- Inspecionar o Layout, recursos do sistema como processador, memória e rede
- Depurar o banco de dados do app e arquivos APKs pré-compilados
- Analisar o build com o analizador de APK

Conteúdos

- `Site` Documentação Android: Depurar seu app (https://developer.android.com/studio/debug?hl=pt-br)
- `Site` Documentação Android: Configurar opções do desenvolvedor no dispositivo (https://developer.android.com/studio/debug/dev-options?hl=pt-br)
- `Site` Documentação Android: Escrever e visualizar registros com o Logcat (https://developer.android.com/studio/debug/am-logcat?hl=pt-br)
- `Site` Documentação Android: Analisar um stack trace (https://developer.android.com/studio/debug/stacktraces?hl=pt-br)
- `Site` Documentação Android: Depurar o layout com o Layout Inspector e a Validação de layout (https://developer.android.com/studio/debug/layout-inspector?hl=pt-br)
- `Site` Documentação Android: Inspecionar o tráfego de rede com o Network Inspector (https://developer.android.com/studio/debug/network-profiler?hl=pt-br)
- `Site` Documentação Android: Depurar seu banco de dados com o Database Inspector (https://developer.android.com/studio/inspect/database?hl=pt-br)
- `Site` Documentação Android: Depurar APKs pré-compilados (https://developer.android.com/studio/debug/apk-debugger?hl=pt-br)
- `Site` Documentação Android: Capturar e ler relatórios de bugs (https://developer.android.com/studio/debug/bug-report?hl=pt-br)
- `Site` Documentação Android: Analisar seu build com o APK Analyzer (https://developer.android.com/studio/debug/apk-analyzer?hl=pt-br)
- `YouTube` Pedro Carvalho: Iniciando com Debug no Android Studio (https://www.youtube.com/watch?v=NoHl04lifOE?hl=pt-br)

Conteúdos Alura:

- `Artigo` O que é logcat e como visualizar logs no Android (https://www.alura.com.br/artigos/o-que-e-logcat-como-visualizar-logs-android)

Android - Segurança e Monitoramento:

- À medida que seu projeto cresce, ele se torna mais visível, atraindo tanto um público maior quanto potenciais invasores. A segurança do aplicativo, desde as fases iniciais de desenvolvimento, é crucial para proteger a integridade do projeto e a privacidade dos usuários. Adotar práticas de segurança robustas e se manter atualizado é essencial para evitar ameaças e garantir uma experiência segura para sua empresa e seu usuário
- Atente-se a entender as necessidades do seu projeto
- Monitoramento é fundamental para compreender como seu aplicativo opera em produção.
- Compreender o fluxo do usuário e identificar possíveis funis ajuda a aprimorar a estratégia do seu projeto e pode ser um fator determinante para o sucesso do aplicativo.
- A capacidade de antecipar e prever cenários de problemas antes que afetem o usuário final é um diferencial crucial.

Conteúdos

- `Site` Documentação: Práticas recomendadas para segurança de apps (https://developer.android.com/topic/security/best-practices?hl=pt-br)
- `Site` Documentação: Executar o app de monitoramento (https://developer.android.com/games/sdk/performance-tuner/unity/run-monitor-app?hl=pt-br)
- `Site` Github(Rods) - Mobile developer security roadmap (https://github.com/rviannaoliveira/mobile-developer-security-roadmap)
- `YouTube` CodandoTV(Rods) - Ofuscação do seu app - Segurança #02 (https://youtu.be/yClh_VTadfY?si=dol_zASzwBCAkflv)
- `YouTube` CodandoTV(Rods) - Malware do seu app - Segurança #03 (https://youtu.be/Qln4S8lYNXc?si=xUYlyrHSvjrHr0pl)
- `YouTube` CodandoTV(Rods) - Autenticação do seu app - Segurança #04 (https://youtu.be/gBAl6okcRVQ?si=NzzCiY2saO52Wiur)
- `YouTube` CodandoTV(Rods) - Análise de código do seu app - Segurança #05 (https://youtu.be/ESkhqn45FoM?si=R8LmgAVbINpMj8v1)
- `YouTube` CodandoTV(Rods) - Comunicação de Tráfego do seu app - Segurança #06 (https://youtu.be/lrCpk7Vckb0?si=4dB78NFRB6lnguGG)
- `Site` Slides sobre Desenvolvimento Android - by Moro (https://github.com/gabrielbmoro/slides-about-android-development/blob/main/README_pt-br.md#seguran%C3%A7a-)

Android - Implantação (Deployment):

- O deployment ou implantação é o processo de disponibilizar um aplicativo Android para os usuários. Existem várias maneiras de fazer isso, um deles é o uso da Play Store. Ela permite que os desenvolvedores distribuam seus aplicativos para um público global.
- Conhecer os conceitos básicos de deployment de aplicativos Android;
- Criar uma conta de Google Developer;
- Usar a Play Store para distribuir aplicativos Android para um público global.

Conteúdos

- `Site` Documentação Android: Publicar o app (https://developer.android.com/studio/publish?hl=pt-br)
- `Artigo` O que é necessário para publicar seu aplicativo no Google Play e na App Store (https://dev.to/altencirsilvajr/o-que-e-necessario-para-publicar-seu-aplicativo-no-google-play-e-na-app-store-em-2023-le6)
- `Artigo` Crie e publique um aplicativo Android na Google Play Store (https://cynoteck.com/pt/blog-post/build-and-publish-an-android-app-on-the-google-play-store/)
- `YouTube` Deivid Rothen: Como Publicar um Aplicativo Android na Play Store (https://www.youtube.com/watch?v=Xk28gXHQ9Bc&ab_channel=DeividRothen)
- `YouTube` Stack Mobile: Como Publicar um Aplicativo Android na Loja da Google Play - Guia Absolutamente Completo (https://www.youtube.com/watch?v=Xk28gXHQ9Bc&ab_channel=DeividRothen)

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 pul 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/MqoqPzjQyCY?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_HRPIwpSEX4A)

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=BAmvmaKQkIQ)
- 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=Z45BTNeZ1l0)
- `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/Lzq3f1DHWcI)
- `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=HiBCv9DoIxI&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=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/8ehIZfyN1S0?si=I7-I5I4zsnYLISjD)

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 Architeture - Hipsters Ponto Tech 254 (https://www.hipsters.tech/clean-architeture-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=EXp0gq9kGxI&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 & Design

Material Design:

- Material Design é um sistema de design de código aberto do Google, onde ele te passa componentes com determinados padrões de uso e certa personalização para seus apps.

- Fundações de experiência de usuário
- Personalizar seus componentes
- Layouts adaptativos

Conteúdos

- `Site` Material 3 - Comece aqui (inglês) (https://m3.material.io/get-started)
- `Site` Fundações do Material 3 (inglês) (https://m3.material.io/foundations)
- `Site` Material 3 - Estilos (inglês) (https://m3.material.io/styles)
- `Site` Material 3 - Componentes (inglês) (https://m3.material.io/components)

Conteúdos Alura:

- `Artigo` Como usar as novas cores dinâmicas no Material Design 3 do Android (https://www.alura.com.br/artigos/usar-novas-cores-dinamicas-material-design-3-android)

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=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)

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-icones-svg-como-fontes-com-o-icomoon-563e)
- YouTube Nadine Fronza: Como escolher a melhor fonte para seu projeto (https://www.youtube.com/watch?v=MSKPUtIdadI)
- 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-incrivel)
- 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=WXfYtneJZzI)
- 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)

Android - Acessibilidade:

- 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.
- Aumentar a visibilidade do texto
- Usar controles grandes e simples
- Descrever cada elemento de IU

Conteúdos

- Site Documentação Android: Tornar os apps mais acessíveis (https://developer.android.com/guide/topics/ui/accessibility/apps?hl=pt-br#text-visibility)

- Site Documentação Android: Princípios para melhorar a acessibilidade do app (https://developer.android.com/guide/topics/ui/accessibility/principles?hl=pt-br)
- Site Documentação Android: Testar a acessibilidade do seu app (https://developer.android.com/guide/topics/ui/accessibility/testing?hl=pt-br)
- YouTube Chico Rasia: View customizada no Android com Kotlin, parte 3: Acessibilidade com Talkback (https://www.youtube.com/watch?v=4nJH3djiVuk&ab_channel=ChicoRasia)
- YouTube CodandoTV: Acessibilidade usando JetPack COMPOSE com Gabriel Moro (https://youtu.be/JtHkmzZDAqo?si=h2qSM-2uCya9icSz)
- Site Slides sobre Desenvolvimento Android - by Moro (https://github.com/gabrielbmoro/slides-about-android-development/blob/main/README_pt-br.md#acessibilidade-%EF%B8%8F)

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/nerdzao/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)

Alura, PM3 e FIAP
O Techguide.sh é um projeto open source