Estratégia de planejamento e execução

Desenvolvendo planos de estudo eficientes

Artigos

Articles

Articles | Swift by Sundell

Repos

GitHub - mattsoccipa/awesome-swift: A collaborative list of awesome Swift libraries and resources. Feel free to contribute!

GitHub - raywenderlich/swift-algorithm-club: Algorithms and data structures in Swift, with explanations!

Ainda não domino | Tenho conhecimento, mas cabe mais aplicação | Me sinto confortável

## iOS Learning Path

Base mínima de conhecimento da plataforma. A ideia aqui é só construir alguma crítica ou visão sobre como sair do chão.

### T0 — Introdução

➡️Consumir todo o conteúdo da formação Swift na Alura rapidamente

Especificação da linguagem em si.

### T1 — Swift (TEORIA)

The Basics - The Swift Programming Language (Swift 5.6)

➡️The Swift Programming Language (Apple Inc. Book)
Pros: Apple Book
Cons: Tempo! Muito snippet, explicação e repetição de tudo que tem na linguagem 2014! talvez não esteja atualizado com as features recentes
⭐ 4.6 | 500 pages | 2014/Jun

➡️Mastering Swift 5 - 5th Edition (Packt - Jon Hoffman)
Pro: Menor (talvez mais focado! mais direto!) de variáveis à padrões de design
Con: Não é da apple!
⭐ 4.5 | 370 pages | 2019/Apr

Conceitos de desenvolvimento móvel na plataforma iOS.

### T2 — iOS Development (TEORIA, PRÁTICA)

➡️iOS Programming The Big Nerd Ranch Guide (Christian Kerr, Aaron Hillegass)
Pros: É base para bootcamp!
Alia prática apresentando features através de projetos guia
Cons: Não é da apple!
⭐ ? | 528 pages | 2020/Apr

➡️Intro to App Development with Swift (Apple Inc. Book)
Pros: Apple book!
Cons: Xcode 10! =(
⭐ 3.7 | 232 pages | 2019/Nov

Conhecimento aprofundado sobre frameworks e SDKs

### T3 — Aprofundamento e Prática (TEORIA, PRÁTICA)

➡️Programming iOS 14: Dive deep into Views, View Controllers and Frameworks (Matt Neuburg)
Pros: Vai bem fundo sobre o funcionamento e detalhes de vários dos frameworks
Cons: Não é da apple!
Aparentemente não foca em SwiftUI
⭐ ? | 1256 pages | 2020/Nov

### Alternativos/Complementares (TEORIA, PRÁTICA DELIBERADA)

CS193p - Developing Apps for iOS

Hacking with iOS: SwiftUI Edition

## Pool de Skills

| Swift | iOS | Infra SDK: APIs e Frameworks | Qualidade | Gestão e Plataforma | Segurança |
|---|---|---|---|---|---|
| Type System, Variáveis, Constantes, Operadores | UIKit | Video, Audio e Biblioteca de Músicas | Testes Unitários | Gerenciamento de Dependências (CocoaPods) | Gestão de Permissões (Apple Guideline) |
| Tipos Opcionais | View Code | Biblioteca de Fotos e Camera | Testes de Integração | Gestão de Certificados e Profiles | Controles de Autenticação e Autorização |
| Coleções | Controles de Navegação | Contatos | Testes e2e | Automação de build e delivery (Fastlane) | 2FA e Tokens de acesso |
| Controle de Fluxo | Auto Layout (possíveis outros sistemas de layout) | Calendar | Análise de Código (SonarQube) | Beta Testing com Testflight | Gestão de Senhas, Chaves e Secrets (iCloud Keychain?) |
| Funções | Animações | Geolocation e Sonsores | Formatação e Styleguide (SwiftLint) | Publicação na AppStore | Retenção Mínima de Dados |
| Classes, Structs e Protocolos | SwiftUI | Social Media SDKs | Design Patterns com Swift | Versionamento e Release Management | Termos de Uso e Política de Privacidade |
| Protocol Extensions | Design System (Voxel) | Mapas | Design de Código | Observability (auto capture) | Patterns e SDKs de Segurança |
| Protocol Oriented Design | Server-Driven UI (Beagle) | Data Fetching HTTP | Arquitetura | Error Tracing (Sentry) | Comunicação Segura com TLS ou VPN |
| Generics | | Deep Linking | Modular Architecture | Logging (Sentry / Rollbar) | SSL Pinning |
| Disponibilidade e Error Handling | | WebView | | Crash Reporting (Crashlytics Firbase) | Obfuscação de Código (IXGuard, Obfuscator) |
| Subscripting | | Comunicação entre App Sandboxes | | Cloud Messaging (Firebase) | |
| Closures | | Eventos de ciclo de vida e Gerenciamento de Estados da App | | Analytics (Firebase) | |
| Concorrência e Paralelismo | | Persistência | | A/B Testing (Firebase) | |
| Tipos Customizados | | In-app Purchases | | Feature Toggle (Firebase Remote Configs) | |
| Gerenciamento de Memória | | Push Notification | | App Store Review Guideline | |

Ter bom conhecimento de Objective-C

## Objetivos de Capacidade

**Ser capaz de desenvolver aplicativos com usabilidade reconhecida através de avaliações de usuário nas lojas acima de 4.5 estrelas**
- Aprender Swift com profundidade
- Ter domínio do iOS SDK e entender objetivos por trás das decisões de design da plataforma
- Ter bons conhecimentos sobre os Guidelines da apple para o desenvolvimento de apps
- Ter domínio sobre ferramentas e práticas de layout e usabilidade (UIKit, SwiftUI)

**Ser capaz de efetuar commits e tê-los aprovados através de Pull Request em projeto relevante para a plataforma iOS ou stack front da Zup/Itaú**
- Aprender Swift com profundidade
- Ter domínio do iOS SDK e entender objetivos por trás das decisões de design da plataforma
- Conhecer interoperabilidade com Objective-C e heranças do modelo de desenvolvimento anterior
- Estudar projetos open-source
- Estudar e começar pelo Beagle para Server-Driven UI

**(1) Ser capaz de aplicar mentoria técnica (para JRs/PLs) com foco em desenvolvimento iOS e (2) receber feedbacks positivos de forma consistente/contínua dos mentorados e de meus pares.**

(1) Objetivo de capacidade; (2) Resultado que vem ao atingir o objetivo

- Aprender Swift com profundidade
- Ter domínio do iOS SDK e entender objetivos por trás das decisões de design da plataforma
- Ter bons conhecimentos sobre os Guidelines da apple para o desenvolvimento de apps
- Conhecer interoperabilidade com Objective-C e heranças do modelo de desenvolvimento anterior
- Ter domínio sobre boas práticas de qualidade com Swift/iOS
- Ter domínio sobre ferramentas de gestão e segurança
- Estudar sobre mentoria técnica

**Ser capaz de conversar de forma produtiva (a contribuir) com especialista iOS da Zup/Itaú/Mercado sobre detalhes da plataforma iOS**
- Aprender Swift com profundidade
- Ter domínio do iOS SDK e entender objetivos por trás das decisões de design da plataforma
- Ter bons conhecimentos sobre os Guidelines da apple para o desenvolvimento de apps
- Conhecer interoperabilidade com Objective-C e heranças do modelo de desenvolvimento anterior
- Conhecer funcionamento da LLVM e runtime environment
- Consumir conhecimento que está sendo produzido pela comunidade em seus eventos
- Seguir pessoas influentes na comunidade e consumir o que produzem. Interagir se possível.
- Conversar com pessoas da comunidade ou próximas dela de forma recorrente
  - digmatic.junior (iap)
  - @felipe.torres (alura)
  - slack ios dev br

## Prática Deliberada

### Objetivos de Desenvolvimento

**Aplicativo da Casa do Código (visão de admin)**
- CRUD de Autor (tab)
- CRUD Livro (tab)

Objetivos de capacidade relacionados: Aprender Swift com profundidade; Ter domínio do iOS SDK e entender objetivos por trás das decisões de design da plataforma

Skills: Type System, Variáveis, Constantes, Operadores; Tipos Opcionais; Coleções; Controle de Fluxo; Funções; Classes, Structs e Protocolos; Disponibilidade e Error Handling; Tipos Customizados; Closures; UIKit; Auto Layout (possíveis outros sistemas de layout); Controles de Navegação; Data Fetching HTTP; Gerenciamento de Dependências (CocoaPods)

**Aplicativo da Casa do Código (o retorno - visão cliente)**
- Home (listagem de livros)
- Tela de Detalhe (push/pop navigation)
- Tela de Checkout (tab navigation)
- Tela de Perfil (crUd nome e foto)

Objetivos de capacidade relacionados: Aprender Swift com profundidade; Ter domínio do iOS SDK e entender objetivos por trás das decisões de design da plataforma

Skills: Todo o set de skills do projeto anterior; Biblioteca de Fotos e Camera; Persistência; View Code; Design de Código; Controles de Autenticação e Autorização

**Aplicativo do Gym.app**
- Splash Screen
- Onboard walkthrough + Location Permissions
- Login
- Home (layout design) + Profile
- Checkin (geolocation service; listagem de academias com layout complexo e animações)

Objetivos de capacidade relacionados: Aprender Swift com profundidade; Ter domínio do iOS SDK e entender objetivos por trás das decisões de design da plataforma; Ter domínio sobre ferramentas e práticas de layout e usabilidade (UIKit, SwiftUI)

Skills: Todo o set de skills do projeto anterior; Design Patterns com Swift; Arquitetura; Formatação e Styleguide (SwiftLint); Testes Unitários; Testes de Integração; Testes e2e; Gestão de Permissões (Apple Guideline); Gestão de Senhas, Chaves e Secrets (iCloud Keychain?); Gestão de Certificados e Profiles; Beta Testing com Testflight; Publicação na AppStore

**Aplicativo do ITI**
- Login com Face Recognition/fingerprint scan
- Abertura de Conta
- Solicitação de Cartão de Crédito
- Pix
- Movimentações da carteira

Skills