

Projetos em Swift que compilam rápido

Um sonho possível

Cícero Camargo - Junho/2021

Timeline de um projeto iOS:

Projeto do zero! Build em segundos!



1 mês depois - build já acima de 1 minuto
Alamofire, Kingfisher, Firebase/*, FBSDK*, etc.



6 meses depois - build acima dos 5 minutos
Faz uma mudança, coloca para compilar,
abre o Twitter, perde o foco... volta.



É sempre assim... né?

Será mesmo? 🤔

Não precisa ser!

Compre um Mac com chip M1!



Dividir para conquistar

Modularização e gerenciamento *fino* de dependências.

Esta é a base de uma solução escalável.

Quem sou eu?

Dev iOS desde 2012

@cicerocamargo.dev no
Instagram

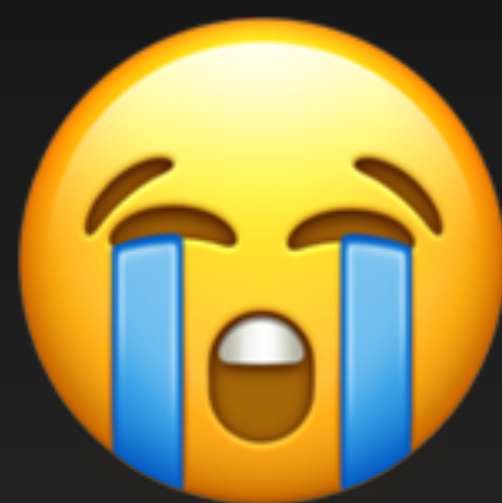


Minha trajetória

- Pelotas (2012 - 2016)
 - Muito ObjC, um pouco de Swift
 - Try harder or walk away?
 - Gerência de projetos...
- São Paulo (2016 - 2017)
 - 50% ObjC / 50% Swift

Minha trajetória

- De volta a Pelotas (2017)
 - Projeto grande, monolítico, feito às pressas
 - Equipe grande, vários freelancers
 - 100% Swift
 - Máquinas velhas
 - Muita lib
 - Build acima de **16 minutos!!**
 - **Baixa produtividade...**



Soluções

Código feio para agradar compilador ✓

Reescrever algumas partes em ObjC 🤔

Máquinas novas ✓

Desenvolver em um projeto à parte ✓

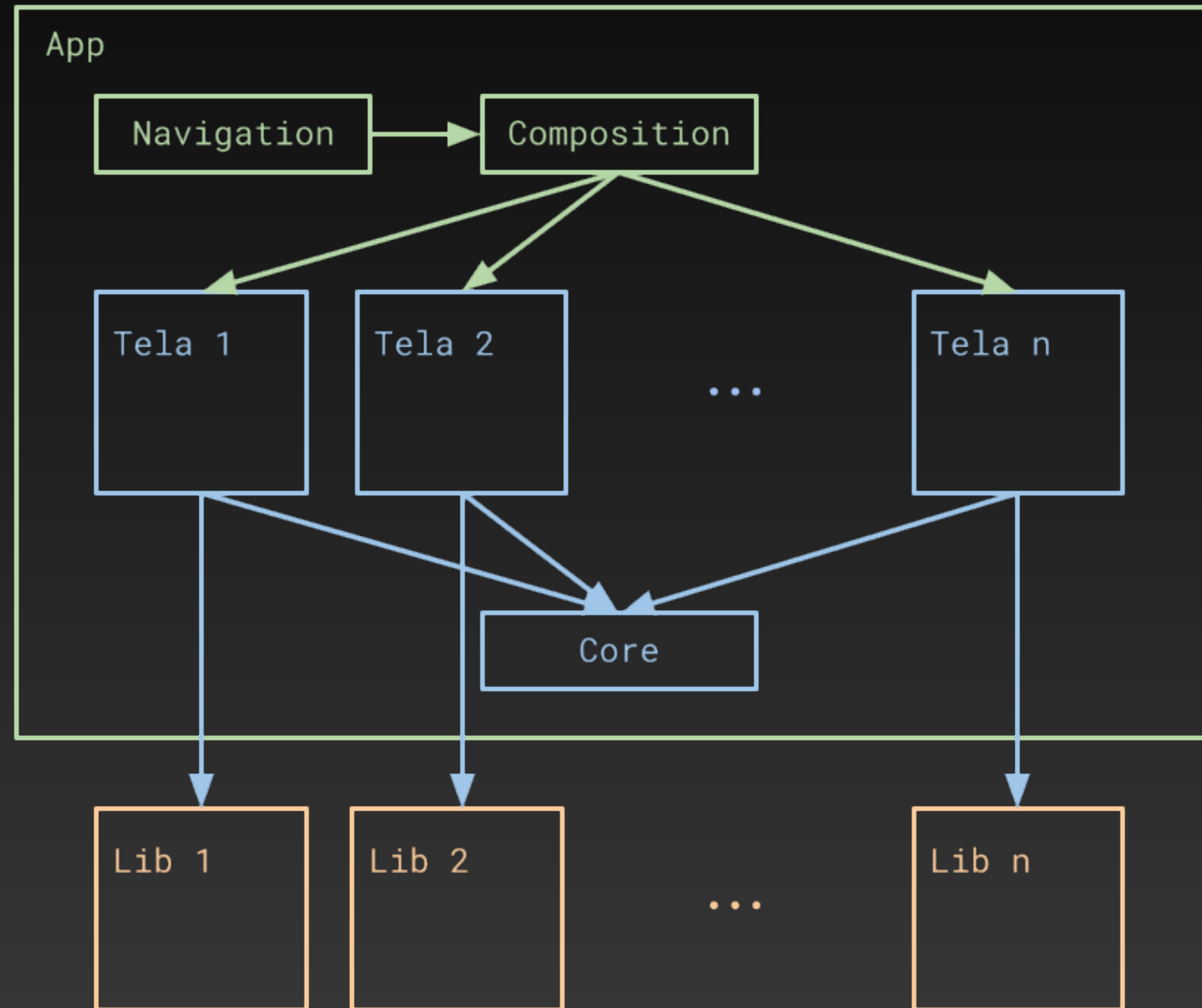


Show me the code!

<https://github.com/cicerocamargo/Workouts>

Versão 1

The majestic monolith



Versão 1

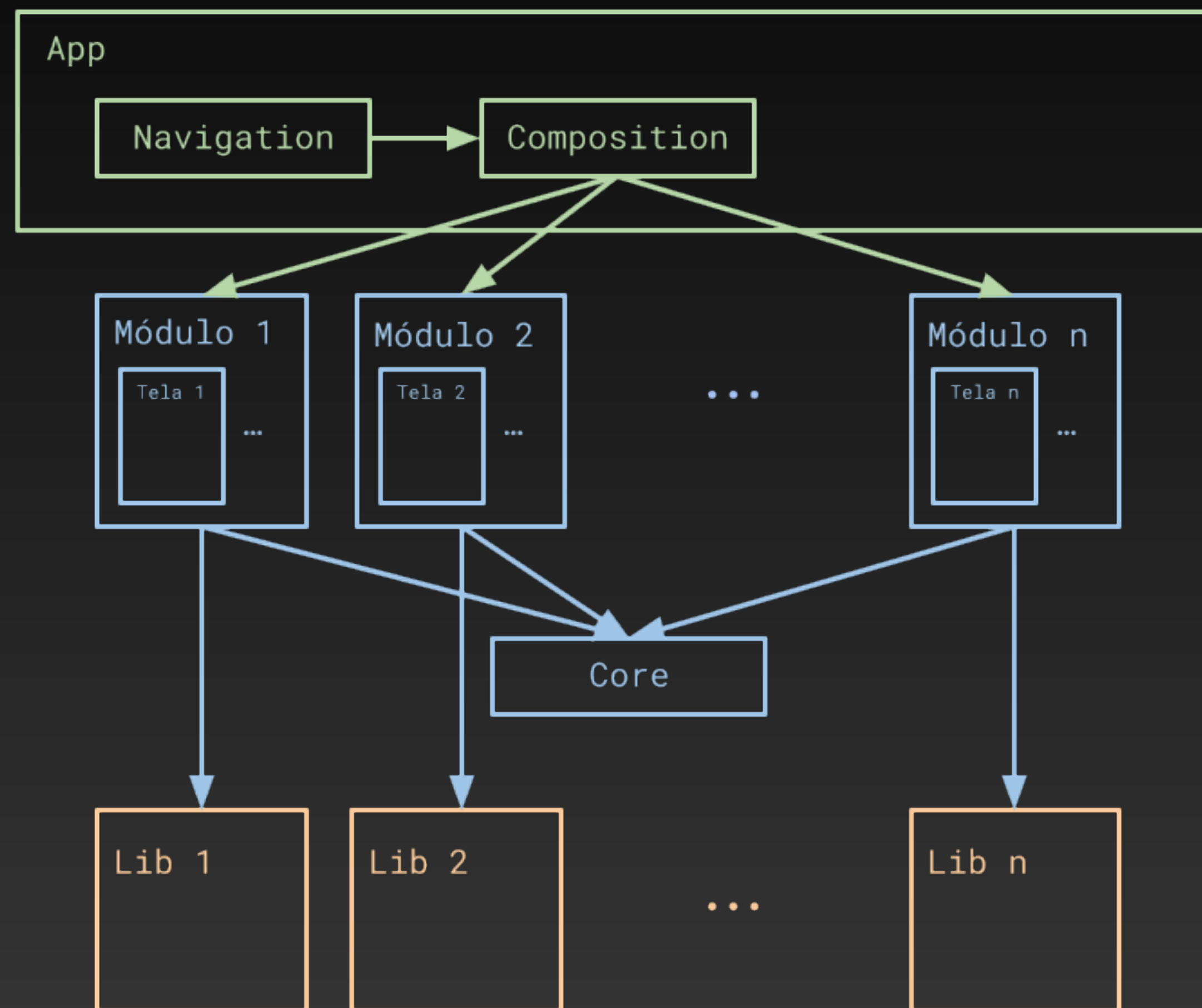
The majestic monolith

Build time: ~32s

MacBook Pro (15-inch, 2017)
2,8 GHz Quad-Core Intel Core i7, 16 GB RAM

Versão 2

Vários módulos importando dependências instáveis



Versão 2

Vários módulos importando dependências instáveis

Build Time:

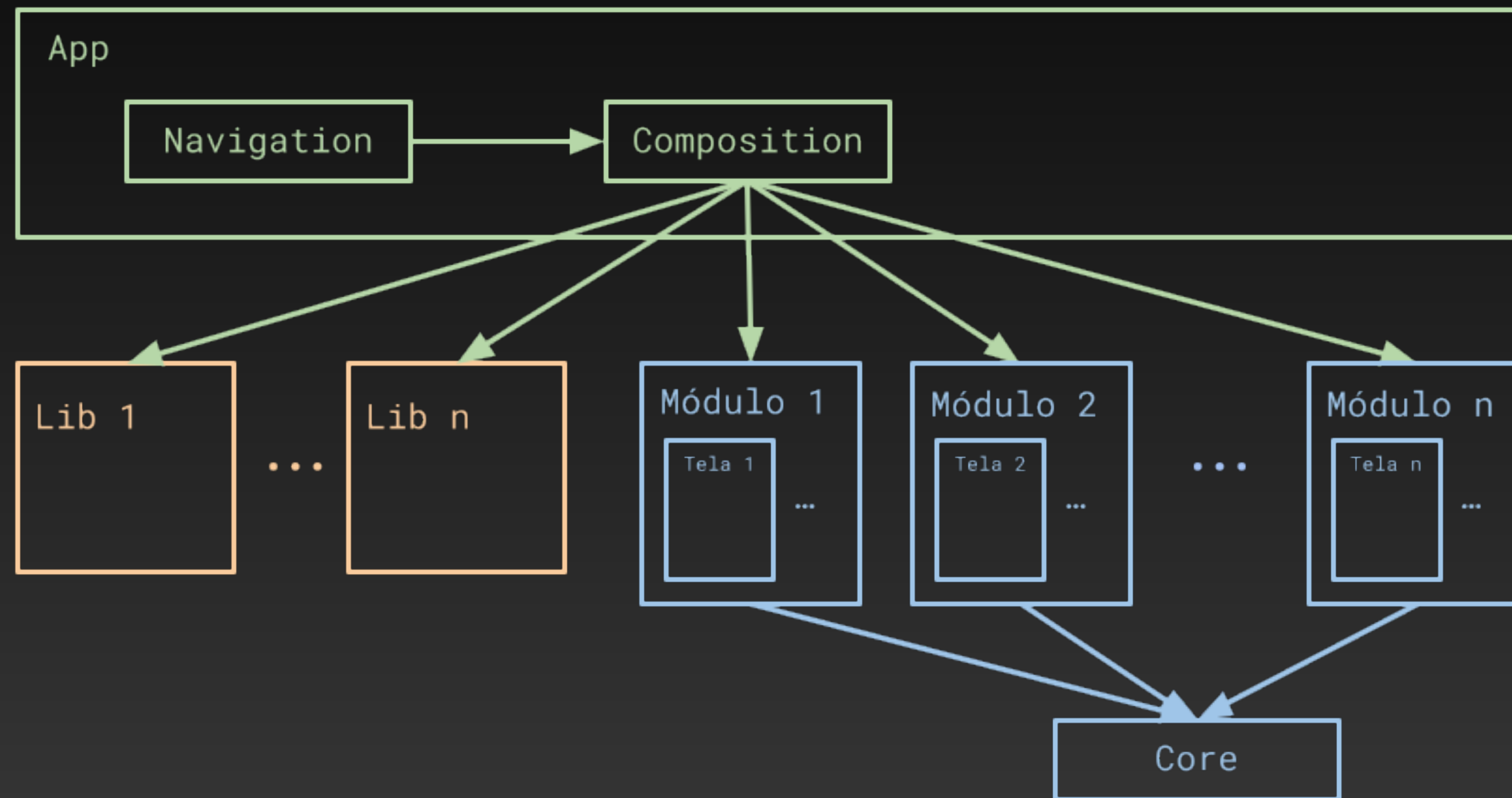
- BrowseWorkouts: ~9s
- Paywall: ~26s
- Settings: ~1s
- WorkoutPlayer: ~1s

App: ~30s

MacBook Pro (15-inch, 2017)
2,8 GHz Quad-Core Intel Core i7, 16 GB RAM

Versão 3

Vários módulos + dependências resolvidas na raiz do app



Versão 3

Vários módulos + dependências resolvidas na raiz do app

Build Time:

Todos os módulos: ~3s
(~0,7s por submódulo)

App: ~33s

MacBook Pro (15-inch, 2017)
2,8 GHz Quad-Core Intel Core i7, 16 GB RAM

Conclusões

Técnicas

- Swift demora mesmo para compilar, então foque no que trará resultados maiores e escaláveis: trabalhe em **módulos pequenos e independentes**;
- Arquitetura limpa e Injeção de dependências não garantem escalabilidade.
- Gerencie com cuidado suas dependências: resolva dependências instáveis somente na raiz do app;
- SPM para para libs e para módulos (adeus conflitos no *pbxproj*)
- Pratiquem TDD

Conclusões

Não técnicas

- Não aceitem um normal ruim, usem o desconforto como motivação
- Não ache que já sabe tudo (*try harder >> walk away*)
- Invista em conhecimento, mas ache fontes confiáveis

Obrigado!
Perguntas?

