Monorepos e Microfeatures

o que nem todo mundo de conta

Sli.do
#TDC_YAS

$ whoami

Yasmin Benatti
iOS Dev @ iFood

@minbenatti

yasminbenatti.com
github.com/yabenatti

co-host @ MovileCast

co-organizer @ CocoaHeads Campinas

Agenda

Quem são? Onde vivem?

Como começou no iFood?

Os novos desafios

E o futuro?

O que são

Monorepos

Monorepos

Monorepo != Monolito

1 único repositório

Múltiplos módulos *(μFeatures)*

O que são

µFeatures

µFeatures

Libraries

Produto ou Foundation

Example App

Testes

Contexto

iFood iOS

2015

1 dev iOS
1 app

2018

13 devs iOS
2 apps
3+ módulos
repos diferentes

Fev
2019

code owner
+ **monorepo**

Julho
2019

BUCK

2020

35+ devs iOS
3 apps
150+ µFeatures

Okay, let's go!

O Começo

Compartilhamento de código

Reutilização

Reutilização

Componentes UI

Strings

Test Utils

Entities

Alinhamento entre

Times

Alinhamento

Trunk-based development

Commit atômico

Feature flags

Liberdade de experimentar

Criação de módulos

Automação ❤️

```
~ ifood > Libraries feature/my-cool-ufeature % _
```

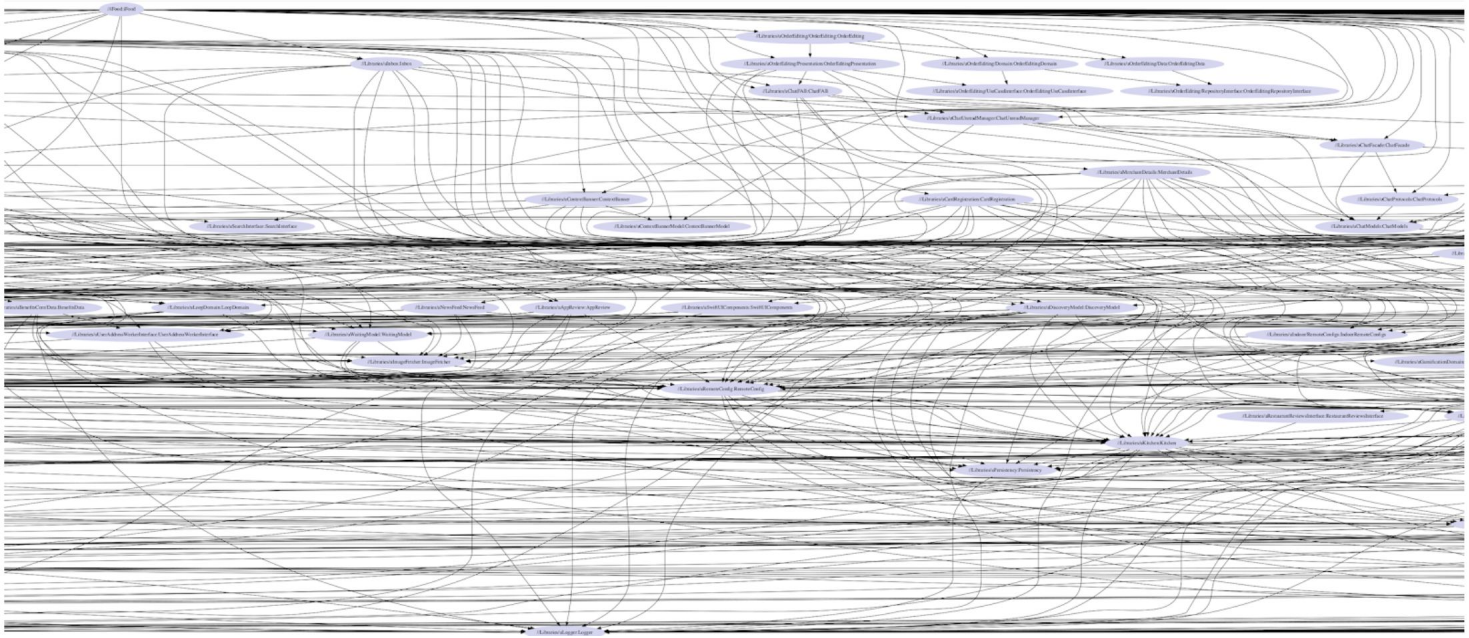
Hmm.. interessante

Novos desafios

Gerenciamento do grafo de

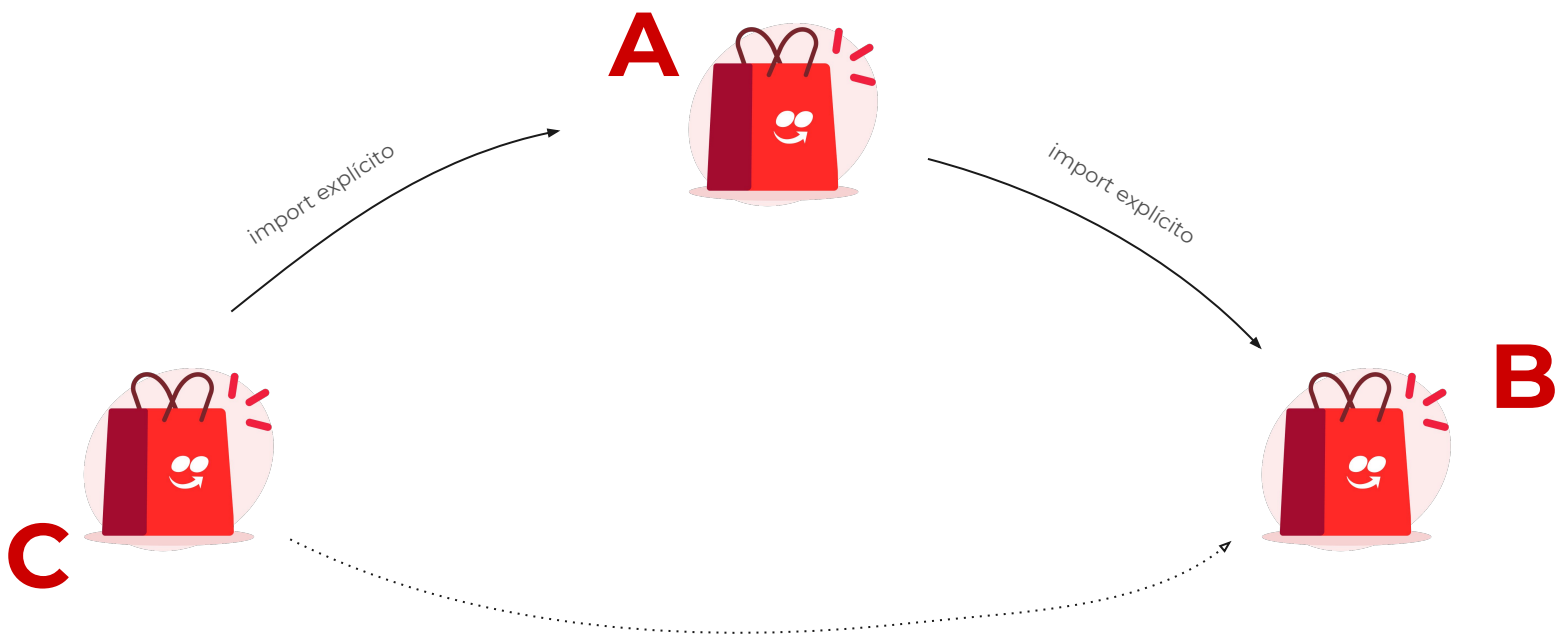
Dependências

The image is too large. Images must be smaller than 25 megapixels.


iOneFood iFood
iLibraries\index.Index
iLibraries\OrderEditing\OrderEditing.OrderEditing
iLibraries\OrderEditing.Presentation.OrderEditingPresentation
iLibraries\OrderEditing.Domain.OrderEditingDomain
iLibraries\OrderEditing.Data.OrderEditingData
iLibraries\ChefAB.ChefFAB
iLibraries\OrderEditing.UseCaseInterface.OrderEditingUseCaseInterface
iLibraries\OrderEditing.RepositoryInterface.OrderEditingRepositoryInterface
iLibraries\ChefUnreadMessages.ChefUnreadMessages
iLibraries\ChefFacade.ChefFacade
iLibraries\AboutUsersDetails.MerchantDetails
iLibraries\ConsumerInner.ConsumerInner
iLibraries\CardRegistration.CardRegistration
iLibraries\ChefProtocols.ChefProtocols
iLibraries\SearchInterface.SearchInterface
iLibraries\ConsumerHomeModel.ConsumerHomeModel
iLibraries\ChefModels.ChefModels
iLibraries\DeliveryOrderDetailsData
iLibraries\LongDomain.LongDomain
iLibraries\NewsFeed.NewsFeed
iLibraries\AppReview.AppReview
iLibraries\iosUIUIComponents.IosUIUIComponents
iLibraries\DiscoveryModel.DiscoveryModel
iLibraries\CurrAddressWorkerInterface.CurrAddressWorkerInterface
iLibraries\PricingRider.PricingModel
iLibraries\shadowBoxesConfigs.ShadowBoxesConfigs
iLibraries\AppFeedbackInsightsApi
iLibraries\GeneralizedInsight
iLibraries\RestaurantCuisineRestaurant.Cuisine
iLibraries\RestaurantReviews.RestaurantReviewsInterface
iLibraries\KitchenKitchen
iLibraries\Persistency.Persistency
iLibraries\Logger.Logger

Dependências

Indiretas



quer usar métodos

Dependência Indireta

Protocol Witness Table

Deixar explícito no BUCK file

Comunicação entre módulos

Router Service

Router Service

Bridges

Injeção de dependências

Rotas

Só mudei uma linha!

Rebuild

Rebuild

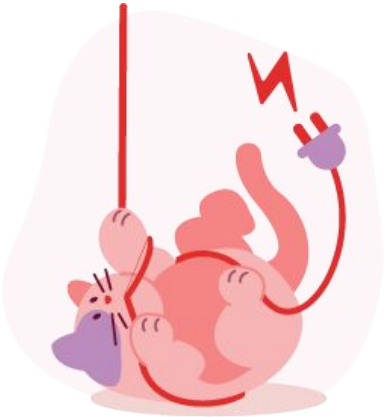
Protocolos

Módulo de interface

You get a problem

Eita...

Novos problemas



Namespaces

SwiftGen

```
first_party_library(
    name = "CommonLocalizable",
    deps = [
        ":uCommonLocalizableStrings"
    ],
    swiftgen_strings = True,
    swiftgen_strings_custom_params = "--param enumName=CommonLocalizable --param publicAccess=true -t structured-swift4"
)

apple_resource(
    name = "uCommonLocalizableStrings",
    visibility = ["PUBLIC"],
    files = glob([
        "Resources/Strings/*.strings",
        "Resources/Strings/*.stringsdict",
    ])
)
```

iFood.strings

Locale

Locale

Múltiplos apps

Língua estática no app

Instabug usa strings em inglês

Manutenção

Bibliotecas

Bibliotecas

FBSnapshot

Quick/Nimble

Diferenças entre pods

kkcry

Cada vez mais complexo

Xcode

Code Coverage

Adding code coverage enable option on buckconfig #1

Edit | Open with

Merged **rockbruno** merged 1 commit into `master` from `feature/code-coverage-xcscheme` on Jul 17, 2019

Conversation 1 | Commits 1 | Checks 0 | Files changed 4 | +28 −6

Changes from **all commits** | File filter... | Jump to... | 0 / 4 files viewed | Review changes

7 src/com/facebook/buck/apple/xcode/XCScheme.java | Viewed

```
@@ -339,17 +339,20 @@ public String getBuildConfiguration() {
339 339     public static class TestAction extends SchemeAction {
340 340       List<TestableReference> testables;
341 341       private final String buildConfiguration;
    342 +     private final boolean codeCoverageEnabled;
342 343       private final Optional<ImmutableMap<String, String>> environmentVariables;
343 344
344 345       public TestAction(
345 346           String buildConfiguration,
346 347           Optional<ImmutableMap<String, String>> environmentVariables,
    348 +         boolean codeCoverageEnabled,
347 349           Optional<ImmutableList<SchemePrePostAction>> preActions,
348 350           Optional<ImmutableList<SchemePrePostAction>> postActions) {
349 351         super(preActions, postActions);
350 352         this.testables = new ArrayList<>();
351 353         this.buildConfiguration = buildConfiguration;
352 354         this.environmentVariables = environmentVariables;
    355 +       this.codeCoverageEnabled = codeCoverageEnabled;
353 356       }
354 357
355 358       public void addTestableReference(TestableReference testable) {
@@ -367,6 +370,10 @@ public String getBuildConfiguration() {
367 370       public Optional<ImmutableMap<String, String>> getEnvironmentVariables() {
368 371         return environmentVariables;
369 372       }
    373 +
    374 +     public boolean getCodeCoverageEnabled() {
```

Continuous Integration

Cache

CI e Cache

Cache remoto e local no CI

~ 50% redução build time

CI e Cache

1 test run == build de todas as dependências

Xcode *path too long*

Build com Buck

Open source Buck Local #134

Merged **xianwen** merged 13 commits into `master` from `xianwen--buck-local` on Jul 8

Conversation 23 | Commits 13 | Checks 0 | Files changed 32

xianwen commented on Jul 2 — Contributor

Buck Local is an internal tool developed in Airbnb.
It generates an Xcode project that invokes Buck to build the App when you hit the Build button in Xcode. This makes local iOS builds 2x faster without Buck Http cache, and iOS complete builds 5~6 times faster when Buck Http cache is enabled.

To apply Buck Local in your own iOS project, you can do so by:

1. Copy `Rakefile` file and `BuckLocal` dir from BuckSample to the root directory of your own iOS project
2. Replace all `apple_asset_catalog` with `buck_local_apple_asset_catalog`, and `apple_resource` with `buck_local_apple_resource` in your BUCK files.
3. Migrate the top level `apple_binary` in your BUCK file to `buck_local_binary`, `xcode_workspace_config` to `buck_local_workspace`, and `apple_bundle` to `buck_local_bundle` as demonstrated in this PR.

After these steps, you'll be able to generate your own Buck Local project by calling `rake buck_local:generate_project`.
You'll be able to build / run / debug your app in Xcode as if it's a normal Xcode project, while the local builds are more than 2 times faster.

- [x] Built and ran in Xcode (`make project`)

🎉 8

cw

E o futuro...

O que nos aguarda?

Futuro

Cache de build para devs

Otimizações

App Clips + WatchOS

~~Toda nova versão de OS~~

Calma,

Vai dar bom

Novos conceitos

Ferramentas complexas

Compiladores/frameworks

Configurações e flags

Outras linguagens

Onde achar informações

Comunidade open source

Criatividade

Dúvidas?

Sli.do
#TDC_YAS

@minbenatti

Obrigada!

#VemProiFood

ifood