

The background is a dark blue-grey color with various abstract geometric patterns. There are several orange circles of different sizes, some with white dots inside. There are also white circles and hexagons, some with orange outlines. A large orange hexagon is in the top right corner. A white dotted pattern is in the bottom right corner. A white dotted pattern is also in the top left corner. A white dotted pattern is in the middle left corner. A white dotted pattern is in the middle right corner. A white dotted pattern is in the bottom left corner. A white dotted pattern is in the bottom right corner. A white dotted pattern is in the top right corner. A white dotted pattern is in the bottom center.

Migrando um aplicativo mobile nativo para Flutter: dificuldades e boas práticas



Marcelo Baccelli

Analista de sistemas do Serpro há 12 anos

Formado pela PUC-SP

Desenvolvedor mobile desde o J2ME

Hobbie: Jogos eletrônicos e filmes



Marcio Frayze David

Analista de sistemas do Serpro há 12 anos

Desenvolvedor Java há mais de uma década

Autor dos projetos:
<https://segunda.tech>
e p de Podcast



Marco Porcho

Analista de sistemas do Serpro há 12 anos

Formado pelo ICMC-USP

Desenvolvedor mobile há 7+ anos

Entusiasta de home office, board games e música



Carteira Digital de Trânsito (CDT)

App de trânsito:

- ★ versões digitais da CNH e CRLV;
- ★ compartilhamento de CRLV;
- ★ pagamento de infrações com desconto de até 40%;
- ★ indicação de principal condutor e mais funcionalidades



Carteira Digital de Trânsito (CDT)

Migrada para Flutter na versão 3.0
(novembro/2019)

20 milhões de usuários ativos

Nota 4,7+ na App Store e Play Store

Prêmio iBest de Melhor Serviço do Governo de acordo com júri popular

.....POR QUE MIGRAMOS PARA FLUTTER?.....

- ★ Spike de 2 semanas: sucesso!
- ★ Dart e Flutter são muito fáceis de aprender
- ★ Interação com as plataformas nativas é simples
- ★ Um único código a se manter



Flutter

+



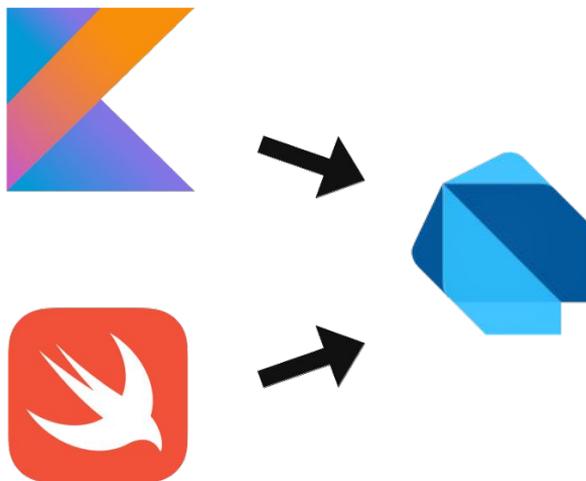
Dart

=



REESCREVENDO O CÓDIGO

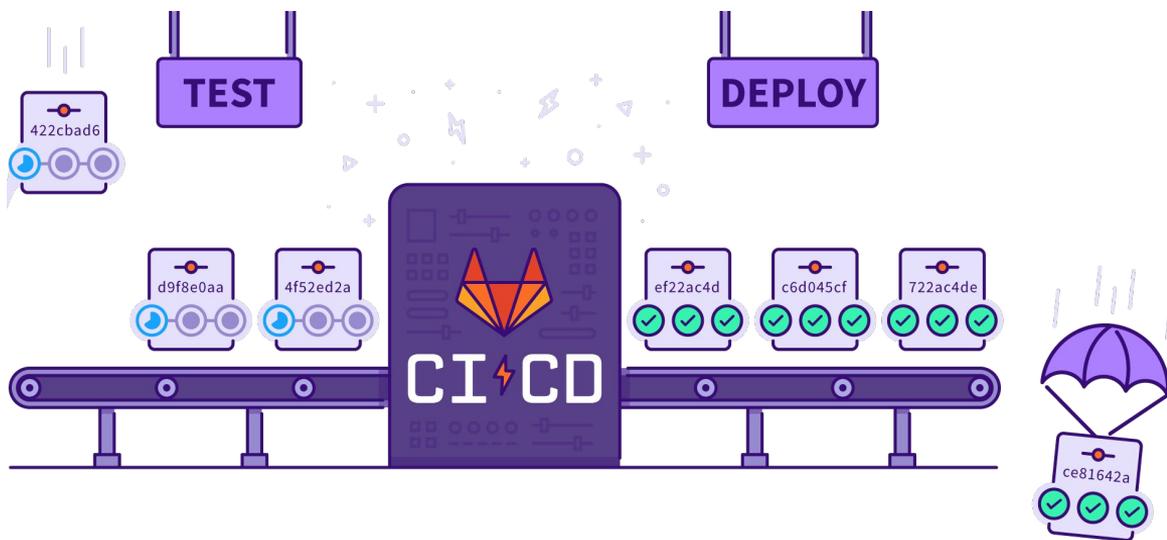
- ★ “Traduzir” levaria a um código de baixa qualidade
- ★ Decidimos por reescrever em Dart
- ★ Hoje em dia, facilita a evolução da arquitetura



INTEGRAÇÃO CONTÍNUA

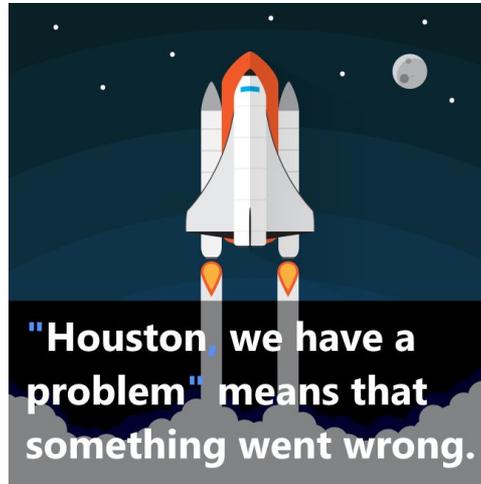
Pipeline de build automatizada

- ★ Feature toggles (no backend e no app)
- ★ Evitamos branches
- ★ Phased release/staged rollout



..... GARANTINDO UMA MIGRAÇÃO SUAVE

- ★ Antes da migração, já tínhamos 3 milhões de usuários
- ★ Migração foi tratada como uma feature da versão 3.0
- ★ Análise de plugins para verificação de compatibilidade



LEITOR BIOMÉTRICO

- ★ Problema: local_auth utiliza Biometric, incompatível com dispositivos antigos
- ★ Solução: fazer nosso próprio plugin para leitura biométrica no Android

Samsung Galaxy S5 (23 modelos)



Instalações em dispositivos ativos (últimos 30 dias) ?

5.831

0,04% de todas as instalações em dispositivos ativos

Samsung Galaxy Note4 (16 modelos)



Instalações em dispositivos ativos (últimos 30 dias) ?

1.912

0,01% de todas as instalações em dispositivos ativos

PREFERÊNCIAS DOS USUÁRIOS

- ★ Problema: plugin shared_preferences incapaz de obter preferência do legado
- ★ Solução: obter as preferências usando platform channels

```
_getString(String key) async {  
  String response = "";  
  response = await platform.invokeMethod('getString', {'key': key});  
  return response;  
}
```

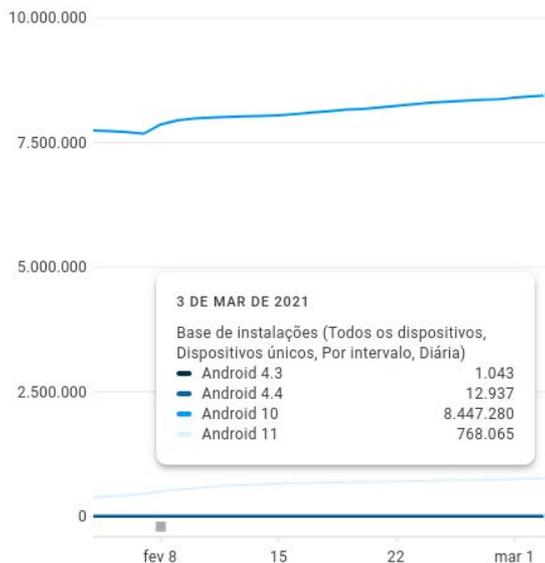
```
class ObterPreferenciasNativeMethod(val context: Context): NativeMethod {  
  override fun execute(call: MethodCall, result: MethodChannel.Result) {  
  
    val prefs: SharedPreferences = PreferenceManager.getDefaultSharedPreferences(context)  
    val key = call.argument<String>("key")  
  
    when(call.method){  
      "getString" -> {  
        val value = prefs.getString(key, "").orEmpty()  
        result.success(value)  
      }  
      //(...)  
    }  
  }  
}
```

```
class ObterPreferenciasNativeMethod: NativeMethod {  
  func execute(_ call: FlutterMethodCall, _ result: FlutterResult) {  
    let defaults = UserDefaults.standard  
  
    let arguments = call.arguments as! NSDictionary  
    let key = arguments["key"] as! String  
  
    switch call.method {  
      case "getString":  
        let value = defaults.string(forKey: key)  
        result(value ?? "")  
        //(...)  
    }  
  }  
}
```



COMPATIBILIDADE COM APIS ANTIGAS

- ★ Problema: plugins do Flutter demandaram APIs mais recentes do que as suportadas pela versão nativa
- ★ Solução: incrementar API mínima e verificação em tempo de execução



GERENCIAMENTO DE ESTADOS

- ★ No início, arquitetura caseira
 - Pouco escalável, baseada em *setState*
- ★ Várias alternativas para gerenciamento de estados
 - Provider, BloC, Cubit, Riverpod, MobX
- ★ MobX foi a escolhida
- ★ Arquitetura nova e antiga convivendo com (relativa) harmonia

TESTE A/B

- ★ Mudanças tem um grande potencial de impacto
- ★ Impossível prever o comportamento do app com detalhes, nos milhares de dispositivos
- ★ Testes A/B auxiliam na avaliação de mudanças e na aceitação dos usuários



Firebase A/B Testing

E O FUTURO?

- ★ Migração para Flutter 2 (null-safety 🍷)
- ★ Aumentar a cobertura de testes automatizados
- ★ Melhorar nosso *bot* de atendimento
- ★ Novas funcionalidades

LIÇÕES APRENDIDAS

- ★ Saber nativo ajuda muito
- ★ Migrar não é pra todo mundo
- ★ Seu usuário típico não tem um Galaxy S21 Ultra
- ★ Nem todo plugin do Flutter serve pra você
- ★ Pense na sua arquitetura
- ★ Compatibilidade com versões antigas pode ser desafiadora

LINKS ÚTEIS

Post sobre migração pra Flutter <https://medium.com/@porcho/c792bfdf0d27>

Flutter Codelab <https://flutter.dev/docs/get-started/codelab>

Lista de abordagens para gerenciamento de estados

<https://flutter.dev/docs/development/data-and-backend/state-mgmt/options>

Curso Flutter Development Bootcamp

<https://www.udemy.com/course/flutter-bootcamp-with-dart>

Flutterando <https://flutterando.com.br>

Tutorial de MobX

<https://circleci.com/blog/state-management-for-flutter-apps-with-mobx/>

Carteira Digital de Trânsito <https://servicos.serpro.gov.br/carteira-digital>

Obrigado!



Baixe essa apresentação: <https://segunda.tech/tdc>

Marcelo Baccelli

 marcelo-baccelli

 marcelo.baccelli@gmail.com

Marcio Frayze

 marciofrayze

 marcio-frayze

 marcio@segunda.tech

 @marciofrayze

Marco Porcho

 marco-porcho

 maporcho@gmail.com