

The background is a dark blue-grey color. It is decorated with various geometric shapes in orange and white. There are circles of different sizes, some with dotted patterns inside. There are hexagons, some solid orange and some outlined in white. There are also triangles and lines. Some shapes are partially cut off by the edges of the frame. The overall style is modern and minimalist.

# Conhecendo os bastidores do App CDT (Carteira Digital de Trânsito)

## 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

## ..... 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



# 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

# Carteira Digital de Trânsito (CDT)



**Primeira versão** lançada em 2017: em Java (Android) e Swift (iOS)

**Segunda versão** lançada em 2018: versão Android migrada para Kotlin

**Terceira versão** lançada em 2019: *facelift* generoso e migração para em Flutter

**Quarta versão** lançada em 2020: possibilidade de pagamento de infrações com 40% de desconto



# Carteira Digital de Trânsito (CDT)

**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

# ..... O QUE É FLUTTER? .....

- ★ Framework do Google para construir aplicações mobile, Web e desktop a partir de uma única base de código
- ★ Utiliza Dart como linguagem
- ★ Open Source
- ★ Estável: versão 1.0 lançada há 2 anos; hoje está na 2.0.3



# O QUE É FLUTTER?

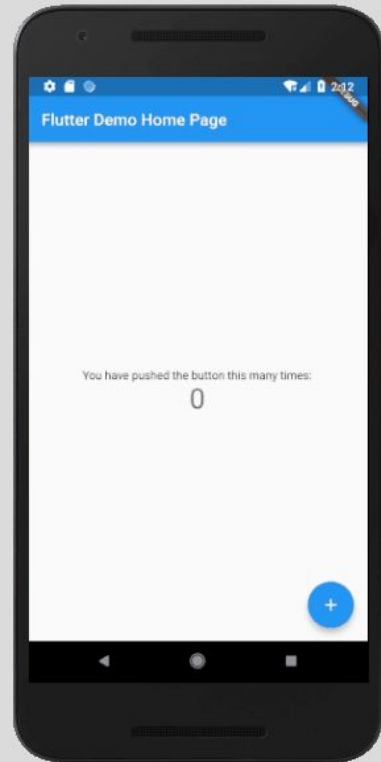
- ★ Desenvolvimento rápido:  
*hot reload*

```
main.dart x
9      return MaterialApp(
10        title: 'Flutter Demo',
11        theme: ThemeData(
12          // This is the theme of your application.
13          //
14          // Try running your application with "flutter run". You
15          // application has a blue toolbar. Then, without quitting
16          // changing the primarySwatch below to Colors.green and
17          // "hot reload" (press "r" in the console where you ran
18          // or simply save your changes to "hot reload" in a Flui
19          // Notice that the counter didn't reset back to zero; t
20          // is not restarted.
21          primarySwatch: Colors.blue,
22        ), // ThemeData
23        home: MyHomePage(title: 'Flutter Demo Home Page'),
24      ); // MaterialApp
25    }
26  }
27
28  class MyHomePage extends StatefulWidget {
29    MyHomePage({Key key, this.title}) : super(key: key);
30
31    // This widget is the home page of your application. It is st
32    // that it has a State object (defined below) that contains f
33    // how it looks.
34
35    // This class is the configuration for the state. It holds th
36    // (see the title) provided by the parent (for this case the A
```

PROBLEMS TERMINAL ... 1: dart

```
Performing hot reload...
Reloaded 1 of 419 libraries in 899ms.
```

Flutter 1.0.0 Android SDK built for x86 (android-x86 Emulator) Formatting: ✓ 2





# ..... O QUE É FLUTTER? .....

- ★ Facilidade em construir interfaces agradáveis
- ★ Alta performance: apps compilados para código de máquina
- ★ Usado pelo Nubank, Serpro, Alibaba e Google



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

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



Flutter

+



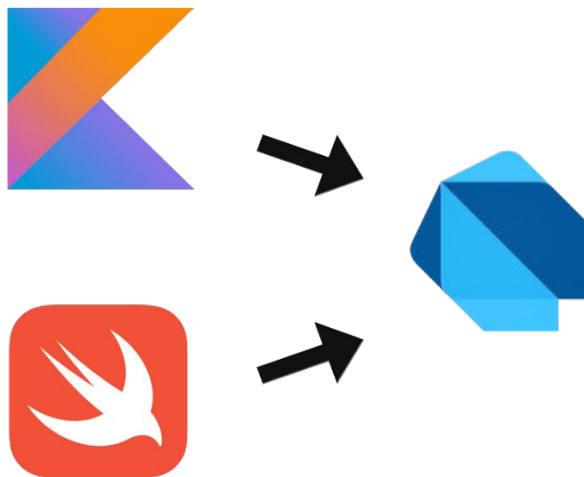
Dart

=



## REESCREVENDO O CÓDIGO

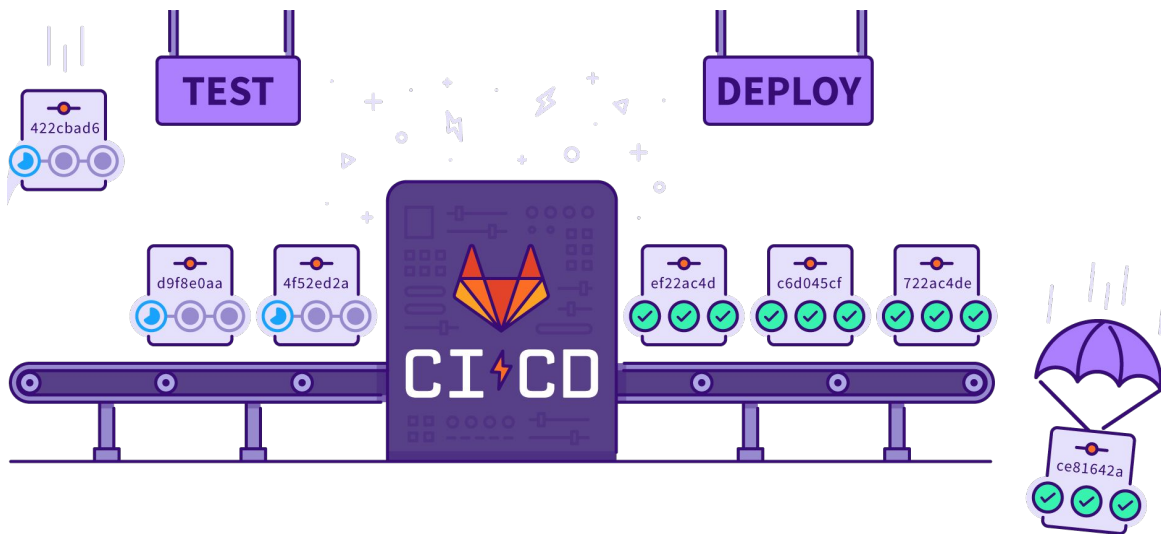
- ★ “Traduzir” levaria a um código de baixa qualidade
- ★ Decidimos por reescrever em Dart
- ★ Hoje em dia, facilita a evolução da arquitetura



# INTEGRAÇÃO CONTÍNUA

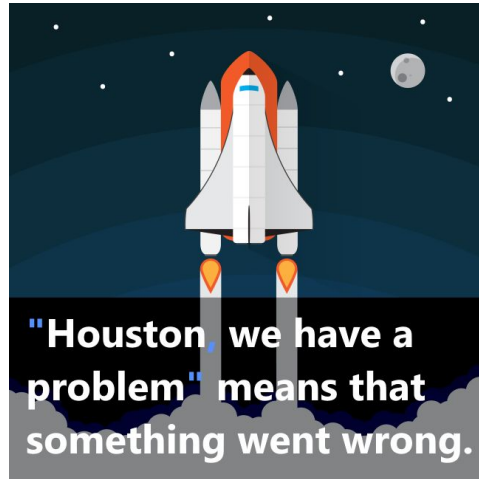
Pipeline de build automatizada

- ★ 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;  
}
```

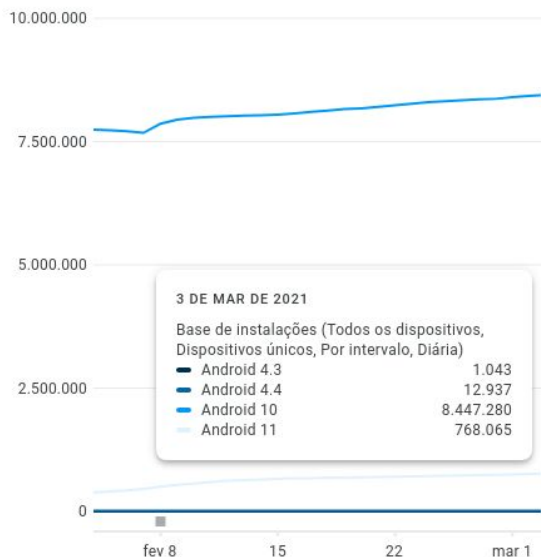
```
class ObterPreferenciasNativeMethod(val context: Context): NativeMethod {  
  override fun execute(call: MethodCall, result: MethodChannel.Result) {  
    val prefs: SharedPreferences = PreferenceManager.getDefaultSharedPreferences(context)  
    val key = call.argument<String>("key")  
  
    when(call.method){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

## ANÁLISE DE COMENTÁRIOS

- ★ Comentários das lojas são monitorados por uma equipe especializada
- ★ Sugestões são consideradas e incorporadas ao app...quando possível



O meu carro está no nome da minha esposa, e por isso o app diz que eu "não tenho veículos", e não consigo adicionar um. O que eu faço se for parado numa blitz e minha esposa não estiver comigo para mostrar o app na conta e no celular dela?

Se o AiQFome for dar cupom, eu quero que o CDT exclua minhas muitas rsrs

## 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>

**Marcio Frayze**



marciofrayze



marcio-frayze



marcio@segunda.tech



@marciofrayze

**Marco Porcho**

marco-porcho



maporcho@gmail.com

