

The background features a complex network of thin grey lines and dots, forming a web-like structure. Scattered throughout are various triangles of different sizes and orientations, some with solid black dots at their vertices. The overall aesthetic is modern and technical.

# **PROGRAMA DEV VENTURE**

---

Desenvolvimento Android

**PROJETO NA VIDA  
REAL**

Resolvendo problemas

**01**

**MVP I**

Lance os Dados

**02**

**GIT**

Sistema de controle de  
código

**03**

**EXECUTAR**

App no celular

**04**

# Aula 02 AGENDA





---

# PRÉ REQUISITOS

## **Android Studio**

Ter o Android Studio instalado na máquina pessoa

## **Emulador**

Ter um emulador configurado

---

## **Git**

Instalar Git no computador

## **Conta no Github**

Quem não tem uma conta, criar



01

**LANCE OS DADOS**

---





---

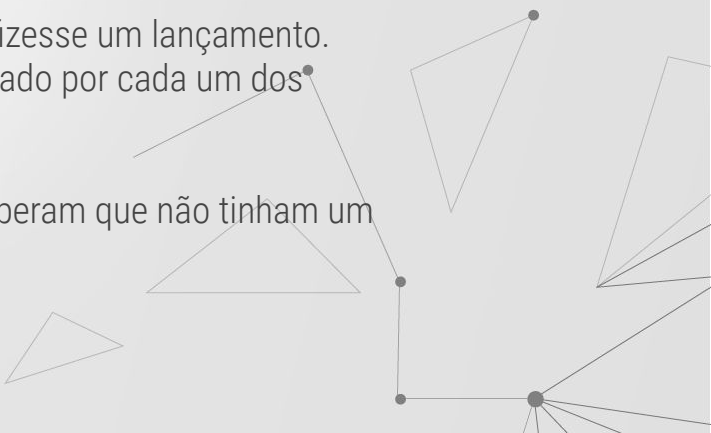
# CONTEXTO


Um grupo de pessoas, resolveram descobrir quem era a mais sortuda do grupo.

Por isso, procuraram um par de dados para que cada uma fizesse um lançamento. Quem conseguisse o maior número somando o valor mostrado por cada um dos dados receberia o título de pessoa mais sortuda.

Todas muito animadas para lançar os dados, até que perceberam que não tinham um par de dados para brincar :(

---

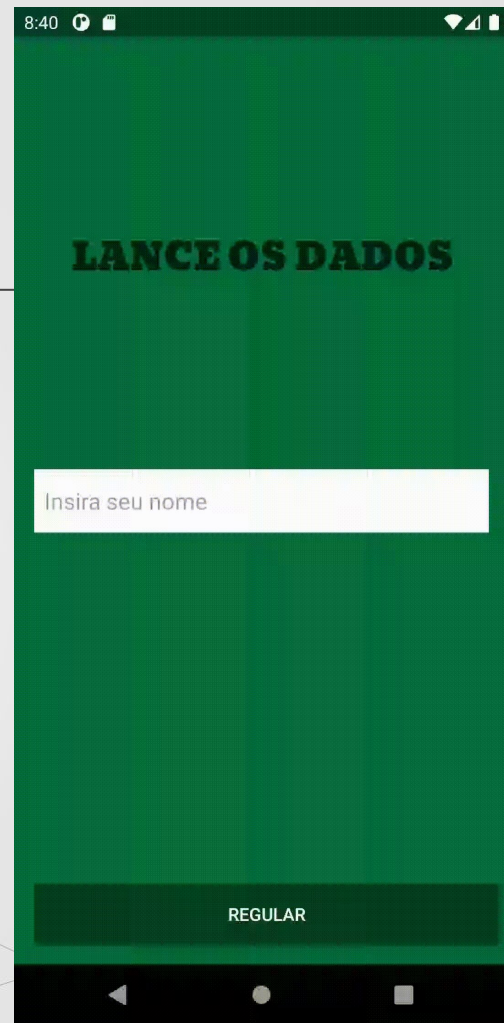




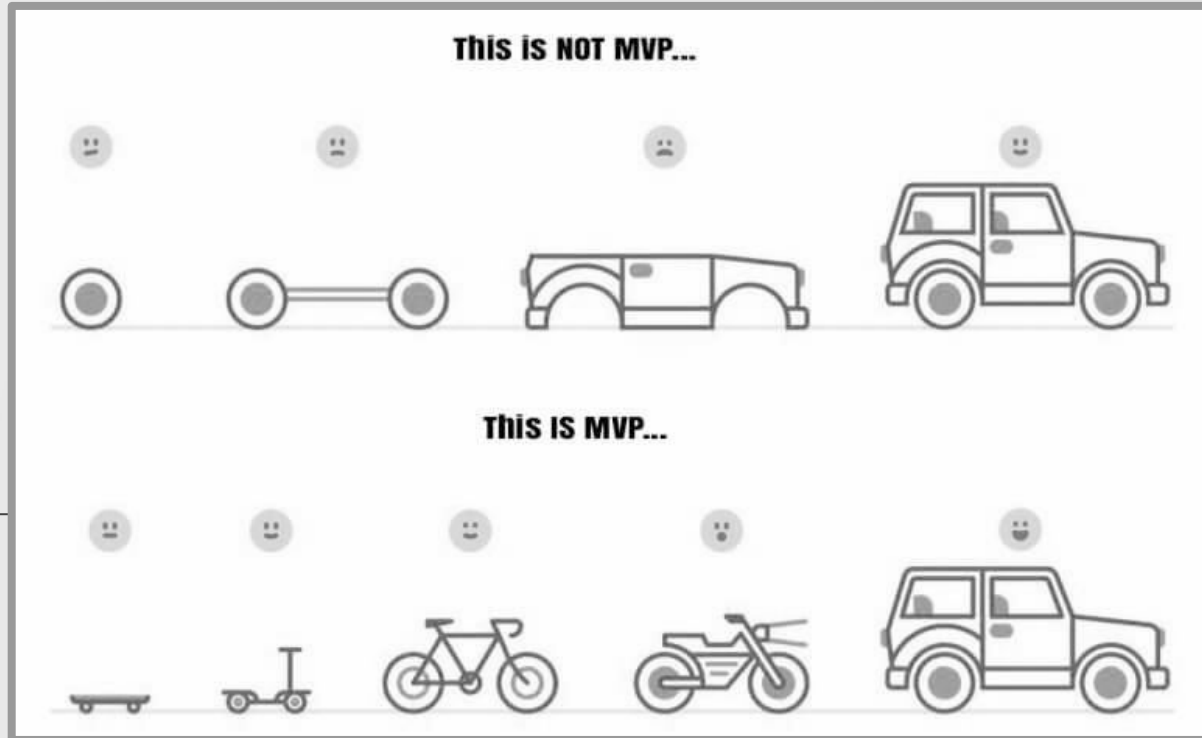
Sorte ou não do grupo, uma das pessoas sabia criar aplicações android. E ela prometeu que da próxima vez o jogo iria acontecer, só que de uma forma mais moderna...

---

## NOSSA PRIMEIRA APP: LANÇADOR DE DADOS

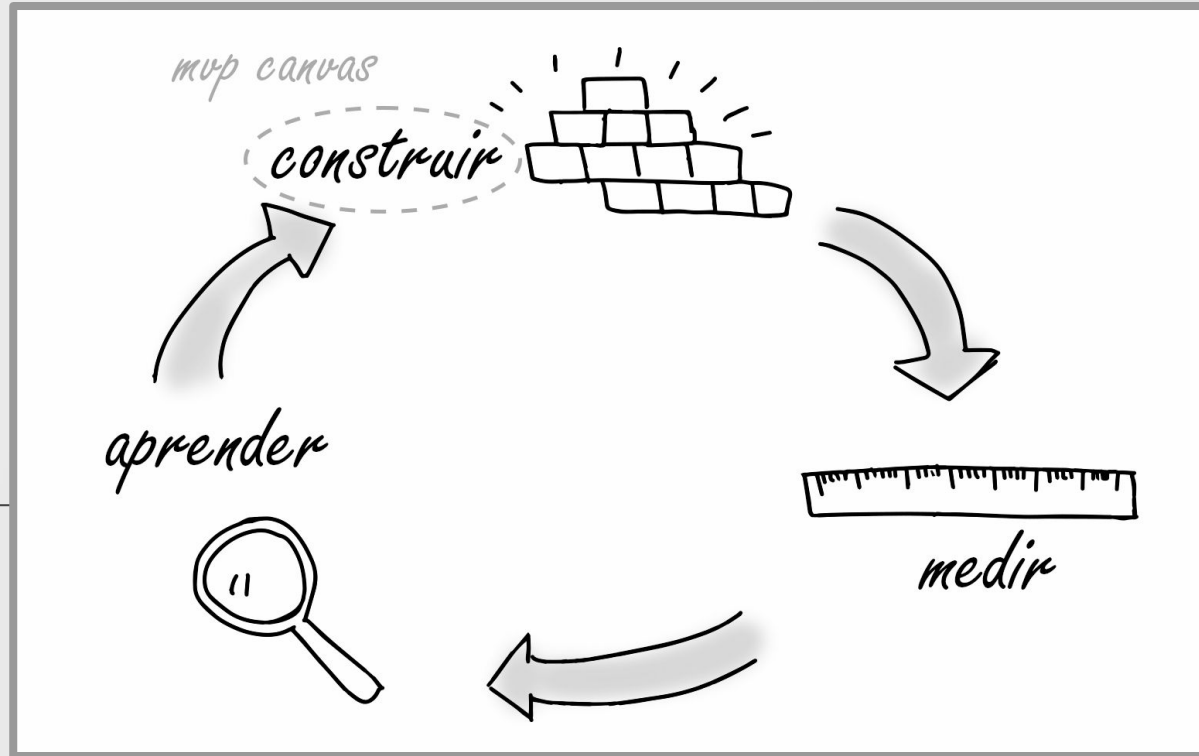


# ETAPA 1 - MVP



[VÍDEO] MVP - como funciona?

# ETAPA 1 - MVP







# REFLITA

---

1. Qual o **problema** que a app Lance Dados se propõe a resolver?
  2. Como resolver esse problema da maneira mais **simples\*** possível?
  3. Como **dividir o trabalho** de construir a app em etapas sem perder a solução do problema principal?
- 

**KISS - KEEP IT SIMPLE STUPID**



# ENTREGAS

I

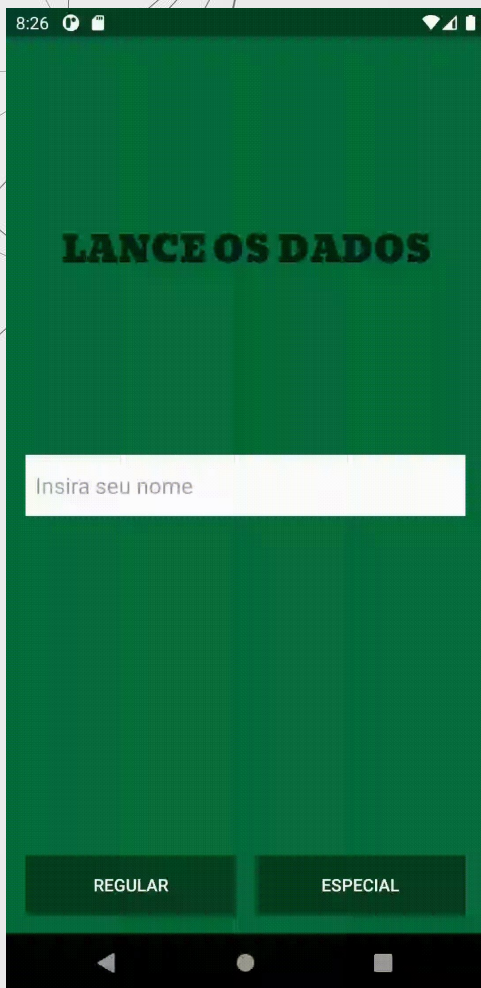
A partir do toque  
no botão mostrar  
dois números  
aleatórios (que  
vão de 1 até 6)

II

A partir do toque  
do botão mostrar  
dois dados com  
valores  
aleatórios (que  
vão de 1 até 6)

III

Cadastrar jogador  
e iniciar o jogo  
com dados já  
lançados.  
Lançar os dados  
novamente quando  
o jogador quiser.



## IV

Disponibilizar  
função de lançar  
dado "especial"  
Que lança 2  
dados de 8 faces

# DESAFIO FINAL



# 02

## MVP I

---

Gerador de número aleatórios



# *dojo* / INTERFACE

---

Crie uma tela com um botão "Jogar" e 2 caixas texto: "Dado 1" e "Dado 2"

---





# MANIFEST.XML

---

Todo projeto de aplicativo precisa ter um arquivo **AndroidManifest.xml** (com esse exato nome) na raiz do conjunto de origem do projeto.

O arquivo de manifesto descreve informações essenciais sobre o aplicativo para as ferramentas de compilação do Android, para o sistema operacional Android e para o Google Play.

---

**MANIFEST**




---

**Gradle é um sistema de automação de compilação** que descrever a estrutura do projeto, configuração e dependências do aplicativo.

Ao compilar e executar seu aplicativo, você vê informações sobre a execução da compilação do Gradle. Você também verá informações sobre o Android Package Kit (APK) sendo instalado. (APK é o formato de arquivo do pacote que o sistema operacional Android usa para distribuir e instalar aplicativos móveis.)

---



# MANIPULANDO ELEMENTOS VISUAIS

---

A vinculação de views permite a programação de códigos que interagem com componentes visuais.

- `findViewById()`
- View Binding\*
- Kotlin synthetic imports\*\*
- Butter knife\*\*

---

\*View Binding - Documentação

\*\* Parte do kapt. Também depreciado

\*\*\*Lib que facilita o vínculo entre os componentes visuais e as classes. Atualmente depreciada

**TextView**







# *dojo* / VINCULANDO UMA AÇÃO

---

Associe a esse botão a ação de alterar o texto da primeira caixa para: "Valor X"

---





# TEXTVIEWS

---

Um elemento da interface que exibe texto para o usuário.

- Tamanho do texto
- Estilo da fonte
- Cor do texto
- Conteúdo
- 

Váaaarias outras coisas! Dá uma futricada aqui:

<https://developer.android.com/reference/android/widget/TextView>

É possível alterar as propriedades desse elemento, nas nossas classes kotlin, no arquivo xml de layout ou ainda nos arquivos de estilo (exploraremos num futuro próximo 😊)

---



# *dojo* / TEXTVIEWS

---

Configure os dois textos da tela com as seguintes propriedades:

fontFamily: sans-serif-medium

textSize: 18sp

textStyle: bold

---





## *dojo* / **BUTTONS**

---

Adicione um botão na tela e associe ao botão as ações de alterar o texto da segunda caixa para: "Valor do Dado 2" e cora o texto de vermelho.

---





# *dojo* / FUNÇÕES

---

Crie uma função que sorteie um número aleatório que varie de 1 até 6

---





# *dojo* / FUNÇÕES

---

Associe a chamada da função ao clique do botão, exibindo o resultado nas caixas de texto.

---





# 03

## GIT

---

Sistema de Controle de Código Fonte

# O QUE É GIT

Git é um Sistema de Controle de Código Fonte (do inglês Source Code Management).

É uma ferramenta importante para nos apoiar na gestão de históricos dos documentos (entre várias outras coisas)

**Arquivo:**  
curso-android.txt

18:35 - 06/07/2021  
| Marcella Criou o  
arquivo

18:40 - 06/07/2021  
| Marcella  
adicionou um título

18:47 - 06/07/2021  
| Marcella  
adicionou uma  
imagem

18:50 - 06/07/2021  
| Marcella coloriu o  
texto de cinza

**GIT**



# O QUE É GIT

## Linux:

```
$ sudo yum install git-all
```

```
$ sudo apt-get install git-all
```

## Mac OS:

```
$ git
```

<http://git-scm.com/download/mac>

## Windows:

<http://git-scm.com/download/win>




# O QUE É GIT

## git init

---

### Num terminal (Git Bash)

- `git init myproject`
- `cd myproject`
- `git config user.email "seu email"`
- `git config user.name "seu nome"`

- Pasta myproject **é criada**
  - Dentro desta pasta a **pasta .git passa a existir**
  - **Tudo que fizermos** dentro desta pasta pode ser versionado pelo Git.
  - Definimos para o projeto git nossa autoria
- 



# O QUE É GIT

## git status

### Num terminal (Git Bash)

- git status
- touch primeiroarquivo.txt
- git status

- **Status:** alguma novidade na minha pasta? Algo que eu queira gravar?

# O QUE É GIT

```
marcelasouza@2045-1B: ~/Personal Projects/testgit/myproject
~/Pe/t/myproject git master git status
branch master

commits yet

nothing to commit (create/copy files and use "git add" to track)
~/Pe/t/myproject git master
```

```
On branch master
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  primeiroarquivo.txt

nothing added to commit but untracked files present (use "git add" to track)
```



# O QUE É GIT

## git add

---

### **Num terminal (prompt comando)**

- git add .
- git status

- add: coloca as mudanças feitas numa espécie de área intermediária (stage área) antes delas serem permanentemente registradas
- Você pode adicionar arquivos específicos ou tudo que foi modificado (.)

# O QUE É GIT

## git add

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   primeiroarquivo.txt
```



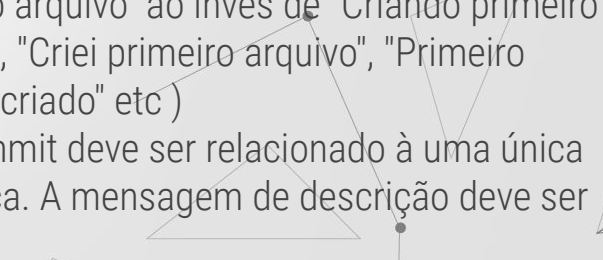
# O QUE É GIT

## git commit

---

### **Num terminal (prompt comando)**

- `git commit -m "Meu primeiro commit"`
- `git status`

- commit: grava permanentemente as alterações no histórico (mas dá pra desfazer se precisar)
  - É comum em projeto termos padrões de como escrever a mensagem do commit:
    - Padronizar a língua
    - Escrever a mensagem no imperativo ("Cria primeiro arquivo" ao invés de "Criando primeiro arquivo", "Criei primeiro arquivo", "Primeiro arquivo criado" etc )
    - Um commit deve ser relacionado à uma única mudança. A mensagem de descrição deve ser curta.
- 

---

### **Conventional Commits**

# O QUE É GIT

## git commit

```
🍏 ~/Pe/t/myproject git master +1 git commit -m "Meu primeiro commit"
[master (root-commit) 2388dc0] Meu primeiro commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 primeiroarquivo.txt

🍏 ~/Pe/t/myproject git master git status ✓ 11:47:13 ☺
On branch master
nothing to commit, working tree clean

🍏 ~/Pe/t/myproject git master ✓ 11:47:27 ☺
```

Conventional Commits





# O QUE É GIT

## git log

**Num terminal (prompt comando)**

- git log

- log: mostra histórico dos commits

**Conventional Commits**



# O QUE É GIT

## git log

```
commit 2388dc024f419609c21145bc797bb056e3c62946 (HEAD -> master)
```

```
Author: Marcella Souza <marcella.souza@ifood.com.br>
```

```
Date: Sat May 8 11:47:13 2021 -0300
```

```
    Meu primeiro commit
```

```
(END)
```


Conventional Commits



# DESAFIO

## **Num terminal (prompt comando)**

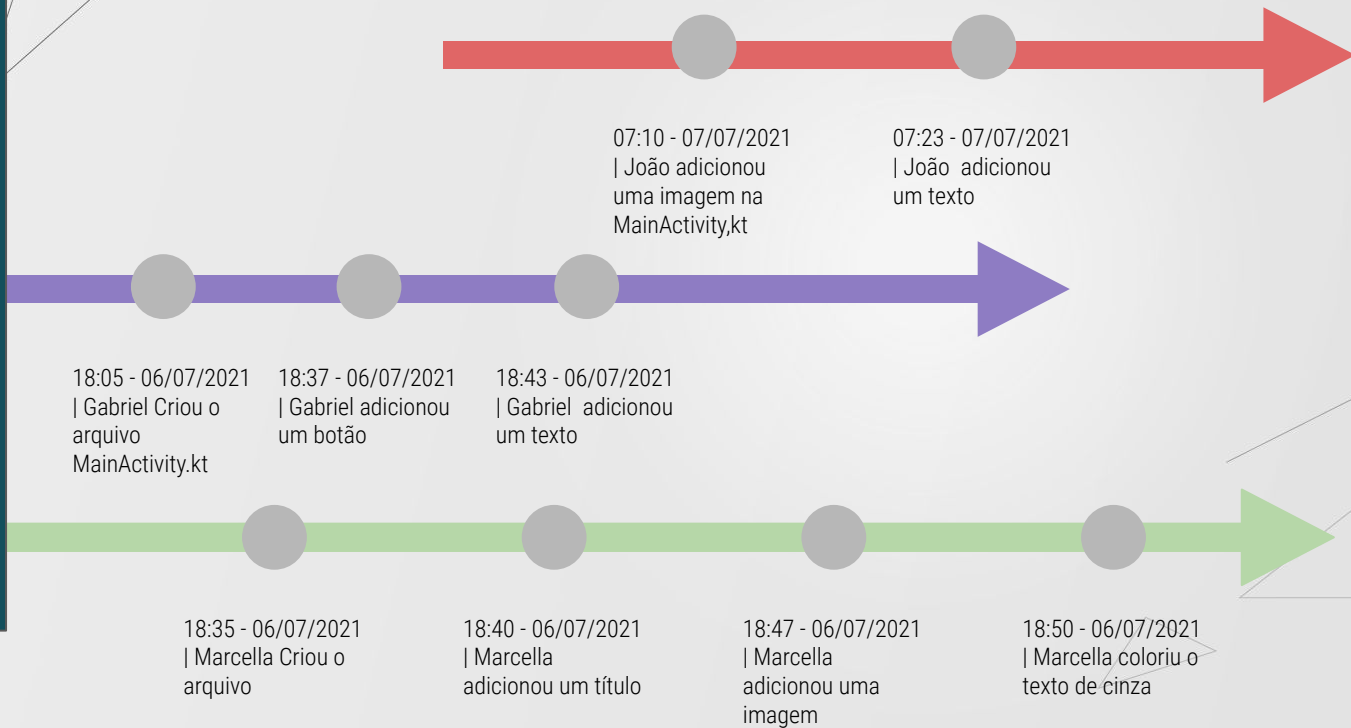
- nano primeiroarquivo.txt

- Adicione no arquivo o texto: meu nome é "seu nome aqui"
  - Salve as alterações
  - Verifique o status do seu repositório
  - Adicione o arquivo no stage
  - Faça um commit
  - Verifique o histórico do seu repositório
- 

# GIT


## TRABALHO DISTRIBUÍDO

Repositório do time





# **GIT** **TRABALHO DISTRIBUÍDO**

- Necessidade de uma "fonte da verdade"
  - Mudanças locais devem ser sincronizadas com a fonte da verdade para que outras pessoas consigam acessá-las
- 

# GITHUB

**GitHub é uma plataforma de hospedagem** de código-fonte e arquivos com controle de versão usando o Git.






# GITHUB

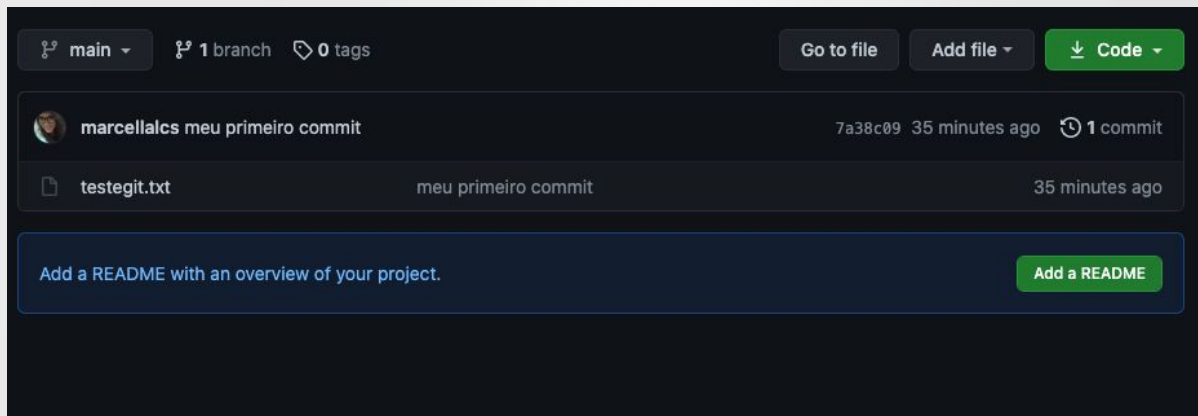
- Crie no Github um repositório
- Sincronize seu repositório local com o repositório remoto que criamos

```
git remote add origin  
https://github.com/suaconta/myproject.git  
git branch -M main  
git push -u origin main
```



# GITHUB

Você vai precisar autorizar o acesso ao repo remoto para o push dar certo







# 04

## APP NO CELULAR

---

APK, distribuição e usando nosso celular como emulador



---

# APK


Android Package (APK) é um arquivo de pacote destinado ao sistema operacional Android. É uma sigla em inglês da palavra Android Application Pack.

```
./gradlew assembleDebug
```

---

```
app/build/outputs/apk/debug/app-debug.apk
```

Para distribuir os apps, empacotamos o APK!





---

# Nosso celular como emulador

- Habilite as opções de desenvolvedor do seu celular
  - Conecte o USB
  - Seu aparelho vai aparecer como uma opção de emulador
  - Play no AndroidStudio
- 



# Palavras chave da aula de hoje:

MVP  
KISS  
Manifest.xml  
Gradle  
Views  
TextViews  
FindViewById  
Git  
Github

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

**Please keep this slide for attribution.**