

PROCEDIMENTOS PARA IMPLEMENTAÇÃO DO ODAS-TRL-FRAMEWORK

Sumário

[1 Configurações 2](#_Toc125981482)

[1.1 Flutter 2](#_Toc125981483)

[1.1.1 Google USB 2](#_Toc125981484)

[1.2 GitHub + VS Code 2](#_Toc125981485)

[1.2.1 Procedimentos 3](#_Toc125981486)

[V. Ir no ícone Accounts (canto esquerdo inferior) e clicar em Turn on Settings Sync... 3](#_Toc125981487)

[VI. Escolher settings para sincronizar e clicar em Sign in & Turn on e depois Sign in with GitHub 3](#_Toc125981488)

[VII. Tem um outro ícone do GitHub para acessar também 3](#_Toc125981489)

[VIII. Criar um novo projeto no VS Code (View – Command Palette – Flutter: New Project – Application - ) 3](#_Toc125981490)

[IX. Adicionar ao README.md o seguinte conteúdo: 3](#_Toc125981491)

[X. Criar arquivo .env.development (as informações do programa que são confidenciais) com o seguinte conteúdo: 4](#_Toc125981492)

[XI. Abrir o Source Control (canto esquerdo) 4](#_Toc125981493)

[XII. Selecionar Publish to GitHub e escolher o repositório (o arquivo .gitignore mostra arquivos que não serão publicados) 4](#_Toc125981494)

[1.3 Firebase 4](#_Toc125981495)

[1.3.1 Procedimentos 4](#_Toc125981496)

[2 Hospedagem 5](#_Toc125981497)

[2.1 Firebase Hosting 5](#_Toc125981498)

[2.1.1 Procedimentos 5](#_Toc125981499)

[3 Bugs 7](#_Toc125981500)

[3.1 Logout 7](#_Toc125981501)

# Configurações

## Flutter

### Google USB

1. Instalar o Google USB Driver (<https://developer.android.com/studio/run/win-usb?hl=pt-br>)
2. Ative as opções do desenvolvedor e a depuração USB no seu dispositivo (Toque em Configurar - Toque em Sobre o telefone -Toque no campo Número da versão 7 vezes) – (Depuração: configurações – sistema – opções do desenvolvedor – depuração usb)
3. Rodar o comando

## GitHub + VS Code

Link: <https://learn.microsoft.com/en-us/training/modules/introduction-to-github-visual-studio-code/>

O Visual Studio Code é um editor de codificação leve e gratuito que muitos desenvolvedores de software usam em todo o mundo. O GitHub é uma plataforma de controle de versão para hospedar projetos de desenvolvimento. O Visual Studio Code fornece uma experiência integrada do GitHub, facilitando o gerenciamento de projetos de dentro do seu editor de código.

Imagine que você deseja começar a contribuir para um projeto hospedado no GitHub. Primeiro, você teria que obter uma cópia do repositório localmente. Em seguida, você gastaria tempo navegando pelos arquivos de código e pelo histórico no GitHub para obter contexto sobre por que uma alteração de código foi feita, quem a fez e há quanto tempo esse código está lá. Para fazer isso, você provavelmente alternaria muito entre seu editor e o GitHub.

Em vez disso, você pode ter uma experiência focada diretamente do Visual Studio Code. Neste módulo, você usará a integração do GitHub dentro do Visual Studio Code para iniciar um novo repositório no GitHub, clonar repositórios existentes e obter contexto para o histórico de cada arquivo, tudo sem sair do editor.

Ao final deste módulo, você saberá como usar o GitHub e os recursos de controle de origem dentro do Visual Studio Code.

### Procedimentos

1. Em uma nova janela do VS Code, selecione o ícone de extensões na barra de atividades. Pesquise a extensão GitHub Pull Requests and Issues e instale-a.

# Ir no ícone Accounts (canto esquerdo inferior) e clicar em Turn on Settings Sync...

# Escolher settings para sincronizar e clicar em Sign in & Turn on e depois Sign in with GitHub

# Tem um outro ícone do GitHub para acessar também

# Criar um novo projeto no VS Code (View – Command Palette – Flutter: New Project – Application - )

# Adicionar ao README.md o seguinte conteúdo:

# My GitHub Project

This is my GitHub project. I published it directly from Visual Studio Code.

# Criar arquivo .env.development (as informações do programa que são confidenciais) com o seguinte conteúdo:

CONNECTION\_STRING=secret-that-should-not-be-pushed-to-github

# Abrir o Source Control (canto esquerdo)

# Selecionar Publish to GitHub e escolher o repositório (o arquivo .gitignore mostra arquivos que não serão publicados)

## Firebase

### Procedimentos

Link: <https://firebase.google.com/codelabs/firebase-get-to-know-flutter>

1. Usar os comandos firebase logout e depois firebase login
2. Criar projeto no site do Firebase
3. Instale a CLI do FlutterFire: dart pub global activate flutterfire\_cli
4. Usar os comandos:

flutter pub add firebase\_core

flutter pub add firebase\_auth

flutter pub add cloud\_firestore

flutter pub add provider

1. Configurar os aplicativos: flutterfire configure. Pode ocorrer o seguinte erro:

**i Found 0 Firebase projects.**

**FirebaseCommandException: An error occured on the Firebase CLI when attempting to run a command.COMMAND: firebase projects:list --json**

**ERROR: Failed to list Firebase projects. See firebase-debug.log for more info.**

Para resolver isso, foi preciso usar o comando: firebase login –reauth

Sucesso: Firebase configuration file lib\firebase\_options.dart generated successfully with the following Firebase apps:

1. Configurar macOS:



# Hospedagem

## Firebase Hosting

Link: <https://firebase.google.com/docs/hosting/quickstart?hl=pt-br>

### Procedimentos

1. Usar os comandos firebase logout e depois firebase login
2. Usar o comando firebase init hosting e escolher o projeto criado do firebase
3. Digitar a pasta do build web (build/web) – lembrar de fazer o flutter build web antes.
4. Decidir se será um website de url única (/index.html)
5. Configurar builds e implantações automáticas com o GitHub (digitar y) - <https://firebase.google.com/docs/hosting/github-integration?hl=pt-br>
6. Escolher repositório do GitHub que você gostaria de configurar um fluxo de trabalho do GitHub

Saída obtida:

**+ Created service account github-action-593748020 with Firebase Hosting admin permissions.**

**+ Uploaded service account JSON to GitHub as secret FIREBASE\_SERVICE\_ACCOUNT\_ODAS\_TRL\_FRAMEWORK.**

**i You can manage your secrets at https://github.com/enfelipesiqueira-br/odas\_trl\_framework/settings/secrets.**

**? Set up the workflow to run a build script before every deploy? Yes**

**? What script should be run before every deploy? npm run build**

**+ Created workflow file C:\Users\User\Google Drive\Felipe\Projetos\Sotwares\VSCode-Flutter\odas-trl-framework\odas\_trl\_framework\.github/workflows/firebase-hosting-pull-request.yml**

**? Set up automatic deployment to your site's live channel when a PR is merged? Yes**

**? What is the name of the GitHub branch associated with your site's live channel? master**

**+ Created workflow file C:\Users\User\Google Drive\Felipe\Projetos\Sotwares\VSCode-Flutter\odas-trl-framework\odas\_trl\_framework\.github/workflows/firebase-hosting-merge.yml**

**i Action required: Visit this URL to revoke authorization for the Firebase CLI GitHub OAuth App:**

**https://github.com/settings/connections/applications/89cf50f02ac6aaed3484**

**i Action required: Push any new workflow file(s) to your repo**

**i Writing configuration info to firebase.json...**

**i Writing project information to .firebaserc...**

**+ Firebase initialization complete!**

1. Configurar o fluxo de trabalho para executar um script de compilação antes de cada implantação
2. Usar o comando firebase deploy para publicar o website - não esquecer de fazer o flutter build web antes

(O deploy pelo github será tratado em outro procedimento futuramente neste relatório)

# Bugs