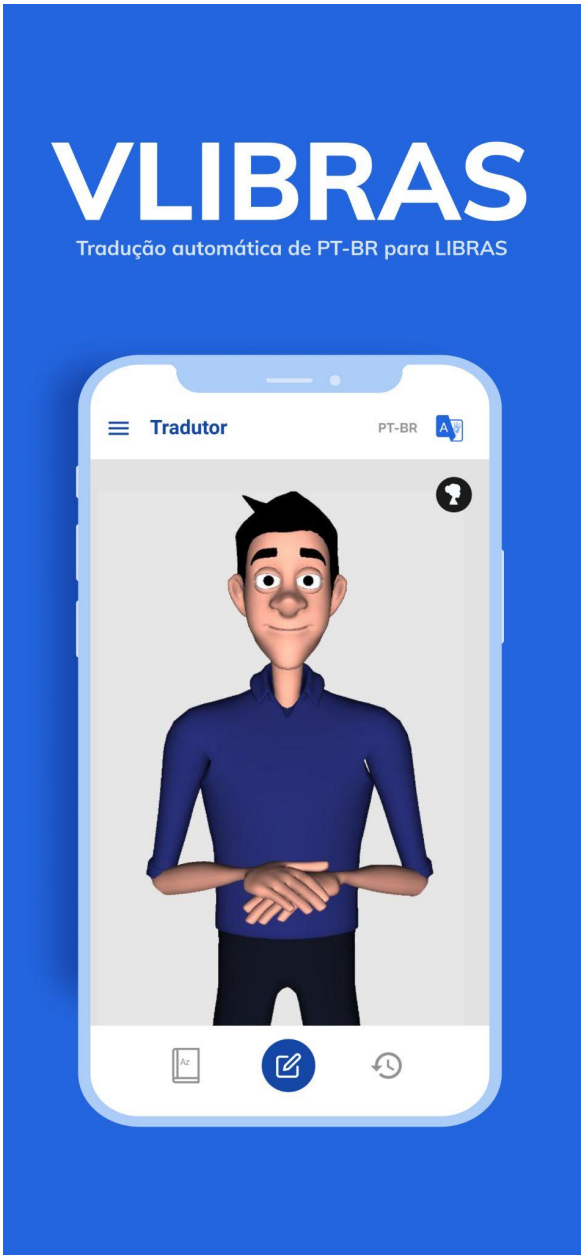
VLibras Cross Platform App

DEV Guide



Out/2021

I. Introdução

Esse documento é um guia de desenvolvimento para entender o aplicativo VLibras Cross Platform App. Nas próximas seções, será apresentada uma introdução geral do aplicativo, arquitetura, manual de testes e preparação de ambiente.

II. Breve descrição

O VLibras Cross Platform App foi desenvolvido utilizando a tecnologia do Ionic 5 React(typescript) para uso de tradução automática de português para LIBRAS nas plataformas IOS e Android, tornando-as acessíveis. O aplicativo busca auxiliar a inclusão de pessoas surdas, além de incentivar o aprendizado da nossa Língua Brasileira de Sinais.

III. Arquitetura

Nesta seção haverão instruções, explicando para que os diretórios são utilizados e também os arquivos de configuração.

- **src**- Diretório contendo todos os arquivos da aplicação, é criado um diretório `src` para que o código da aplicação possa ser isolado em um diretório e facilmente portado para outros projetos, se necessário;
    - **config** - Diretório para guardar os arquivos de configuração da aplicação.
    - **assets** - Diretório para armazenar imagens, icons, gifs e o index.ts que serve para exportar o conteúdo da pasta.
    - **components -** Diretório para criação dos componentes que serão utilizados na aplicação. Dentro do diretório, possui os diretórios dos componentes com a seguinte estrutura: index.ts(componente), styles.css(estilização do componente) e strings.ts(textos utilizados no componente).
        - **index.ts** - Exportador dos componentes.
    - **constants** - Diretório onde são armazenadas as constantes que serão utilizadas na aplicação.
    - **data** - Diretório onde são armazenados os dados de customização do player(corpo, olho, cabelo, calça e camisa) e dados da funcionalidade do Regionalismo(recurso utilizado para filtrar termos regionalistas).
    - **environment** - Diretório onde são armazenados as variáveis de ambiente da aplicação.

- **hooks** - Diretório onde armazena o useContext utilizado para a tradução.

- **layouts** - Diretório onde é armazenado o MenuLayout(menu para navegação da aplicação).
    - **index.ts** - Exportador do MenuLayout

- **models** - Diretório que serve para armazenar o construtor de uma palavra(Words) que será utilizado posteriormente na tradução.

- **pages** - Diretório onde ficam as páginas (telas) da aplicação, como forma de padronização e boas práticas toda página fica dentro de um diretório com seu nome; Cada página possui sua pasta do qual contém o index.ts(arquivo principal), styles.css(estilização da página) e um strings.ts(textos que serão utilizados ná página).

    - **index.js** - Exportador das páginas.

- **routes** - Diretório onde armazena as rotas da aplicação.

- **store** - Diretório onde é armazenado diversas estruturas Redux de "State Management.

- **themes** - Diretório que armazena os temas que serão utilizados na aplicação.

- **services** - Diretório onde serão criados os arquivos relacionados a serviços utilizados na aplicação, por exemplo, requisições HTTP, e qualquer outro serviço que for utilizado;

    - **api.ts** - Arquivo com a configuração da biblioteca Axios para envio de requisições HTTP, o endereço que vem configurado por padrão é para a API do dicionário Vlibras.
        a) Endereço: "https://dicionario2-dth.vlibras.gov.br/api"

    - **suggestionGloss.ts** - API utilizada para realizar requisições de sugestões das traduções.

    - **translate.ts -** Utilizado para enviar requisições das traduções de vídeos.

    - **unity.ts -** Utilizado para requisitar as funcionalidades do player(Ícaro/Rosana) Ex: Play/Pause.

- **utils** - Diretório possui funções que serão reaproveitadas e utilizadas em várias partes dos códigos, que estão nos arquivos dataFormat.ts, file.ts(conversão de dado binário para String), reloadHistory(carregador de histórico) e index.ts(exportador da classe).

Obs.: Para ocorrer a execução da aplicação com a feature de tradutor de vídeo ativada, deve-se ir em src/environment/env.ts e alterar a flag videoTranslator para **true**.

V. Manual de testes e preparação de ambiente

Esta seção serve para descrever os passos que são necessários para preparar e executar o projeto VLibras Cross Platform App.

Sistemas Operacionais

Para o desenvolvimento e/ou manutenção do Vlibras Cross Platform App você pode usar qualquer um dos principais Sistemas Operacionais (Windows, Linux ou MacOS).

> **Nota:** É recomendado o uso de alguma distribuição Linux pela simplicidade da instalação das dependências do VLibras Cross Platform App.

Plugins

Independentemente se a aplicação será executada no Android ou iOS, os plugins do Ionic precisam ser instalados para ocorrer o funcionamento correto e desempenhar funções nativas, como capturar um vídeo, compartilhar ou acessar um arquivo da galeria, entre outros. Abaixo, segue a lista dos plugins utilizados e a página oficial dos mesmos na documentação do Ionic, indicando instruções e melhor detalhamento de cada uma.

1. SocialSharing: Responsável por compartilhar os vídeos de traduções.
2. NativeStorage: Responsável por armazenar ou acessar dados no storage do smartphone.
3. VideoCapturePlus: Responsável por capturar vídeos e salvá-los na galeria.
4. VideoEditor: Responsável pela criação de thumbnails e consulta de metadados dos vídeos.
5. File: Responsável por acesso e manuseio de arquivos.

Versão Android

Esta seção apresentará os passos necessários para testagem via Android Studio(android emulator) e mobile device.

Pré-requisitos

Antes de executar o projeto VLibras Cross Platform App precisamos cumprir alguns pré-requisitos, são eles:

- Yarn (1.22.10) - O Yarn é um gerenciador de pacotes para aplicar comandos prontos ao código de uma aplicação.
- Ionic (6.16.3) - O Ionic é um kit de ferramentas móveis de código aberto para a construção de experiências de aplicativos da web e nativas de plataforma cruzada de alta qualidade.
- NodeJS(14.16.0) - Node.js é um software de código aberto, multiplataforma, que executa códigos JavaScript no backend/servidor e frontend/interface.
- Download Android Studio(4.1) - Para prosseguimento é necessário baixar o Android Studio ou ter um celular compatível com o sistema Android. Inicialmente, veremos os passos para baixarmos e instalarmos o Android Studio.

Instalando

***OBS:** *Lembre-se de clonar o repositório VLibras Cross Platform App*;

Dentro da pasta de seu repositório via terminal da IDE de sua preferência, basta instalar o app com o comando:

$ npm install ou **$ yarn install**

***OBS:** *Utilize o package manager de sua preferência*

Execução via Android Emulator

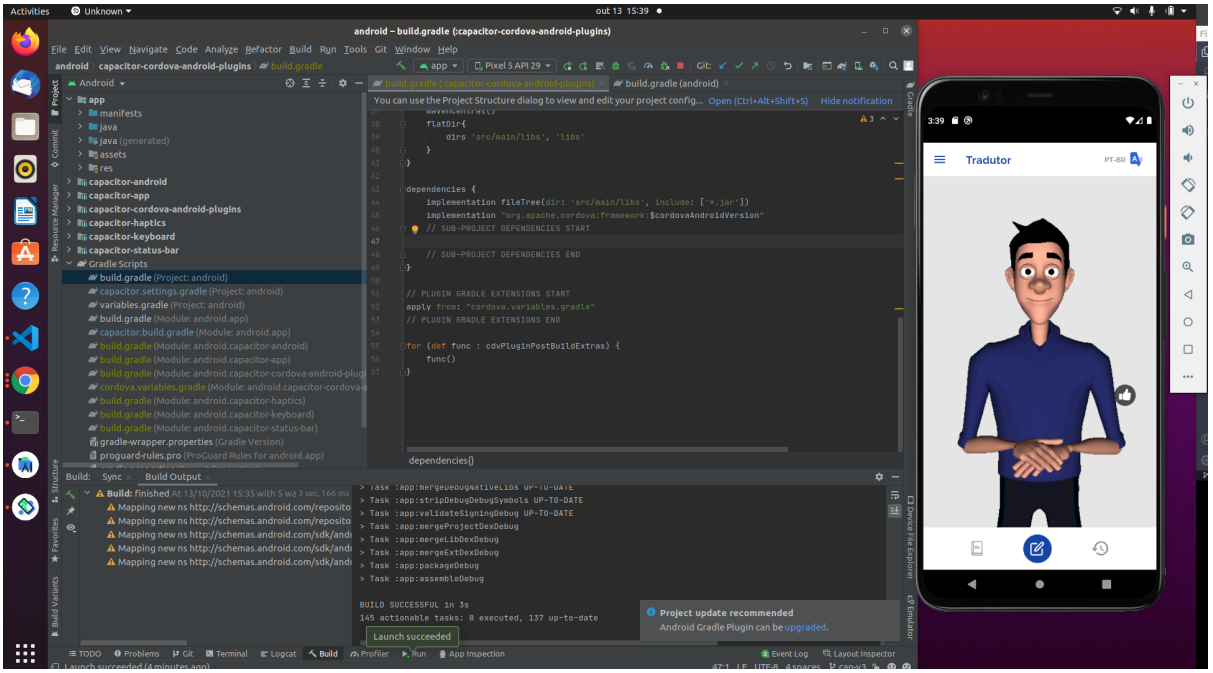
Para execução no **emulador** do Android Studio, basta seguir os passos recomendados no site do android developer: Link.

Após a configuração ter sido feita, basta executar os comandos do ionic capacitor:

$ ionic capacitor run android

Dica: Verifique este link para maiores dúvidas.

Dentro do Android Studio, o projeto será buildado. A partir daí, basta escolher o emulador para executá-lo conforme imagem abaixo:



Execução via Mobile Device(hardware)

Para execução em um mobile device(android), basta conectá-lo ao seu notebook/computador e configurar conforme descrito neste link.

Após devida configuração, basta executar os comandos do ionic capacitor:

$ ionic capacitor run android

Selecionando o device que está conectado em sua máquina, seja apresentado para ti via terminal ou pelo próprio Android Studio.

Dica: Verifique o link para maiores dúvidas.

- **Execução da aplicação com o tradutor de vídeo ativado:**

Quando executa-se a aplicação com a flag do tradutor de vídeo ativada, são utilizadas bastantes features nativas. Por isso, são necessários passos a mais para conseguir executar e testar a aplicação devidamente via device. Alguns deles são:

1) Ir em node_modules/cordova-plugin-video-capture-plus/plugin.xml e comentar o bloco de código abaixo. Esse processo é necessário devido a um bug da lib ao dar build no projeto no Android Studio.

```
<provider android:name="android.support.v4.content.FileProvider"
android:authorities="nl.x-services.plugins.videocaptureplus.provide
r" android:exported="false" android:grantUriPermissions="true">
<meta-data
android:name="android.support.FILE_PROVIDER_PATHS"
android:resource="@xml/provider_paths"/>

</provider>
```

2) Para conseguir testar em dev apontando para um endpoint em http, é necessário adicionar duas linhas no <application> do AndroidManifest.xml da build Android. Também é necessário habilitar uma flag de external storage, a fim de conseguirmos ter acesso aos arquivos de mídia capturados do usuário. Dessa maneira, segue abaixo:

```
<application
...
android:usesCleartextTraffic="true"
tools:targetApi="28"
android:requestLegacyExternalStorage="true">
```

Versão iOS

Esta seção apresentará os passos necessários para testagem via Xcode no emulador.

Pré-requisitos

Antes de executar o projeto VLibras Cross Platform App precisamos de alguns pré-requisitos, são eles:

- Yarn (1.22.10) - O Yarn é um gerenciador de pacotes para aplicar comandos prontos ao código de uma aplicação.
- Ionic (6.16.3) - O Ionic é um kit de ferramentas móveis de código aberto para a construção de experiências de aplicativos da web e nativas de plataforma cruzada de alta qualidade.
- NodeJS (14.16.0) - Node.js é um software de código aberto, multiplataforma, que executa códigos JavaScript no backend/servidor e frontend/interface.
- Xcode (12.5) - Para prosseguimento é necessário baixar o Xcode disponível apenas para o sistema operacional macOS.

Instalando

***OBS:** *Lembre-se de clonar o repositório VLibras Cross Platform App*;

Dentro da pasta de seu repositório via terminal da IDE de sua preferência, basta instalar o app primeiramente:

$ npm install ou **$ yarn install**

***OBS:** *Utilize o package manager de sua preferência*

Execução via Simulator

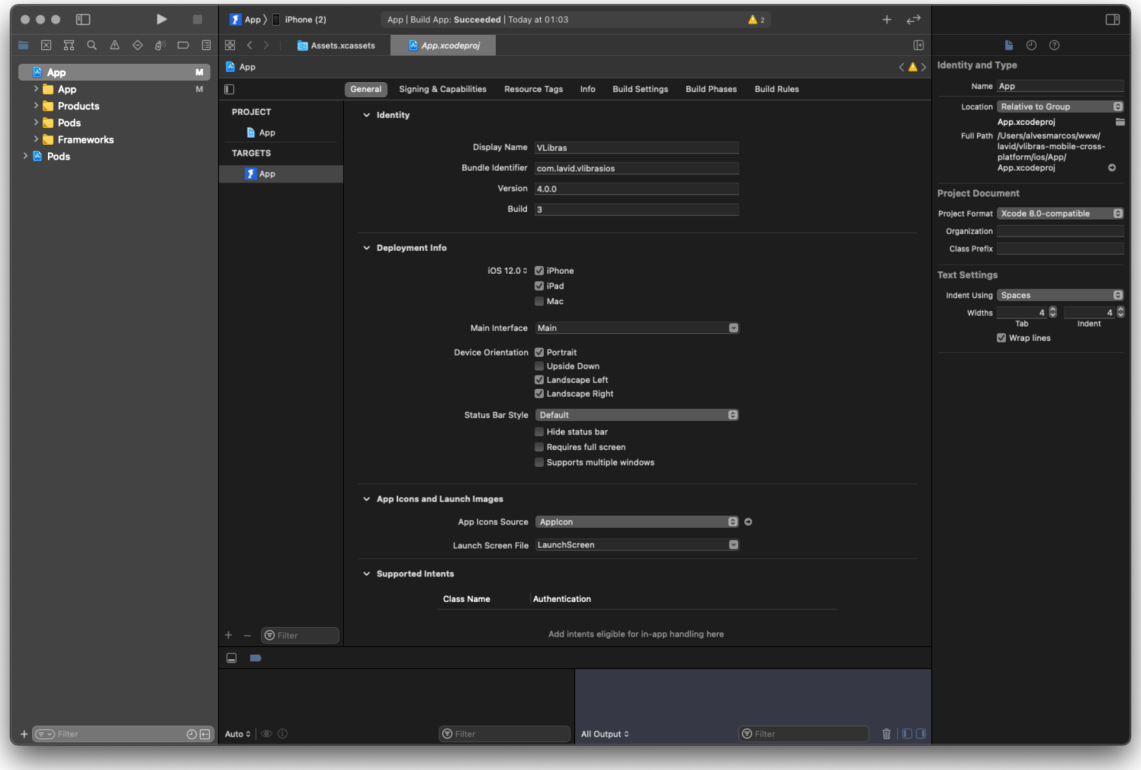
Para execução no **emulador** do Xcode, basta seguir os passos recomendados do próprio site do android developer: link

após devida configuração, basta executar os comandos do ionic capacitor:

$ ionic capacitor run ios

Dica: Verifique o link para maiores dúvidas.

Dentro do próprio Xcode, o projeto será buildado e basta escolher o emulador para executá-lo conforme imagem abaixo:



Versão Web

É necessário ter um navegador instalado para testar o VLibras Cross Platform App em seu computador.

> **Nota:** É recomendado o Google Chrome ou Mozilla Firefox.

Pré-requisitos

Antes de executar o projeto VLibras Cross Platform App precisamos de alguns pré-requisitos, são eles:

- Yarn(1.22.10) - O Yarn é um gerenciador de pacotes para aplicar comandos prontos ao código de uma aplicação.
- Ionic(6.16.3) - O Ionic é um kit de ferramentas móveis de código aberto para a construção de experiências de aplicativos da web e nativas de plataforma cruzada de alta qualidade.
- NodeJS(14.16.0) - Node.js é um software de código aberto, multiplataforma, que executa códigos JavaScript no backend/servidor e frontend/interface.

Instalando

Após ter feito a clonagem do repositório VLibras Cross Platform App. Siga para o diretório do projeto em sua máquina e insira no cmd:

```
cd vlibras-mobile-cross-platform
```

Em seguida instale todas as dependências do projeto com o comando:

```
yarn install
```

ou simplesmente

```
yarn
```

Se tudo foi executado corretamente você já está apto para executar o VlibrasMovel.

Realização de testes

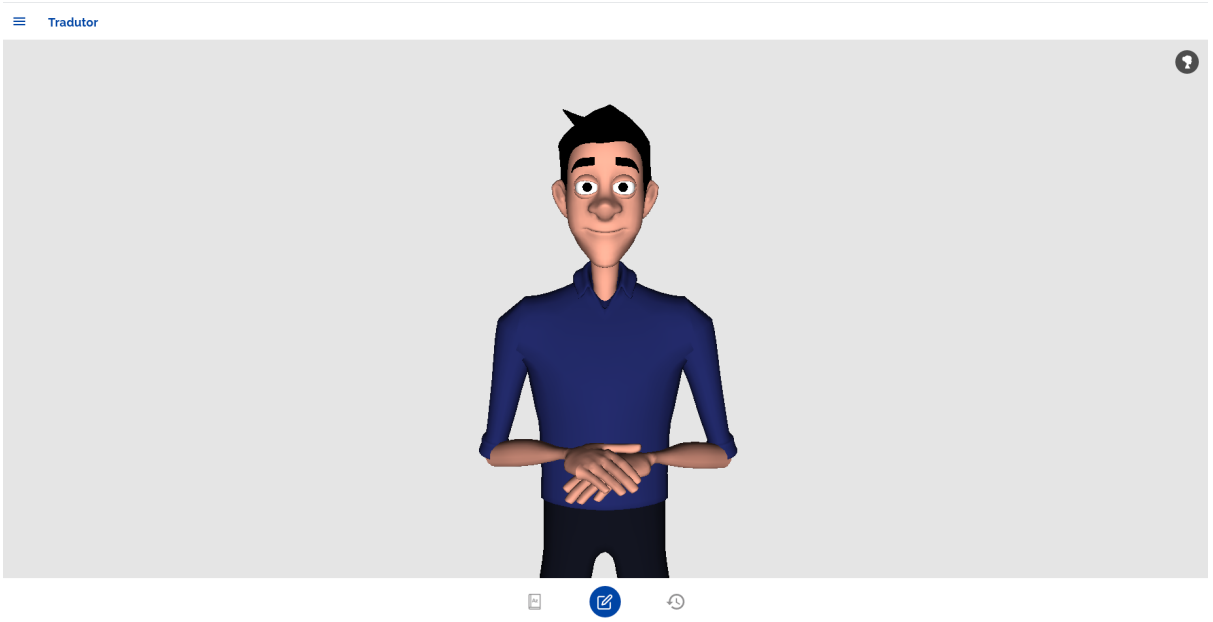
Agora que todas as dependências foram instaladas, basta usar um editor de códigos de sua preferência e realizar as alterações desejadas. Para executar o aplicativo em um navegador web e acompanhar as mudanças feitas em tempo real é necessário ir ao terminal da sua máquina e digitar:

```
cd vlibras-mobile-cross-platform
```

E em seguida abrir o servidor ionic utilizando o comando:

```
ionic serve
```

Logo após seguir os passos anteriores irá abrir em seu navegador uma aba com aplicação:



Nota: Para que funcione perfeitamente, fique atento que é necessário já ter instalado com sucesso um navegador, o Ionic e o Nodejs.

Dica: Para que se simule os tamanhos de um smartphone, no Google Chrome aperte Ctrl + Shift + J.

Equipe:

Marcos Alves

Suanny Fabyne da Silva Vieira

Caio Carvalho

Pedro Lucas Santos Silva