

Tutorial de Instalação

Este documento tem o objetivo de apresentar a lista de softwares e pacotes para a instalação e a configuração do ambiente de desenvolvimento para a Disciplina de Programação Móvel, com foco na abordagem híbrida de desenvolvimento e na plataforma nativa Android (devido ao número de dispositivos Android em sala de aula).

Os principais conceitos referentes aos softwares e aos pacotes / bibliotecas foram explicados na Aula 04 (18/08/2016). Se necessário, o aluno pode revisar os conceitos teóricos nos arquivos desta aula - disponibilizados no Moodle.

Observações Importantes

1. O tutorial é genérico para cobrir questões multiplataforma (Mac, Linux e Win);
2. Particularidades do computador do aluno e da plataforma devem ser buscadas no suporte e nos tutoriais de instalação do site de cada software ou pacote.
3. Por exemplo, no site de cada software há informações específicas de cada plataforma para configurar variáveis de ambiente para terminal e linha de comando. O aluno deve verificar essas questões direto no site do produto.
4. Checar a documentação de instalação de cada software se necessário.

Softwares e Pacotes Necessários

Os softwares que são requisitos para a Plataforma Ionic e para o Ambiente de Desenvolvimento Híbrido estão listados a seguir. Os pacotes devem ser instalados na ordem informada. No site do Ionic, também existem alguns guias para te ajudar caso necessário:

- <http://ionicframework.com/docs/v2/resources/platform-setup/mac-setup.html>
- <http://ionicframework.com/docs/v2/resources/platform-setup/windows-setup.html>

Node e NPM

Site: <https://nodejs.org/en/>

Versão: Utilizar > 4.0

Node é uma tecnologia que permite executar JavaScript no lado do servidor. O Node utiliza o mecanismo [Chrome's V8 JavaScript](#), usando um mecanismo baseado em eventos e I/O não-bloqueante, sendo uma ótima tecnologia para aplicativos que requerem velocidade.

O Node oferece uma comunidade ativa que desenvolve diversas bibliotecas e pacotes de software. Estes pacotes e bibliotecas são centralizados no site <https://www.npmjs.com/> que consiste no repositório utilizado pelo **NPM** - o gerenciador de pacotes específico do Node.

Ao instalar o Node em seu computador, você será capaz de executar comandos no seu Terminal ou CMD (Windows) e, com isso, instalar pacotes Node no projeto local em que está trabalhando. É exatamente desta forma que iremos instalar o Ionic, Angular, Banco de Dados, Recursos Nativos, Componentes Visuais, entre outros. Por isso, este é o primeiro software que você deve instalar.

A instalação é simples. Basta baixar o arquivo e seguir o passo a passo. Após a instalação, abra seu Terminal ou CMD (Windows) para testar os comandos. A seguir uma imagem para ilustrar o resultado dos comandos de teste:

A screenshot of a macOS Terminal window. The title bar shows the user '1. drantunes@iMac-DRA' and the directory '~/Desktop (zsh)'. The terminal has a dark background with light green text. It shows two commands being executed: 'node -v' which returns 'v5.10.1', and 'npm -v' which returns '3.8.6'.

```
1. drantunes@iMac-DRA: ~/Desktop (zsh)
→ Desktop node -v
v5.10.1
→ Desktop npm -v
3.8.6
```

IOS

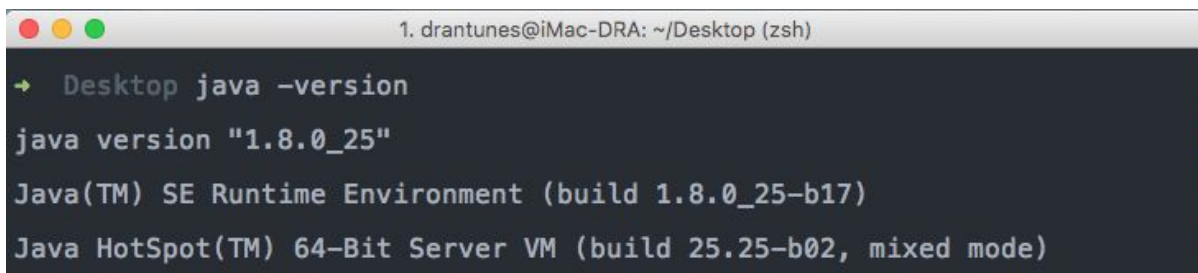
Caso você pretenda desenvolver para a plataforma IOS é necessário baixar e instalar o XCode do [portal de desenvolvedores da Apple](#). O XCode instala a IDE da Apple, o SDK para IOS, o CLI da Apple e você pode trabalhar com funções nativas.

Java JDK 8

Site: www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151

Versão: Utilizar ≥ 1.8

Caso você já utilize a plataforma Java ou tenha instalado alguma IDE tal como o Netbeans, seu Java provavelmente já estará configurado para você. Adicionalmente é importante configurar variáveis de ambiente para você poder executar o comando **java** ou **javac** em seu terminal ou CMD (Windows). Você pode testar com:

A screenshot of a terminal window on a Mac. The title bar shows the user '1. drantunes@iMac-DRA' and the path '~/Desktop (zsh)'. The terminal has a dark background with light green text. The command 'Desktop java -version' has been entered and executed. The output shows 'java version "1.8.0_25"', 'Java(TM) SE Runtime Environment (build 1.8.0_25-b17)', and 'Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)'.

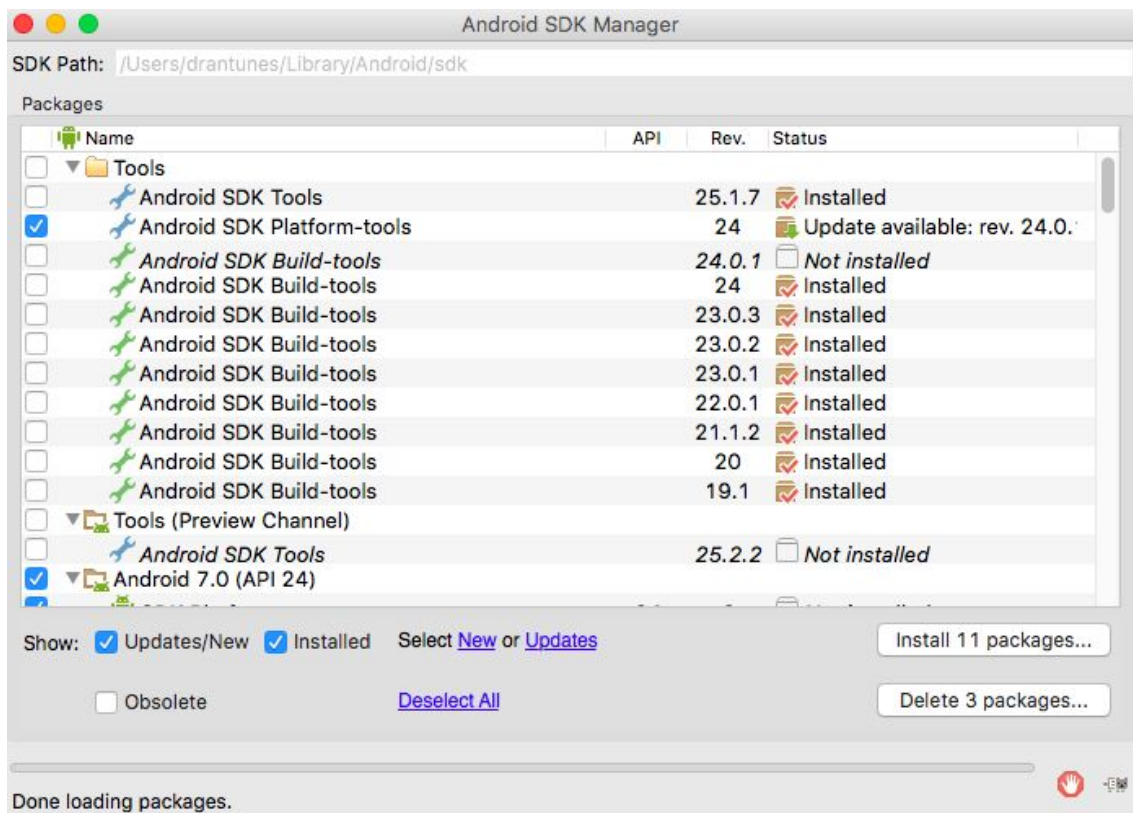
Android Studio e Android SDK

Site: <https://developer.android.com/studio/install.html?pkg=tools>

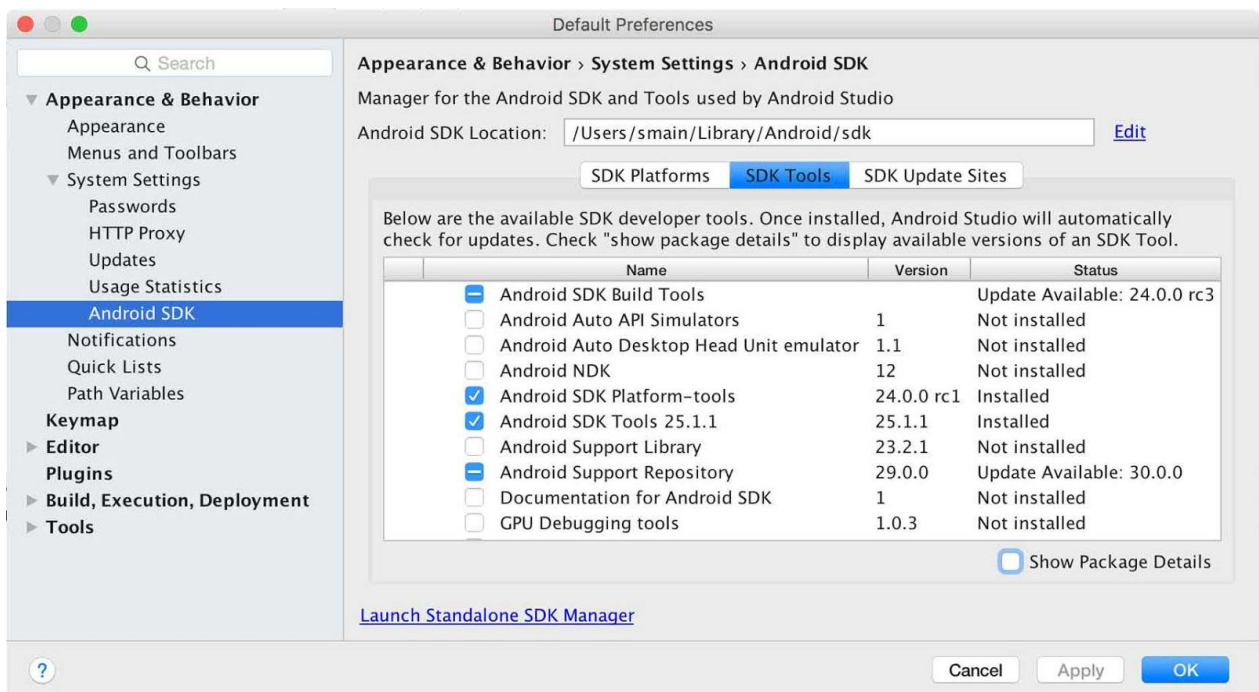
Dica: Neste link você pode escolher qual a plataforma deseja instalar. Assim você obtém dicas e recomendações para a sua plataforma: Windows, Mac ou Linux.

Após você instalar o Android Studio, você deve ir nos arquivos instalados e abrir a Ferramenta de Gerenciamento do SDK. Para acessar o [Painel do SDK via Android Studio](#), você deve abrir o Android Studio:

1. Abra a janela Preferences clicando em **File > Settings** (no Mac, **Android Studio > Preferences**) OU clique em **Tools > Android > SDK Manager**;



OU



Então você deve marcar nos ícones e instalar os seguintes pacotes:

- Android Platform SDK para sua versão alvo do Android (ex: Versão 6);
- Android Platform-Tools;
- Android SDK build-tools version 19.1.0 or higher
- Android Support Repository (No tópico “Extras”)

Em seguida, você deve configurar suas variáveis de ambiente para funcionamento dos comandos android e java. Para Mac e Linux:

```
# Create a JAVA_HOME Variable, use, the name of the JDK folder
export JAVA_HOME=/Library/Java/JavaVirtualMachines/{SUA_VERSAO_JAVA}/Contents/Home
# Add that to the global PATH variable
export PATH=${JAVA_HOME}/bin:$PATH
# Add the Android SDK to the PATH variable
export PATH=${PATH}:~/Library/Android/sdk/tools:~/Library/Android/sdk/platform-tools
```

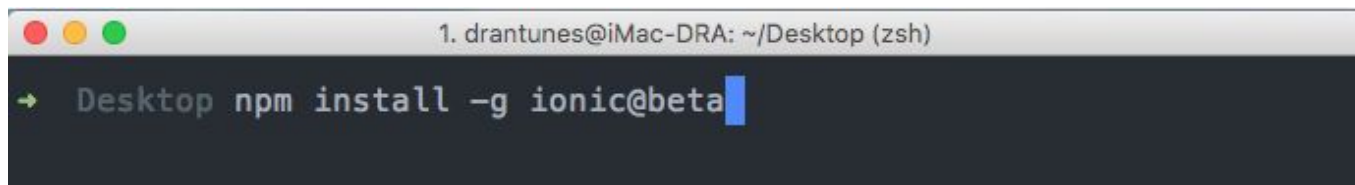
Para Windows:

No menu iniciar, pesquise por “**System Environment Variables**” ou “Variáveis de Ambiente”. Você irá atualizar a variável PATH criando uma nova variável chamada **JAVA_HOME** (Caminho onde o Java está instalado) e **PATH** que terá dois caminhos (um para as ferramentas instaladas pelo Android SDK e outra para a Platform Tools).

Ionic

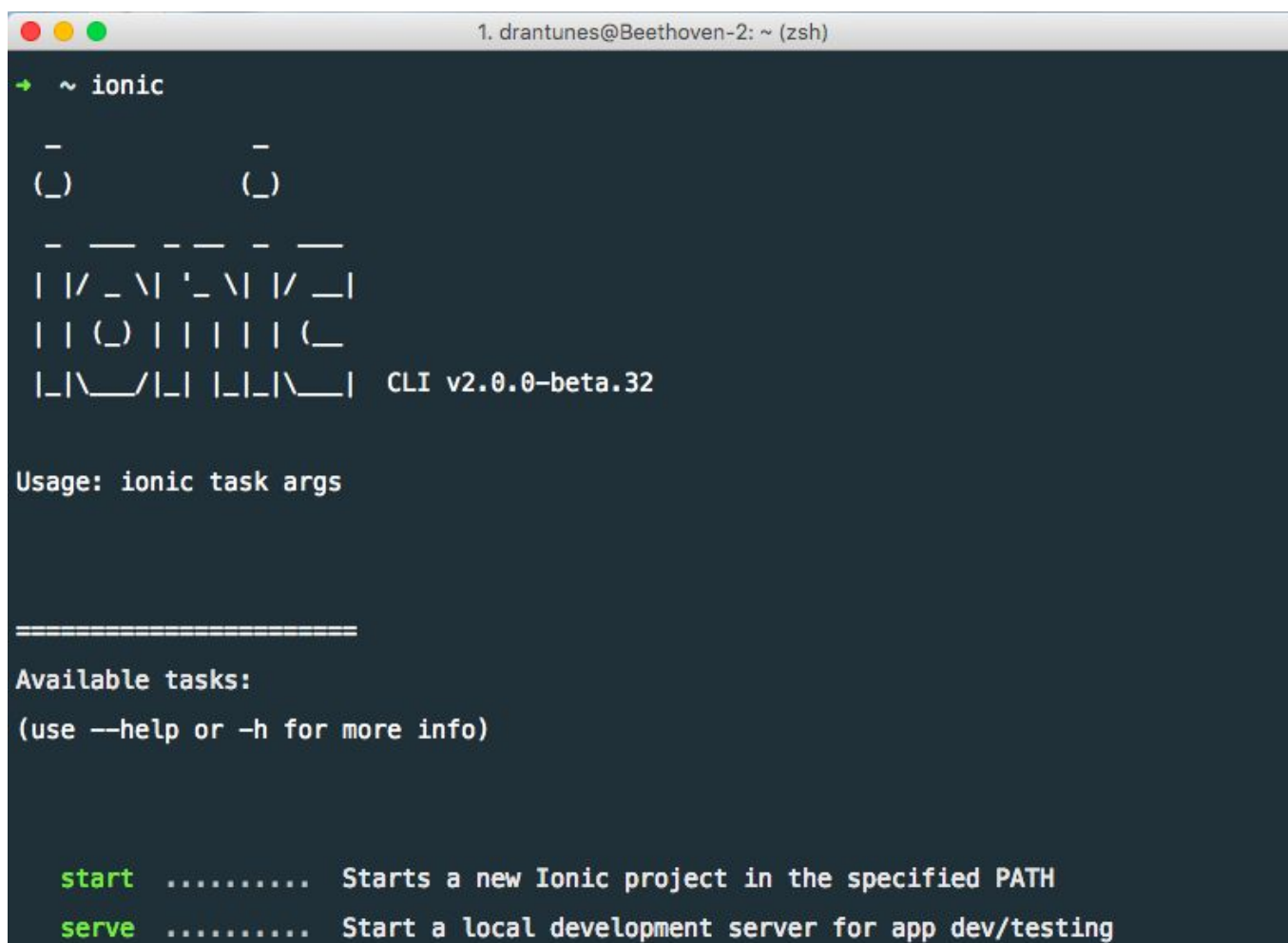
Site: <http://ionicframework.com/docs/v2/getting-started/installation/>

Vamos instalar agora os pacotes e o CLI do Ionic. Como vamos utilizar a versão beta do Ionic 2, vamos utilizar o seguinte comando para instalar via Terminal ou CMD (Windows).



```
1. drantunes@iMac-DRA: ~/Desktop (zsh)
→ Desktop npm install -g ionic@beta
```

Após instalar o pacote node do Ionic, você pode rodar um teste no Terminal ou CMD (windows) para verificar se o Ionic está instalado corretamente. Ao rodar o comando **ionic**, você deve ver:



```
1. drantunes@Beethoven-2: ~ (zsh)
→ ~ ionic

  _           _
(_)_         (_)_

  _  _  _  _  _  _
| /  _ \  ' _ \ /  _ |
| | ( ) | | | | | (
| _ \ / | | | _ \ | CLI v2.0.0-beta.32

Usage: ionic task args

=====

Available tasks:
(use --help or -h for more info)

start ..... Starts a new Ionic project in the specified PATH
serve ..... Start a local development server for app dev/testing
```

Vamos criar agora um projeto de exemplo para aprendermos a usar o Ionic CLI e também para rodar o projeto e testá-lo diretamente em nosso browser.

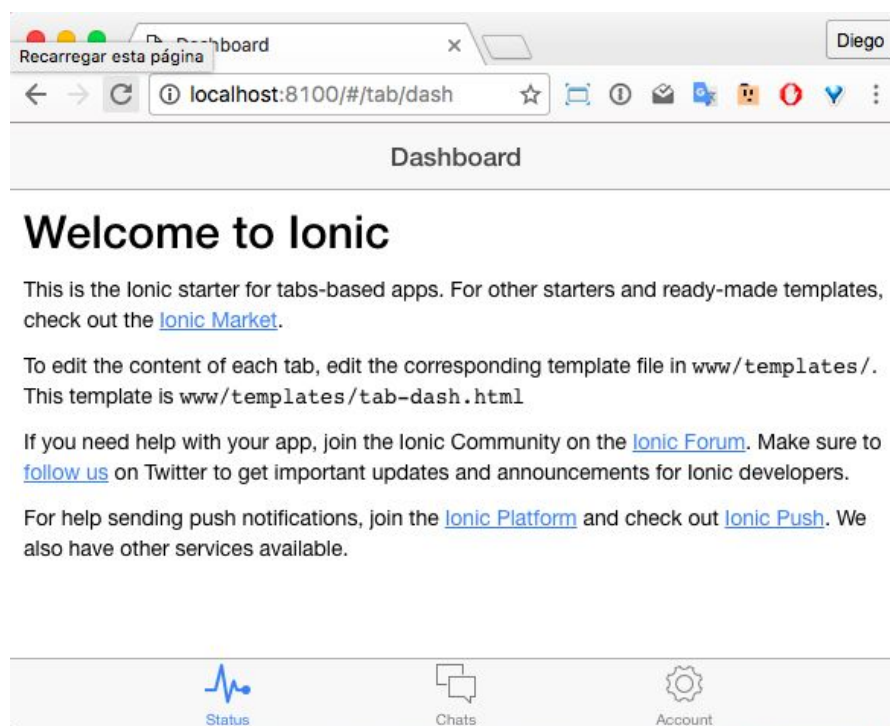
```
1. drantunes@Beethoven-2: ~/Desktop (zsh)
→ Desktop ionic start meuProjeto --v2
```

Importante: o Ionic CLI instala uma série de pacotes via NPM. Estes pacotes são instalados na pasta do seu projeto dentro de **node_modules**. Portanto, toda vez que você fizer backup ou enviar para o professor o trabalho você deve fazer sem esta pasta, pois os pacotes ocuparão muito espaço desnecessário. Para instalar novamente os pacotes, dentro da pasta do projeto, você pode rodar o comando **npm install** no Terminal ou CMD (Windows).

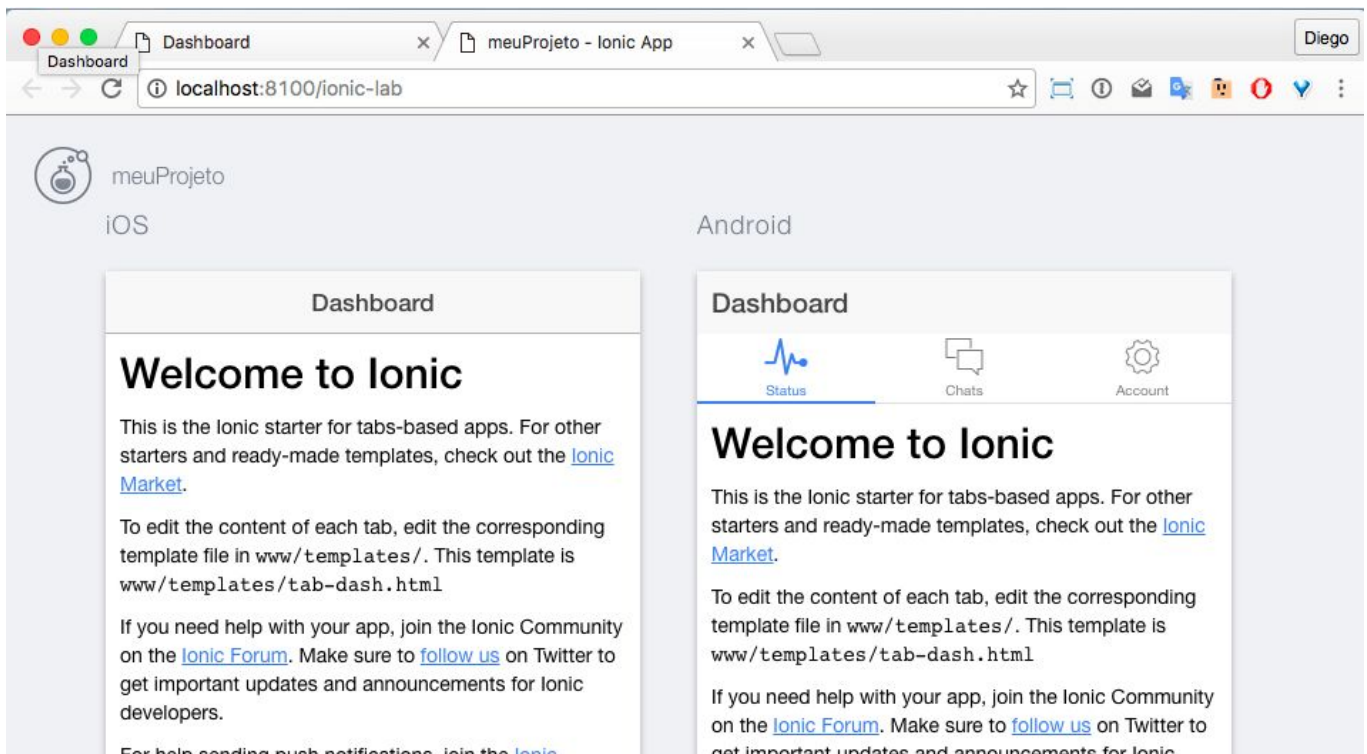
Para testar o seu projeto execute os seguintes comandos via Terminal ou CMD (Windows):

```
1. ionic serve (node)
→ Desktop cd meuProjeto
→ meuProjeto ionic serve
```

Você deve ver no navegador a seguinte tela.



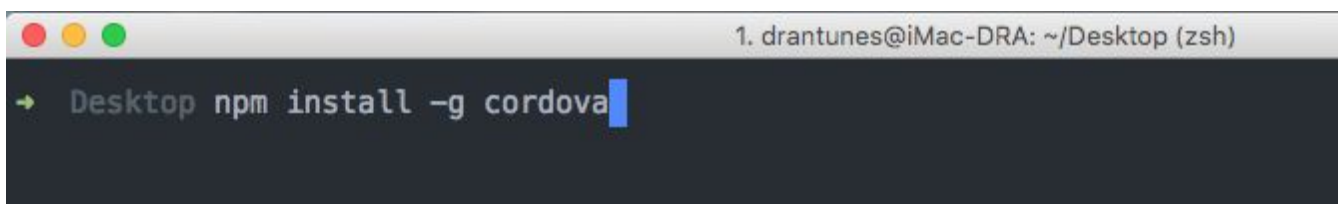
Se você rodar o comando `ionic serve --lab`, você poderá ver duas renderizações no navegador para seu projeto (iOS e Android).



Sobre o Ionic View: http://ionicframework.com/docs/cli/uploading_viewing.html

Cordova

Para ter acesso às funções nativas, é necessário instalar o Apache Cordova. Para isso, podemos rodar um comando via NPM:



Dentro do seu projeto, você deve mudar a plataforma padrão do Ionic para android. Você pode fazer isso com o seguinte comando:

A screenshot of a macOS terminal window. The title bar shows three colored window control buttons (red, yellow, green) on the left and the text '1. drantunes@iMac-DRA: ~ (zsh)' on the right. Below the title bar, there are two tabs: 'ionic (node)' with a magnifying glass icon and 'zsh' with a magnifying glass icon. The main area of the terminal is dark gray, and the text 'ionic platform add android' is typed in a light blue font, with a blue cursor at the end of the line.

Para mais informações acessar:

<http://cordova.apache.org/docs/en/latest/guide/support/index.html>

Proxy e Endereços Locais [Atualizado]

Na UTFPR é necessário configurar alguns itens para o correto funcionamento do aplicativo, uma vez que a Instituição utiliza um servidor de Proxy. Para tanto, precisamos configurar o proxy para o NPM. Ainda assim, é possível que alguns recursos, que não possuem opções para a configuração de proxy, não funcionem. É necessário também adicionar o proxy no browser (para navegação).

Estando na UTFPR você precisa executar estes comandos no Terminal ou CMD (Windows):

Comandos para o NPM:

```
npm config set https-proxy "http://10.20.10.50:3128/"
```

```
npm config set proxy "http://10.20.10.50:3128/"
```

```
npm config set registry "http://registry.npmjs.org/"
```

```
npm config set strict-ssl false
```

Comandos para o Github:

```
git config --global http.proxy "http://10.20.10.50:3128"
```

Comandos para o APM da IDE Atom (para instalar pacotes):

```
apm config set http-proxy "http://10.20.10.50:3128"
```

```
apm config set https-proxy "http://10.20.10.50:3128/"
```

```
apm config set proxy "http://10.20.10.50:3128/"
```

```
apm config set strict-ssl false
```

Proxy Geral

```
export http_proxy=http://10.20.10.50:3128
```

```
export https_proxy=http://10.20.10.50:3128
```

O CLI do Ionic também enfrenta alguns problemas em relação ao Proxy na UTFPR. Para executar corretamente o comando, você deve iniciar o proxy na frente do comando **ionic**, por exemplo:

```
PROXY=http://10.20.10.50:3128 http_proxy=http://10.20.10.50:3128 https_proxy=http://10.20.10.50:3128 ionic
```

Dica para Mac e Linux: você pode criar um alias (pesquisar no Google), como um atalho, para ligar e desligar as configurações de proxy em um único comando.

Em sua casa você deve desabilitar o proxy executando os comandos

```
npm config delete https-proxy
```

```
npm config delete proxy
```

```
git config --global http.proxy ""
```

```
apm config delete http-proxy
```

```
apm config delete https-proxy
```

```
apm config delete proxy
```

```
export https_proxy=""
```

```
export http_proxy=""
```

Adicionalmente, você deve ir em Configurações de Internet e permitir que o endereço local (LOCALHOST) passe livre pelo proxy.



Infelizmente ainda estamos tendo problemas com o comando ionic upload, para enviar os dados para o Ionic Cloud. Neste caso, vocês deverão fazer o upload de casa para compartilhar os aplicativos com o professor e colegas.

Atom IDE

Site: <https://atom.io/>

Dica: No site do Ionic há outras opções de IDE, mas o Atom oferece uma série de plugins para nos ajudar a desenvolver aplicações móveis focada em Ionic.

O Atom permite o uso de alguns plugins ou pacotes adicionais. Você pode pesquisar os pacotes no site <https://atom.io/packages>. A seguir uma lista de pacotes que podem ser interessantes para lhe ajudar no desenvolvimento (instalar por critério próprio).

- <https://atom.io/packages/atom-typescript>
- <https://atom.io/packages/autocomplete-ionic2-framework>
- <https://atom.io/packages/ionic-preview>
- <https://atom.io/packages/ionic-atom>
- <https://atom.io/packages/angular2-snippets-atom>
- <https://atom.io/packages/angular2-snippets>
- <https://atom.io/packages/angular-2-typescript-snippets>
- <https://atom.io/packages/fuzzy-finder>
- [https://atom.io/packages/linter \(erros\)](https://atom.io/packages/linter (erros))
- <https://atom.io/packages/pigments>
- <https://atom.io/packages/color-picker>
- <https://atom.io/packages/minimap>
- <https://atom.io/packages/atom-beautify>
- <https://atom.io/packages/less-than-slash>
- <https://atom.io/packages/emmet>
- <https://atom.io/packages/file-icons>
- <https://atom.io/packages/javascript-snippets>
- <https://atom.io/packages/advanced-new-file>
- <https://atom.io/packages/highlight-selected>
- <https://atom.io/packages/simple-drag-drop-text>
- <https://atom.io/packages/docblockr>
- <https://atom.io/packages/autocomplete-modules>
- <https://atom.io/themes/seti-ui>