

Curso Técnico Superior Profissional em Tecnologias e Programação de Sistemas de Informação

Unidade Curricular: Desenvolvimento Web – Front-End

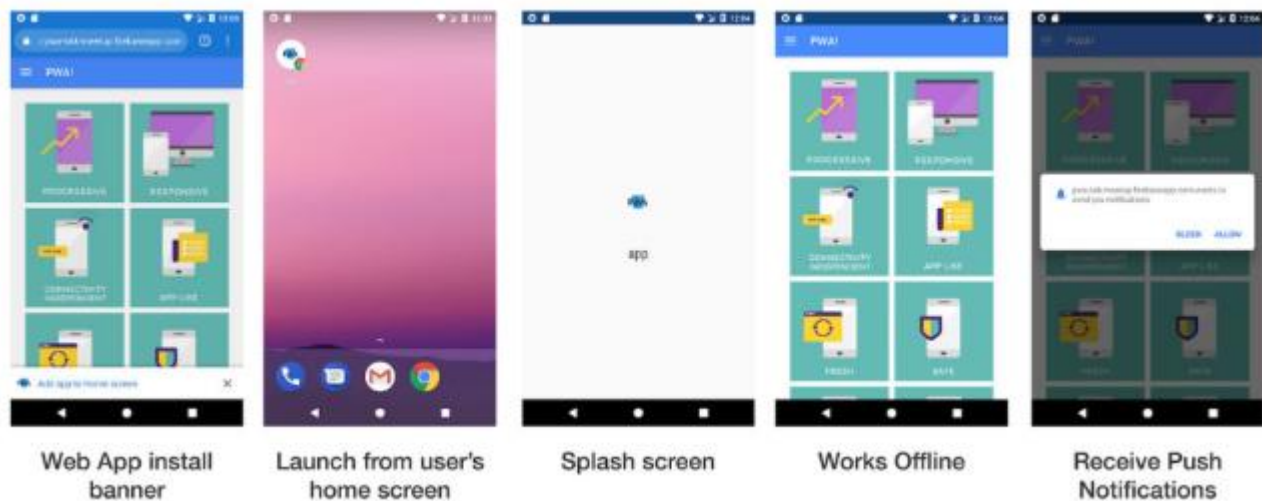
1.º Ano/2.º Semestre

Docente: Marco Miguel Olival Olim

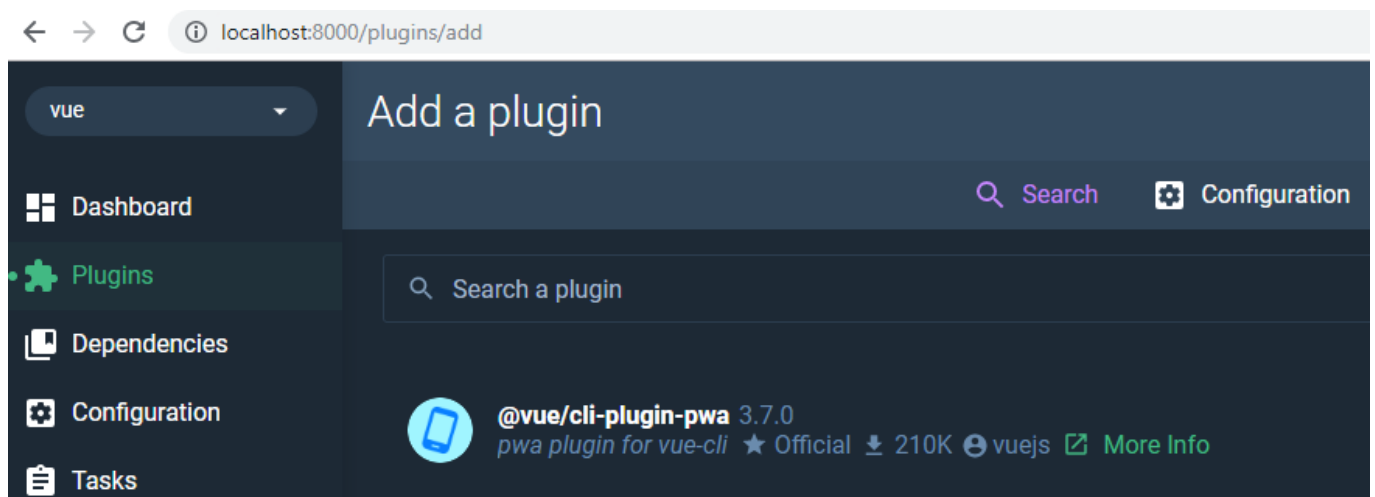
Data 10/05/2019

ESTE EXERCÍCIO INTRODUZ A ABORDAGEM COM PWA PARA A CRIAÇÃO DE APLICAÇÕES PARA DISPOSITIVOS MÓVEIS

Uma Aplicação Web Progressiva (**PWA** - *Progressive Web App*) é uma página web que dispõe de funcionalidades tradicionalmente só disponíveis em aplicações para dispositivos móveis, tais como acesso ao hardware (sensores, camera, etc), funcionamento offline, instalação no dispositivo e *push notifications*. Saliente-se que PWA não se refere a uma nova tecnologia ou framework mas apenas um conjunto de boas práticas que conduzem a uma experiência similar a uma App num telemóvel.

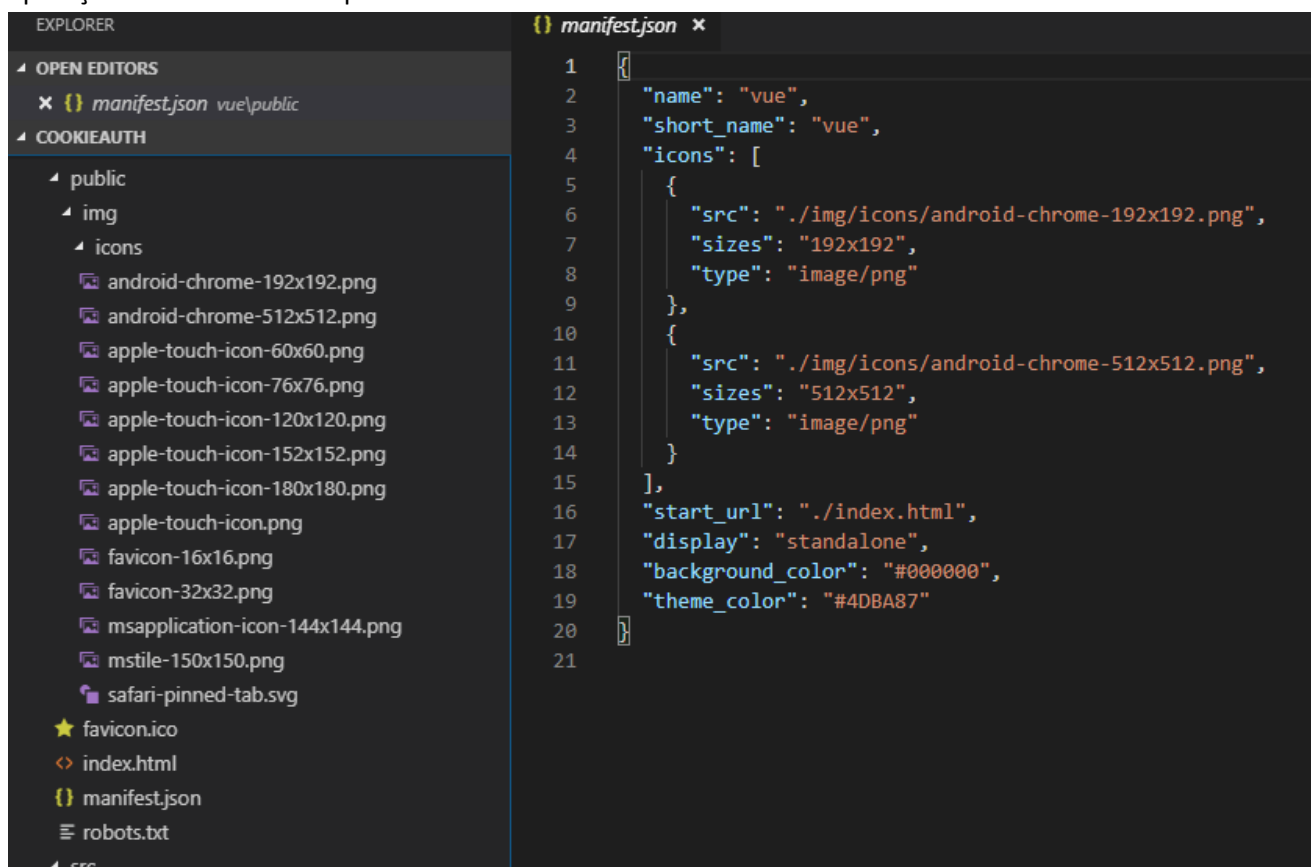


O Vue CLI já dispõe de suporte para PWA, podendo ser instalado como plugin no Vue UI em qualquer fase do projeto, ou então pela consola com **npm install @vue/cli-plugin-pwa**

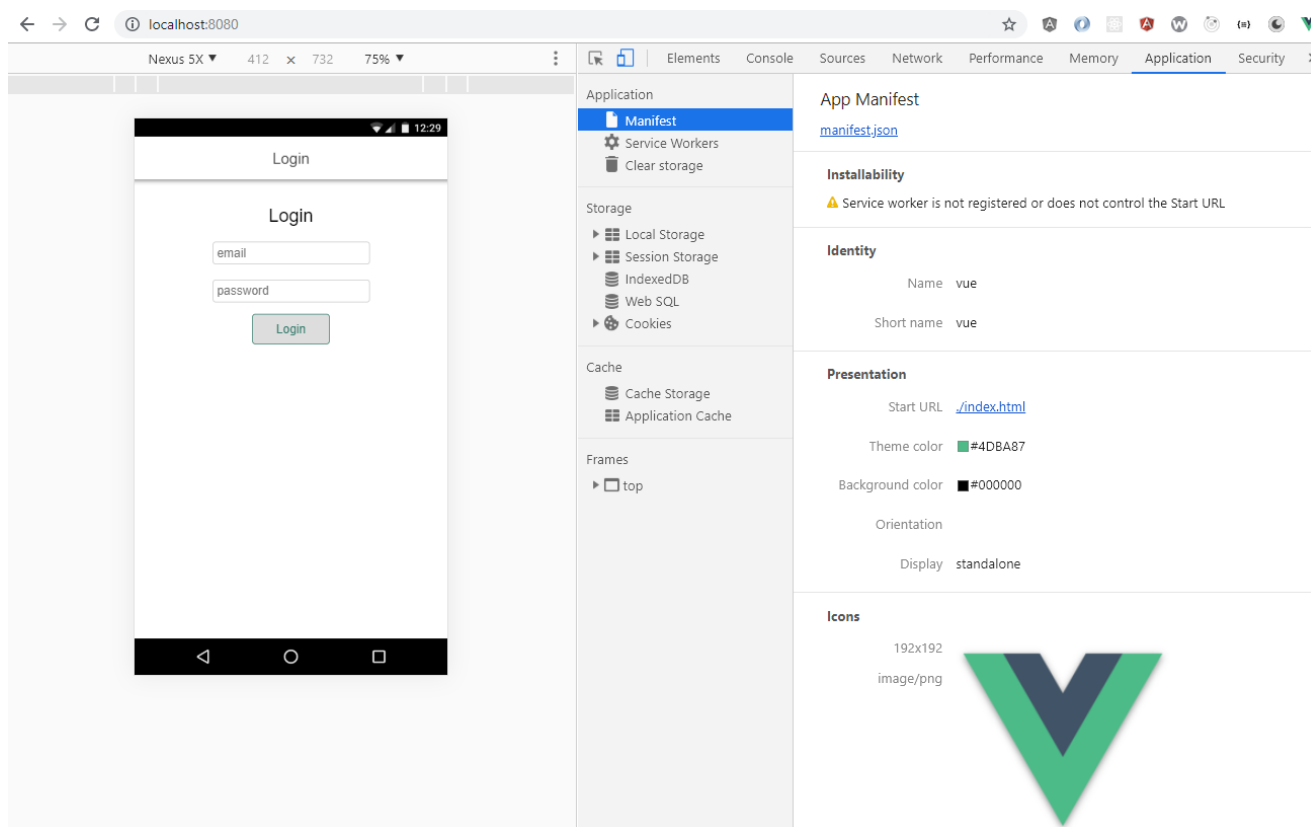


Cofinanciado por:

Depois da instalação do plugin de PWA é criado na pasta public um conjunto de ícones adaptados às dimensões dos diversos dispositivos móveis e um Web App Manifest que define alguma informação sobre a aplicação da mesma forma que o **manifest** em Android



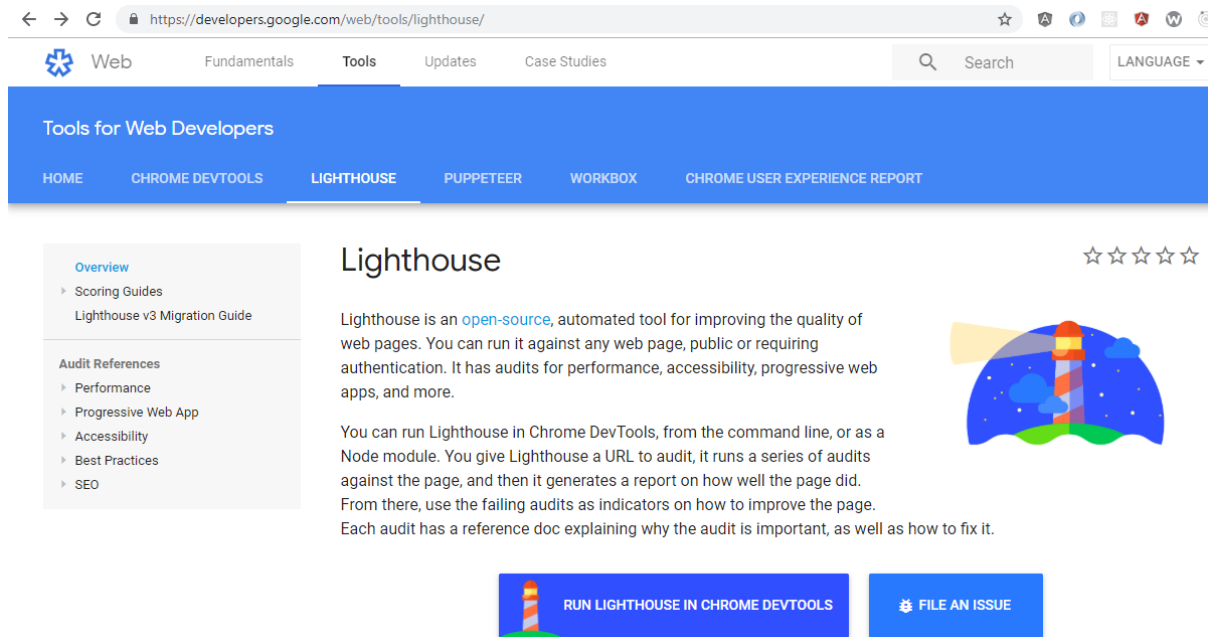
Confira agora no browser (em Application) se a PWA já está registado em no localhost:8080. Saliente-se que os *service workers* não funcionam em desenvolvimento.



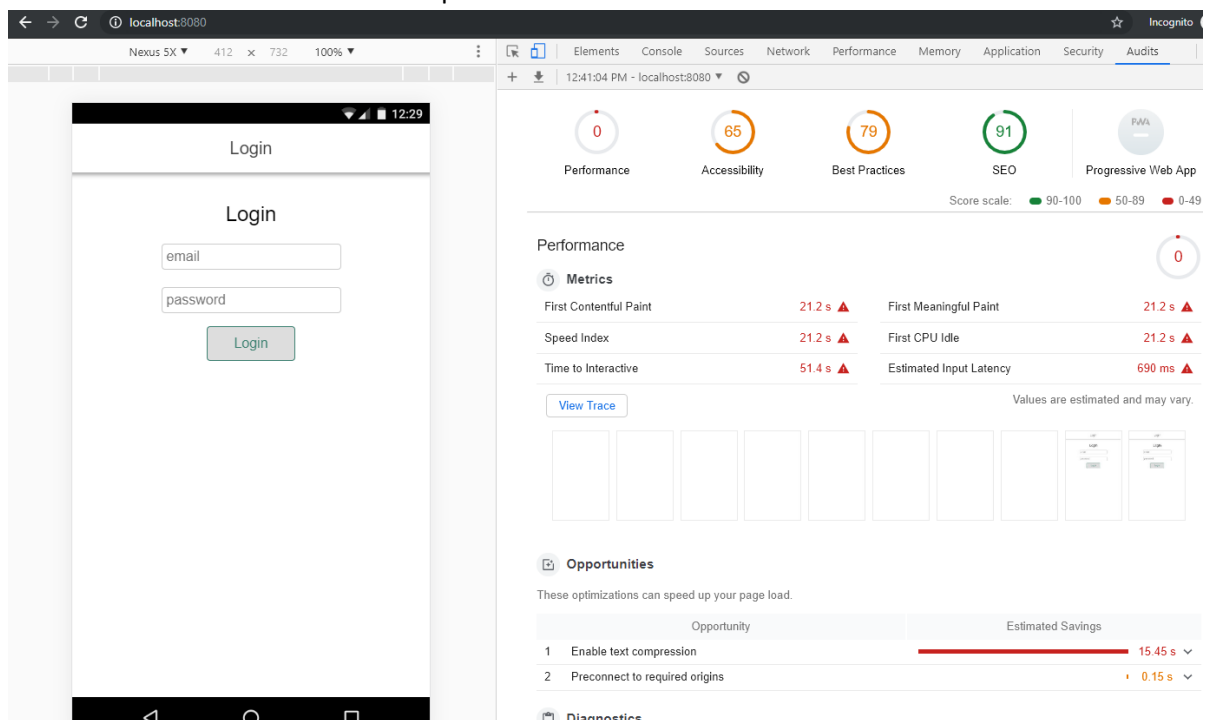
Cofinanciado por:



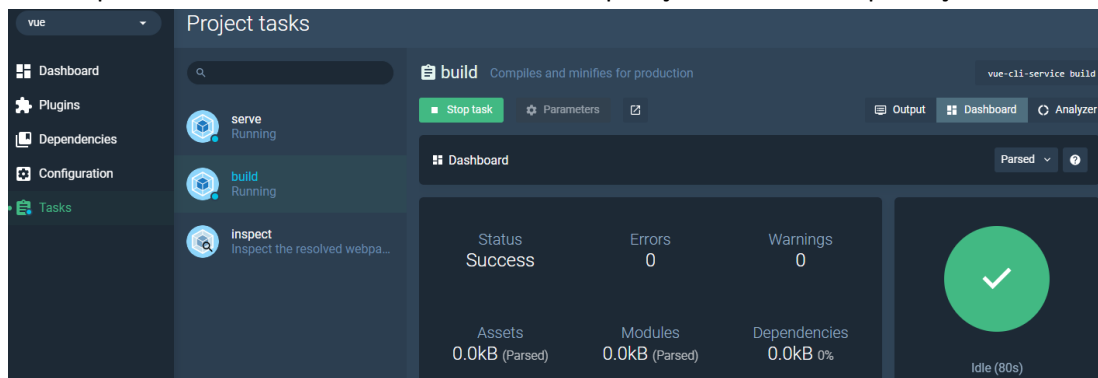
Vamos agora aferir a performance desta PWA utilizando uma ferramenta de auditoria da Google: o Lighthouse. Podemos utilizá-la no Chrome devtools ou instalando o cli com **npm install -g lighthouse** e executando para um relatório local em html com **lighthouse <http://localhost:8080> --view**



Neste momento a auditoria retorna péssimos resultados:



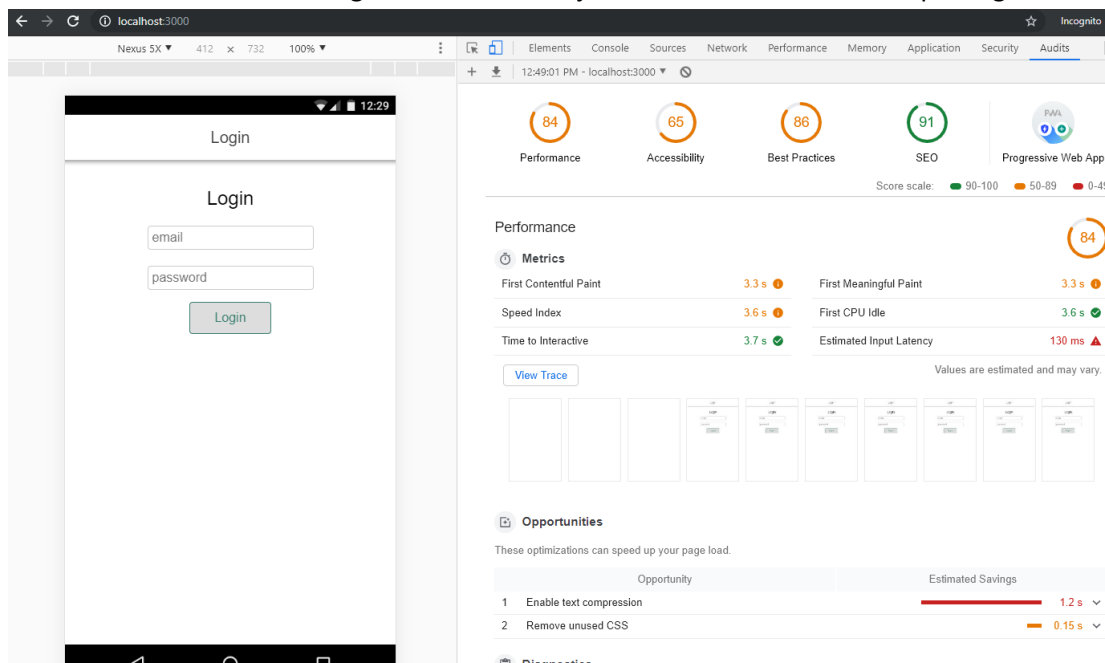
Vamos proceder ao build e auditar novamente a aplicação na versão de produção no **localhost:3000**



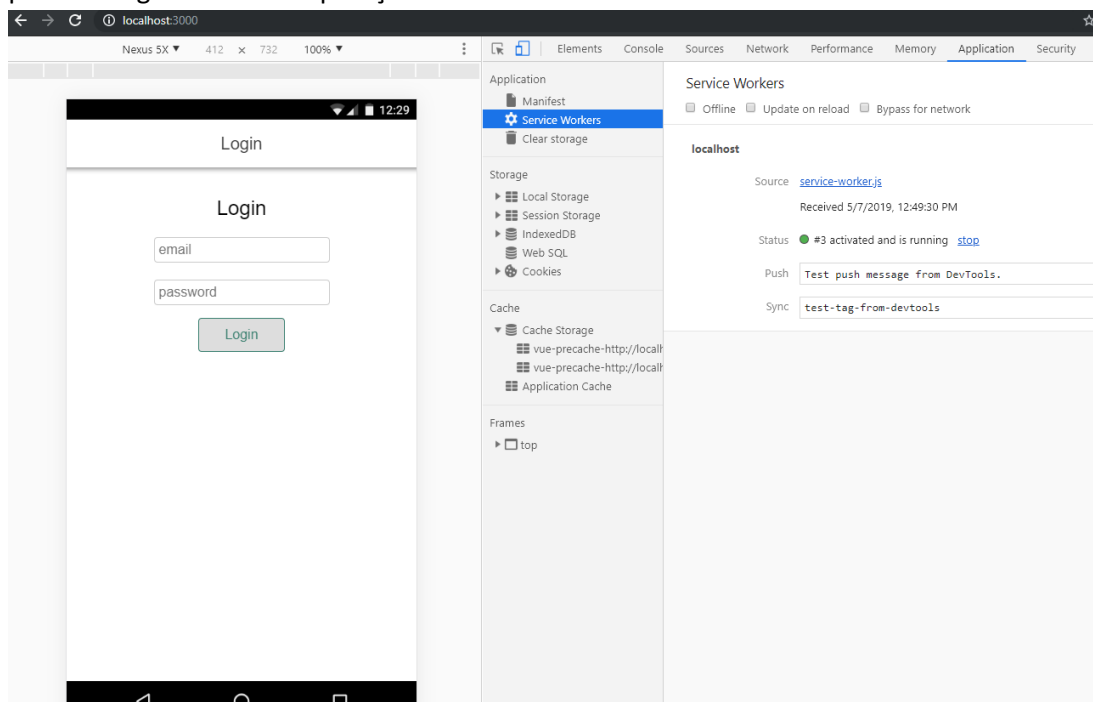
Cofinanciado por:



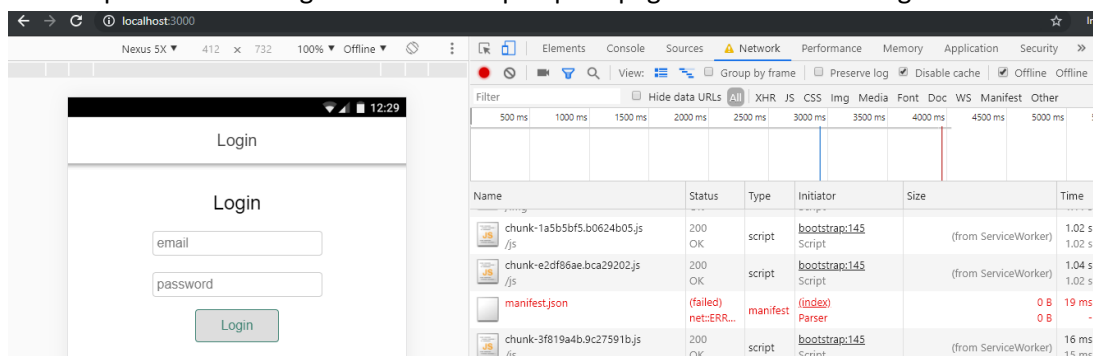
Os resultados melhoraram significativamente e já é reconhecido como PWA pelo lighthouse



O *service worker* já está registado. Este em particular permite que seja efetuada cache dos conteúdos, podendo agora utilizar a aplicação no browser mesmo sem acesso à internet



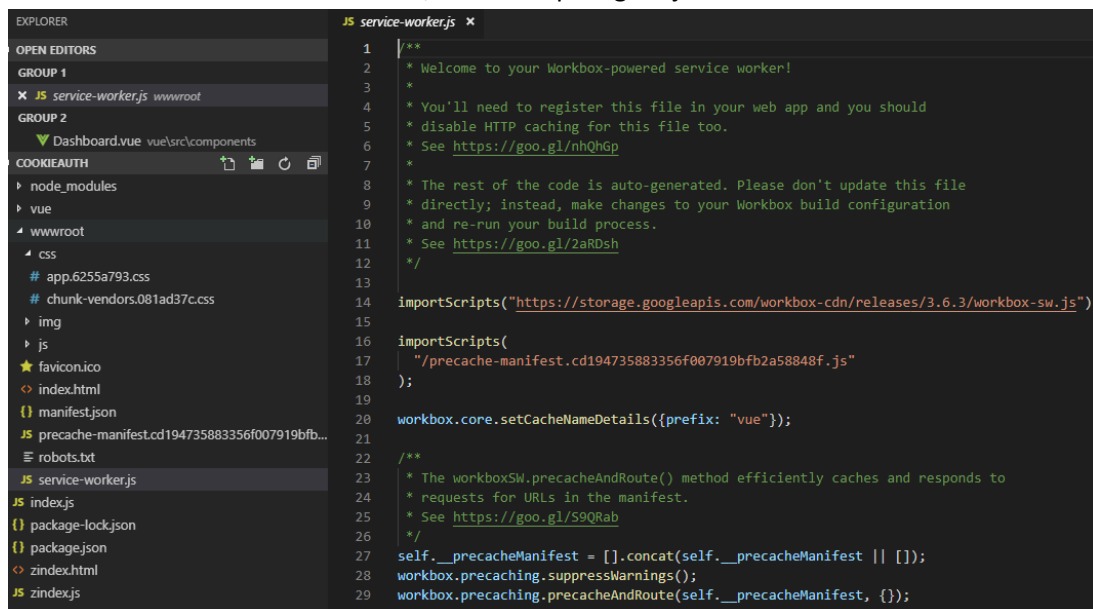
Pode experimentar desligar a net e verifique que a página continua a carregar



Cofinanciado por:

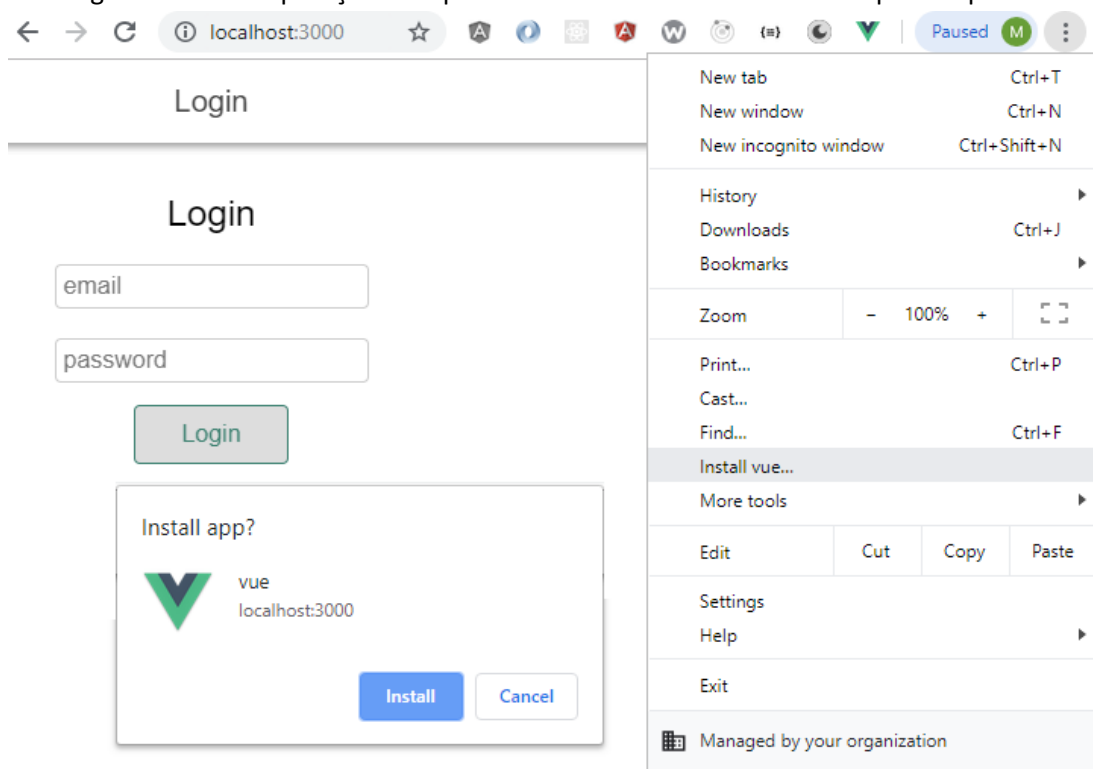


Se verificar os ficheiros no servidor, constata que agora já lá existe um *service worker*

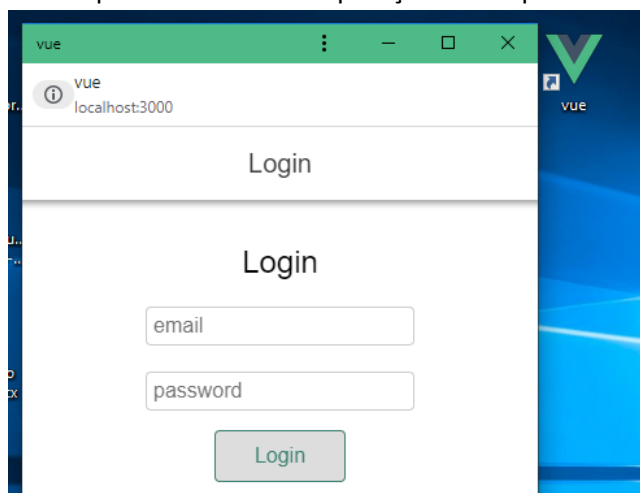


```
1 /**
2  * Welcome to your Workbox-powered service worker!
3  *
4  * You'll need to register this file in your web app and you should
5  * disable HTTP caching for this file too.
6  * See https://goo.gl/nhQhGp
7  *
8  * The rest of the code is auto-generated. Please don't update this file
9  * directly; instead, make changes to your Workbox build configuration
10 * and re-run your build process.
11 * See https://goo.gl/2aRDsh
12 */
13
14 importScripts("https://storage.googleapis.com/workbox-cdn/releases/3.6.3/workbox-sw.js");
15
16 importScripts(
17   "/precache-manifest.cd194735883356f007919bfb2a58848f.js"
18 );
19
20 workbox.core.setCacheNameDetails({prefix: "vue"});
21
22 /**
23  * The workboxSW.precacheAndRoute() method efficiently caches and responds to
24  * requests for URLs in the manifest.
25  * See https://goo.gl/S9QRab
26  */
27 self.__precacheManifest = [].concat(self.__precacheManifest || []);
28 workbox.precaching.suppressWarnings();
29 workbox.precaching.precacheAndRoute(self.__precacheManifest, {});
30
```

Pode agora instalar a aplicação no a partir do chrome ou outro browser que o suporte PWA



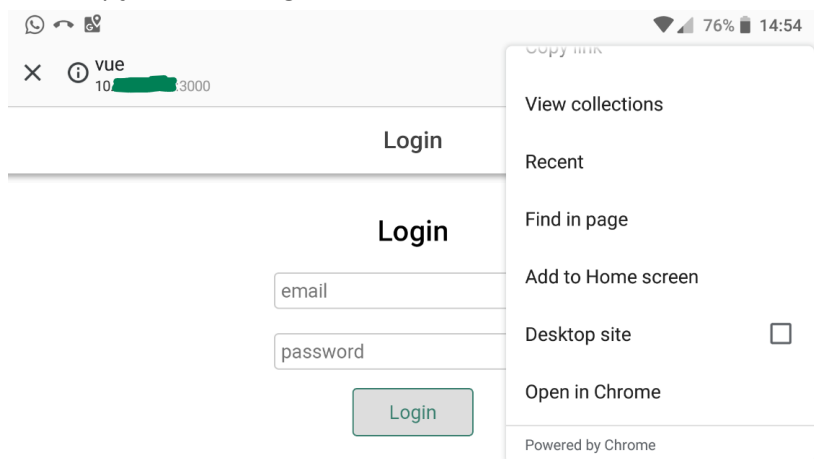
Fica disponível como uma aplicação desktop no Windows



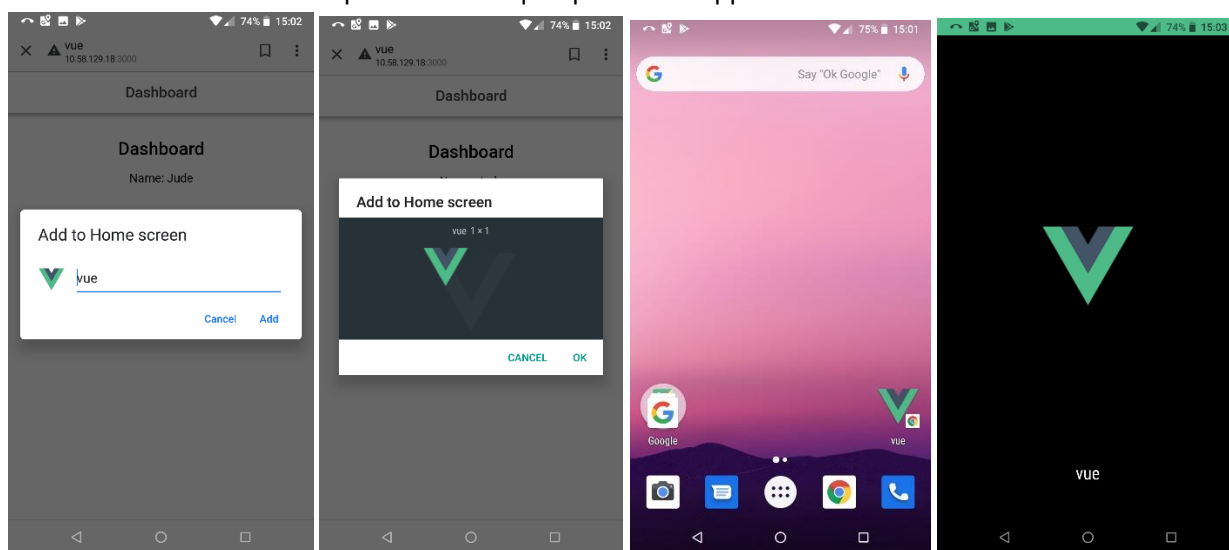
Cofinanciado por:



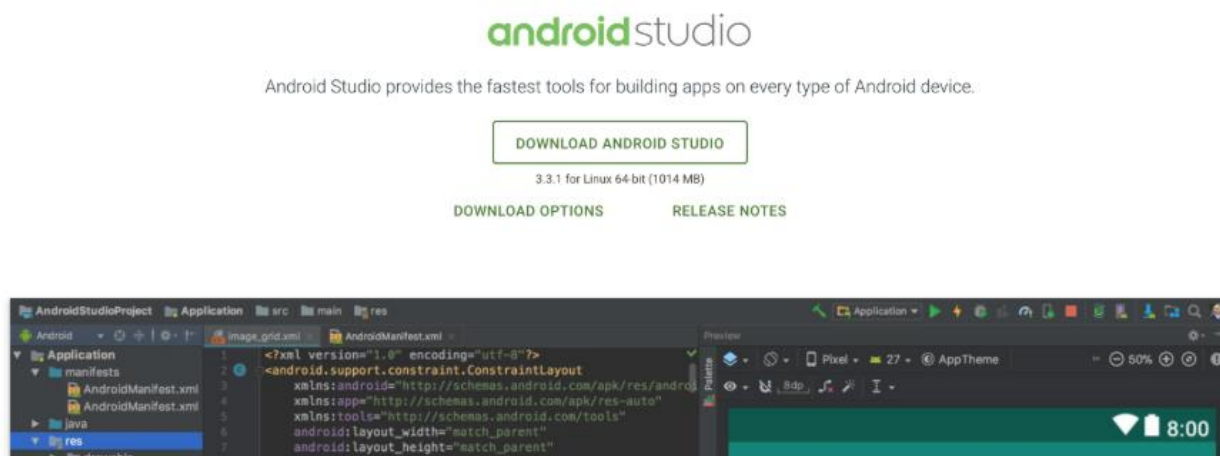
Num telemóvel também pode instalar como App. Ao abrir com o chrome no telemóvel surge um pop-up *Add to home screen*. Pode a PWA não estar ainda configurada para ativar o pop-up, mas é sempre possível aceder a esta opção nos settings do browser.



No telemóvel a PWA fica disponível como qualquer outra App nativa



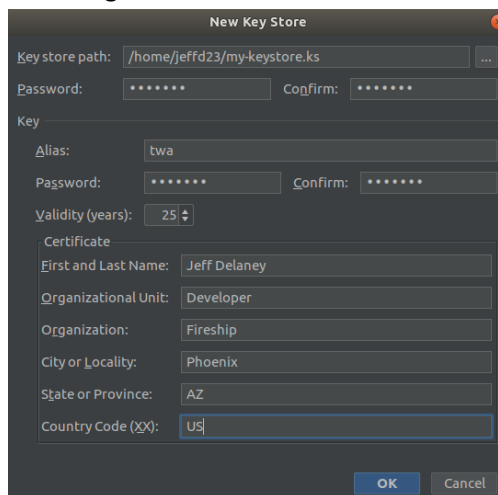
Esta App também pode ser publicada para a Play Store (desde que tenha conta de Developer), agora que a Google disponibilizou as Custom Tabs, através de **Trusted Web Activities**. Para tal necessita do **Android Studio** com a TWA Support Library. O procedimento de instalação encontra-se em <https://developers.google.com/web/updates/2019/02/using-twa>



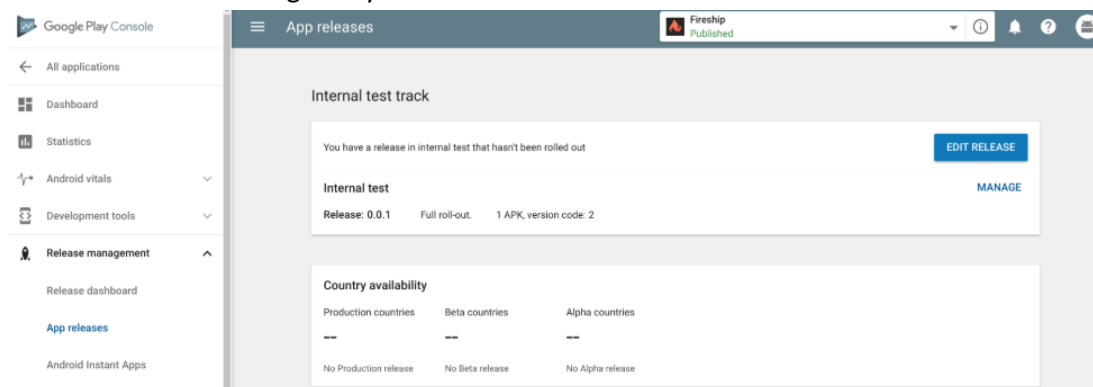
Cofinanciado por:



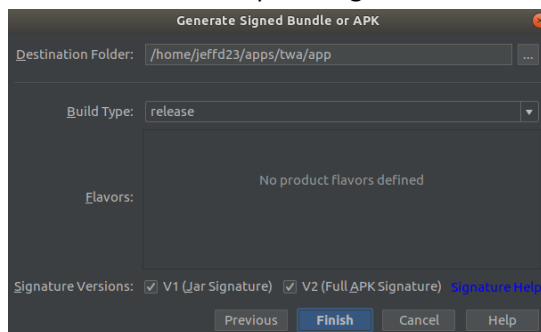
Tem de gerar uma chave SHA-256 com **keytool -list -v -keystore -alias -storepass -keypass**



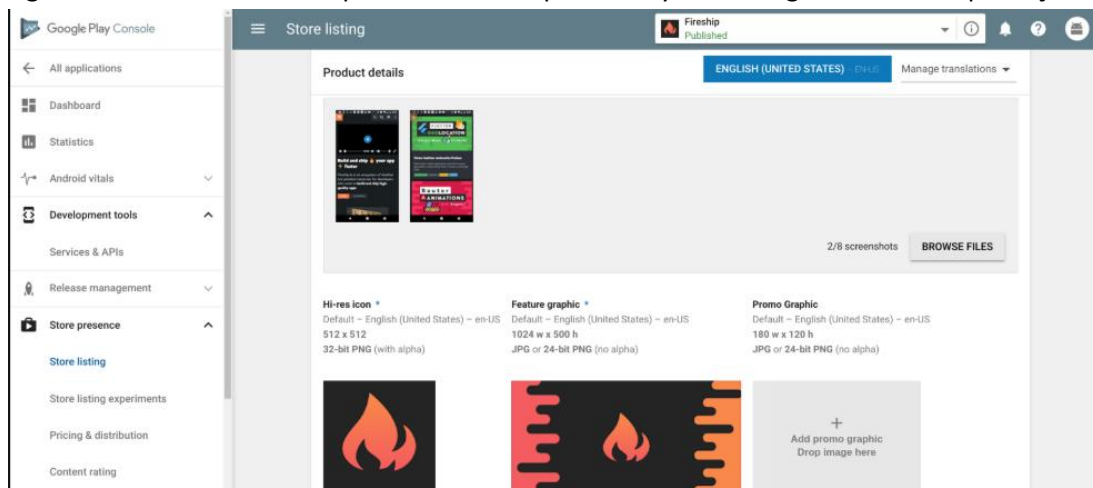
Crie um release na Google Play Console



No Android Studio é preciso gerar o APK com a assinatura criada anteriormente



Agora só tem de efetuar o upload deste APK para a Play Store e aguardar a sua aprovação



Cofinanciado por:

