PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Curso de Ciência da Computação

Douglas Magalhães de Oliveira

RA:479580

TRABALHO DE APP HÍBRIDO

POÇOS DE CALDAS

2019

SISTEMA DE CADASTRO DE PRESTADORES DE SERVIÇO

# OBJETIVO

Neste trabalho irei cadastrar pessoas que prestam serviços na modalidade de autônomo, ou seja, funcionário que cobre folgas ou férias sem vínculos empregatícios.

# PASSO A PASSO

# BANCO DE DADOS

A função save foi criada em edit-contact.ts utilizando o ionic storage do qual salva os campos nome, telefone, data de nascimento e se está ou não disponível, ou seja, esta ativo para o trabalho.

As funções seguem abaixo:

save() {

this.saveContact()

.then(() => {

this.toast.create({ message: 'Contato salvo.', duration: 3000, position: 'botton' }).present();

this.navCtrl.pop();

})

.catch(() => {

this.toast.create({ message: 'Erro ao salvar o contato.', duration: 3000, position: 'botton' }).present();

});

}

private saveContact() {

if (this.key) {

return this.contactProvider.update(this.key, this.model);

} else {

return this.contactProvider.insert(this.model);

}

}

# **RECURSO NATIVO**

O recurso nativo utilizado foi o vibration, que após preencher o formulário, ativa a vibração do aparelho utilizando o recurso nativo, a função foi implementada no edit-contact-ts, porem antes foi instalado o pacote cordova com o comando:

ionic cordova plugin add cordova-plugin-vibration --save

npm install --save @ionic-native/vibration

Segue o código:

vibrar(){

this.vibration.vibrate(1000)

}

# SERVICE WORKER

Foi criado um service worker para que funções que não dependem de internet sejam instaladas no dispositivo móvel, segue o código:

/\*\*

\* Check out https://googlechrome.github.io/sw-toolbox/ for

\* more info on how to use sw-toolbox to custom configure your service worker.

\*/

'use strict';

importScripts('./build/sw-toolbox.js');

self.toolbox.options.cache = {

name: 'ionic-cache'

};

// pre-cache our key assets

self.toolbox.precache(

[

'./build/main.js',

'./build/vendor.js',

'./build/main.css',

'./build/polyfills.js',

'index.html',

'manifest.json'

]

);

// dynamically cache any other local assets

self.toolbox.router.any('/\*', self.toolbox.cacheFirst);

// for any other requests go to the network, cache,

// and then only use that cached resource if your user goes offline

self.toolbox.router.default = self.toolbox.networkFirst;

# 4.TESTES E RESULTADOS

A aplicação foi testada utilizando o ionic devapp, que foi instalado no meu celular lg k4 e foi testada através dos comandos:

Ionic serve --devapp