ESTUDOS REACTJS

Primeiro passo para começar um projeto Javascript é criar o packge.json, ele pode ser criado com um dos dois comandos abaixo:

* yarn init -y
* Npm init -y

O packege.json é o arquivo onde fica as pricipais informações do projeto e dependências de terceiros, principais bibliotecas são:

* React
  + yarn add react
* React Dom
  + yarn add react-dom
* Babel (converte JS para um JS que os Browsers entendão)
  + yarn add @babel/core @babel/cli @babel/preset-env @babel/preset-react babel-loader -D
* WebPack
  + yarn add webpack webpack-cli webpack-dev-server -D
* Plugin pra injetar tag scrip no html
  + yarn add html-webpack-plugin -D

Para o Babel é necessário criar um aquivo de configuração chamado babel.config.js; No contéudo deste arquivo estará os exports dos módulos do babel.

module.exports = {

presets: [

'@babel/preset-env',

'@babel/preset-react'

]

}

Para converter o código com JS modermo para um JS entendível pelos browsers deve-se rodar no terminal o comando:

* yarn babel (caminho do arquivo) -o (flag -o indica a saída) (saída do arquivo)
  + yarn babel src/index.js -o dist/bundle.js

Para o WebPack é necessário criar o arquivo de configuração webpack.config.js; No conteúdo do arquivo deve-se configurar os módulos que serão exportados:

const path = require('path');

const HtmlWebpackPlugin = require('html-webpack-plugin');

module.exports = {

mode: 'development',

entry: path.resolve(\_\_dirname, 'src', 'index.jsx'),

output: {

Path: path.resolve(\_\_dirname, 'dist'),

filename: 'bundle.js'

},

resolve: {

extensions: ['.js','.jsx'],

},

plugins: [

new HtmlWebpackPlugin({

template: path.resolve(\_\_dirname, 'public', 'index.html'),

}),

],

module: {

rules:

[

{

test: /\.jsx$/,

exclude: /node\_modules/,

use: 'babel-loader'

}

]

}

};

Para executar o webpack na aplicação basta rodar no terminal: yarn webpack.