

# Electron

Do web ao desktop em poucos comandos

## AGENDA

01

Electron – O que é

02

Quem utiliza o Electron

03

Exemplos e Aplicação  
Prática

04

Build e Deploy

# APRESENTAÇÃO

---

## Julio Sieg

Desenvolvedor de Software na Nelogica  
Bacharel em Ciência da Computação pela UPF  
MBA em Gestão de Projetos pela Universidade La Salle



# ELECTRON

## O QUE É?

Permite o desenvolvimento de aplicações desktop, usando Javascript, HTML e CSS.

Desenvolvido pelo **Github**.

Utiliza tecnologias como o **Node.js** e o Chromium

# ELECTRON

## QUEM UTILIZA?



### VISUAL STUDIO CODE

Microsoft

Editor de código-fonte desenvolvido para Windows, macOS e Linux.



### ATOM EDITOR

Github

Editor de texto de código aberto disponível para Linux, macOS e Windows.



### SLACK

Slack

Plataforma proprietária de mensagens baseada na nuvem.

## ELECTRON QUEM UTILIZA?



### WHATSAPP DESKTOP

Whatsapp Inc.

Aplicativo multiplataforma de mensagens instantâneas e chamadas de voz.



### WEBTORRENT DESKTOP

Webtorrent

O WebTorrent é um cliente torrent de streaming ponto a ponto.



### WORDPRESS DESKTOP

Wordpress

Manutenção de sites Wordpress através do aplicativo desktop

“É mais fácil do que você pensa. Se você pode construir um website, você pode construir um aplicativo desktop.”

— Documentação do Electron

## Tecnologias Web

O Electron usa o Chromium e o Node.js, assim você pode construir sua aplicação com HTML, CSS e JavaScript.

## Código Aberto

Electron é um projeto de código aberto mantido pelo GitHub e por uma comunidade ativa de colaboradores.

## Multiplataforma

Compatíveis com Mac, Windows, e Linux, aplicativos Electron são construídos e executados nas três plataformas.



## SIMPLIFICANDO

---

**ATUALIZAÇÕES  
AUTOMÁTICAS**

**NOTIFICAÇÕES E MENU  
NATIVOS**

**RELATÓRIO DE  
FALHAS**

**DEPURAÇÃO E CRIAÇÃO  
DE PERFIL**

**INSTALADORES DO  
WINDOWS**



LET'S DO  
THIS

---

# POR ONDE COMEÇAR

```
# Clone o repositório Quick Start
$ git clone https://github.com/electron/electron-quick-start

# Ir para o repositório
$ cd electron-quick-start

# Instale as dependências e execute
$ npm install && npm start
```

Hello World!  
File Edit View Window Help

**Hello World!**

We are using Node.js 12.8.1, Chromium 78.0.3905.1, and Electron 7.0.0.

Hello World!  
File Edit View Window Help

**Hello World!**

We are using Node.js 12.8.1, Chromium 78.0.3905.1, and Electron 7.0.0.

```
const {app, BrowserWindow} = require('electron')
const path = require('path')

let mainWindow

function createWindow () {
  mainWindow = new BrowserWindow({
    width: 800,
    height: 600,
    webPreferences: {
      preload: path.join(__dirname, 'preload.js')
    }
  })

  mainWindow.loadFile('index.html')

  mainWindow.on('closed', function () {
    mainWindow = null
  })
}

app.on('ready', createWindow)
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Hello World!</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    We are using Node.js <span id="node-version"></span>,
    Chromium <span id="chrome-version"></span>,
    and Electron <span id="electron-version"></span>.

    <!-- You can also require other files to run in this process -->
    <script src="./renderer.js"></script>
  </body>
</html>
```

```
// All of the Node.js APIs are available in the preload process.  
// It has the same sandbox as a Chrome extension.  
window.addEventListener('DOMContentLoaded', () => {  
  const replaceText = (selector, text) => {  
    const element = document.getElementById(selector)  
    if (element) element.innerText = text  
  }  
  
  for (const type of ['chrome', 'node', 'electron']) {  
    replaceText(`${type}-version`, process.versions[type])  
  }  
})
```

# RENDERIZANDO PÁGINA WEB

```
# Criando nova pasta para o projeto
$ mkdir projeto-tdc

# Ir para o repositório
$ cd projeto-tdc

# Instalar o electron
$ npm install -g electron

# Inicializa
$ npm init

# Executar
$ electron .
```



```
const { app, BrowserWindow } = require('electron')

let win

app.on('ready', function(){
  win = new BrowserWindow({ width: 800, height: 600 })
  win.loadURL('https://www.thedevelopersconference.com.br/tdc/2019/portoalegre/trilhas')
})
```



TDC 2019 ▾

+TDC ▾

DATA & LOCAL

INSCRIÇÕES

PATROCÍNIO ▾

TDC ONLINE ▾



Grade clique nos botões para ver a programação e conteúdo de cada atividade

QUARTA | 27/11

KANBAN E LEAN

RH ÁGIL

ANÁLISE DE NEGÓCIOS

SOFTWARE SECURITY

DEVOPS

QUINTA | 28/11

EXTREME PROGRAMMING XP

MANAGEMENT 3.0 E GESTÃO ÁGIL

TRANSFORMAÇÃO DIGITAL E INOVAÇÃO

SAÚDE 4.0

DEVOPS TOOLS



Acessível em Libras

# BUILD E DEPLOY

```
# Instalar o Electron Builder  
$ npm install electron-builder --save-dev
```





## package.json

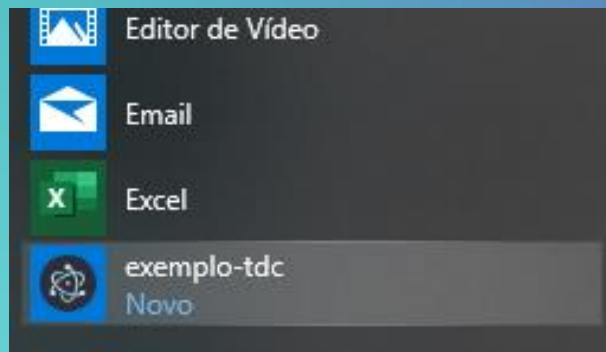
```
{
  "name": "exemplo-tdc",
  "version": "1.0.0",
  "description": "",
  "main": "main.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node main.js",
    "pack": "electron-builder --dir",
    "dist": "electron-builder"
  },
  "build": {
    "appId": "your.id",
    "mac": {
      "category": "your.app.category.type"
    }
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "electron": "^7.0.0",
    "electron-builder": "^21.2.0"
  }
}
```

# BUILD E DEPLOY

```
# Rodar o electron-builder  
$ npm run dist
```

# BUILD E DEPLOY

	win-unpacked	27/10/2019 02:53	Pasta de arquivos	
	builder-effective-config.yaml	27/10/2019 02:53	Arquivo YAML	1 KB
	exemplo-tdc Setup 1.0.0	27/10/2019 02:54	Aplicativo	43.639 KB
	exemplo-tdc Setup 1.0.0.exe.blockmap	27/10/2019 02:54	Arquivo BLOCKM...	47 KB



# OBRIGADO!

---

Alguma pergunta?

julio.sieg@gmail.com  
+55 51 98652-5088

