

Mão na massa: criando a janela Sobre

Começando deste ponto? Você pode fazer o [DOWNLOAD \(https://s3.amazonaws.com/caelum-online-public/electron/cap01/stages/alura-timer-fim-cap01.zip\)](https://s3.amazonaws.com/caelum-online-public/electron/cap01/stages/alura-timer-fim-cap01.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

Vamos começar criando a janela de sobre, para praticarmos com o Browser Window.

1- Crie a página `sobre.html` dentro da pasta `app`.

2- No `index.html`, coloque um link para a página Sobre que vamos criar.

```
<a href="#" id="link-sobre">Sobre</a>
```

3- No `renderer.js`, vamos vincular o evento de click do link criado no item anterior a uma função que irá criar a janela Sobre. Para isso, é necessário importar o módulo `ipcRenderer` do Electron. Em seguida, dentro da função, vamos disparar um evento próprio chamado `abrir-janela-sobre`, através do método `send` do `ipcRenderer`.

```
const { ipcRenderer } = require('electron');

let linkSobre = document.querySelector('#link-sobre');

linkSobre.addEventListener('click', function(){
  ipcRenderer.send('abrir-janela-sobre');
});
```

4- Registre o `listener` para o evento `abrir-janela-sobre` no módulo `ipcMain` dentro do arquivo `main.js`, criando uma função que vai criar a janela Sobre com as características definidas na aula (tamanho, sempre no topo das outras janelas e sem frame). Defina a variável `sobreWindow` fora do `listener`, para que apenas uma janela Sobre seja criada. Não se esqueça de importar o módulo `ipcMain` e carregar a URL da página `sobre.html` depois de garantir que a janela está instanciada.

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

//...trecho de código omitido...

let sobreWindow = null;
ipcMain.on('abrir-janela-sobre', () => {
  if(sobreWindow == null){
    sobreWindow = new BrowserWindow({
      width: 300,
      height: 220,
      alwaysOnTop: true,
      frame: false
    });
  }
  sobreWindow.loadURL(`file://${__dirname}/app/sobre.html`);
});
```

5- Registre o evento `closed` dentro do condicional de criação da instância da janela Sobre, para que a variável `sobreWindow` não seja destruída pelo Javascript.

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

//...trecho de código omitido...

let sobreWindow = null;
ipcMain.on('abrir-janela-sobre', () => {
  if(sobreWindow == null){
    sobreWindow = new BrowserWindow({
      width: 300,
      height: 220,
      alwaysOnTop: true,
      frame: false
    });
    sobreWindow.on('closed', () => {
      sobreWindow = null;
    })
  }
  sobreWindow.loadURL(`file://${__dirname}/app/sobre.html`);
});
```

6- Crie um link para fechar a janela Sobre com `id link-fechar`. Registre o evento de click do link em um arquivo novo chamado `sobre.js`, disparando um evento próprio `fechar-janela-sobre` através do `ipcRenderer`. Será necessário importar o módulo `ipcRenderer`.

```
const { ipcRenderer } = require('electron');

let linkFechar = document.querySelector("#link-fechar");

linkFechar.addEventListener('click', function () {
  ipcRenderer.send('fechar-janela-sobre');
})
```

7- Registre o `listener` para o evento `fechar-janela-sobre` no módulo `ipcMain` dentro do arquivo `main.js`, criando uma função que vai fechar a janela Sobre.

```
//...trecho de código omitido do arquivo main.js...

ipcMain.on('fechar-janela-sobre', () => {
  sobreWindow.close();
});
```

8- Na janela Sobre, adicione a logo do Alura Timer (baixe a imagem [aqui \(https://s3.amazonaws.com/caelum-online-public/electron/cap02/stages/icone-alura-timer.zip\)](https://s3.amazonaws.com/caelum-online-public/electron/cap02/stages/icone-alura-timer.zip)), nome autor e seu twitter, além do número das versões usadas. Para isso, modifique sua página `sobre.html` para ficar com o código abaixo. Crie também um arquivo CSS `sobre.css` para definir o estilo da página, linkando-o através da tag `link`.

```
<!DOCTYPE html>
<html lang="en">
```

```

<head>
  <meta charset="UTF-8">
  <title>Sobre o Alura Timer</title>
  <link rel="stylesheet" href="css/sobre.css">
</head>
<body>

  <div class="container">
    
    <p>O Alura timer é para você guardar seus tempos de estudo na Alura.</p>
    <p>Versão: 1.0.0</p>
    <p>Feito por <a href="#" id="link-twitter">Douglas Quintanilha</a></p>
    <p>Versão do Electron <span id="versao-electron"></span></p>
    <a href="#" id="link-fechar">Fechar</a>

  </div>
  <script src="js/sobre.js"></script>
</body>
</html>

```

O código do arquivo `sobre.css` está listado abaixo:

```

.container{
  text-align: center;
}
.container p {
  margin: 5px 0;
}
.container img{
  width: 50px;
}

```

9- Recupere a versão do Electron dinamicamente quando a página carregar e a insira na tag `span` identificada por `versao-electron`. Isso será feito no arquivo `sobre.js`. Será necessário carregar o módulo `process` do Node.js.

```

//...trecho omitido do arquivo sobre.js...

const process = require('process');

let versaoElectron = document.querySelector('#versao-electron');

window.onload = function(){
  versaoElectron.textContent = process.versions.electron;
}

```

10- Faça com que o link para o Twitter do autor abra num navegador normal do sistema operacional. Para isso, importe o módulo `shell` do Electron e use-o no evento de click do link `link-twitter`, através do método `openExternal`.

```

//...trecho omitido do arquivo sobre.js...

const { ipcRenderer, shell } = require('electron');

let linkTwitter = document.querySelector("#link-twitter");

```

```
linkTwitter.addEventListener('click', function () {
  shell.openExternal("https://www.twitter.com/dquintanilhas");
})
```