

Entendendo a criação de menus

Transcrição

Para conseguirmos criar o menu de cursos, ao clicar no *try icon* do Alura Timer, vamos utilizar o módulo [Menu](https://electron.atom.io/docs/api/menu/) (<https://electron.atom.io/docs/api/menu/>) do Electron. Então, já podemos importá-lo no **main.js**:

```
// main.js

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

// restante do código comentado
```

E para criar o menu, vamos utilizar a função `Menu.buildFromTemplate`, que recebe um array como parâmetro. Esse array é um array de objetos, e cada objeto representa um item do menu, em que podemos configurar o seu texto (`label`), o seu tipo (`type`), por exemplo:

```
// main.js

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

app.on('ready', () => {
  console.log('Aplicação iniciada');
  mainWindow = new BrowserWindow({
    width: 600,
    height: 400,
  });
  tray = new Tray(__dirname + '/app/img/icon-tray.png');
  let trayMenu = Menu.buildFromTemplate([
    {label: 'Teste1', type: 'radio'},
    {label: 'Teste2', type: 'radio'}
  ]);

  mainWindow.loadURL(`file://${__dirname}/app/index.html`);
});

// restante do código comentado
```

Após isso, podemos atribuir o menu ao *try icon*, passando-o como parâmetro para a função `setContextMenu` do ícone:

```
// main.js

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

app.on('ready', () => {
  console.log('Aplicação iniciada');
  mainWindow = new BrowserWindow({
    width: 600,
    height: 400,
  });
});
```

```
tray = new Tray(__dirname + '/app/img/icon-tray.png');
let trayMenu = Menu.buildFromTemplate([
  {label: 'Teste1', type: 'radio'},
  {label: 'Teste2', type: 'radio'}
]);
tray.setContextMenu(trayMenu);

mainWindow.loadURL(`file://${__dirname}/app/index.html`);
});

// restante do código comentado
```

Ao rodar a nossa aplicação e clicar o ícone na *tray bar*, o menu é exibido. Como o tipo dos itens é `radio`, só podemos selecionar um item do menu, o que nos atende perfeitamente. Há várias outras propriedades que podemos configurar no item do menu, que podem ser vistas na documentação do [MenuItem](https://electron.atom.io/docs/api/menu-item/#new-menuitemoptions) (<https://electron.atom.io/docs/api/menu-item/#new-menuitemoptions>).

Por exemplo, o tipo `separator` transforma o item em uma linha fina:

```
// main.js

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

app.on('ready', () => {
  console.log('Aplicação iniciada');
  mainWindow = new BrowserWindow({
    width: 600,
    height: 400,
  });
  tray = new Tray(__dirname + '/app/img/icon-tray.png');
  let trayMenu = Menu.buildFromTemplate([
    {label: 'Cursos'},
    {label: '', type: 'separator'},
    {label: 'JavaScript', type: 'radio'},
    {label: 'Java', type: 'radio'},
    {label: 'Photoshop', type: 'radio'}
  ]);
  tray.setContextMenu(trayMenu);

  mainWindow.loadURL(`file://${__dirname}/app/index.html`);
});

// restante do código comentado
```

Entendida a criação de menus, vamos começar a montar o nosso próprio menu a partir do próximo vídeo.