08

Mãos na Massa: Finalizando o app

Começando deste ponto? Você pode fazer o DOWNLOAD (https://s3.amazonaws.com/caelum-online-public/electron/cap08/stages/alura-timer-fim-cap08.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

Vamos agora finalizar os últimos detalhes de nosso app antes de embalarmos para os 3 sistemas operacionais.

1- Não queremos que o usuário adicione um curso com o nome vazio, então por isso vamos evitar isso na hora em que ele clica no botão de adicionar. Em seu `renderer.js` faça:

```
// renderer.js
...
botaoAdicionar.addEventListener('click', function() {
    if(campoAdicionar.value == ''){
        console.log('Não posso adicionar um curso com nome vazio');
        return;
    }
    ...
}
```

2- Agora vamos cuidar do caso de quando o usuário adiciona um curso e trocar para um outro sem iniciar o anterior previamente, o curso novo fica com o tempo do curso trocado. Isto ocorre pois o nosso console está jogando um erro, afinal se o curso não foi iniciado, ele também não gerou um JSON com o tempo do curso, logo ele não consegue trocar o tempo para um curso que não tem JSON. Em seu `renderer.js` , capture este erro e coloque o timer para zero caso o curso não tenha tempo:

```
//renderer.js
...
ipcRenderer.on('curso-trocado', (event,nomeCurso) => {
    data.pegaDados(nomeCurso)
        .then((dados) => {
            tempo.textContent = dados.tempo;
        //Adicione o catch abaixo
        }).catch((err) => {
            console.log('O curso ainda não possuí um JSON');
            tempo.textContent = "00:00:00";
        })
    curso.textContent = nomeCurso;
});
```

3- Para que o timer seja parado previamente também antes de cada trocada de curso, fazendo assim com que um timer de um curso não sobreescreva o de outro, vamos chamar a função `timer.parar()` :

```
//renderer.js
...
ipcRenderer.on('curso-trocado', (event,nomeCurso) => {
    //Adicionar a chamada a timer.parar() aqui
```

```
    timer.parar(curso.textContent);

    data.pegaDados(nomeCurso)
    ...

    curso.textContent = nomeCurso;
});
```

4- Vamos também já inicializar a nossa variável segundos com o valor de `0`, evitando assim um erro ao adicionar novos cursos devido a nossa última mudança. Em seu `timer.js` atribua:

```
const { ipcRenderer } = require('electron');
const moment = require('moment');
let segundos = 0;
...
```

Agora com a nossa aplicação mais redonda, vamos aprender como gerar para os três sistemas operacionais.