

## Padronizando o método que envia os alunos internos

Como toque final, vamos modificar a forma como estamos salvando os alunos por meio do método `sincronizaAlunosInternos()` da classe `AlunoSincronizador`.

Atualmente fazemos ainda no modo manual e repetitivo que não passa pelo processo de verificação que implementamos anteriormente, sendo assim, modifique o `onResponse()` para que utilize o método `sincroniza()`, e então, chame o `close` do `AlunoDAO` antes da chamada da `call` que envia todos os alunos internos.

## Projeto Android

O projeto final do curso, isto é, com todas as alterações realizadas em aula, pode ser acessado por meio [deste link](https://github.com/alura-cursos/android-sync-parte3/archive/aula-5.zip) (<https://github.com/alura-cursos/android-sync-parte3/archive/aula-5.zip>).

## Projeto Web

Caso tenha interesse em verificar o arquivo fonte do projeto web segue o [link do github](https://github.com/alura-cursos/agenda-web) (<https://github.com/alura-cursos/agenda-web>). Lembrando que esse é um projeto Spring Boot baseado em Maven, portanto, para executá-lo será necessário ter o Maven na sua máquina. O único comando que você precisará executar é o `maven package` dentro do diretório raiz.

Assim que o processo do Maven finalizar, basta apenas executar o Arquivo `Application.class` (fonte dele `agenda-web/src/main/java/br/com/caelum/alura/Application.java`). Recomendo que utilize uma IDE como [Spring Tool Suite](https://spring.io/tools) (<https://spring.io/tools>) ou [Eclipse](https://eclipse.org/downloads/) (<https://eclipse.org/downloads/>) para visualização ou edição do projeto.