

Implementando a prioridade

Caso você não tenha o projeto com as alterações realizadas na aula anterior, você pode acessá-lo nesse [link](https://github.com/alura-cursos/android-sync-parte3/archive/aula-3.zip) (<https://github.com/alura-cursos/android-sync-parte3/archive/aula-3.zip>).

Agora que sabemos sobre como podemos lidar com o merge das informações, precisamos deixar uma prioridade para um dos lados, isto é, seja servidor ou App Android.

Para o nosso cenário atual vimos que a prioridade no servidor pode fazer mais sentido justamente porque vários clientes podem alterar o mesmo recurso, ou seja, se um cliente offline tiver uma alteração mais antiga não faz sentido mantê-la.

Sendo assim, no final do `onResponse()` do callback do método `buscaAlunoCallback()`, chame o método `sincronizaAlunosInternos()`. Dessa forma, apenas depois de buscar todas as informações do servidor que iremos enviar todos os alunos internos.

Além disso, para garantir que a sincronização vai ser feita apenas dessa maneira, deixe o método `sincronizaAlunosInternos()` com o modificador de acesso `private`, então, remova as chamadas desse método na `ListaAlunosActivity`, pois no nosso contexto atual, só fará sentido esse comportamento depois de buscarmos as informações do servidor.