

## Realizando a migration para adicionar o campo sincronizado

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-1.zip) (<https://github.com/alura-cursos/android-sync-parte3/archive/aula-1.zip>).

Precisamos agora fazer com que a nossa App funcione também em modo offline. Para que isso seja possível, o nosso primeiro passo é justamente fazer com que cada aluno, no momento que é editado seja marcado como não sincronizado. Pois dessa forma conseguimos saber quais alunos foram editados em modo offline.

Portanto, na classe `Aluno` adicione o atributo `sincronizado` do tipo `int`. Aproveite também e adicione os getters e setters.

Em seguida, precisamos fazer com que esse novo campo também seja persistido no SQLite, ou seja, será necessário realizar uma migration. Dentro do método `onUpgrade()`, adicione um novo `case` para adicionar o novo campo `sincronizado` com o tipo `int` e valor padrão `0`, pois, dessa forma, evitamos o caso de lidar com valores nulos, e também, para considerar que todo dado novo, será considerado como não sincronizado.

Lembre-se de modificar a tabela do `onCreate()` para que ela também tenha esse novo campo, e também, modifique a versão do SQLite para que agora ele execute essa migration.

Por fim, adicione a referência desse novo campo tanto no método `pegaDadosDoAluno()` como também no método `populaAlunos()` para que seja possível tanto enviar como buscar as informações referentes a esse marcador.

Após realizar todos os passos, adicione um log no `foreach` dentro do método `carregaLista()` da `ListaAlunosActivity` para verificar a informação do atributo `sincronizado`.

Depois de toda a implementação, realize o teste da App e veja qual é o valor impresso.