

## Implementando Adapter personalizado

Crie o Adapter personalizado para que seja possível migrar do `ArrayAdapter` para o novo Adapter e manter os comportamentos de antes, como notificar a atualização e remoção de alunos no dataset.

Para isso, crie a classe `ListaAlunosAdapter` no pacote `br.com.alura.agenda.ui.adapter`. Faça extensão da classe `BaseAdapter` e envie todos os métodos sobrescritos da `BaseAdapter` que está na `ListaAlunosActivity` para o adapter.

Lembre-se de aplicar os imports necessários para que tudo compile.

Muito provavelmente haverá um problema de compilação no processo de inflar o layout. Isso acontece pelo fato de não obter uma referência de `Context`.

Crie um atributo do tipo `Context` e o envie no `from()` do `LayoutInflater`. Então, peça o `Context` via construtor.

Na `ListaAlunosActivity`, mude a referência do `adapter` para que utilize a `ListaAlunosAdapter`. Em seguida, no método de configuração do adapter, faça a inicialização do `adapter` com a nova classe.

Implemente os métodos `clear()`, `addAll()` e `remove()`.

As implementações de ambos são chamadas equivalentes, porém, para a lista de alunos, por exemplo, no `clear()` basta apenas chamar o `clear()` da lista de alunos.

Teste o App e veja se aparece as views personalizadas. Considerando essa implementação, deve apenas o espaço ocupado com base na altura ocupada pelo `item_aluno.xml`.

Caso você tenha deixado o layout comentado, para que apresente o espaço ocupado pelo layout.