

## Atualizando os alunos com o SwipeRefreshLayout

Atualmente, para verificarmos alguma mudança que ocorreu lá no servidor, precisamos fechar a App e abrir denovo, ou seja, todas as vezes que o usuário quiser saber sobre as últimas atualizações que foram feitas na sua agenda ele precisará abrir e fechar a App... Com toda certeza esse tipo de ação da uma péssima experiência para o usuário. Para melhorar a experiência do usuário, vamos adicionar um componente do Android conhecido como [SwipeRefreshLayout](https://developer.android.com/reference/android/support/v4/widget/SwipeRefreshLayout.html) (<https://developer.android.com/reference/android/support/v4/widget/SwipeRefreshLayout.html>).

A partir deste componente assim que puxamos a lista pra baixo aparece aquele circulozinho que fica girando, ou seja, indica que algo está carregando!

### Adicionando o SwipeRefreshLayout

Vá até o arquivo `res/layout/activity_lista_alunos.xml` que é justamente o XML de layout da lista de alunos. Dentro do XML adicione o `SwipeRefreshLayout` conforme o código abaixo:

```
<android.support.v4.widget.SwipeRefreshLayout
    android:id="@+id/swipe_lista_aluno"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/lista_alunos"/>
</android.support.v4.widget.SwipeRefreshLayout>
```

Repara que o swipe está com o id `swipe_lista_aluno` que vai ser a forma a qual recuperaremos este componente na activity. Além disso, para que funcione como esperamos, isto é, para colocarmos o comportamento esperado do swipe, é necessário deixar a lista, neste caso a `ListView`, dentro do swipe.

Agora, dentro do `onCreate()` da `ListaAlunosActivity` faça um `findViewById()` para o id do swipe, e então, atribua o retorno para um atributo da activity.

### Adicionando ação ao SwipeRefreshLayout

Com o swipe em mãos, precisamos adicionar uma ação assim que ele for acionado, ou seja, configurar um listener que fique esperando a ativação dele! Para isso basta apenas utilizar o método `setOnRefreshListener()` enviando como parâmetro uma implementação da interface `SwipeRefreshLayout.OnRefreshListener`, a implementação pode ser feita por meio do recurso de classe anônima conforme o código abaixo:

```
swipe.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {
    @Override
    public void onRefresh() {
    }
});
```

Agora é necessário realizar a mesma call que está contida no `onCreate()` dentro do método `onRefresh()`. Ou melhor, extraia o código que faz a call para um método chamado `buscaAlunos()`, então, invoque esse método dentro do `onRefresh()` também!

### Testando a App

Execute a App novamente, em seguida, vá até a aplicação web e adicione um ou mais alunos. Volte até a App Android, e arraste a lista para baixo e veja se o componente do `SwipeRefreshLayout` aparece e também se os alunos que foram adicionados no servidor também!

