

08

Sincronizando alunos editados em offline

Transcrição

Até o momento, fizemos apenas uma unica implementação que funciona em modo offline, porém, apenas a inserção não é o suficiente para a aplicação, também precisa funcionar tanto a edição quanto a remoção dos alunos.

Vamos testar colocando o celular no modo avião. Agora que estamos offline podemos editar um aluno trocando o nome, o aluno será alterado localmente. Tirando o modo avião, e com a conexão estabelecida, fazemos um *swipe* e vemos que os dados não são atualizados no servidor.

Se analisarmos o *log* percebemos que **todos** os alunos estão como "sincronizados". Mas um aluno editado é sincronizado? Na verdade não deveria, justamente por conter novas informações. Precisamos garantir que todas as informações que forem editadas ficarão como "não sincronizadas".

Informações editadas estão na classe `FormularioActivity`, podemos acessar mais rapidamente a classe usando o atalho "Shift + Shift", no campo aberto basta digitar o nome da classe.

As informações editadas ficam no método `onOptionsItemSelected()`, onde utilizamos o `helper`, popula os alunos e manda para o banco de dados. O problema é que quando executamos esse processo não estamos informando que o dado **não** é mais sincronizado, afinal estamos fazendo uma edição dentro da aplicação.

Por conta disso logo após a utilização do `helper` vamos *settar* o valor do atributo `sincronizacao` como **0**. Mas lembrando que isso é função da própria classe aluno, por isso chamaremos o método `aluno.desincroniza()`. Nosso método ficará:

```
// ...  
  
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.menu_formulario_ok:  
            Aluno aluno = helper.pegaAluno();  
  
            aluno.desincroniza();  
  
            AlunoDAO dao = new AlunoDAO(this);  
            if (aluno.getId() != null) {  
                dao.altera(aluno);  
            } else {  
                dao.insere(aluno);  
            }  
            dao.close();  
  
            Call call = new RetrofitInicializador().getAlunoService().insere(aluno);  
  
            call.enqueue(new Callback() {  
                @Override  
                public void onResponse(Call call, Response response) {  
                    Log.i("onResponse", "requisicao com sucesso");  
                }  
            });  
    }  
}
```

```
        }

    @Override
    public void onFailure(Call call, Throwable t) {
        Log.e("onFailure", "requisicao falhou");
    }
});

Toast.makeText(FormularioActivity.this, "Aluno " + aluno.getNome() + " salvo!", Toast.LENGTH_SHORT).show();

finish();
break;
}

return super.onOptionsItemSelected(item);
}

// ...
```

Como o método `aluno.desincroniza()` ainda não existe, vamos criá-lo na classe `Aluno`.

```
// ...

public void sincroniza() {
    this.sincronizado = 1;
}

public void desincroniza() {
    this.sincronizado = 0;
}

// ...
```

Vamos executar a aplicação, colocar no modo avião e editar um aluno em modo offline. Após editar, retiramos o modo avião e forçamos a atualização com um *swipe*. Atualizando o servidor vimos que o aluno editado foi enviado.