04

Exercício 2

Vamos melhorar a usabilidade de nossa aplicação colocando um `ProgressDialog` na tela para avisar que uma tarefa está sendo executada em *background*.

Como vamos impactar na tela precisaremos de um `Context`. Altere o construtor da classe `EnviaContatosTask` para pedir um `Context`. Guarde-o como atributo para que possa ser usado no `onPreExecute`:

```
public class EnviaContatosTask extends AsyncTask<Object, Object, String>{

    private final Context context;
    private final String ENDERECO = "http://www.caelum.com.br/mobile";

    public EnviaContatosTask(Context context) {
        this.context = context;
    }

    @Override
    protected void onPreExecute() {
        // aqui vamos criar nosso ProgressDialog!
    }

    //...
}
```

Implemente o método `onPreExecute` da classe `EnviaContatosTask` e crie dentro do método um `ProgressDialog` que aparecerá enquanto a requisição estiver sendo feita para o nosso servidor:

```
ProgressDialog progress = ProgressDialog.show(context, "Aguarde...",
           "Envio de dados para a web", true, true);
```

Não esqueça de guardar o progress criado para que ele possa ser cancelado ao fim da requisição para o servidor:

```
public class EnviaContatosTask extends AsyncTask<Object, Object, String>{
    //...

    @Override
    protected void onPostExecute(String result) {
        // cancelar o ProgressDialog criado!

        //...
    }
}
```

Dica: Para cancelar programaticamente um `ProgressDialog` chame o seu método `dismiss`.

Rode novamente sua aplicação e teste a nova funcionalidade.

Como ficou sua implementação completa do EnviaContatosTask?

Responda

INSERIR CÓDIGO		FORMATAÇÃO