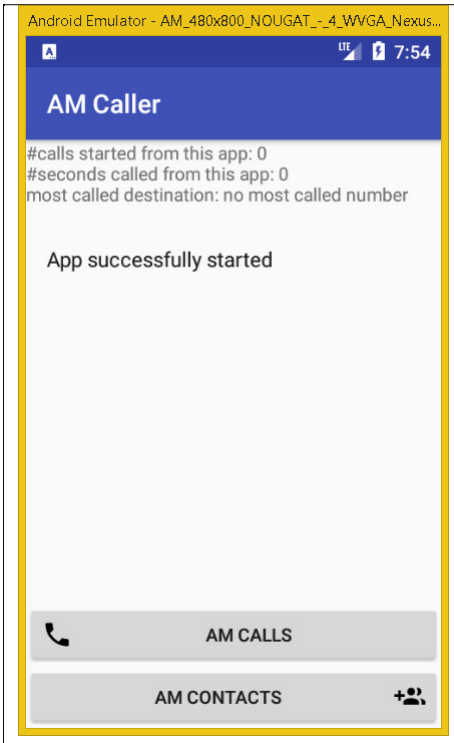

1. [Desenho de layout]
Escreva em XML o código de um RelativeLayout que permita a interface ilustrada pela imagem.

Existem, de cima para baixo, e da esquerda para a direita, com exceção da barra de título, as seguintes zonas:
Z0: uma zona de texto identificada por idTvDashboard, com mais do que uma linha, que em runtime receberá frase(s) com 3 linhas/informações;
Z1: uma ListView identificada por idLvOps, que em runtime poderá receber feedback de operações feitas com a app, a mais recente à cabeça;
Z2: um botão idBtnCalls, para encaminhar o utilizador para uma Activity "ActivityAmCalls";
Z3: um botão idBtnContacts", para encaminhar o utilizador para uma Activity "ActivityBasicContacts".

Note que esta descrição refere algumas situações de runtime, que o XML sozinho não consegue capturar - são referidas apenas para melhor contextualizar a app.

Saiba que os ícones que estão na imagem, nos botões, são Drawables vetoriais com ids "ic_calls" e "ic_contacts", que poderão ser embebidos nos Button via atributos "android:drawableLeft" e "android:drawableRight", para aparecerem à esquerda ou à direita, respetivamente.

2. [padrão “init”]
Admita que o XML da questão anterior é o layout da Activity "SimpleCallerStart", que é a Activity de arranque da app.

Escreva, em Java, um método de nome "init", invocável em onCreate, que permita:
- inicializar os membros de dados Java da class, com nomes que correspondam aos ids do XML (por exemplo "mXpto" para o elemento que no XML tenha id "idXpto");
- confira, pelo mesmo listener, comportamentos diferentes "goCalls" e "goBasicContacts" aos botões em Z2 e Z3, respetivamente, devendo incluir na sua resposta a invocação das Activities adequadas;
- tenha código defensivo e reutilizável que evite quaisquer operações com objetos null;
- organize a abordagem em momentos de associação de variáveis a objetos XML e de atribuição de comportamentos;
- no final invoque "updateDashboard()", que deve admitir disponível e perfeito, tendo apenas que referir como escreveria String(s) com campos variáveis adequados em strings.xml.

3. [utilização de SQLite para Android]
Admita que quer categorizar contactos e chamadas, por exemplo com categorias como "family" e "work".
Escreva parcialmente as classes "Category" e "CategoryDB". Uma Category é apenas um "name" (String) e uma "dateStamp" (String); CategoryDB deve seguir o padrão SQLiteOpenHelper.

3.1. Escreva uma estrutura SQLite adequada para CategoryDB.

3.2. Escreva
public long insertCategory (Category pC); //que deve retornar o id onde o insert foi feito, ou -1 se falhar.

3.3. Escreva
ArrayList<Category> selectCategoriesWithNameMatching (String pExpression); //que deve retornar um ArrayList de Category objects cujo nome satisfaça exatamente a expressão recebida.

4. [Execução assíncrona]
Assuma que é necessário fazer a importação de contactos "legacy", disponíveis para serem consumidos desde o URL: https://site/dados.TSV
Note que é um recurso remoto, a ser consumido por https; note também o formato TSV exemplificado:
John * 123456789 * family\n
Rita * 987654321 * work\n

Admita que tem disponível um método ler_https(String pUrl) que retorna a String do conteúdo lido por https.

4.1. Escreva, em SimpleCallerStart, em método “legacyLoader”, todo o código necessário para:
Ler os contactos e depois criar as categorias lidas, em base de dados, sem repetições;

4.2. Escreva, na Activity principal, uma classe adequada a executar legacyLoader em thread própria.

5. [Integração]
Admita que na Activity ActivityAmCalls está disponível um membro de dados mCalls, do tipo ArrayList<Call>, que em runtime descreve todas as chamadas já feitas a partir da aplicação.

Cada objeto do tipo Call, que pode assumir disponível e perfeito, representa:
- em membro mWhen, o momento de início de um telefonema;
- em membro String mDestination, o número de destino.

Com liberdade criativa, utilizando o padrão SAFR (Start Activity For Result), explique como faria para:
- chamar ActivityAmCalls a partir da Activity principal, deixando a “chamadora” atenta a um resultado, quando eventualmente chegar;
- na chamada deve seguir um “extra” inteiro em chave KEY_YEAR, correspondente a um ano (exemplo: 2023);
- ao terminar ActivityAmCalls, por algum mecanismo que decida, deve ser produzido como resultado o número de telefone de destino mais utilizado no ano indicado em KEY_YEAR, por análise de mCalls;
- ao receber o resultado (desde que não cancelado), a “chamadora” deve dar feedback, por algum mecanismo à sua escolha.