A classe Application

## **Sobre a classe Application do android, julgue as afirmativas abaixo:**

* Ela é instanciada assim que o seu aplicativo é iniciado.
* Para ela funcionar, basta apenas criar uma classe que herde da classe Application, sem precisar de configurações a mais.
* Ela é morta assim que o seu aplicativo é fechado.
* É uma instancia para cada Activity.

=============================================

O framework Dagger

## **Qual é a funcionalidade principal do framework Dagger ?**

*Marque 1 alternativa*

* É um framework de injeção de dependências
* É um framework para facilitar o gerenciamento de cache no Android
* É um framework para ajudar na hora de gerar o código.
* É um framework para facilitar a comunicação entre componentes.

O Dagger é um framework de injeção de dependências para o Android, que é open source e mantido pela Google. Ele veio com o objetivo de substituir frameworks antigos que dependiam de Reflection em suas soluções, sendo o Dagger bem mais rápido por ser completamente estático e fazer a injeção de dependência em tempo de compilação.

=============================================

Configurando o Dagger

## **Marque as opções abaixo que são necessárias para configurar o Dagger para ser usado em nosso projeto:**

* É preciso criar um novo build.graddle com as dependências do Dagger.
* Precisamos colocar no arquivo build.graddle as linhas com as dependências do Dagger.
* Importar o plugin android-apt no build.graddle do projeto.
* Informar com a instrução apply plugin no build.graddle do módulo que queremos utilizar o plugin do android-apt.

Para utilizar o Dagger, devemos primeiro importar suas dependências no build.graddle do **app** :

dependencies {  
 ...  
 compile 'com.google.dagger:dagger-compiler:2.5'  
 apt 'com.google.dagger:dagger:2.5'  
}

Mas além disso também precisamos importar o plugin do android-apt no build.graddle **do projeto** que é quem será responsável por processar as anotações em tempo de compilação, já que o Dagger não utiliza reflection e depende das anotações para poder injetar as dependências:

dependencies {  
 ...  
 classpath 'com.neenbedankt.gradle.plugins:android-apt:1.8'  
 ...  
}

E fica faltado apenas habilitar o uso do plugin do android-apt no módulo do **app**, com a seguinte instrução:

apply plugin: 'com.neenbedankt.android-apt'

=============================================

Anotando o provedor

## **O Dagger precisa saber como criar os objetos que queremos que ele injete, e para isto trabalhamos com um conceito de módulo. Quais das opções abaixo trazem as anotações do Dagger e suas respectivas funcionalidades para produzir um "criador" de objetos injetáveis ?**

* @Provides, para anotar o método de um módulo que é responsável por retornar um objeto que queremos injetar.
* @Injectable, para anotar a classe que é um módulo do Dagger.
* @Module, para anotar a classe que é um módulo do Dagger.
* @DaggerModule, para anotar a classe que é um módulo do Dagger.
* @Inject, para anotar o método de um módulo que é responsável por retornar um objeto que queremos injetar.

O Dagger utiliza as anotações @Module e @Provides para indicar a classe que é um módulo do Dagger e o método que consegue prover o objeto a ser injetado, respectivamente.