**TESTE PARA PRATICAR**

# Exercício de Fixação

1. **Marque as alternativas verdadeiras.**
   - Usamos match_parent quando queremos que a View seja grande o suficiente para conter todas as views que estão dentro dela.
   - **Usamos match_parent quando queremos que a View seja tão grande quanto o parent.**
   - Usamos wrap_content quando queremos que a View seja tão grande quanto o parent.
   - **Usamos wrap_content quando queremos que a View seja grande o suficiente para conter todas as views que estão dentro dela.**

2. **Marque as alternativas verdadeiras.**
   - Usamos o atributo layout_alignParentRight para alinhar o lado direito de uma View com o lado direito de outra.
   - **Usamos o atributo layout_alignParentRight para alinhar o lado direito de uma View com o lado direito do parent.**
   - Usamos o atributo layout_alignRight para alinhar o lado direito de uma View com o lado direito do parent.
   - Usamos o atributo layout_toRightOf para posicionar uma View à direita de outra View.
   - **Usamos o atributo layout_alignRight para alinhar o lado direito de uma View com o lado direito de outra.**

3. **Quais atributos são mandatórios na tag LinearLayout no código em XML.**
   - layout_behaviour
   - **layout_width**
   - **layout_height**
   - **orientation**
   - paddingBotton
   - context

4. **Assinale as alternativas verdadeiras.**
   - No LinearLayout, devemos adicionar uma id para as Views que serão usadas como referência.

- **Nós podemos adicionar views ao GridLayout da mesma forma que fizemos para o LinearLayout, ou seja, simplesmente adicionando views no arquivo XML e deixando que o android posicione as views na tabela na ordem em que aparecem no XML.**
- **Para fazer com que uma view ocupe várias colunas em um GridLayout, usamos layout_row e layout_column para indicar onde a view começa e, em seguida, usamos layout_columnSpan para indicar quantas colunas a nossa view irá ocupar.**
- No GridLayout, os atributos obrigatórios são layout_width, layout_height, columnCount e rowCount.
- **No RelativeLayout, apenas os atributos layout_width e layout_height são obrigatórios.**

5. **Assuma que estamos em um LinearLayout, como devemos informar que duas Views A e B dividirão o espaço não utilizado, sendo que a View A usará 2/3 desse espaço e a View B usará 1/3?**
    - **Podemos colocar o atributo layout_weight em A e em B para depois atribuir os valores 0.666 e 0.333, respectivamente.**
    - Podemos colocar o atributo layout_weight em A e em B para depois atribuir os valores 1 e 2, respectivamente.
    - **Podemos colocar o atributo layout_weight em A e em B para depois atribuir os valores 2 e 1, respectivamente.**
    - Podemos colocar o atributo layout_weight em A e em B para depois atribuir os valores 0.333 e 0.666, respectivamente.