**Regras de boas práticas de programação**

1. Nomes de variáveis:  
   -O nome de uma variável deve descrever precisamente a sua ideia central, melhor dizendo ir direto ao ponto;  
   -O tamanho do nome não deve ser restrito, e sim contemplar o primeiro tópico, porém deve ser usado de bom senso;  
   -Utilizar a convenção de nome camel case, por exemplo “nomeVariavel”;
2. Regra do escoteiro:  
   -Sempre que algum dos desenvolvedores for fazer alguma alteração em código, deve deixá-lo mais limpo que estava antes;
3. DRY(Don´t Repeat Yourself):  
   -Não é interessante existir replicação, ou duplicação, de código, por isso, não devem existir duas unidades que realizam a mesma função. E também ser totalmente livre de ambiguidade.
4. Comente apenas o necessário:  
   -Se seguirmos bem as outras regras aqui apresentadas, provavelmente o código não precisará ser comentado, melhor dizendo se temos um código intuitivo e limpo, não existe a necessidade de comentários;
5. Tratamento de exceções:  
   -Erros são passíveis de acontecer, porém se tratarmos os mesmos de forma correta, o sistema não irá parar de funcionar, irá continuar executando suas funções, ou seja temos de garantir que o mesmo continue funcionando.
6. Testes limpos:  
   -Os testes devem seguir algumas regras, como:

-*Fast*: O teste deve ser rápido, permitindo que seja realizado várias vezes e a todo momento;

-*Independent*: Ele deve ser independente, a fim de evitar que cause efeito cascata quando da ocorrência de uma falha – o que dificulta a análise dos problemas;

-*Repeatable*: Deve permitir a repetição do teste diversas vezes e em ambientes diferentes;

-*Self-Validation*: Os testes bem escritos retornam com as respostas true ou false, justamente para que a falha não seja subjetiva;

-*Timely*: Os testes devem seguir à risca o critério de pontualidade. Além disso, o ideal é que sejam escritos antes do próprio código, pois evita que ele fique complexo demais para ser testado.