AULA 1

Teste UnitárioDescrição: É a forma de se testar unidades individuais de código fonte. Unidades podem ser métodos, classes, funcionalidades, módulos, etc. Depende muito do que é a menor parte que seu Software pode ser testado.Objetivo: dos testes unitários é mostrar que cada unidade atende corretamente sua especificação.Exemplo: testes automáticos criados para darem entradas e conferirem saídas a cada método, permitindo que se saiba que está funcionando de acordo com o esperado.Benefícios: encontrar problemas o quanto antes, facilitam a mudança da unidade, simplificam a integração e melhoram a documentação.Teste IntegradoDescrição: É a forma de se testar a combinação das unidades em conjunto.Objetivo: Nesse caso, a ideia é encontrar falhas na junção destas unidades. Pode ser que a classe X funcione bem sozinha, mas ao ser utilizada pela classe Y, ela deixe de funcionar.Exemplo: Colocar todo o software para rodar e começar a usar diversas funcionalidades consideradas centrais no seu programa para confirmar que ele roda e as principais funcionalidades fazem o esperado.Benefícios: além de testar funcionalmente, pode assegurar performance e confiabilidade. Ajudam a garantir que o trabalho de um desenvolvedor não está afetando o trabalho de outro e em equipes grandes isso pode fazer toda a diferença se forem realizados com frequência.

http://www.basef.com.br/index.php/Atalhos\_do\_IntelliJ\_Idea

Seleção múltipla, ALT + mouse

GIVEN\_WHEN\_THEN Dado um cenário, quando fazemos algo, então resultado

dadoPessoa\_quandoAdicionamosNomes\_entaoAdicionaComSucesso

BeforeEach executa antes de todos os testes, BeforeAll uma única vez na inicialização, temos também o afterEach e afterAll