



## Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Departamento de Sistemas e Computação

Graduação em Ciência da Computação

### Exercícios de Revisão de Java - JUnit

**Objetivo:** Praticar e revisar o framework JUnit.

Material básico sobre JUnit disponível em

- <http://www.jup-net.de/djux/JUnit40QuickStartGuide.pdf>
- <http://www.cavdar.net/2008/07/21/junit-4-in-60-seconds>
- <http://code.google.com/p/t2framework/wiki/JUnitQuickTutorial>.

Tente compreender o funcionamento do framework JUnit. Ele será usado para testar suas classes. No momento você vai criar uma classe de teste no JUnit para testar a classe Vetor implementada.

Relembre do exercício anterior que existia uma classe `Vetor`. `TestarVetor.java` contendo um método **main** com o código de teste de sua classe Vetor (você pode sobrescrever a classe `TestarVetor` fornecida com este exercício com a que você implementou). Com o estudo de JUnit é possível não apenas melhorar o design através de testes sistemáticos, mas também seguir a teoria de testes de unidade, tentando produzir situações específicas para cada método a ser testado.

A classe Vetor tem diversos métodos. A idéia é que cada método (exceto o construtor, gets e sets) sejam tratados por pelo menos um caso de teste (pense num caso de teste como um método que vai realizar um teste específico num método da classe desejada).

1. Crie uma classe de teste usando o Wizard do Eclipse. Neste caso a classe a ser testada será `Vetor.java`. No próprio Wizard você pode informar/escolher que métodos devem ser testados que o wizard já gera um caso de teste para cada um deles.
2. Note que os métodos auxiliares dos casos de testes já estará criado (provavelmente apenas o **setUp**). Preencha o método com as inicializações necessárias. O método **setUp** será executado antes de cada caso de teste, de forma que os casos de testes funcionem de forma independente com uma mesma inicialização.
3. Em seguida preencha o código de cada caso de teste procurando isolar o aspecto/método testado dos demais para evitar efeitos colaterais. Procure colocar nos casos de testes corretos o código que você usou no método **main**. Você perceberá que ficará muito melhor de você realizar manutenções em seu código de testes quando se usa JUnit. Tenha em mente sempre testar seu código nas situações de limite: **onde deve funcionar, onde não deve funcionar**. Isso será essencial para detectar erros (lembre-se que o objetivo do teste é encontrar erros e não mostrar a ausência deles!).
4. Em seguida execute sua classe como um JUnit Test Case e observe os erros reportados durante a execução. A retirada de erros em casos de teste JUnit com Eclipse pode ser feita (e isso é recomendado) com o uso do Debugger. Neste caso os breakpoints podem ser colocados no próprio caso de teste.