

Instruções:

- Essa prática deve ser resolvida em **até duas pessoas**, sem exceção;
- A resolução (o código fonte) deve ser anexado juntamente a um arquivo texto (.TXT) com informações sobre os componentes do seu *PSasaS*, bem como instruções de como e onde executá-lo (caso julgue necessário);
- Plágio e soluções copiadas de outros colegas, implicarão na nota ZERO a todos os envolvidos;

Processo de *Software as a Software*

Quando desenvolvemos ou evoluímos um *software*, transformamos em código uma enorme quantidade de informações provenientes de um (ou mais) processos de negócio (ou analíticos). A execução desse código representa o instanciamento desse processo que poderia estar envolvido em levar o homem a Marte. Isto posto, representar um processo de *software* por código (ou programação) é análogo ao cenário supracitado.

Nesse trabalho você deverá criar um *software* que materialize um processo de software completo. Calma, não são todos os sub-processos. Para esse trabalho, seu *software* deverá contemplar os elencados abaixo.

- Processo Primário
 - Requisitos → Projeto → Construção → Teste → Implementação

Quando materializamos um processo de *e-Commerce* (por exemplo), combinamos vários componentes (objetos ou funções) responsáveis por atividades específicas. Materializar um processo de *software* é análogo! Você precisará:

1. decompor os sub-processos em componentes menores e atribuir sua responsabilidade (quais são elementos essenciais fazem parte do processo de requisitos?)
2. determinar ordem de chamadas e dependências entre os componentes (como os elementos de requisitos se relacionam?)
3. combinar todos os componentes de cada sub-processo

Para proporcionar um pequeno *insight*, o pseudo-código a seguir reflete uma possível interpretação de um dos elementos do sub-processo de “Teste”: o “Teste Unitário”. Note que utilizei mensagens (o famoso *print*) para destacar os momentos.

Algorithm 1 TesteUnitario

Require: ComponenteExecutavel, CasosTeste

```
1: numCasos ← Conta(CasosTeste)
2: print “Início de teste unitário de componente...”
3: for N = 1 to numCasos do
4:   print “Teste unitário número: ”numCasos
5:   resultadoObtido ← ExecutaTesteCódigo (ComponenteExecutavel, CasosTeste[N].dados)
6:   if resultadoObtido ≠ CasosTeste[N].resultadoEsperado then
7:     CasosTeste[N].situacaoTeste ← falhou
8:   else
9:     CasosTeste[N].situacaoTeste ← ok
10:  end if
11: end for
12: return CasosTeste
```

Seu *PSasaS* poderá utilizar a abordagem estruturada, orientada a objeto ou funcional. Além disso, construa-o com a linguagem de programação de sua escolha (Java, Python, C++, R, por exemplo). Não é necessário qualquer tipo de

persistência dos dados (banco de dados. XML, etc.). Importante enfatizar que a **qualidade do código não será avaliada**, mas sim o quanto esse comunica o seu entendimento de um processo de *software*.

Bom trabalho!

Critérios de Avaliação

Serão utilizados critérios de avaliação elencados abaixo cada qual com o valor igual a 10. A nota final desse trabalho será a média aritmética das notas atribuídas a cada critério.

- Completude: Foram contemplados os principais elementos de cada sub-processo?
- Consistência: Os elementos contemplados representam cada sub-processo de modo consistente?
- Apresentação: São apresentados os momentos dos processo, isto é, a transição do processo (de requisitos para projeto e assim por diante) é claramente apresentada?
- Funcionamento: o PSasaS executa?

Bibliografia de Referência

Consulte o final da página de apresentação da disciplina e o último *slide* da “Aula 1 - Fundamentação da Engenharia”. Nessas são apresentadas boas referências.