

- 1) O que significa uma relação de uso entre módulos no projeto de software? Qual é a importância das relações de uso na melhoria de qualidade do projeto? (Valor: 2,0 pontos)

Uma relação de uso é uma relação direcionada e estática que ocorre entre dois módulos quando um módulo depende do outro. A dependência pode ocorrer pelo uso de uma rotina, uma variável ou um tipo definido no segundo módulo. Em uma relação de uso, um módulo pode ser visto como cliente e o outro módulo como fornecedor de serviços. Em um bom projeto de software, o comportamento de um módulo cliente deve ser compreensível sem a análise detalhada de seus fornecedores.

- 2) Porque dizemos que a geração de um “resultado único” é condição para caracterizar um trabalho como um projeto ao invés de uma operação continuada? (Valor: 2,0 pontos)

Operações e projetos compartilham diversas características. Ambos são realizados por pessoas, são sujeitos a limites em sua capacidade de consumir recursos e são (ao menos, deveriam ser) planejados, executados e controlados. Enquanto as operações representam tarefas que se repetem no dia-a-dia da organização, um projeto é definido como um esforço temporário, realizado com o intuito de criar um produto ou serviço único. Sendo assim, a realização de um projeto presume a criação de algo novo, que diferencie o trabalho realizado no contexto do projeto daquele realizado para entregar produtos ou serviços anteriores.

- 3) Considerando o requisito descrito abaixo, especifique os casos de teste (valor de entrada; resultado esperado) utilizando **particionamento por equivalência** que permita testar o tempo de disparo do disjuntor quando simulado por software (valor 2,0 pontos):

".... um disjuntor elétrico (sistema de proteção elétrico que desliga a energia em caso de excesso de uso) funciona em diferentes valores de tensão e corrente elétrica e seu tempo de disparo é relativo a carga (potencia) instalada. As faixas de tensão e corrente para seu funcionamento são [50...380] volts e [2...40] amperes. A potência de funcionamento (em VA) é calculada pela formula $P = \text{Tensão (V)} \times \text{Corrente (A)}$. O tempo de disparo (em ms) é dado pela formula $TP = 1 / P$, ou seja, quanto maior a potência, mais rápido o disjuntor deve disparar. Valores abaixo da potência mínima não provocam sua atuação. Usualmente, o disjuntor é utilizado em instalações com tensão de 127, 220 e 380 V e qualquer corrente dentro dos limites estabelecidos"

	Inválida	Válida	Inválida
Tensão	< 50	$50 \leq T \leq 380$	>380
Corrente	< 2	$2 \leq C \leq 40$	>40

Entretanto, considerando as 3 faixas de tensão de trabalho (127, 220 e 380V) e suas correspondentes correntes mínima (2 A) e máxima (40 A) vemos que as faixas válidas e inválidas devem ser redistribuídas, da seguinte forma:

	Inválida	Válida	Inválida
Tensão	< 50	$50 \leq T \leq 380$	> 380
Corrente	< 2	$2 \leq C \leq 40$	> 40

Assim teremos os seguintes casos de teste (exemplos, valores pertencentes as classes de equivalência válida e inválida são corretos):

CT={ (49, 1, Não Dispara); (380,40, Não Dispara); (127,42, Dispara em 1/5334); (381, 41, Disparo Automático) }

- 4) Explique que diferenças existem entre atividades de manutenção de correção e melhoria, e indique 3 fatores que podem afetar o esforço de manutenção de software e 3 responsabilidades da equipe de manutenção. (valor: 2,0 pontos).

Manutenção de Correção: Corrige defeitos – i.e. diferenças (não conformidades) entre o comportamento requerido para um produto/aplicação e o comportamento observado.

Manutenção de Melhorias: Implementa mudanças no sistema que evolui seu comportamento ou implementação. Melhorias podem ser: Trocar requisitos (Manutenção Perfectiva), Adicionar novos requisitos ao sistema (Manutenção Adaptativa) e Trocar a implementação mas não o requisito (Manutenção Preventiva)

Fatores: Tipo de aplicação, Novidade do sistema, Rotatividade e disponibilidade do pessoal de manutenção, Duração da vida útil do sistema, Dependência de um ambiente que se modifica, Característica de hardware, Qualidade do projeto, Qualidade do código, Qualidade da documentação, Qualidade dos testes.

Responsabilidades da Equipe: entender o sistema, localizar informação na documentação do sistema, manter a documentação do sistema atualizada, estender as funções existentes para acomodar novos requisitos ou modificações nos requisitos, acrescentar novas funções para o sistema, encontrar a fonte de falhas ou problemas no sistema, localizar e corrigir faltas, responder questões sobre a forma como o sistema funciona, reestruturar o projeto e codificação dos componentes, reescrever o projeto e código dos componentes, apagar os projetos e códigos de componentes que não são mais úteis, gerenciar trocas para o sistema.

- 5) Defina o que representam os conceitos de acoplamento e coesão em software. Apresente os problemas que podem ocorrer quando estes são observados no software. Por exemplo, considere o que pode ocorrer quando se apresentam em excesso ou deficiência. Sugira uma métrica que pode ser usada para observar cada conceito e exemplifique, através de um diagrama de classes, o conceito de acoplamento através de classes acopladas e não acopladas. (Valor: 2,0 pontos)

Acoplamento: Intuitivamente, refere-se ao grau de interdependência entre diferentes pedaços de um projeto.

Comentário: Desde que objetos de uma mesma classe possuem as mesmas propriedades, duas classes estão acopladas quando métodos declarados numa classe utilizam métodos ou variáveis de instância de outra classe.

Coesão: Intuitivamente, refere-se à consistência interna dos pedaços de um projeto.

Comentário: O grau de similaridade de métodos pode ser visto como o maior aspecto de coesividade das classes/objetos. Se uma classe/objeto tem diferentes métodos executando diferentes operações para um mesmo conjunto de variáveis de instância, a classe é coesa.

Uma métrica que pode ser usada para medir acoplamento é a CBO (*coupling between objects*). Basicamente ela mede o numero de relacionamentos existente entre as classes que irão instanciar os objetos.

```

classDiagram
    class Customer {
        name : Strings
        address : Strings
        SSN : Strings
        birthday : DateTime
        account_number : long
        create_customers()
        create_a_customer()
    }
    class "gas station" {
        name : Strings
        gasstation code : int
        dormant_accounts()
    }
    class services {
        discount rate : float = 0.0
        price()
    }
    class bill {
        issue date : DateTime
        payment date : DateTime
        price()
        taxes()
        customer()
        list_purchases()
    }
    class Purchase {
        date : DateTime
        tax : float = 0.05
        price()
        taxes()
    }
    class refuel {
        gallons : float = 0.0
        price()
    }
    class gas {
        min_quantity = 100
        current_quantity
        price = 1.09
    }
    class product {
        min_quantity : float
        current_quantity : float
        price : float
    }
    class inventory {
        order_gas()
        order_part()
    }
    class parking_spot {
        place : Strings
        is_available()
        next_available()
    }
    class parking {
        price : float = 5.00
    }
    class Part {
        part_code
        discount rate
    }
    class "message" {
        text : Strings
    }
    class "warning letters"
    class "periodic messages"
    class "CC system (from External)"
    class "parts ordering system (from External)"
    class "gas ordering system (from External)"

    Customer "0..*" -- "1" Purchase
    Customer "0..*" -- "1..*" services
    Customer "0..*" -- "1..*" "gas station"
    "gas station" "1..*" -- "1" services
    "gas station" "1..*" -- "1" parking_spot
    "gas station" "1..*" -- "1" parking
    "gas station" "1..*" -- "1" inventory
    "gas station" "1..*" -- "1" "gas ordering system (from External)"
    "gas station" "1..*" -- "1" "parts ordering system (from External)"
    "gas station" "1..*" -- "1" "CC system (from External)"
    bill "1..*" -- "1" Purchase
    Purchase "1" -- "0..*" services
    Purchase "1" -- "0..*" "gas station"
    refuel --|> services
    gas --|> services
    gas --|> product
    product --|> inventory
    product --|> Part
    parking_spot --|> parking
    parking --|> inventory
    message --|> "warning letters"
    message --|> "periodic messages"
    
```