

### ***Perguntas conceituais rápidas:***

1. O que é um módulo? Quais as duas características importantes?
2. Por que desenvolvemos programas de forma modularizada? Explique e compare brevemente acoplamento, coesão e encapsulamento. Dê um exemplo para ilustrar a diferença.
3. Explique falta, erro e falha.
4. O que é programa correto? É possível criar um programa correto sem especificação?
5. Quais as diferenças entre teste manual e teste automatizado?
6. Explique o que é análise estática e análise dinâmica. Dê exemplos de como revisões e testes se complementam.
7. Dê 2 exemplos de como revisões e testes se complementam.
8. Explique a diferença entre requisito funcional e requisito não-funcional.
9. Explique benefícios da identificação explícita de requisitos e modelagem da arquitetura antes de iniciar a implementação dos módulos.
10. Qual a diferença entre modelagem conceitual e modelagem física?
11. O que são: declaração e definição de variáveis? Dê 3 exemplos diferentes (e explique) de cada caso.

### ***Questões práticas***

#### **Questão 1**

Um artigo é formado por um título e um ou mais capítulos. Um capítulo é formado por um cabeçalho e um ou mais parágrafos. Cada parágrafo é formado por zero ou mais palavras. Cada título, cabeçalho e parágrafo possui uma formatação que vale para o parágrafo inteiro. Os componentes da formatação são sempre os mesmos e estão contidos em um descritor de formato.

Produza um modelo **conceitual** para a estrutura de um artigo.

**Questão 2.** Considere o trecho de código abaixo que contém anomalias e viola certos padrões de programação.

- a) realize uma inspeção do código abaixo e explique 3 destes problemas.
- b) uma inspeção de código rigorosa elimina a necessidade de testes? Dê 2 justificativas para sua resposta.

```
int TOTAL_CONVERSOES;

long Conversao (short baseCalculo, int opcao) {
    long Resultado;

    if (opcao == 1 || opcao == 2 || opcao == 3) {
        Resultado = *((long *) &baseCalculo );
    }
}
```

```

    } else {
        Resultado = *(( int *) &baseCalculo );
    }

    TOTAL_CONVERSOES += 1;
    return Resultado;
}

```

**Questão 3.** Um dos requisitos não-funcionais típicos que são inerentemente associados com programação modular é manutenibilidade, isto é, a facilidade de modificar um sistema. Redija um texto explicativo (não exceda uma página) descrevendo quais foram os princípios e técnicas da programação modular, adotados no desenvolvimento do seu 1º trabalho da disciplina. Como a aplicação de tais princípios e técnicas contribui para que módulos e casos de teste sofram poucas alterações caso a modificação para dois cursos fosse implementada? Caso anomalias de modularidade existam no seu trabalho, você também pode indicar no seu texto quais re-trabalhos inúteis poderiam ter sido evitados se você tivesse seguido decisões diferentes. Use exemplos concretos do seu trabalho que dêem suporte aos seus argumentos. Sua resposta deve mencionar, ao menos, técnicas e princípios básicos da programação modular, tais como: encapsulamento, abstração, interface, acoplamento, coesão, especificação, e teste de módulo.

**Questão 4.** Considerando um módulo que implementa uma árvore binária genérica, cite 2 vantagens da opção de implementar uma árvore com cabeça sobre a opção de implementar uma árvore identificada pela referência a sua raiz?

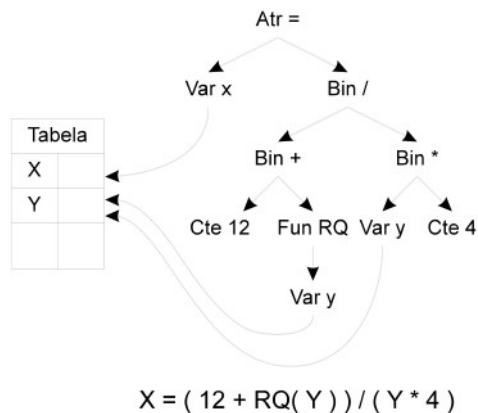
**Questão 5.** Projete a linguagem do script para testar completamente o módulo lista ligadas a listas das disciplinas e suas respectivas turmas que constam no trabalho 2. Exemplos de script não serão aceitos.

### Questão 6

Por que não devem estar no módulo de implementação as declarações dos dados e das funções globais externas, bem como as respectivas especificações?

### Questão 7

A figura a seguir exemplifica uma árvore de derivação obtida a partir da análise sintática de uma fórmula. Desenhe o modelo físico e redija as assertivas estruturais utilizados para armazenar essa estrutura em memória real, assumindo que a tabela de símbolos seja implementada utilizando uma árvore n-ária ordenada.

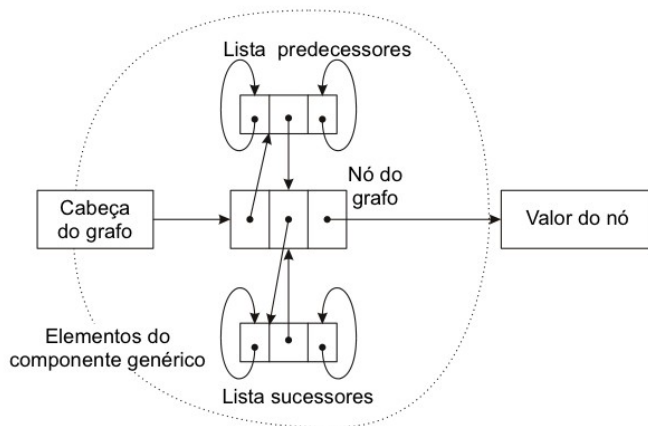


### Questão 8

Desenhe o modelo físico da estrutura que armazene um livro com índice analítico (índice utilizado para encontrar a página de determinado capítulo ou seção). Apresente um exemplo e as assertivas estruturais.

### Questão 9

A figura abaixo modela um grafo genérico. Como você complementaria o protótipo da função `void GRF_ProcurarNo( void * ValorProcurado , ... );` de modo que um módulo cliente do módulo grafo possa solicitar a procura por um nó contendo um determinado valor cujo tipo somente o cliente conhece. A função atualiza o campo `noCorrente` contido na cabeça do grafo. E como o módulo cliente chamaria esta função?



### Questão 10

O que são padrões de gerenciamento? Do ponto de vista qualidade de software, qual é a finalidade dos padrões de gerenciamento? Ilustre a resposta com um padrão que tenha sido fundamental para o sucesso do 2o. trabalho.

### Questão 11

A quem se destina um módulo de definição? Em no máximo 10 linhas, qual é o seu conteúdo típico? Explique, através de exemplos, a importância dos módulos de definição no segundo trabalho.

### Questão 12

O que é encapsulamento? Qual a relação entre cabeças de estruturas de dados e encapsulamento? Exemplifique.

### Questão 13

A quem se destina um módulo de definição? Redija o módulo de definição de um módulo que tenha por objetivo criar e manter árvores binárias de pesquisa. A organização da estrutura de dados usada deve estar encapsulada.

### Questão 14

Assuma que o comando utilizado para compilar um módulo de nome `mod.lng` seja:

```
cmplng mod.lng _mod.obj
```

O comando utilizado para ligar os módulos `m1.obj` e `m2.obj` gerando o programa `p.exe` é:

```
ligar m1.obj, m2.obj _p.exe
```

A tabela a seguir identifica os arquivos de inclusão dos módulos necessários para gerar o programa `XPTO.EXE`:

```
abc.lng abc.def, x1.inc, const.inc
cde.lng cde.def, zyz22.inc
ghj.lng ghj.def, abc.def, cde.def, const.inc
zyz22.inc prk30.inc
```

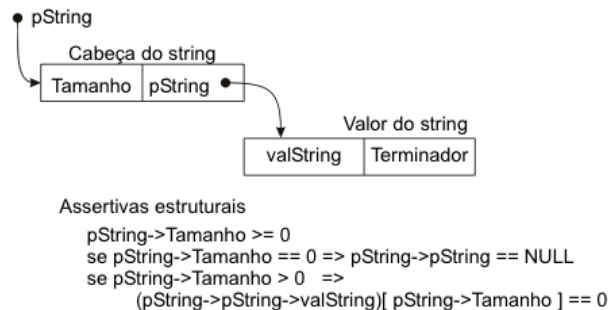
Esboce o *script* (arquivo `xpto.mak`) que seria utilizado pelo programa `MAKE` para controlar a compilação do programa `XPTO.EXE`.

### Questão 15

Considere uma lista duplamente encadeada contendo uma cabeça que, além de apontar para a origem da lista (`pOrigem`) contém outras referências para nós da lista. Com pequenas alterações essa estrutura pode ser utilizada como uma fila. Quais são essas alterações? Qual é o modelo físico da fila e quais as respectivas assertivas estruturais?

### Questão 16

A seguir é apresentada o modelo da estrutura que implementa um *string* seguro.



(1) Os nomes contidos nesse modelo satisfazem os padrões estabelecidos no livro? Explique cada caso.

### Questão 17

Quais as diferenças e similaridades entre as seguintes técnicas de garantia de qualidade: teste, medição e inspeção de código fonte?

### Questão 18

Explique **sucintamente** o que é um módulo, ressaltando a diferença entre sua definição física e lógica. Como a utilização de programas modulares pode contribuir para organizações de desenvolvimento de software? Utilize de forma coerente em sua explicação os conceitos: encapsulamento, coesão, tipo abstrato de dados, e acoplamento. Dê exemplos relacionados aos trabalhos 1 e 2 para ilustrar suas explicações.

**Questão 19**

Explique os diferentes tipos de coesão. Quais ocorreram mais recorrentemente no código dos módulos do trabalho 1. Por quê?

**Questão 20**

Suponha que você é solicitado para modelar uma estrutura para construir um programa para posicionar peças sobre um tabuleiro. É possível utilizar esta mesma estrutura para um jogo de damas? Apresente o modelo físico da estrutura, um exemplo físico para xadrez e outro para damas.

**Questão 21**

O que é uma cabeça (*header*) de uma estrutura de dados? Por que e quando utilizá-las? Quais são os cuidados a tomar quando se atribui uma variável contendo um valor tipo estrutura de dados (por exemplo lista) a uma outra variável? Não se esqueça de dar exemplos.