

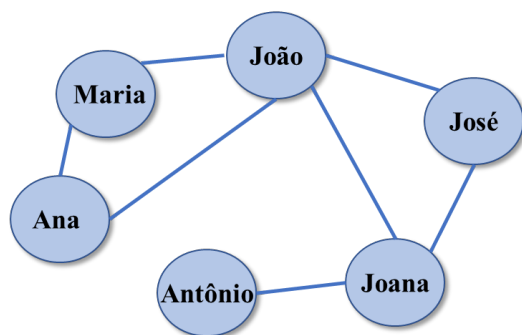
Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação de Aplicações Web
Professores: Flávio L. Seixas e Miguel Elias M. Campista
AD1 – 2º Semestre de 2021

Aviso: As respostas da AD devem ser entregues obrigatoriamente em um único arquivo, no formato PDF. Outros formatos não serão aceitos, i.e., a prova será zerada. Assinale a que questão se refere cada resposta ou coloque o enunciado acima da resposta. Em caso de cópia ou plágio, a nota da AD será dividida pelo número de alunos que realizaram a prova em grupo.

Esta AD avalia o uso das estruturas básicas de repetição e condição, a criação e uso de funções, manipulação de vetores e programação orientada a objetos na linguagem PHP.

- 1) Uma estrutura em grafo é utilizada para modelar relações entre pares de elementos de um determinado conjunto. Os elementos são chamados de vértices e as relações entre pares, quando existirem, definem o conjunto de arestas. Tomemos um exemplo prático: um conjunto de alunos do CEDERJ pode ser modelado como um conjunto de vértices de um grafo enquanto as arestas podem representar relações entre pares de alunos que pertençam a uma mesma turma. A figura abaixo ilustra o grafo resultante do seguinte conjunto de alunos $V = \{\text{Maria, João, José, Ana, Antônio, Joana}\}$, sabendo que Maria, João e Ana foram da mesma turma de PAW em 2021; João, José e Joana da mesma turma de redes em 2020; e Antônio e Joana da mesma turma de sistemas operacionais também em 2020.

Note que é possível representar o grafo através de uma matriz de adjacências $M = V \times V$, onde $M[i][j]=1$ quando a aresta entre i e j existir. Na figura abaixo, $M[i][j]=1$ indica que eles pertenceram a uma mesma turma; enquanto $M[i][j]=0$ indica que eles nunca pertenceram a uma mesma turma.



$$M[i][j] = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- A) **(2,0)** Deseja-se desenvolver em PHP uma função para desenvolver grafos cujas arestas entre os vértices possuem uma probabilidade p de existir (grafo conhecido pelo nome de Erdős-Rényi, $G(n, p)$), onde n é o número de vértices. Esta função deve retornar a matriz de adjacências correspondente. Note que a implementação das funções deve respeitar as assinaturas abaixo, na qual o parâmetro `$vetorVertices` contém o vetor de vértices (p.ex., o vetor v da figura acima) e o parâmetro `$probabilidade` é um número real entre 0 e 1 usado para definir se cada uma das arestas entre os elementos de v existem ou não.

Dica: A implementação da função deve considerar cada uma das possíveis arestas do grafo através da combinação dois a dois de todos os vértices. A aresta existe se o valor retornado por uma função aleatória for menor que o valor da probabilidade utilizada. A função aleatória deve retornar valores pertencentes ao mesmo intervalo real de probabilidades, ou seja, entre 0 e 1.

```
function cria_grafo($vetorVertices, $probabilidade)
```

- B) **(2,0)** Deseja-se desenvolver em PHP uma função para atualizar uma matriz de adjacências, considerando a adição de um novo vetor de vértices. Note que todas as arestas já representadas na matriz de adjacências devem ser preservadas. Note também que o novo vetor de vértices pode conter arestas para vértices já existentes na matriz original. Esta função deve utilizar passagem de parâmetro por referência para atualizar a matriz de adjacências, gerando arestas considerando a mesma probabilidade p . Note que a implementação das funções deve respeitar as assinaturas abaixo, nas quais o parâmetro `$vetorVerticesNovos` contém o vetor com os novos vértices, `$matriz` é a matriz de adjacência atual e `$probabilidade` é a mesma probabilidade do item anterior.

```
function atualiza_matriz($vetorVerticesNovos, &$matriz, $probabilidade)
```

- 2) (2,0) A função `esconde_senha($texto)` deve esconder a senha, de tamanho arbitrário e escrita em claro, em uma string de mesmo tamanho, mas composta apenas por asteriscos. Para isso, a função DEVE seguir as seguintes etapas: receber uma string, encontrar por meio de expressão regular o rótulo "senha:", ignorando espaços intermediários ("senha: " ou "senha:"), substituir a sequência de caracteres que aparece em seguida por * e retornar a string produzida.

Exemplo:

Para `$texto = "Esta conta tem senha: 12345."` a função deve retornar `"Esta conta tem senha: *****."` ou ainda para o `$texto = "Este outro exemplo tem uma senha:123 que pode ser mais curta e sem espaço após os dois pontos."` a função deve retornar `"Este outro exemplo tem uma senha:*** que pode ser mais curta e sem espaço após os dois pontos."`

Note que a substituição ocorre até que um espaço em branco ou uma pontuação (.,!?) é encontrada.

- 3) (4,0) Você foi contratado para desenvolver em PHP o sistema de controle de alunos do CEDERJ. Utilizando o paradigma de orientação a objetos, você mapeou as seguintes classes: `Aluno`, `Disciplina` e `TurmasDisciplina`. A dinâmica é a seguinte: os alunos se inscrevem nas disciplinas e o sistema deve informar quantas turmas serão necessárias e quais os alunos inscritos em cada uma delas. O número de turmas depende do número máximo de alunos por turma (`nmax`). Portanto, se o número de inscrições passar o número máximo de alunos por turma, o sistema deve abrir uma nova turma e assim por diante até que todos os alunos estejam em uma turma.

Os alunos são organizados por turma por ordem de CR (Coeficiente de Rendimento - média ponderada de todas as disciplinas cursadas). Assim, os alunos com os `nmax` maiores CRs são alocados na primeira turma e assim por diante. Ao final, garante-se que todos os alunos são alocados em uma turma da disciplina. Não há limite máximo para o número de turmas.

A listagem abaixo mostra os atributos das classes, o código usado para instanciar as classes e testar a lógica da programação. Ao final, espera-se que seja exibido na tela o número de turmas e a lista, por turma, de nomes de alunos e respectivos CRs. Pede-se para apresentar a implementação completa, levando em conta que a listagem abaixo pode omitir alguns métodos e atributos, que devem ser parte da resposta.

```
class Aluno {  
  
    private $nome, $cr;
```

```

    public $getNome();

    public inscreveDisciplina($disciplina);

    ...

class Disciplina {

    private $alunosInscritos = array();

    private $nomeDisciplina;

    public getAlunosInscritos();

    ...

class TurmasDisciplina {

    private $turmas = array(); // array de arrays de turmas

    private $nmax;

    public imprimeTurmas();

    public calculaTurmas($disciplina);

    ...

$aluno1      = new Aluno ("João", 8.5);

$aluno2      = new Aluno ("Ana", 9.0);

$aluno3      = new Aluno ("Maria", 9.5);

    ...

$disciplinal = new Disciplina("PAW");

$turmasDisciplina = new TurmasDisciplina(2); // $nmax = 2

$aluno1->inscreveDisciplina($disciplinal);

$aluno2->inscreveDisciplina($disciplinal);

$aluno3->inscreveDisciplina($disciplinal);

```

```
$turmasDisciplina->calculaTurmas($disciplinal);
```

```
$turmasDisciplina->imprimeTurmas();
```

```
// Imprime na tela o número de turmas e as listas de alunos com  
respectivos CRs.
```