

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUI</p>	<p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ</p> <p>Curso: ADS</p> <p>Disciplina: Programação Orientada a Objetos</p> <p>Professor: Ely</p>
--	--

Trabalho Final

Crie um sistema de rede social em equipes de até 3 pessoas com a possibilidade de gerenciar perfis e publicações conforme as instruções a seguir.

1) Implementação das Classes Básicas (2,0 pontos)

- **Classe Perfil:**
 - Atributos:
 - ID único, apelido, foto (emojis), email, status (ativo/inativo), array de amigos, e array de postagens.
 - Métodos:
 - Adicionar amigo;
 - Remover amigo;
 - Adicionar publicação;
 - Listar amigos;
 - Listar postagens;
 - Ativar/desativar perfil.
- **Classe PerfilAvancado:**
 - Herança da classe Perfil.
 - Método:
 - Habilitar ou desabilitar outros perfis.
- **Classe Publicacao:**
 - Atributos:
 - ID único, conteúdo, data/hora da publicação e perfil associado.
- **Classe PublicacaoAvancada:**
 - Herança da classe Publicacao.
 - Atributo:
 - Lista de interações.
 - Métodos:
 - Adicionar interação.
 - Listar iterações.
- **Classe Interacao:**
 - Atributos:
 - ID único, tipo de interação (emoji), perfil autor da interação.
- **Enumeração TipoInteracao:**

- Valores permitidos: "curtir", "não curtir", "riso", "surpresa" representados através de emojis.

Criar as classes com atributos privados, inicializados nos construtores e com métodos de leitura e escrita, quando necessários.

2) Implementação da Classe RedeSocial (2,0 pontos)

- Atributos:
 - Array de perfis cadastrados;
 - Array de todas as publicações da rede social;
 - Mapa solicitações de amizade pendentes (map<Perfil, Perfil>)
- Métodos:
 - Gerenciamento de Perfis:
 - Adicionar perfil;
 - Buscar perfil por email, apelido ou id;
 - Listar todos os perfis;
 - Ativar/desativar perfil por meio de perfil avançado.
 - Gerenciamento de Publicações:
 - Adicionar publicação simples ou avançada associadas a perfis;
 - Listar publicações ordenadas em ordem decrescente, podendo ser filtradas por perfil.
 - Gerenciamento de solicitações:
 - Enviar solicitações de amizade;
 - Aceitar solicitações;
 - Recusar solicitações.
 - Gerenciamento de Interações:
 - Adicionar interações a publicações avançadas, garantindo que um usuário não interaja mais de uma vez na mesma publicação.

Nessa classe, devem ser passados aos métodos preferencialmente ids, apelidos, e-mails ou outros atributos únicos. Daí os objetos são consultados nos arrays e a partir daí serão manipulados.

3) Simulação Interativa (2,0 pontos)

- Implemente uma interface interativa que permita executar os métodos de Gerenciamento de Perfis, publicações, solicitações e interações;
- Utilize uma interface de terminal que permita ao usuário informar as opções e fornecer os dados necessários;
- Seja criativo e, sempre que possível, liste opções para o usuário escolher previamente, minimizando a necessidade de memorizar códigos/e-mails e outros identificadores;
- Dê o feedback a cada interação, exibindo o que foi solicitado com sucesso ou que houve erro;
- Para funcionalidades mais “sérias”, solicite confirmação.

4) Persistência de Dados (2,0 pontos)

- Salvar Dados em Arquivo:
 - Perfis e publicações devem ser salvos em arquivos JSON, separados por tipo e evitando redundância.
- Recuperar Dados de Arquivo:
 - Os dados salvos devem ser lidos e reconstruídos em memória, preservando as associações entre perfis e publicações.

5) Validações e Tratamento de Exceções (2,0 pontos)

Adicione pelo menos as exceções personalizadas abaixo para tratar casos específicos como:

- Gerenciamento de Perfis:
 - IDs ou emails duplicados devem lançar a exceção `PerfilJaCadastradoError`;
 - Perfis normais tentando ativar/desativar status de outro perfil devem lançar a exceção `PerfilNaoAutorizadoError`.
- Gerenciamento de Publicações:
 - Perfis inativos tentando criar publicações devem lançar a exceção `PerfilInativoError`;
 - Interações duplicadas devem lançar a exceção `InteracaoDuplicadaError`.
- Gerenciamento de solicitações:
 - Amizades duplicadas devem lançar a exceção `AmizadeJaExistenteError`.
- Gerenciamento de interações:
 - Interações duplicadas devem lançar a exceção.
- Entradas de dados inválidas e demais exceções:
 - Lançar a exceção `ValorInvalidoException` quando um valor inapropriado for informado;
 - Capturar os demais erros de forma genérica e apropriada exibindo uma mensagem amigável.

Esteja atento a nunca deixar a aplicação parar de executar por conta de erros não tratados.

Entrega

- Código no github;
- Apresentação ao professor por toda a equipe. Apenas membros presentes pontuarão;
- Vídeo no repositório de até 15 min demonstrando as 5 questões. Edite o vídeo para torná-lo mais dinâmico.