

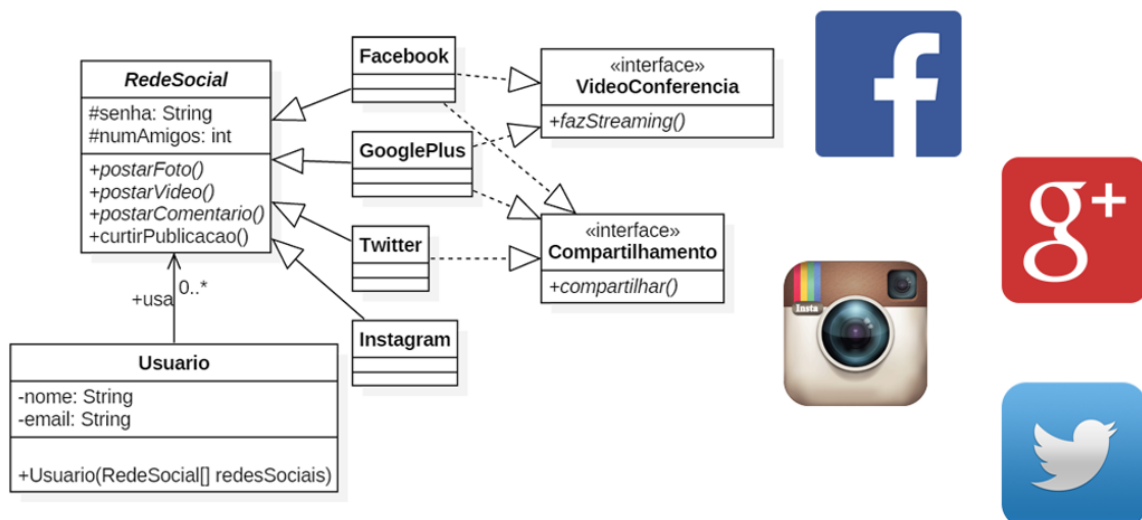
**C206/C06 - Programação Orientada a Objetos**  
**Exercício prático**  
**Professor: Christopher Lima**

**Instruções:**

- Utilize os slides, livros e aulas como referência para resolver o exercício.
- Códigos realizados durante a aula podem ser encontrados em:  
[https://github.com/chrislima-inatel/C206\\_C125](https://github.com/chrislima-inatel/C206_C125)
- Objetivos: Aprimorar os conhecimentos na manipulação de estruturas de dados e orientação a objetos.
- Ao final, o link do repositório com a solução deve ser enviada. Caso seu repositório seja privado, adicionar os seguintes membros como colaboradores do repositório (utilizando o nome no github): **chrislima**, **mathzpereira** (turma A) e **LauraPivoto** (turma B). Envie o link correto com a solução desenvolvida.
- Não esquecer de enviar o projeto corretamente (não enviar somente as classes).
- Atividade individual.
- Não deixe de contactar o professor/monitor caso tenha **QUALQUER** dúvida.

Bons Estudos!

As Redes Sociais são ferramentas essenciais nos dias de hoje! Apesar de serem ferramentas muito interessantes e possuírem características específicas, elas também compartilham muitas funcionalidades em comum. Crie Classes em Java que atendam às seguintes regras e especificações abaixo:



## Especificações importantes:

- Nas reescritas de métodos, simplesmente deverá ser colocado um `System.out.println` demonstrando suas ações: Ex: “postou uma foto no Facebook!”, “curtiu uma publicação no Facebook!”, “realizou uma vídeo conferência no Facebook” e assim por diante;
- A Classe `RedeSocial` é abstrata, e seus métodos podem ser abstratos ou não (observar quem está em *itálico*);
- Após realizada a implementação do modelo, deverá ser criado um usuário e deveremos fazê-lo utilizar de duas redes sociais (*Main*);
- Todas as informações do modelo deverão ser preenchidas;
- Na criação do usuário é mandatório que se use do construtor com parâmetros para adicionar suas redes sociais;
- É mandatório que se use pelo menos uma rede social que seja capaz de prover o compartilhamento de posts;
- Depois de adicionadas as redes sociais, cada rede social deverá executar 3 (três) métodos diferentes (ou seja, os métodos que forem executados em uma rede social não poderão ser chamados na outra) a partir da coleção de redes sociais do usuário;
- Crie 2 Classes para representar duas Exceções diferentes. Uma delas deve ser do tipo *Unchecked* e a outra *Checked*. Trate-as corretamente no seu código (*Try-Catch*);
- Crie um pacote para separar as classes de exceções e outro para o código da funcionalidade.
- Utilize um ***HashSet*** ou ***HashMap*** para representar a coleção de `RedeSocial` presente em `Usuario`.