Exercícios - Semana 14

Nas semanas 10 e 11, você teve contato com o padrão de projeto *Strategy,* um **padrão comportamental**. Nas semanas 12 e 13, você teve contato com o padrão de projeto *Bridge,* um **padrão estrutural**. Nesta semana, você terá contato com um **padrão de criação**: o padrão *Factory*, e suas variações: Simple Factory, Factory Method e Abstract Factory.

No projeto desta semana (redes sociais), cada rede social é instanciada em profile (main.py). Há duas opções: linkedin e facebook.

Desafio 1 (3 pontos)
Os códigos estão funcionando, mas não foi criado testes. Crie os testes para as implementações de redes sociais (redes_sociais.py) e sessões (sessoes.py) .

Desafio 2 (3 pontos)
Implementar a rede social **Github**. Crie uma sessão UploadCode.
Crie testes para suas implementações.

Desafio 3 (2 pontos)
Implementar a rede social **Instagram**.
Crie testes para suas implementações.

Desafio 4 (Resposta livre – até 2 pontos)
"Herança é provavelmente a maneira mais fácil de estender o comportamento padrão de uma biblioteca ou framework. Mas como o framework reconheceria que sua subclasse deve ser usada em vez de um componente padrão?
A solução é reduzir o código que constrói componentes no framework em um único método fábrica e permitir que qualquer pessoa sobrescreva esse método, além de estender o próprio componente. O Factory Method separa o código de construção do produto do código que realmente usa o produto. Portanto, é mais fácil estender o código de construção do produto independentemente do restante do código."

Trecho do livro "Mergulho nos padrões de projeto"

Você identifica algum código/cenário dentro da Napp em que o padrão de projeto estudado poderia ser aplicado para melhora do produto ou redução de manutenção ?

R: Pelo que vi até o momento da minha jornada na NAPP, a biblioteca napplib já está aplicando este conceito adequadamente, porém novos códigos são desenvolvidos todos os dias, sendo assim esse conceito pode ser aplicado em novos códigos.