

Nome: _____

Nº Mec. _____

1. [2] Descreva os princípios GRASP de "Low Coupling" e de "High Cohesion".

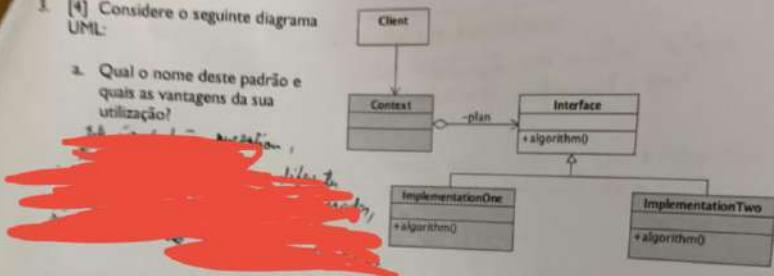
(1 pt) _____

2. [3] Considere o seguinte código que pretende fornecer uma solução para copiar ficheiros entre diferentes dispositivos. Complete-o e/ou reescreva-o, atendendo nomeadamente aos princípios GRASP "Program to an interface, not to an implementation" e/ou "High-level modules should not depend upon low-level modules. Both should depend upon abstractions."

```
public class PenDrive {  
    // attributes and other methods  
    public byte[] read(File fn) { /* read and return data */ }  
    public boolean write(File fn, byte[] data) { /* write data in file fn */ }  
}  
public class NetDrive {  
    // attributes and other methods  
    public byte[] read(File fn) { /* read and return data */ }  
    public boolean write(File fn, byte[] data) { /* write data in file fn */ }  
}  
public class TransferUtils {  
    public static void copyFile(PenDrive usb, File orig, NetDrive ssd, File dest) {  
        byte[] data = usb.read(orig);  
        // work on data e.g transform, compress, encrypt etc..  
        ssd.write(dest, data);  
    }  
    // other static functions...  
}
```

3. [4] Considere o seguinte diagrama UML:

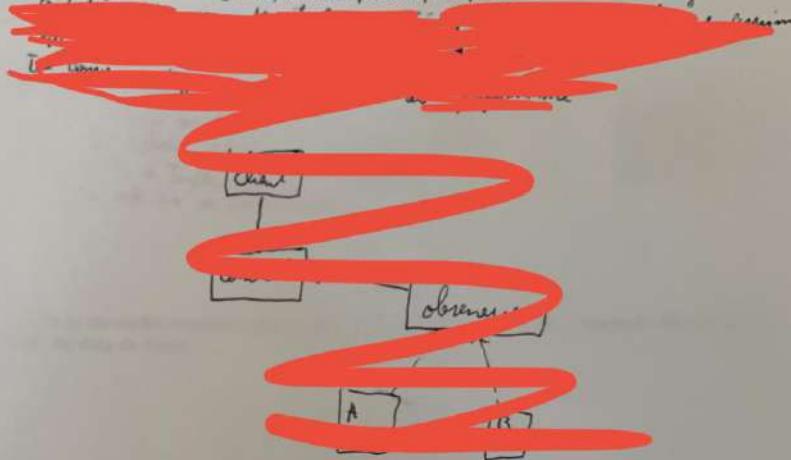
- a. Qual o nome deste padrão e quais as vantagens da sua utilização?



- b. Usando as entidades representadas no diagrama, apresente um pequeno exemplo de código, do ponto de vista de Client.

4. [4] Considere o padrão Proxy e o padrão Decorator. Represente os dois modelos em UML e explique a diferença entre eles (use exemplos).

5. [4] Explique a organização do padrão *Observer*, descrevendo, nomeadamente: a) o modelo L as suas principais vantagens; c) um exemplo em que o padrão é usado.



6. [3] Considere a arquitetura de software apresentada na figura. Identifique e descreva este modelo, indicando as suas principais vantagens e desvantagens.

