

Data: DD/MM/AAAA

Nome: Fulano de Tal

Assinatura: #####

Matrícula: XX/XXXX-X

EXEMPLO DE PROVA (GABARITO)

Disciplina: Arquitetura & Desenho de Software - Aula de Revisão - Profa. Milene Serrano

Questão Única

Contexto:

Um Engenheiro de Software foi contratado recentemente em uma empresa.

Para ele, foi conferido um primeiro desafio.

Esse desafio consistiu na aplicação de padrões de projeto, visando solucionar algumas demandas específicas.

Na empresa, setor financeiro, percebeu-se que muitas requisições ocorrem a partir de operações de entrada do sistema, sendo as principais:

- *consultar_saldo(...)*
- *consultar_extrato(...)*
- *sacar_valor(...)*
- *depositar_valor(...)*

Cada requisição demanda conversar com diferentes classes, implementadas especificamente para lidar com os devidos retornos e as devidas ações.

Entretanto, da forma como o sistema foi modelado/implementado, em um primeiro momento, está bastante confuso esse direcionamento inicial das requisições de entrada, gerando muitas dependências e, portanto, alto acoplamento e baixa coesão.

No intuito de lidar com essa situação, bem como visando reduzir o acoplamento e aumentar a coesão, o Engenheiro de Software atuou orientando-se por padrões de projeto. Na solução, o foco foi nas operações de entrada e no despacho para as respectivas classes internas capazes de tratar cada uma dessas requisições.

PARTE I:

O novo funcionário conferiu uma solução bem elegante para isso.

Para tanto, ele estudou bem a demanda, e logo pensou em uma solução orientada a um padrão de projeto da categoria:

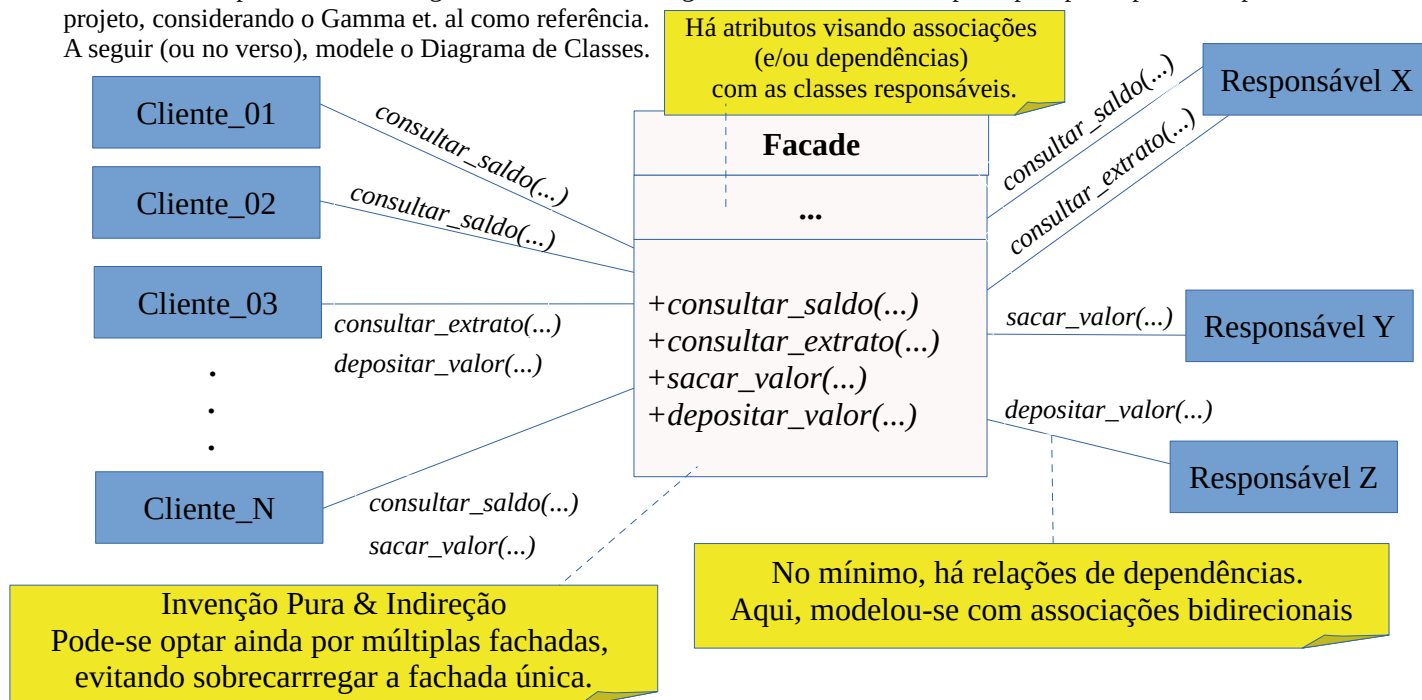
- ☐ GoFs Criacionais
- ☒ GoFs Estruturais
- ☐ GoFs Comportamentais

Qual é o padrão? Facade

PARTE II:

Além disso, ele pensou na modelagem, e elaborou um diagrama de classes com os principais participantes do padrão de projeto, considerando o Gamma et. al como referência.

A seguir (ou no verso), modele o Diagrama de Classes.



PARTE III:

Por fim, ele implementou essa solução.

A seguir, escreva o pseudocódigo da programação, SEM OMITIR PASSOS. Caso prefira, pode usar JAVA (exclusivamente essa linguagem).

- **OBS_01:** Não é para usar apenas comentários, omitindo os pontos de complexidade do código.
- **OBS_02:** O código precisa revelar claramente sobre as particularidades do padrão e seus participantes. Caso contrário, infelizmente, a nota será reduzida.

BOA PROVA! :)

```
// Representando os clientes, cada qual com classes diferentes
// Poderia ser diferente, usando um caixa eletrônico ou outros dispositivos de entrada
class Cliente_01 {
// ...
}
class Cliente_02 {
// ...
}
class Cliente_03 {
// ...
}
class Cliente_N {
// ...
}

// Fachada Única
class Fachada {
// Atributos
+consultar_saldo(...){
// Chama Responsável X
}
+consultar_extrato(...){
// Chama Responsável X
}
+sacar_valor(...){
// Chama Responsável Y
}
+depositar_valor(...){
// Chama Responsável Z
}
```

Em cada cliente,
há necessidade de instanciar a fachada,
visando chamar
o método específico
para a operação de entrada.

Ao longo da programação,
escopos (*public* ou *private*)
podem ser ajustados conforme aspectos
de autenticação e autorização,
bem como uso de *getters* e *setters*
ou métodos equivalentes.

```
// Representando os responsáveis, cada qual com classes diferentes
// Poderia ser diferente, caso as responsabilidades sejam atribuídas de outra forma
class Responsavel_X {
// Atributos
+consultar_saldo(...){
// Trata a consulta ao saldo
}
+consultar_extrato(...){
// Trata a consulta ao extrato
}
}
class Responsavel_Y {
// Atributos
+sacar_valor(...){
// Trata o saque do valor
}
}
class Responsavel_Z {
// Atributos
+depositar_valor(...){
// Trata o depósito do valor
}
}
```

Ao longo da programação, escopos (*public* ou *private*) podem ser ajustados conforme aspectos de autenticação e autorização, bem como uso de *getters* e *setters* ou métodos equivalentes.

Na main, tem-se o seguinte passo a passo:

Instanciar a fachada, sendo algo feito em cada cliente...

Com a instância de fachada em um dado cliente, requisitar a(s) operação(ões)...

Em cada operação da fachada, será chamada a classe responsável correspondente ao tratamento da operação...

Em cada classe responsável, haverá o tratamento da operação, permitindo os retornos, que serão informados para a fachada que, por sua vez, retornará a cada cliente...