

# Trabalho Prático do Módulo 3

**Entrega** 8 abr em 21:00      **Pontos** 25      **Perguntas** 15  
**Disponível** até 8 abr em 21:00      **Limite de tempo** Nenhum

## Instruções

O Trabalho Prático do Módulo 3 está disponível!

### 1. Instruções para realizar o trabalho prático

Consulte a data de entrega no teste e em seu calendário.

Reserve um tempo para realizar a atividade, leia as orientações e enunciados com atenção. Em caso de dúvidas utilize o "Fórum de dúvidas do Trabalho Prático do Módulo 3".

Para iniciá-lo clique em "Fazer teste". Você tem somente **uma** tentativa e não há limite de tempo definido para realizá-lo. Caso precise interromper a atividade, apenas deixe a página e, ao retornar, clique em "Retomar teste".

Clique em "Enviar teste" **somente** quando você concluí-lo. Antes de enviar confira todas as questões.

Caso o teste seja iniciado e não enviado até o final do prazo de entrega, a plataforma enviará a tentativa não finalizada automaticamente, independente do progresso no teste. Fique atento ao seu teste e ao prazo final, pois novas tentativas só serão concedidas em casos de questões médicas.

O gabarito será disponibilizado partir de sexta-feira, **09/04/2021**, às 23h59.

Bons estudos!

### 2. O arquivo abaixo contém o enunciado do trabalho prático

[Enunciado do Trabalho Prático – Módulo 3 - Bootcamp Arquiteto\(a\) de Software.pdf](#)

[Arquivos complementares do Trabalho Prático.zip](#)

## Histórico de tentativas

|              | Tentativa                   | Tempo      | Pontuação |
|--------------|-----------------------------|------------|-----------|
| MAIS RECENTE | <a href="#">Tentativa 1</a> | 30 minutos | 25 de 25  |

! As respostas corretas estarão disponíveis em 9 abr em 23:59.

Pontuação deste teste: **25** de 25

Enviado 7 abr em 21:38

Esta tentativa levou 30 minutos.

### Pergunta 1

1,67 / 1,67 pts

Sobre o padrão *Singleton* NÃO podemos afirmar:

- ☐ Trata-se de um padrão de projeto criacional.
- ☐ Oferece um ponto único de acesso à classe que gerencia.
- ☐ Garante que uma classe possui uma única instância.
- ☒ Facilita a execução de testes automatizados em qualquer circunstância.

### Pergunta 2

1,67 / 1,67 pts

Sobre o padrão Facade temos as seguintes afirmações:

- I. O padrão oferece uma interface unificada e de alto nível, que torna mais fácil o uso de um sistema.
- II. Devemos utilizar o padrão Facade quando necessitamos de uma interface limitada, mas simples para um subsistema complexo.
- III. É um padrão que funciona como um intermediário que controla o acesso a um objeto base.

Quais das opções estão CORRETAS?

- ☐ Todas as alternativas.
- ☒ I e II.
- ☐ I e III.
- ☐ II e III.

### Pergunta 3

1,67 / 1,67 pts

No projeto da Asaideira, a classe *PadraoDeProjeto* deveria apresentar as seguintes características para implementar o padrão de projeto mais adequado para o problema proposto:

☐

A classe deveria garantir a existência de uma única instância dela mesmo durante toda a execução do código.

☐

A classe deveria implementar uma interface que tenha os métodos `getPreco` e `getDescricao`.

☒

A classe deveria estender (palavra reservada `extends`) da classe abstrata *Açaí* e ainda ter um atributo do tipo *Açaí*.

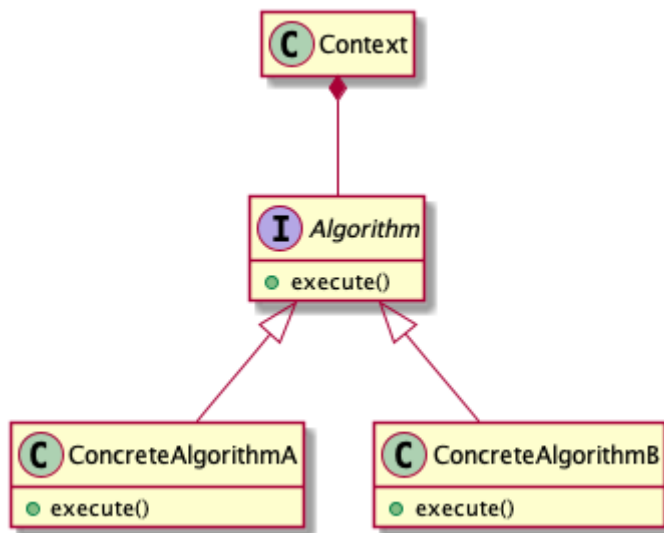
☐

A classe deveria disponibilizar métodos que ajudem na construção de objetos que herdam da classe *Açaí*.

### Pergunta 4

1,67 / 1,67 pts

Qual padrão de projeto é representado pelo diagrama UML abaixo?



- ☒ Strategy.
- ☐ Observer.
- ☐ Abstract Factory.
- ☐ Singleton.

### Pergunta 5

1,67 / 1,67 pts

Ao executarmos o projeto Asaideira, é exibido o preço dos diferentes tipos de açaí vendidos pela empresa. Qual é o preço do açaí na tigela com morango do tamanho Grande?

- ☐ 17,99.
- ☐ 36,97.
- ☒ 17,98.
- ☐ 9,99.

### Pergunta 6

1,67 / 1,67 pts

O ocultamento de informações é uma propriedade desejada no projeto de software. Essa propriedade traz as seguintes vantagens, EXCETO:

- ☒ Confidencialidade dos dados.
- ☐ Facilidade de entendimento.
- ☐ Desenvolvimento em paralelo.
- ☐ Flexibilidade a mudanças.

### Pergunta 7

1,67 / 1,67 pts

De acordo com o Princípio Aberto/Fechado, é CORRETO afirmar que uma classe:

- ☐ Deve ser fechada para extensões
- ☒ Pode ser customizada ou estendida por meio de herança ou padrões de projeto.
- ☐ Deve ser aberta para modificações
- ☐ Deve ter atributos privados

### Pergunta 8

1,67 / 1,67 pts

São propriedades desejáveis em projetos de software, EXCETO:

- ☐ Ocultamento de Informação.
- ☒ Coesão baixa.

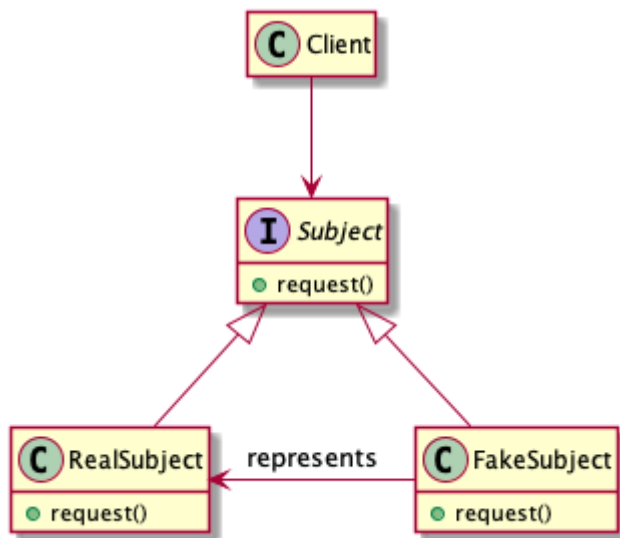
☐ Acoplamento baixo.

☐ Integridade Conceitual.

### Pergunta 9

1,67 / 1,67 pts

Qual padrão de projeto é representado pelo diagrama UML abaixo:



☐ Strategy.

☒ Proxy.

☐ Abstract Factory.

☐ Singleton.

### Pergunta 10

1,67 / 1,67 pts

Sobre o padrão Observer, podemos afirmar:

☐ É a base da arquitetura orientada a serviços.



É um padrão que permite que um objeto, avise outros objetos de que seu estado mudou.



Cria um acoplamento entre os sujeitos (observados) e os observadores.



Os objetos que herdam da classe Subject – também conhecido como observados, precisam ser alterados sempre que houver mudanças nos objetos que são os observadores.

### Pergunta 11

1,67 / 1,67 pts

Qual padrão de projeto é o mais adequado para o problema enfrentado pelo sistema da Asaideira?



Decorator.



Strategy.



Abstract Factory.



Proxy.

### Pergunta 12

1,67 / 1,67 pts

A respeito dos padrões de projeto, podemos afirmar, EXCETO:



Um Adaptador oferece uma interface diferente para o objeto adaptado.



Abstract Factory é um padrão que oferece um mecanismo para a criação de uma família de objetos relacionados.



Um Proxy possui a mesma interface que o seu objeto base.



Decorator é um padrão que permite converter a interface de uma classe para outra interface esperada pelos clientes. Permite que classes trabalhem juntas, o que não seria possível devido à incompatibilidade entre suas interfaces.

### Pergunta 13

1,67 / 1,67 pts

Os itens listados a seguir descrevem alguns padrões arquiteturais:

- a. Oferece uma interface unificada e de alto nível que torna mais fácil o uso de um sistema.
- b. Garante que uma classe possui no máximo uma instância e oferece um ponto único de acesso a ela.
- c. Facilita a construção de objetos complexos com vários atributos, sendo alguns deles opcionais.

Tais descrições representam, respectivamente, os seguintes padrões:



Iterator, Visitor e Abstract Factory.



Visitor, Singleton e Builder.



Facade, Strategy e Abstract Factory.



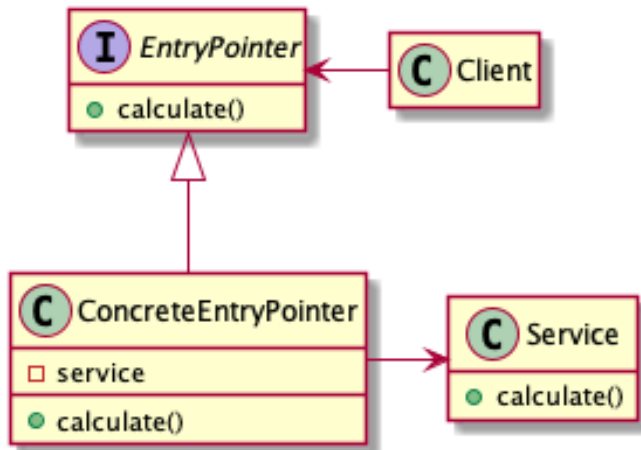
Facade, Singleton e Builder.

### Pergunta 14

1,67 / 1,67 pts



Qual padrão de projeto é representado pelo diagrama UML abaixo?



- ☒ Adapter.
- ☐ Strategy.
- ☐ Proxy.
- ☐ Abstract Factory.

### Pergunta 15

1,62 / 1,62 pts

Para implementar o padrão de projeto no código da Asaideira, é necessário criar um relacionamento entre as classes PadraoDeProjeto às novas versões das classes Morango e LeiteEmPo (pacote me.clementino.solution.acompanhamentos). Sobre esse relacionamento entre as classes é CORRETO afirmar:

- ☐ A classe PadraoDeProjeto deve estender (palavra reservada extends) da classe Morango ou LeiteEmPo.
- ☒ As classes Morango e LeiteEmPo devem estender (palavra reservada extends) a classe PadraoDeProjeto.



As classes Morango e LeiteEmPo devem ter um atributo do tipo PadraoDeProjeto.



Para implementar o padrão de projeto não é necessário criar relacionamentos entre as classes PadraoDeProjeto, Morango e LeiteEmPo.

Pontuação do teste: **25** de 25