Exercício - outros patterns e princípios

- Entrega 31 out em 23:59
- Pontos 1
- Perguntas 9
- Disponível 29 out em 0:00 - 31 out em 23:59
- Limite de tempo Nenhum
- Tentativas permitidas Sem limite

Fazer o teste novamente

Histórico de tentativas

	Tentativa	Tempo	Pontuação
MAIS RECENTE	Tentativa 1	5 minutos	1 de 1

Enviado 31 out em 7:46

Pergunta 1
0 / 0 pts

Sobre frameworks MVC (Model/View/Controller), analise as afirmativas a seguir e assinale a alternativa correta:

I. MVC é um modelo idealizado na década de 80 e que continua sendo usado até a atualidade.
II. MVC é um método em conformidade com boas práticas de separação de códigos, fazendo com que se tenha mais segurança, facilidade de manutenção e escalabilidade.
III. Embora agilize o carregamento dos recursos no servidor, MVC é um método que não se encontra em conformidade com boas práticas de separação de códigos, fazendo com que se tenha, geralmente, baixos níveis de encapsulamento e, portanto, menos segurança.

- Somente a afirmativa I é verdadeira

Correto!

- Somente as afirmativas I e II são verdadeiras.
- Somente as afirmativas I e III são verdadeiras.
- As afirmativas I, II e III são verdadeiras.
- Somente as afirmativas II e III são verdadeiras.

Pergunta 2
0 / 0 pts

Em relação aos padrões de projeto de software e princípios arquiteturais, em programação orientada a objetos, existe um princípio denominado de SOLID. Ele, por sua vez, é composto por 05 princípios de acordo com as suas iniciais, sendo eles:

S (Solid principle) – O (Open principle) – L (Library principle) – I (Integration principle) – D (Double principle)

S (Security closed principle) – O (Open extend principle) – L (Liskov include principle) – I (Interface duplication principle) – D (Duplicate structure principle)

S (Single closed principle) – O (Open-closed principle) – L (Library exclusive principle) – I (Integration case principle) – D (Dependency inversion principle)

Correto!

S (Single responsibility principle) – O (Openclosed principle) – L (Liskov substitution principle) – I (Interface segregation principle) e D (Dependency inversion principle)

Pergunta 3

0 / 0 pts

I. O cliente não sabe o que está por trás de uma tela bonita, ele simplesmente quer ver tudo funcionando e porquê complicar e inventar a roda quando podemos fazer o simples em tempo recorde e com uma lógica boa.

II. Se você usa muitas repetições, você não tem total conhecimento do assunto, ou seja você está enrolando. Pensar em apenas não repetir o código não significa que você está usando plenamente este principio. Temos que trabalhar em cima da lógica também.

As afirmativas acima dizem respeito aos seguintes princípios:

Correto!

KISS, DRY

WET, DRY

YAGNI, KISS

DRY, YAGNI

Pergunta 4

0 / 0 pts

Considere que em um projeto baseado em XP (eXtreme Programming) pratica-se a propriedade coletiva de código,
de forma que todos os desenvolvedores podem fazer alterações e refatoração de qualquer parte do código a qualquer momento.

Para isso, é necessário que também haja:

releases grandes

Correto!

- uso de padrões.
- muita documentação
- testes apenas depois da codificação.

Pergunta 5
0 / 0 pts

O framework Ionic facilita o processo de criação inicial do desenvolvimento passando pelos testes em browser sem a necessidade de instalar o aplicativo em um dispositivo. Além dessa facilidade, o framework Ionic contém alguns templates já prontos, em que é gerado todo artefato apenas com um comando. Se, por exemplo, você desejar construir um aplicativo com tabs de navegação, poderia gerá-lo com o seguinte comando:
ionic start [nome_de_ sua_aplicação] tabs.

Este exemplo de framework pode ser caracterizado como:

- Biblioteca de classes
- PWA (Progressive Web App)

Correto!

- SDK de código aberto para o desenvolvimento de aplicativos móveis híbridos
- Gerador de aplicativos

Pergunta 6
0 / 0 pts
São variações do padrão MVC, exceto:
Correto!

- DIM
- MVP
- PM
- MVVM

Pergunta 7
0 / 0 pts

“MVC, MVP, PM, MVVM all are different ways of implementing the Separation of Concerns (SoC) principle”. Martin Fowler

Segundo Martin Fowler, sobre o princípio SoC, são corretas as afirmativas, exceto:

Uma preocupação pode ser tão geral quanto "os detalhes do hardware de um aplicativo" ou tão específica quanto "o nome de qual classe instanciar.

A modularidade e, portanto, a separação de interesses, é obtida encapsulando informações dentro de uma seção de código que possui uma interface bem definida.

é um princípio de design para separar um programa de computador em seções distintas, de modo que cada seção trate de uma preocupação separada.

Correto!

- Um programa que incorpora bem o SoC não é chamado de programa modular.

Pergunta 8

0 / 0 pts

O uso de microserviços popularizou-se pois, EXCETO:

Você respondeu

- Escalar requer o redimensionamento de todo o aplicativo, ao invés de partes que exigem mais recursos.

Ao longo do tempo muitas vezes é difícil manter uma boa estrutura modular, o que dificulta mantermos as alterações que devem afetar apenas um módulo de dentro do módulo.

Resposta correta

- Microserviços não podem ser gerenciados por equipes diferentes.

como os serviços têm deploy independente e são escaláveis, cada serviço também possui um limite de módulo firme, que permite inclusive que diferentes serviços sejam escritos em diferentes linguagens de programação.

Pergunta 9

1 / 1 pts

"Dependency Inversion Principle (DIP): DIP states that High level modules should not depend upon low-level modules. Both should depend upon abstractions. Abstractions should not depend upon details. Details should depend upon abstractions. "

Esse princípio é :

- DRY

Correto!

- SOLID e GRASP
- GoF
- KISS