

Pergunta 1

Incorreta

Nota: -0,50 em 2,50

🚩 Marcar pergunta

A norma ISO/IEC25010 define um modelo de qualidade para sistemas de software. Qual a sua importância para as empresas?

Selecione uma opção de resposta:

- ☐ A. Facilitar a integração (compatibilidade) entre sistemas que seguem este modelo de qualidade.
- ☐ B. Define estratégias para garantir a integridade dos dados, o não repúdio das ações e a reutilização dos módulos.
- ☐ C. Serve para avaliar o grau até onde o conjunto de funcionalidades implementadas cobre a totalidade das necessidades e objetivos dos utilizadores.
- ☒ D. Determinar um conjunto de características de referência que devem ser consideradas para maximizar a facilidade de manter um produto (*maintainability*), sem introduzir defeitos. ✖
- ☐ E. Pode ser usado com base para um sistema de avaliação de qualidade, uma vez que o modelo estabelece as características a considerar na avaliação de um produto de software.

Your answer is incorrect.

A resposta correta é: Pode ser usado com base para um sistema de avaliação de qualidade, uma vez que o modelo estabelece as características a considerar na avaliação de um produto de software.

Pergunta 2

Correta

Nota: 2,50 em 2,50

🚩 Marcar pergunta

Considere a seguinte descrição sobre um teste incluído numa aplicação Spring Boot: "este teste procura confirmar que os objetos JSON da resposta estão a ser mapeados corretamente quando é usado um objeto válido no pedido, e se não forem válidos os pedidos, que estamos a enviar o erro HTTP correto, juntamente com uma razão descritiva para a falha." [In: relatório de um projeto de TQS]

Qual o tipo de teste (predefinido na Spring Boot) mais eficiente/adequado para o propósito enunciado?

Selecione uma opção de resposta:

- ☐ A. @MockMvc
- ☐ B. @SpringBootTest
- ☒ C. @WebMvcTest ✔
- ☐ D. @ExtendWith(MockitoExtension.class)
- ☐ E. @SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)

Your answer is correct.

A resposta correta é: @WebMvcTest

Pergunta 3

Incorreta

Nota: -0,50 em 2,50

🚩 Marcar pergunta

O excerto junto mostra o conteúdo do ficheiro "application-integration.properties" de um projeto SpringBoot.

Qual a sua utilidade no contexto dos testes?

```
spring.datasource.url=jdbc:mysql://localhost:3306/demo
spring.jpa.hibernate.ddl-auto=create-drop
spring.datasource.username=demo
spring.datasource.password=demo
```

Selecione uma opção de resposta:

- ☐ A. Não é usado nos testes; teria de se designar "application.properties" para ser utilizado pela ação da anotação `@AutoConfigureTestDatabase`.
- ☐ B. Serve para criar um contexto de dados persistente ao longo das várias execuções dos testes, de maneira a que cada ciclo de CI possa usar os dados (na base de dados) do ciclo anterior.
- ☐ C. Serve para configurar uma base de dados de teste temporária, em memória, que é destruída em cada execução dos testes.
- ☒ D. Não é usado nos testes; serve para configurar a base de dados de *runtime* (produção). ✖
- ☐ E. Serve para definir uma base de dados temporária, por exemplo, configurada na anotação `@TestPropertySource`

Your answer is incorrect.

A resposta correta é: Serve para definir uma base de dados temporária, por exemplo, configurada na anotação `@TestPropertySource`

Pergunta 4

Incorreta

Nota: 0,00 em 2,50

🚩 Marcar pergunta

Uma das regras do SonarQube sinaliza como uma vulnerabilidade do código, em projetos de Spring Boot, a utilização de direta de entidades como argumentos de métodos de uma API: «*Persistent entities should not be used as arguments of "@RequestMapping" methods*»

O programador pode resolver esta vulnerabilidade de várias formas, mas qual destas ideias **NÃO** é adequada?

Selecione uma opção de resposta:

- ☒ A. Usar identificadores das entidades, em vez da entidade em si, na definição dos argumentos, quando isso seja possível. ✖
- ☐ B. Os DTO ("Data Transfer Objects") auxiliares são versões mais leves que as Entidades (que mantêm os mapeamentos do JPA), e tornam mais rápido e seguro a sua utilização.
- ☐ C. Especificar, na configuração do `@RequestMapping`, que o método usa o formato JSON para codificar os dados.
- ☐ D. Usar um POJO auxiliar, sem contexto de persistência, como argumento do método da API.
- ☐ E. Utilizar objetos específicos para fazer o mapeamento dos atributos das entidades para os parâmetros dos métodos da API, através de "Data Transfer Objects".

Your answer is incorrect.

A resposta correta é: Especificar, na configuração do `@RequestMapping`, que o método usa o formato JSON para codificar os dados.

Pergunta 5

Correta

Nota: 2,50 em 2,50

🚩 Marcar pergunta

Os testes unitários são escritos pelo programador, como parte da estratégia de QA.

É **FALSO** que estes testes o ajudem:

Selecione uma opção de resposta:

- ☐ A. Na revisão do seu código pelos pares (em *code reviews*).
- ☒ B. A escrever menos testes de aceitação. ✔
- ☐ C. A aumentar a cobertura do código.
- ☐ D. A documentar a utilização pretendida de um componente;
- ☐ E. A entender o contrato do módulo (requisitos do que vai construir);

Your answer is correct.

A resposta correta é: A escrever menos testes de aceitação.

Pergunta 6

Correta

Nota: 2,50 em 2,50

🚩 Marcar pergunta

Há alguma vantagem na utilização do Maven para configurar um projeto ("Maven project") quando se está a usar um ambiente de CI?

Selecione uma opção de resposta:

- ☐ A. Não, o ambiente de CI precisa sempre de analisar o projeto e proceder à respetiva construção (*build*)
- ☐ B. Sim, o Maven garante que o código é compilado e testado e passa os binários ao serviço de CI livres de erros.
- ☒ C. Sim, o Maven contribui para uma descrição do projeto neutra em relação aos IDE facilitando a construção do projeto no ambiente de CI. ✓
- ☐ D. Não, a linguagem para a escrita do *pipeline* de CI é diferente e não tira partido da configuração descrita no POM.xml
- ☐ E. Não, a maior parte dos ambientes de CI não está preparado para integrar com projetos Maven.

Your answer is correct.

A resposta correta é: Sim, o Maven contribui para uma descrição do projeto neutra em relação aos IDE facilitando a construção do projeto no ambiente de CI.

Pergunta 7

Correta

Nota: 2,50 em 2,50

🚩 Marcar pergunta

"Uma das maiores vantagens de usar o TDD é que estamos a fazer pequenos incrementos ao escrever código e, portanto, os problemas são mais fáceis de corrigir, uma vez que apenas abordam um pedaço de código limitado." [In: Relatório de projeto de TQS]

A vantagem referida pode ser considerada real e fundamentada?

Selecione uma opção de resposta:

- ☐ A. Não, se cada teste só incidir sobre pedaços limitados do código a correção de problemas é ilusória, porque não verifica a integração dos módulos.
- ☐ B. Não, porque é preciso escrever bastante mais código em produção do que aquele que é necessário para passar os testes.
- ☒ C. Sim, o código de produção é feito na medida do suficiente para satisfazer os testes e a causa dos eventuais problemas é fácil de localizar. ✓
- ☐ D. Sim, o código de produção está sempre protegido por testes pequenos e rápidos de executar.
- ☐ E. Sim, em TDD, é frequente reutilizar pedaços limitados do código dos próprios testes para escrever o código de produção.

Your answer is correct.

A resposta correta é: Sim, o código de produção é feito na medida do suficiente para satisfazer os testes e a causa dos eventuais problemas é fácil de localizar.

Pergunta 8
Correta
Nota: 2,50 em 2,50

🚩 Marcar pergunta

A que é que se refere a ideia de continuidade subjacente às designações de integração contínua e entrega contínua (CI/CD)?

Selecione uma opção de resposta:

- ☐ A. É uma ideia exigente, que transmite o princípio de que a equipa deve entregar novos incrementos ao cliente de forma regular e frequente, ou seja, contínua, ao longo do projeto.
- ☐ B. Através de *triggers* (como *webhooks*), os sistemas estão continuamente a monitorizar o repositório de código para reagir a alterações e proceder a novas instalações, em cada entrega de código.
- ☐ C. Os programadores são incentivados a fazer a entrega e integração de código com grande frequência e, daí, a ideia de continuidade.
- ☒ D. A cultura da equipa, os processos de trabalho e as ferramentas estão preparados para, a qualquer altura, integrar os contributos dos programadores e entregar novas funcionalidades do produto. ✓
- ☐ E. Uma equipa que trabalha com rotinas de CI/CD é capaz de propagar as novas funcionalidades para produção, sem interromper as operações; do ponto de vista dos utilizadores, as operações decorrem de forma contínua, sem disrupção.

Your answer is correct.

A resposta correta é: A cultura da equipa, os processos de trabalho e as ferramentas estão preparados para, a qualquer altura, integrar os contributos dos programadores e entregar novas funcionalidades do produto.

Pergunta 9
Parcialmente correta

Nota: 1,25 em 2,50

🚩 Marcar pergunta

Várias ferramentas, como o Jenkins, Circle CI, GitLab CI, GitHub Actions, entre outras, utilizam a noção de *pipeline*. O que é um *pipeline* de Integração Contínua (*continuous integration*)?

Selecione uma opção de resposta:

- ☐ A. É uma visualização das várias etapas (*stages*) configuradas no projeto, que devem ser realizadas sempre que há um novo incremento, de forma sequencial ou paralela.
- ☒ B. É uma implementação automatizada do processo de compilação, montagem, teste e instalação dos ambientes de ensaio e de produção de uma aplicação. ✓ A pergunta refere-se a integração contínua mas esta opção inclui já componentes de CD. De qualquer forma, atendendo à natureza das ferramentas mencionadas e uso dos pipelines, esta hipótese vale 1/2 da cotação.
- ☐ C. É um ficheiro (na notação .yaml) usado para configurar as etapas de compilação e teste de um projeto, quando ocorrem incrementos de código num repositório.
- ☐ D. É um ficheiro de configuração que deve acompanhar o repositório Git, de modo a que as novas entregas sejam detetadas de imediato.
- ☐ E. É uma especificação dos passos que devem ser feitos sempre que é detetado um novo incremento, e que decisões devem ser tomadas quando certas condições ocorrem (e.g.: partes do processo passam ou falham).

Your answer is partially correct.

A resposta correta é: É uma especificação dos passos que devem ser feitos sempre que é detetado

Pergunta 10

Incorreta

Nota: -0,50 em 2,50

Marcar pergunta

A utilização de *personas* (personagens) ajuda a focar a criação de valor nas motivações de utilizadores concretos e cria contextos reutilizáveis para escrever as histórias. Como é que os processos de QA podem beneficiar do recurso às *personas*?

Selecione uma opção de resposta:

- ☐ A. As *personas* podem ser usadas para criar os ramos ("branches") de um repositório (um "branch" para cada *persona*), facilitando a distribuição do trabalho e a revisão pelos pares.
- ☒ B. Apesar de serem úteis no processo de descoberta de requisitos e fixação das histórias, o conceito de *persona* não tem continuidade para a definição de testes funcionais. ✖
- ☐ C. As *personas* fixam um conjunto de características do utilizador e ajudam a criar exemplos de utilização coerentes, na escrita dos critérios de aceitação.
- ☐ D. Embora as *personas* ajudem a concretizar quem beneficia do produto, é preferível usar os papéis do utilizador diante do sistema (e.g.: um cliente, um aluno) para não enviesar os critérios de aceitação.
- ☐ E. As *personas* são úteis para os projetos que seguem BDD, em que são usadas para incluir nos ficheiros de "features", que veiculam os exemplos de utilização.

Your answer is incorrect.

A resposta correta é: As *personas* fixam um conjunto de características do utilizador e ajudam a criar exemplos de utilização coerentes, na escrita dos critérios de aceitação.

Pergunta 11

Incorreta

Nota: -0,50 em 2,50

Marcar pergunta

O suporte para testes existente no Spring Boot integra bem com o *framework* Mockito. O que seria uma situação natural para usar o Mockito no teste de componentes da Spring Boot?

Selecione uma opção de resposta:

- ☐ A. O Mockito pode ser usado num teste `@DataJpaTest` de um componente `@Repository` para definir os resultados esperados das queries que seriam colocadas à base de dados.
- ☐ B. O Mockito pode ser usado num teste `@WebMvcTest` de um componente do tipo `@RestController` para substituir a lógica do serviço, injetando um "*bean*" sintetizado (via `@MockBean`).
- ☐ C. O Mockito é um framework para usar no contexto de testes unitários e, genericamente, não é útil em testes da SpringBoot que precisam de injetar componentes adicionais ("*beans*").
- ☒ D. O Mockito pode ser usado num teste `@DataJpaTest` de um componente `@Service` para substituir as chamadas à base de dados por respostas sintetizadas, fazendo *mock* do componente repositório. ✖
- ☐ E. O Mockito pode ser usado num teste do tipo `@SpringBootTest` para predefinir os resultados das invocações do `TestRestTemplate`, usado para invocar métodos de uma API.

Your answer is incorrect.

A resposta correta é: O Mockito pode ser usado num teste `@WebMvcTest` de um componente do tipo `@RestController` para substituir a lógica do serviço, injetando um "*bean*" sintetizado (via `@MockBean`).

Considere o excerto de código que testa um componente Spring Boot (`EmployeeRestController.class`), relativo ao método POST acessível em `"/api/employees"`.

Este teste está a falhar, com a indicação de erro na injeção da dependência na variável `"service"`. Do que se pode inferir, qual é o problema do código que provoca o erro do teste?

```
@WebMvcTest(EmployeeRestController.class)
public class EmployeeControllerTest {

    @Autowired
    private MockMvc mvc ;

    @MockBean
    private EmployeeService service = new EmployeeServiceImpl();

    @Test
    public void whenPostEmployee_thenCreateEmployee() throws Exception {
        Employee alex = new Employee("alex");
        given(service.save(Mockito.any())).willReturn(alex);

        mvc.perform(post("/api/employees")
            .contentType(MediaType.APPLICATION_JSON)
            .content(JsonUtil.toJson(alex)))
            .andExpect(status().isCreated())
            .andExpect(jsonPath("$.name", is("alex"))));
        verify(service, VerificationModeFactory.times(1))
            .save(Mockito.any());
        reset(service);
    }
}
```

Selecione uma opção de resposta:

- ☐ A. Na declaração da variável `"service"` deveria estar a anotação `"@InjectMocks"` e não `"@MockBean"`
- ☐ B. As instâncias marcadas com `@MockBean` deve ser criadas (injetadas) pelo ambiente e não com o operador `"new"`.
- ☐ C. O parâmetro `Mockito.any()` não é adequado como argumento do método `service.save()`.
- ☐ D. A anotação `@WebMvcTest` não é adequada para este teste, uma vez que o teste acede a um `endpoint` REST (`EmployeeRestController.class`)
- ☐ E. A operação `Mockito.verify()` deve ser executada no início do teste (e não no final), uma vez que verifica se as instâncias são nulas e, nesse caso, procede à injeção das dependências.

Your answer is incorrect.

A resposta correta é: As instâncias marcadas com `@MockBean` deve ser criadas (injetadas) pelo ambiente e não com o operador `"new"`.

Pergunta 13

Incorreta

Nota: -0,50 em 2,50

🚩 Marcar pergunta

Relativamente ao conceito de “*user story*”, tal como é entendido nos processos de Garantia de Qualidade em metodologias ágeis, qual é a afirmação verdadeira?

Selecione uma opção de resposta:

- ☐ A. A *user story* descreve uma funcionalidade, segunda a perspetiva de quem dela beneficia, e é usada como unidade para priorizar o trabalho e a entrega de valor.
- ☒ B. As *user stories* são criadas no início de cada iteração, apresentado de forma breve as novas funcionalidades a desenvolver na presente iteração/sprint. ✖
- ☐ C. As *user stories* são preparadas pelos representantes do negócio/cliente para apresentarem as histórias que pretendem realizar no sistema.
- ☐ D. As *user stories* são preparadas pelos testers, para detalhar os requisitos funcionais do sistema e os critérios de aceitação.
- ☐ E. As *user stories* são escritas pelo dono do produto (*product owner*) e entregues à equipa no início de cada iteração/sprint.

Your answer is incorrect.

A resposta correta é: A *user story* descreve uma funcionalidade, segunda a perspetiva de quem dela beneficia, e é usada como unidade para priorizar o trabalho e a entrega de valor.

Pergunta 14

Correta

Nota: 2,50 em 2,50

🚩 Marcar pergunta

“Os testes de sistema irão ser conduzidos pelos *stakeholders*, com base na informação presente no manual de qualidade do software e no relatório de especificações do produto. Estes testes irão indicar se cada componente é capaz de satisfazer os requisitos definidos nas especificações do produto.” [In: relatório de projeto de TQS]

Esta declaração apresenta vários problemas, assinalados a seguir. Neste contexto, qual das afirmações é **FALSA**?

Selecione uma opção de resposta:

- ☐ A. Os *stakeholders* nunca realizam testes de sistema, apesar de participar na elaboração de critérios de aceitação.
- ☐ B. O Manual de Qualidade não tem informação suficiente para determinar os testes de sistema que podem ser usados na verificação de um produto.
- ☐ C. Os testes de sistema são, principalmente, orientados para a qualidade interna do produto e não para as necessidades dos utilizadores.
- ☐ D. Os testes de sistema servem para validar o sistema globalmente, não cada componente por si.
- ☒ E. Os testes são automatizados; os *stakeholders* nunca participam na realização de testes. ✔

Your answer is correct.

A resposta correta é: Os testes são automatizados; os *stakeholders* nunca participam na realização de testes.

Pergunta 15

Incorreta

Nota: -0,50 em 2,50

🚩 Marcar pergunta

Os projetos baseados em práticas ágeis colocam maior ênfase na automação dos testes do que os projetos “tradicionais” (desenvolvimento sequencial), porque:

Selecione uma opção de resposta:

- ☐ A. As alterações aos requisitos acontecem diariamente e os testes precisam de ser regerados de forma automática, para garantir a velocidade da equipa e a entrega diária de valor.
- ☐ B. Os projetos ágeis dependem sobretudo de testes unitários, em detrimento de outros. Como os testes unitários devem ser em grande número, não podem ser executados manualmente.
- ☒ C. As iterações/sprints são de duração fixa. A equipa deve garantir que todos os testes podem ser completamente executados no final de cada iteração/Sprint. ✖
- ☐ D. Os testes são executados regularmente, tanto para novos incrementos como para o código já existente, inviabilizando o recurso a testes manuais no ciclo de CI.
- ☐ E. Os testes geram as métricas de qualidade do código que determinam o grau de prontidão/confiança na qualidade do código e a passagem nos controlos de qualidade (“quality gates”) definidos.

Your answer is incorrect.

A resposta correta é: Os testes são executados regularmente, tanto para novos incrementos como para o código já existente, inviabilizando o recurso a testes manuais no ciclo de CI.

Pergunta 16

Correta

Nota: 2,50 em 2,50

🚩 Marcar pergunta

Qual a interpretação mais adequada do conceito “dívida técnica” (*technical debt*), tal com é usada no ambiente de qualidade do SonarQube?

Selecione uma opção de resposta:

- ☐ A. É uma estimativa do tempo de trabalho necessário para fazer passar os testes que estão a falhar.
- ☐ B. É a diferença do nível de cobertura atual para cobertura plena (100%).
- ☒ C. É uma estimativa do tempo necessário para corrigir os problemas encontrados durante a análise estática do código. ✔
- ☐ D. É o diferencial do nível atual de qualidade do projeto, determinado pela ferramenta de análise estática, para o nível de qualidade plena (sem quaisquer problemas).
- ☐ E. É o número de *user stories* não aceites na iteração corrente, por não passarem os testes definidos.

Your answer is correct.

A resposta correta é: É uma estimativa do tempo necessário para corrigir os problemas encontrados durante a análise estática do código.

Pergunta 17

Correta

Nota: 2,50 em 2,50

 Marcar pergunta

O guia de estilo para os colaboradores do Android Open Source Project indica que a situação ilustrada no exemplo junto não é aceitável.

Em termos gerais, qual é a alternativa preferencial para resolver esta situação?

```
void setServerPort(String value){  
    try{  
        serverPort =Integer.parseInt(value);  
    } catch( NumberFormatException e){ }  
}
```

Selecione uma opção de resposta:

- ☐ A. Receber a exceção e lançar uma nova ocorrência da exceção genérica (Exception).
- ☐ B. Proteger o código com asserções (assert), para garantir que a condição de erro nunca é atingida.
- ☐ C. Ignorar a exceção, sem a tratar, desde que se inclua um comentário no código a documentar a razão para o fazer.
- ☐ D. Receber a exceção e lançar uma nova ocorrência de `RuntimeException`.
- ☒ E. Lançar uma nova exceção, mas apropriada ao nível de abstração da operação (com a semântica adequada). ✓

Your answer is correct.

A resposta correta é: Lançar uma nova exceção, mas apropriada ao nível de abstração da operação (com a semântica adequada).

Pergunta 18

Correta

Nota: 2,50 em 2,50

 Marcar pergunta

A complexidade ciclomática do código influencia a facilidade com que pode ser lido e mantido. Que fatores podem influenciar a métrica de complexidade, tal como é definida no SonarQube?

Selecione uma opção de resposta:

- ☒ A. Ocorrência de instruções que dividem o controlo de fluxo, tais como `if`, `for`, `while`, `case`. ✓
- ☐ B. Número de iterações realizadas em ciclos (repetições de um ciclo).
- ☐ C. Tamanho de uma classe (número de linhas).
- ☐ D. Número de variáveis declaradas num método.
- ☐ E. Número de variáveis declaradas num método que não são utilizadas.

Your answer is correct.

A resposta correta é: Ocorrência de instruções que dividem o controlo de fluxo, tais como `if`, `for`, `while`, `case`.

Pergunta 19

Não
respondida

Nota: 2,50

🚩 Marcar
pergunta

É muito frequente a utilização de virtualização por *containers* (contentores), como o Docker, em ambientes de CI/CD. Qual das seguintes afirmações é **ERRADA** em relação ao contributo dos *containers* nos processos de QA:

Selecione uma opção de resposta:

- ☐ A. As ferramentas de CI/CD integram suportam operações com *containers*, na definição dos respetivos *pipelines* de CI/CD.
- ☐ B. Os *containers* oferecem uma forma natural e eficiente de isolar elementos da arquitetura, que podem ser ativados ou desativados pelos processos de CI/CD.
- ☐ C. Os *containers* são muito úteis para instanciar localmente, mas limitados, quando é necessário a ativação de vários *containers* em vários servidores (i.e., não é possível trabalhar com *clusters* de *containers* distribuídos).
- ☐ D. Os *containers* podem ser geridos num repositório (um *registry*), de modo a que *containers* criados em certas etapas do processo de CI/CD podem ser reusados em outras.
- ☐ E. A definição dos *containers* pode ser sujeita a controlo de versões juntamente com o resto do código e recursos do projeto, de forma a que a configuração da infraestrutura evolua com história do próprio projeto.

Your answer is incorrect.

A resposta correta é: Os *containers* são muito úteis para instanciar localmente, mas limitados, quando é necessário a ativação de vários *containers* em vários servidores (i.e., não é possível trabalhar com *clusters* de *containers* distribuídos).

Pergunta 20

Correta

Nota: 2,50 em
2,50

🚩 Marcar
pergunta

Os testes funcionais implementados com recurso a Selenium/WebDriver podem recorrer ao padrão "*page object model*" (POM). Qual a vantagem de utilizar este padrão?

Selecione uma opção de resposta:

- ☐ A. A programação da página web deve usar uma linguagem por objetos, com injeção de dependências, que é facilitador para os testes.
- ☐ B. O padrão POM utiliza exemplos escritos em linguagem do domínio (*features*) para alimentar a execução dos testes de aceitação, tornando os testes mais expressivos.
- ☐ C. Cada página (web) é abstraída por uma classe; os testes de aceitação podem ser vistos como testes unitários sobre métodos de uma classe, permitindo reutilizar ferramentas e ambientes de teste.
- ☐ D. É vantajoso quando a implementação da camada web é baseada em Spring Boot MVC, mas não tem uma aplicação geral (e.g.: React JS).
- ☒ E. Cada página (web) é abstraída por uma classe própria; as interações possíveis com essa página são expostas como métodos da classe (e.g.: *searchByFirstName*), tornando os testes mais expressivos e legíveis. ✓

Your answer is correct.

A resposta correta é: Cada página (web) é abstraída por uma classe própria; as interações possíveis com essa página são expostas como métodos da classe (e.g.: *searchByFirstName*), tornando os testes

Pergunta 21

Incorreta

Nota: -0,50 em 2,50

🚩 Marcar pergunta

Nem sempre é necessário, ou conveniente, testar uma unidade de software (ou partes). Quando é que pode ser contraproducente/desaconselhável escrever testes (unitários)?

Selecione uma opção de resposta:

- ☐ A. Objetos que não incluem lógica relevante, como entidades persistentes geradas, dispensam a realização de testes.
- ☒ B. Os testes escritos com a preocupação de aumentar a cobertura passam a ser inúteis e não testam a lógica do problema. ✖
- ☐ C. É sempre necessário e útil escrever testes unitários para a totalidade do código de um projeto.
- ☐ D. É útil aguardar pelo *feedback* da revisão do código pelos pares para então escrever os testes.
- ☐ E. O código que foi automaticamente gerado (e.g.: vários *"toString"* e *"equals"*), também deve ser testados para atingir a cobertura de 100%.

Your answer is incorrect.

A resposta correta é: Objetos que não incluem lógica relevante, como entidades persistentes geradas, dispensam a realização de testes.

Pergunta 22

Correta

Nota: 2,50 em 2,50

🚩 Marcar pergunta

A abordagem BDD explicita a necessidade de testes numa estratégia de QA. Qual das seguintes características NÃO se aplica ao BDD?

Selecione uma opção de resposta:

- ☐ A. Facilita a compreensão do comportamento esperado do software, ao fomentar a colaboração entre a equipa técnica e a equipa do "negócio".
- ☒ B. Dispensa a utilização de outras técnicas de especificação de requisitos, uma vez que permite centralizar a especificação do produto na descrição da "feature", que é sujeita a controlo de versões, com o restante código. ✔
- ☐ C. As "features" podem ser chamadas de "especificações executáveis" porque alimentam a realização de testes (em código).
- ☐ D. É suportada por uma linguagem natural que permite descrever o teste na área do domínio do problema, entendida pela equipa alargada.
- ☐ E. Os (novos) cenários tornam-se mais fáceis de implementar à medida que mais "passos" ficam implementados, pois há a possibilidade de partilhar/reusar passos comuns.

Your answer is correct.

A resposta correta é: Dispensa a utilização de outras técnicas de especificação de requisitos, uma vez que permite centralizar a especificação do produto na descrição da "feature", que é sujeita a controlo de versões, com o restante código.

Pergunta 23

Não
respondida

Nota: 2,50

🚩 Marcar
pergunta

Qual das seguintes práticas é mais recomendável na implementação de rotinas de revisão de código pelos pares:

Selecione uma opção de resposta:

- ☐ A. A revisão do código deve ser feita preferencialmente pelos membros juniores na equipa, de modo a familiarizarem-se com os padrões de trabalho da equipa e com a próprio código/módulos já existentes.
- ☐ B. Concentrar a revisão do código num período específico da semana, tomando o tempo que for necessário, para aumentar a produtividade da revisão.
- ☐ C. O código a analisar deve já ter sido objeto de análises automáticas de qualidade (e.g.: análise estática) para descartar problemas prévios à própria revisão do código.
- ☐ D. A pessoa que vai rever deve começar por incluir no código fonte os seus pressupostos e observações (como comentários), para reduzir ambiguidades na comunicação.
- ☐ E. Quando a equipa pratica a revisão de código pelos pares, a aprovação do código através de *pull /merge requests* torna-se dispensável, eliminado esse possível ponto de bloqueio.

Your answer is incorrect.

A resposta correta é: O código a analisar deve já ter sido objeto de análises automáticas de qualidade (e.g.: análise estática) para descartar problemas prévios à própria revisão do código.

Pergunta 24

Correta

Nota: 2,50 em
2,50

🚩 Marcar
pergunta

É uma boa prática de SQA incluir na lista de condições para concluir um incremento ("definition of done") a obrigação do programador suplementar o seu código de produção com testes?

Selecione uma opção de resposta:

- ☒ A. Sim, porque o programador tem conhecimento da lógica interna do módulo e assim forma-se cumulativamente uma "rede de segurança" para prevenir futuras regressões. ✓
- ☐ B. Não; existem várias ferramentas de análise de qualidade do código que compensam a eventual falta de testes e assim aumentam a produtividade do programador.
- ☐ C. Não; se a equipa adotar a revisão de código pelos pares (*peer code review*), os testes são facultativos e apenas necessários em alguns casos mais complexos.
- ☐ D. Sim, porque os ambientes de CI não podem realizar as respetivas *pipelines* se não houver testes no projeto.
- ☐ E. Sim, porque a possibilidade de fazer entrega contínua depende de um elevado grau de confiança na qualidade do código, com níveis de cobertura muito próximos de 100%.

Your answer is correct.

A resposta correta é: Sim, porque o programador tem conhecimento da lógica interna do módulo e assim forma-se cumulativamente uma "rede de segurança" para prevenir futuras regressões.

Pergunta 25

Não
respondida

Nota: 2,50

🚩 Marcar
pergunta

O *framework* Mockito permite sintetizar objetos de substituição (mocks) que são usados em lugar das verdadeiras implementações. Duas das principais anotações são `@Mock` e `@Spy`.

Assinale a afirmação verdadeira.

Selecione uma opção de resposta:

- ☐ A. `@Mock` serve para marcar o objeto que deve ser sintetizado; `@Spy` serve para marcar a classe sob teste que precisa de usar o objeto sintetizado (e injeta o *mock*).
- ☐ B. `@Spy` é usado para observar um objeto real; podemos verificar se certos métodos foram ou não utilizados nesse objeto, mas não é possível declarar, com expectativas (*expectations*), o valor predefinido que um determinado método deve retornar.
- ☐ C. Se o objeto `spyList` foi anotado com `@Spy`, então `spyList.add("one")` não adiciona realmente o objeto à lista e o tamanho dela não é alterado.
- ☐ D. `@Spy` é usado para observar um objeto real; permite chamar e executar os métodos normais do objeto, e rastrear todas as interações ocorridas, tal como fariamos com um objeto de substituição (*mock*).
- ☐ E. Se o objeto `mockList` foi anotado com `@Mock`, então `mockList.add("one")` não adiciona realmente o objeto à lista; o tamanho da lista só é incrementado quando for invocado `Mockito.verify(mockList).add("one")`;

Your answer is incorrect.

A resposta correta é: `@Spy` é usado para observar um objeto real; permite chamar e executar os métodos normais do objeto, e rastrear todas as interações ocorridas, tal como fariamos com um objeto de substituição (*mock*).

Pergunta 26

Incorreta

Nota: -0,50 em
2,50🚩 Marcar
pergunta

O "V-model" é uma forma conhecida de relacionar diferentes tipos de testes com diferentes âmbitos do sistema.

Qual das afirmações é característica de uma estratégia de testes que segue o "V-Model"?

Selecione uma opção de resposta:

- ☒ A. Os testes das camadas superiores devem usar os testes das camadas inferiores. ✖
- ☐ B. O esforço da equipa com as atividades de teste é cumulativo e aumenta de iteração para iteração.
- ☐ C. Os testes são definidos antes da implementação do código, associados ao trabalho da etapa respetiva do ciclo de desenvolvimento.
- ☐ D. Em cada iteração, os testes são executados depois da implementação, de forma ascendente, i.e., dos unitários para os de aceitação.
- ☐ E. Existem diferentes tipos de teste, que devem ser executados numa ordem bem definida, segundo um esquema "top-down": âmbito dos requisitos, âmbitos dos serviços, âmbito das unidades.

Your answer is incorrect.

A resposta correta é: Os testes são definidos antes da implementação do código, associados ao trabalho da etapa respetiva do ciclo de desenvolvimento.

Pergunta 27

Correta

Nota: 2.50 em 2.50

🚩 Marcar pergunta

Um relatório de cobertura, por exemplo, gerado pelo Jacoco, apresenta diversas dimensões segundo as quais a cobertura pode ser calculada (e.g.: métodos, instruções, linhas, ramos alternativos). Qual das seguintes opções é a descrição mais adequada do conceito de cobertura de instruções do código?

Selecione uma opção de resposta:

- ☐ A. É uma métrica que traduz o número de defeitos corrigidos sobre o total de defeitos encontrados.
- ☐ B. É uma métrica usada para medir a percentagem de testes que foram executados com sucesso.
- ☐ C. É uma métrica utilizada para medir a percentagem de linhas de código fonte que são realmente executadas.
- ☒ D. É uma métrica que determina a relação das instruções que foram executadas num conjunto de testes, em relação ao total de instruções do código analisado. ✓
- ☐ E. É uma métrica que traduz a percentagem de código que foi inspecionado pelo ambiente de análise estática (e.g.: SonarQube) em relação à totalidade do projeto.

Your answer is correct.

A resposta correta é: É uma métrica que determina a relação das instruções que foram executadas num conjunto de testes, em relação ao total de instruções do código analisado.

Pergunta 28

Não respondida

Nota: 2.50

🚩 Marcar pergunta

Considerando os testes de integração, identifique a afirmação **FALSA**:

Selecione uma opção de resposta:

- ☐ A. Os testes de integração podem ter de esperar que certas etapas do ciclo de CI estejam concluídas, para garantir que houve etapas prévias que prepararam o ambiente necessário para os testes de integração.
- ☐ B. Os testes de integração podem ser otimizados recorrendo ao uso generalizado de objetos de substituição (*mocks*) em lugar das dependências.
- ☐ C. Quando um teste de integração falha, isso indicia um problema na articulação dos componentes ou na disponibilidade dos serviços.
- ☐ D. Os testes de integração, para além da unidade sob teste, usam componentes ou infraestruturas adicionais como a rede, base de dados ou sistema de ficheiros.
- ☐ E. Os testes de integração devem ser utilizados para testar as arquiteturas de microserviços

Your answer is incorrect.

A resposta correta é: Os testes de integração podem ser otimizados recorrendo ao uso generalizado de objetos de substituição (*mocks*) em lugar das dependências.

Pergunta 29

Incorreta

Nota: -0,50 em 2,50

🚩 Marcar pergunta

Em que situação é mais provável que o programador recorra ao uso de objetos de substituição (mocks) num plano de teste?

Selecione uma opção de resposta:

- ☒ A. A unidade sob teste utiliza uma base de dados em memória. ✖
- ☐ B. A unidade sob teste ainda não está pronta (implementada) e é necessário criar uma implementação provisória do seu comportamento.
- ☐ C. A unidade sob teste é um serviço (*endpoint*) de uma API REST.
- ☐ D. A equipa adotou uma metodologia BDD para os testes.
- ☐ E. A unidade sob teste apresenta uma dependência de um serviço remoto, com um resultado variável.

Your answer is incorrect.

A resposta correta é: A unidade sob teste apresenta uma dependência de um serviço remoto, com um resultado variável.

Pergunta 30

Correta

Nota: 2,50 em 2,50

🚩 Marcar pergunta

Que problemas poderiam ser apontados a um portfolio de testes baseado numa “pirâmide invertida” (poucos testes unitários, muitos testes de aceitação)?

Selecione uma opção de resposta:

- ☐ A. Não assegura os testes de integração associados, por exemplo, à verificação de serviços complexos.
- ☐ B. Não é escalável porque os testes de aceitação requerem a participação dos *stakeholders*.
- ☐ C. Depende de ambientes gráficos para a execução dos processos de CI (usualmente assegurados por servidores remotos)
- ☒ D. É limitativo quanto à localização da causa dos erros, no caso de eles existirem. ✔
- ☐ E. É impeditivo da adoção de práticas de *refactoring* de código, pois não oferece a necessária “rede de segurança” para prevenir regressões.

Your answer is correct.

A resposta correta é: É limitativo quanto à localização da causa dos erros, no caso de eles existirem.