

Consolidated Exercises

Consolidated Exercises

Consolidated Exercises

Lista de Exercícios Consolidados

PDF 1: Metodologia RAD

****Questão 1: Princípio do RAD - "Todas as mudanças durante o desenvolvimento são**

Consolidated Exercises

reversíveis"

- A. A versão atual do protótipo pode não corresponder à expectativa do cliente, portanto é adequado retornar a versão anterior que já está aprovada pelo cliente. *
- B. A não flexibilidade da metodologia RAD pode não facilitar o desenvolvimento de protótipos, portanto o desenvolvedor nem sempre pode criar uma versão.
- C. Nas iterações com os usuários, os desenvolvedores podem detectar uma funcionalidade que não atende a um requisito não funcional. Portanto é necessário voltar à versão anterior.
- D. Dada a flexibilidade da metodologia RAD, a utilização de ferramentas de controle de versões de software é desnecessária, pois os protótipos sempre podem ser recuperados.
- E. As novas versões devem ter o incremento de apenas uma nova funcionalidade, pois, caso não seja validada pelo usuário, é mais simples retornar à versão anterior que já foi validada.

****Questão 2: Características da metodologia RAD****

- A. Diferente de métodos tradicionais de desenvolvimento, a RAD demanda por intensa colaboração entre desenvolvedores e usuários. *
- B. Apesar das diferenças dos métodos tradicionais, se assemelham no sentido de que os sistemas-alvo devem ser modularizados.
- C. No sentido de atingir a meta da rapidez do desenvolvimento, a metodologia RAD possui poucas exigências.
- D. Os requisitos funcionais são determinados no início do projeto e não podem ser alterados ao longo do projeto de modo a atingir a rapidez do desenvolvimento.
- E. Uma das vantagens mais importantes da metodologia RAD é sua adequação para projetos complexos de grande escala.

****Questão 3: Princípio do RAD - "Adequação para fins comerciais é o critério essencial"**

- A. Este princípio garante que o produto terá menos erros e, conseqüentemente, terá menos vulnerabilidades.

Consolidated Exercises

- B. O foco deve ser voltado para cumprir os prazos pré-estabelecidos.
- C. O foco é fazer o desenvolvimento para atender aos requisitos funcionais. *
- D. Significa que as melhores práticas para desenvolver um software devem ser aplicadas.
- E. Permite que os desenvolvedores tomem a decisão sobre a aplicação das melhores soluções para o projeto.

****Questão 4: Escolha de framework para projetos RAD****

- A. Deve considerar se o framework suporta a linguagem Python por ser enxuta e possuir mais pacotes do que as demais linguagens de programação.
- B. A principal característica a ser considerada refere-se ao tipo de licença do framework.
- C. Deve ser feita para frameworks que deem suporte ao desenvolvimento de aplicações WEB e paração de gráficos.
- D. Deve levar em consideração a qualidade dos componentes visuais, de modo a manter a motivação das partes interessadas no projeto.
- E. Deve estar alinhada com os princípios da RAD, ou seja, suportar o desenvolvimento rápido. *

****Questão 5: Característica relevante da equipe para eficácia da metodologia RAD****

- A. Especialização em uma única área.
- B. Habilidades diversificadas. *
- C. Experiência em projetos de longa duração.
- D. Preferência por processos formais.
- E. Foco em desenvolvimento independente.

****Questão 6: Princípio do RAD - Entrega contínua dentro do prazo****

- A. Isso permite que a equipe do projeto defina rapidamente a abordagem ideal para entregar o produto no tempo disponível. *
- B. Que, pelo menos, uma parte do sistema esteja funcionando corretamente.

Consolidated Exercises

- C. Este princípio garante que o produto terá menos erros.
- D. O prazo sempre deve ser respeitado, mesmo que não seja possível entregar o produto conforme programado.
- E. O desenvolvimento rápido sempre vai reduzir o custo do projeto.

****Questão 7: Benefício do uso da metodologia RAD em aplicativo web customizável****

- A. Menor custo de desenvolvimento.
- B. Independência de feedback do cliente.
- C. Maior cobertura no código.
- D. Rápida adaptação a mudanças. *
- E. Menos necessidade de testes.

****Questão 8: Aspecto da equipe que contribui para o sucesso da metodologia RAD****

- A. Grande número de desenvolvedores.
- B. Comunicação ágil e eficiente. *
- C. Uso extensivo de documentação.
- D. Longo prazo de desenvolvimento.
- E. Alta complexidade do projeto.

****Questão 9: Elemento crucial para sucesso de projeto RAD em ambiente governamental****

- A. Estrutura hierárquica rígida.
- B. Colaboração constante com stakeholders. *
- C. Enfoque em tecnologias open-source.
- D. Evitar feedback frequente.
- E. Priorizar o desenvolvimento a longo prazo.

PDF 2: Python, Tkinter e Kivy

****Questão 1: Substituição de lacunas no código Tkinter para exibir texto****

Consolidated Exercises

- A. text e "Este é um texto. Esta é a segunda linha.\nEsta é a terceira linha\n"
- B. insert e "Este é um texto.\nEsta é a segunda linha.\nEsta é a terceira linha\n" *
- C. insert e "Este é um texto. Esta é a segunda linha.\nEsta é a terceira linha\n"
- D. text e "Este é um texto. Esta é a segunda linha. Esta é a terceira linha\n"
- E. Nenhuma das outras alternativas.

****Questão 2: Componente Tkinter para exibir mensagem após tarefa****

- A. Dialog
- B. Checkbox
- C. Label
- D. Button
- E. Message *

****Questão 3: Características da linguagem Python para RAD****

- A. A linguagem Python estabelece padrões de simplicidade que exigem dos pacotes e frameworks certo nível de adaptação estrutural.
- B. A linguagem Python permite criar rapidamente aplicações flexíveis que permitem o desenvolvedor reprogramar conforme a necessidade. *
- C. A linguagem Python possui mecanismos embutidos para construção de interfaces ricas, padronizando assim os projetos de software.
- D. A linguagem Python possui uma grande quantidade de recursos, o que torna a adaptação às funções de negócio mais demorada.
- E. A linguagem Python se destaca pela quantidade de pacotes e frameworks, no entanto, a aplicação do RAD fica prejudicada devido à falta de padronização.

****Questão 4: Substituição de lacuna no código Kivy****

- A. ClassApp
- B. buildApp

Consolidated Exercises

C. EApp

D. build *

E. EbuildApp

PDF 3: RAD e Python

****Questão 1: Modelagem de Negócios na RAD****

- A. Identifica as entidades de dados e como se relacionam com o negócio.
- B. É caracterizada por um processo rígido que vai determinar como será a versão final.
- C. É uma fase voltada para os desenvolvedores do sistema com interações técnicas.
- D. Identifica as informações relevantes sobre o negócio. *
- E. Nela, os testes do sistema são tratados de modo a validar os requisitos do sistema.

****Questão 2: Escolha de ferramentas/frameworks para interfaces****

- A. Licença, pois trata dos direitos de uso de um software.
- B. A ferramenta escolhida sempre deve ser a mais difícil de aprender para que se tenha um desenvolvimento rápido pelos membros da equipe.
- C. A ferramenta escolhida não deve fornecer capacidade de colaboração entre os membros da equipe.
- D. Integração, uma vez que é muito importante que o código esteja disponível no controlador de versões git. *
- E. Facilidade de uso e conforto, pois o usuário final precisa de atalhos para navegar no sistema.

****Questão 3: Modelagem de dados na RAD****

- A. O diagrama de entidade-relacionamento pode ser utilizado nesta fase. *
- B. Não é esperado que ao final da fase seja produzida uma documentação.
- C. Ao final da fase, o modelo de dados é concluído e não deve passar por modificações.

Consolidated Exercises

D. O sistema gerenciador de banco de dados sempre deve ser o mais eficiente na gerência de dados.

E. O modelo de dados deve ser baseado no diagrama de classes da UML (Unified Modeling Language).

****Questão 4: Python e projetos RAD****

A. Não é ideal para projetos RAD, dado que é orientada a objetos.

B. Por ter uma sintaxe muito simples, não é necessário que o desenvolvedor precise programar linhas de código.

C. Aplica-se perfeitamente bem para a RAD, dado que é considerada a melhor linguagem de programação.

D. É usada para construir a lógica do sistema, mas não a interface gráfica com o usuário.

E. Licença e simplicidade de sintaxe fazem da linguagem Python uma forte candidata para desenvolvimentos de projetos RAD. *

****Questão 5: Técnicas para levantamento de requisitos****

A. Os casos de uso são usados para descrever como os usuários devem usar o sistema. *

B. Simular situações de desastre e como usuários devem proceder.

C. A observação e análise social auxilia a mapear o comportamento do usuário.

D. Os grupos focais participam de debates para analisar possíveis vulnerabilidades do sistema.

E. O brainstorming estimula que as partes interessadas apresentem suas ideias sobre o sistema e como devem ser priorizadas.

PDF 4: Manipulação de Arquivos e Exceções em Python

****Questão 1: Tratamento de exceções****

A. Em branco

Consolidated Exercises

B. Comuns

C. Específicas *

D. Incomuns

E. Genéricas

****Questão 2: Método join com conector para nova linha****

A. Dividir coluna espaço.

B. Excluir célula tabulação.

C. Combinar linha '\n'. *

D. Separar tabela vírgula.

E. Marcar posição ponto.

****Questão 3: Saída do script com tratamento de exceções****

A. O programa não irá gerar nenhuma saída. *

B. Erro geral.

C. Erro de sistema.

D. Diretório não encontrado.

E. O mesmo erro apresentado na execução do script exercício_9a.py.

****Questão 4: Método para unir elementos de uma lista****

A. split()

B. concat()

C. append()

D. join() *

E. merge()

****Questão 5: Método para combinar elementos da lista****

A. join, separado. *

B. split, início, exclusão.

Consolidated Exercises

C. append, visualização, edição.

D. remove, exclusão, início.

E. find, programação, visualização.

****Questão 6: Manipulação de arquivos e exceções****

A. Strings, boas práticas, tratamento. *

B. Caracteres, diretrizes de codificação, lançamento.

C. Códigos, estilos de programação, registro.

D. Arquivos, padrões de segurança, rastreamento.

E. Dados, normas de desenvolvimento, log.

****Questão 7: Ler conteúdo de um arquivo****

A. `data = open("informacoes.txt", "w").read()`

B. `data = open("informacoes.txt", "a").read()`

C. `data = open("informacoes.txt", "r").read()` *

D. `data = open("informacoes.txt", "r").read()`

E. `data = open("informacoes.txt", "r").read()`

****Questão 8: Método para dividir string em lista****

A. `split()` *

B. `divide()`

C. `break()`

D. `join()`

E. `detach()`

****Questão 9: Ferramentas para manipulação de dados****

A. Compilação, interpretação, padrões de codificação.

B. Leitura, gravação, boas práticas. *

C. Análise, codificação, estilos de programação.

Consolidated Exercises

D. Execução, teste, diretrizes de codificação.

E. Configuração, instalação, normas de segurança.

****Questão 10: Manipulação de arquivos e exceções****

A. Strings, boas práticas, tratamento. *

B. Endereços IP, diretrizes de codificação, fluxo.

C. Códigos, estilos de programação, fluxo.

D. Sockets, padrões de segurança, fluxo.

E. Ponto flutuante, normas de desenvolvimento, log.

PDF 5: Bancos de Dados com Python e SQLite

****Questão 1: Tipos de dados em SQLite****

A. Atributos - Adicionar - Excluir. *

B. Correspondências - Remover - Criar.

C. Relações - Modificar - Eliminar.

D. Atributos - Modificar - Criar.

E. Correspondências - Adicionar - Eliminar.

****Questão 2: Comando SQL para selecionar nome e quantidade****

A. SELECT nome, quantidade FROM Produtos. *

B. INSERT INTO Produtos (nome, quantidade).

C. UPDATE Produtos SET nome, quantidade.

D. DELETE FROM Produtos WHERE nome, quantidade.

E. CREATE TABLE Produtos (nome, quantidade).

****Questão 3: Atualizar email de um membro****

A. SELECT email FROM membros WHERE id = 123.

B. DELETE FROM membros WHERE email = 'novo@email.com'.

Consolidated Exercises

C. INSERT INTO membros (email) VALUES ('novo@email.com') WHERE id = 123.

D. UPDATE membros SET email = 'novo@email.com' WHERE id = 123. *

E. CREATE TABLE membros_new AS SELECT * FROM membros.

****Questão 4: Inserção de dados com INSERT INTO****

A. Excluída - dicionário - foreign_keys.

B. Incluída - lista - check constraints.

C. Explícita - tupla - foreign_keys. *

D. Removida - array - primary keys.

E. Mantida - cat - unique keys.

****Questão 5: Técnica para evitar SQL Injection****

A. Hardening da comando SQL.

B. Utilização de procedimentos.

C. Uso de parâmetros dinâmicos. *

D. Aplicação de criptografia nos dados.

E. Implementação de chaves.

****Questão 6: Remover tabela Marca****

A. DELETE FROM Marca.

B. DROP TABLE Marca. *

C. REMOVE TABLE Marca.

D. TRUNCATE TABLE Marca.

E. ERASE TABLE Marca.

****Questão 7: Parâmetros dinâmicos em PostgreSQL****

A. Usando delimitadores '%' na string SQL e passando um dicionário como parâmetro.

B. Inserindo diretamente os valores na string SQL.

C. Utilizando o delimitador '?' na string SQL e passando os valores como uma lista.

Consolidated Exercises

D. Empregando a função `execute()` com parâmetros adicionais.

E. Utilizando delimitadores '%' na string SQL e passando os valores como uma tupla. *

****Questão 8: Formatar data de nascimento****

A. Usando a função `strftime` para converter a data em uma string. *

B. Utilizando o formato 'YYYY/MM/DD'.

C. Convertendo a data para timestamp UNIX.

D. Aplicando o método `date()` no objeto de data.

E. Formatando a data como 'DD-MM-YYYY'.

****Questão 9: Comando SQL para criar tabela****

A. CREATE TABLE. *

B. INSERT INTO.

C. UPDATE.

D. DELETE.

E. JOIN.