Matéria: Tecnologia da Informação

Assunto: Programação e Desenvolvimento de Software

Resumo Teórico do Assunto

Para ter sucesso nas questões sobre Programação e Desenvolvimento de Software, é fundamental compreender os conceitos-chave de diferentes ecossistemas e ferramentas. Abaixo, apresentamos um resumo teórico conciso, focado nos tópicos abordados pelas questões.

---

1. Manipulação de Dados com Pandas (Python)

Pandas é uma biblioteca de código aberto para a linguagem de programação Python, amplamente utilizada para manipulação e análise de dados. Sua estrutura de dados principal é o **DataFrame**.

• **DataFrame:** É uma estrutura de dados tabular bidimensional, com rótulos para linhas e colunas, semelhante a uma planilha ou tabela de banco de dados.
• **Seleção de Dados:**
* **Máscaras Booleanas (Boolean Indexing):** Permitem selecionar linhas de um DataFrame com base em uma condição lógica. Uma máscara booleana é uma Série de valores `True` ou `False` do mesmo tamanho do DataFrame, onde `True` indica as linhas a serem selecionadas.
* Exemplo: `df[df['coluna'] == valor]` seleciona todas as linhas onde a coluna 'coluna' tem o 'valor' especificado.
* **Seleção de Colunas:**
* Para selecionar uma única coluna, usa-se `df['nome_da_coluna']`. O resultado é uma **Series**.
* Para selecionar múltiplas colunas, é necessário passar uma **lista de nomes de colunas** dentro dos colchetes: `df[['coluna1', 'coluna2']]`. O resultado é um **DataFrame** contendo apenas as colunas especificadas.

2. Desenvolvimento Mobile (Android e React Native)

 # 2.1. Desenvolvimento Android (Java/Kotlin)

O desenvolvimento de aplicações nativas para Android envolve componentes específicos que definem a estrutura e o comportamento da aplicação.

• **Activity:** É um componente fundamental que representa uma **única tela com uma**

interface de usuário em uma aplicação Android. Cada tela que o usuário vê e interage é geralmente uma `Activity`. Elas são autônomas e podem ser iniciadas e gerenciadas pelo sistema.
• **Widgets:** São elementos da interface do usuário (UI) que o usuário pode ver e interagir. Exemplos incluem `TextView` (para exibir texto), `Button` (botões), `EditText` (campos de entrada de texto), etc.
• **Manipulação de Widgets:** Para interagir com os widgets programaticamente (por exemplo, alterar seu texto, habilitá-los/desabilitá-los), são utilizados métodos específicos da classe do widget. Para um `TextView`, o método para definir ou alterar o texto exibido é **`setText()`**.
* Exemplo: `nomeDoTextView.setText("Novo Texto");`

• **Outros Componentes Android (Contexto da Questão 46):**
* **Fragment:** Representa uma parte modular da interface do usuário de uma `Activity`. Pode ser reutilizado em múltiplas `Activities`.
* **Content Provider:** Gerencia o acesso a um conjunto estruturado de dados.
* **Intent:** Um objeto de mensagem usado para solicitar uma ação de outro componente do aplicativo ou de outro aplicativo.
* **Manifest (AndroidManifest.xml):** Um arquivo XML que descreve a estrutura essencial do aplicativo, seus componentes, permissões e requisitos.

 # 2.2. Desenvolvimento com React Native

React Native é um framework JavaScript para construir aplicativos móveis nativos para iOS e Android a partir de uma única base de código.

• **Componentes Funcionais:** Uma forma moderna de escrever componentes em React/React Native, que são funções JavaScript que retornam elementos React.
• **Hooks:** Introduzidos no React 16.8, os Hooks são funções que permitem "enganchar" (hook into) recursos do React, como o estado e o ciclo de vida, a partir de componentes funcionais, sem a necessidade de escrever classes.
• **`useState` Hook:** É um dos Hooks mais importantes. Ele permite que componentes funcionais tenham **estado local**. O `useState` retorna um par de valores: o estado atual e uma função para atualizá-lo. É fundamental para gerenciar dados que mudam ao longo do tempo dentro de um componente e que afetam sua renderização.
* Exemplo: `const [count, setCount] = useState(0);`

3. Automação e Gerenciamento de Configuração com Ansible

Ansible é uma ferramenta de automação de TI de código aberto, utilizada para gerenciamento de configuração, implantação de aplicações e orquestração de tarefas. É conhecido por sua simplicidade e por ser "agentless" (não requer software cliente nos servidores gerenciados).

• **Playbooks:** São o coração do Ansible. São arquivos escritos em **YAML** (YAML Ain't Markup Language) que descrevem as tarefas de automação a serem executadas.

• **Estrutura de um Playbook:** Um playbook é composto por uma ou mais **plays**.
* **Play:** Uma play é uma sequência de **tasks** que são executadas em um grupo específico de hosts (servidores). Cada play define em quais hosts as tarefas serão executadas e quais usuários serão usados.
* **Task:** Uma task é uma única ação a ser executada. Cada task chama um **módulo** Ansible.
* **Module:** São as unidades de código que o Ansible executa nos hosts gerenciados para realizar uma tarefa específica (ex: instalar um pacote, copiar um arquivo, iniciar um serviço, gerenciar usuários).

• **Hierarquia:** A estrutura lógica de um playbook Ansible segue a seguinte hierarquia:
Playbook → **Plays** → **Tasks** → **Modules**

Compreender esses conceitos e a forma como eles se interligam em cada tecnologia é crucial para analisar e responder corretamente às questões de programação e desenvolvimento de software.

Questões de Provas Anteriores

Fonte: escriturario_agente_de_tecnologia.pdf, Página: 14

pcimarkpci MjgwNDowMTRkOjE0YTU6OTI1ODozOGQ2OjNhMGM6NTM0MzplZmI1:U3VuLCAyNyBKdWwgMjAyNSAyMzo0NzozMSAtMDMwMA==
www.pciconcursos.com.br
AGENTE DE TECNOLOGIA - Microrregião 158 -TI
14
GABARITO 1
BANCO DO BRASIL
45
Ao programar em Python com Pandas, é possível usar máscaras para selecionar linhas específicas, de acordo com um
padrão.
Nesse cenário, analise o seguinte código:

```
import pandas as pd
data = {'x':[1,2,3], 'y':[3, 7, 11], 'z': [False, True, False]}
df = pd.DataFrame(data)
m = df['z'] == False
ef = df[m]
# a fazer
print(ff)
```

Ao executar esse código, deseja-se obter a seguinte saída:

x y
0 1 3
2 3 11
O fragmento de código que deve substituir o comentário # a fazer para obter a saída desejada é

(A) ff = ef['x','y']
(B) ff = ef[] == 'x' or 'y'
(C) ff = ef[['x','y']]
(D) ff = ef.cols('x','y')
(E) ff = ef.cols(['x','y'])

46

Um programador recebeu a incumbência de desenvolver uma aplicação móvel segundo a API 30 do Android, correspondente ao Android 11. Seguindo as melhores práticas, cada tela dessa aplicação, incluindo sua funcionalidade, foi construída como um módulo único e autônomo, totalmente independente de outros módulos similares.
Esse módulo único e autônomo é conhecido como

(A) activity
(B) content provider
(C) fragment
(D) intent
(E) manifest

47

O React Native 0.59 introduziu o conceito de Hooks.
Entre os Hooks, tem-se o usestate, que permite

(A) calcular o estado de um CEP ou ZIP de acordo com o Locale.
(B) chamar estados específicos do engine React para alterar seu comportamento.
(C) declarar uma classe que segue o padrão de design state.
(D) criar uma enumeration que representa estados.
(E) manter um estado local em uma função de um componente funcional.

48

Durante o desenvolvimento de uma aplicação mobile em Java para Android, um programador detectou a necessidade de alterar o texto de um widget da classe TextView, chamado resultado, para "Sucesso!".
Para realizar essa ação, esse programador deve usar o seguinte fragmento de código:

(A) TextView resultado = "Sucesso!";
(B) resultado := "Sucesso!";
(C) resultado.setValue("Sucesso!");
(D) resultado.setText("Sucesso!");
(E) resultado = TextView.setValue("Sucesso!");

49

Ansible é uma ferramenta configurável por playbooks, escritos em YAML.

Um playbook é composto de

(A) plays, que são sequências de modules que, por sua vez, chamam tasks.

(B) plays, que são sequências de tasks que, por sua vez, chamam modules.

(C) tasks, que são sequências de modules que, por sua vez, chamam plays.

(D) tasks, que são sequências de plays que, por sua vez, chamam modules.

(E) modules, que são sequências de tasks que, por sua vez, chamam play.