

Matéria: Tecnologia da Informação

Assunto: Sistemas, Desenvolvimento e Infraestrutura de TI

Resumo Teórico do Assunto

Este resumo aborda conceitos fundamentais em **Sistemas, Desenvolvimento e Infraestrutura de TI**, essenciais para a compreensão de tópicos como Big Data, desenvolvimento mobile, lógica de programação e automação de infraestrutura.

1. Pré-processamento e Qualidade de Dados em Big Data

Em ambientes de **Big Data**, onde lidamos com um volume massivo, variedade e velocidade de dados, a qualidade dos dados é crucial.

- **Dados Ruidosos (Noisy Data):** São dados inconsistentes, incompletos, irrelevantes ou fora de sentido que podem surgir em grandes conjuntos de dados. Eles podem prejudicar a análise e a interpretação, levando a resultados imprecisos.

- **Pré-processamento de Dados:** É uma etapa fundamental no ciclo de vida do Big Data e da mineração de dados. Seu objetivo é transformar os dados brutos em um formato limpo e adequado para análise. Isso inclui tarefas como limpeza, integração, transformação e redução de dados.

- **Suavização de Dados (Data Smoothing):** É uma técnica de pré-processamento utilizada para reduzir o efeito de dados ruidosos. O objetivo é remover o "ruído" e destacar os padrões ou tendências subjacentes nos dados.

- * **Binning (Compartimentação):** Um método comum de suavização. Consiste em dividir os valores dos dados originais em pequenos intervalos, chamados **compartimentos (bins)**. Em seguida, os valores dentro de cada compartimento são substituídos por um valor geral ou genérico, como a média, a mediana ou os limites do compartimento. Isso ajuda a agrupar valores próximos e a reduzir a variabilidade causada pelo ruído.

2. Lógica de Programação e Linguagem Swift

A **lógica de programação** é a base para a criação de qualquer software, e as linguagens de programação fornecem as ferramentas para expressá-la.

- **Linguagem Swift:** É uma linguagem de programação moderna, desenvolvida pela Apple, utilizada principalmente para criar aplicativos para os sistemas operacionais **iOS**, macOS, watchOS e tvOS.

- **Variáveis:** São espaços na memória do computador usados para armazenar dados. Em Swift, são declaradas com a palavra-chave ``var`` (para variáveis mutáveis) ou ``let`` (para

constantes imutáveis).

* **Tipos de Dados:** Definem o tipo de valor que uma variável pode armazenar. Exemplos incluem:

* ``Int``: Para números inteiros.

* ``String``: Para sequências de caracteres (texto).

• **Estruturas de Controle de Fluxo:** Determinam a ordem em que as instruções de um programa são executadas.

* **Laços de Repetição (``for...in``):** Permitem executar um bloco de código repetidamente para cada item em uma sequência (como um intervalo de números ou elementos de uma coleção).

* **Estruturas Condicionais (``switch``):** Permitem executar diferentes blocos de código com base no valor de uma variável ou expressão. O ``switch`` compara um valor com vários ``case``s possíveis. O ``default`` é executado se nenhum ``case`` corresponder.

• **Operadores:** Símbolos que realizam operações em valores e variáveis.

* **Operadores Aritméticos:** Como ``+`` (adição), ``-`` (subtração), ``*`` (multiplicação), ``/`` (divisão).

* **Operadores de Intervalo:** Como ``...`` (intervalo fechado, incluindo ambos os limites) usados em laços ``for`` ou ``case``s de ``switch``.

• **Saída de Dados (``print``):** Função utilizada para exibir informações (texto, valores de variáveis) no console ou em outra saída padrão.

3. Desenvolvimento de Aplicativos Mobile (iOS/Xcode)

O desenvolvimento de aplicativos para dispositivos móveis exige ferramentas específicas para criar a **Interface Gráfica do Usuário (GUI)** e gerenciar o projeto.

• **Desenvolvimento Mobile:** Processo de criação de software para smartphones, tablets e outros dispositivos móveis.

• **iOS:** É o sistema operacional móvel desenvolvido pela Apple Inc. para seus dispositivos iPhone, iPad e iPod Touch.

• **Xcode:** É o **Ambiente de Desenvolvimento Integrado (IDE)** oficial da Apple para desenvolver aplicativos para iOS, macOS, watchOS e tvOS. Ele inclui um editor de código, compilador, depurador e ferramentas para design de interface.

• **Interface Gráfica do Usuário (GUI):** Refere-se à parte visual de um aplicativo com a qual o usuário interage (botões, campos de texto, imagens, etc.).

• **Interface Builder:** É uma ferramenta visual integrada ao Xcode que permite aos desenvolvedores projetar e construir a GUI de seus aplicativos arrastando e soltando elementos visuais (como botões, rótulos, imagens) em uma tela. Ele gera o código XML (ou outro formato) que descreve o layout da interface, que é então carregado pelo aplicativo em tempo de execução.

4. Automação e Gerenciamento de Configuração com Ansible

Em ambientes de TI modernos, especialmente em **clusters** e infraestruturas complexas, a **automação** é essencial para gerenciar e configurar servidores de forma eficiente.

- **Automação de TI:** O uso de software para automatizar tarefas repetitivas e processos de gerenciamento de infraestrutura, como provisionamento de servidores, implantação de software e gerenciamento de configurações.
 - **Gerenciamento de Configuração:** Disciplina que garante que os sistemas de computador (servidores, redes, etc.) estejam em um estado consistente e desejado, seguindo padrões e políticas predefinidas.
 - **Ansible:** É uma ferramenta de **automação de TI** e **gerenciamento de configuração** de código aberto. Ele é conhecido por ser **agentless**, o que significa que não requer a instalação de software cliente nos servidores gerenciados; ele se comunica via SSH (para Linux/Unix) ou WinRM (para Windows).
 - **YAML (YAML Ain't Markup Language):** É uma linguagem de serialização de dados legível por humanos, frequentemente usada para arquivos de configuração. O Ansible utiliza YAML para definir suas tarefas e configurações.
 - **Playbook:** É o principal componente do Ansible. Um **Playbook** é um arquivo escrito em **YAML** que descreve um conjunto de tarefas que o Ansible deve executar em um ou mais servidores. Ele define os "plays" (conjuntos de tarefas para um grupo de hosts) e as "tasks" (ações específicas a serem realizadas, como instalar um pacote, copiar um arquivo, iniciar um serviço). Os Playbooks permitem automatizar processos complexos de forma declarativa e repetível.
-

Questões de Provas Anteriores

Fonte: escriturario_agente_de_tecnologia (1).pdf, Página: 22

pcimarkpci MjgwNDowMTRkOjE0YTU6OTI1ODozOGQ2OjNhMGM6NTM0MzplZmI1:U3VuLCAyNyBKdWwgMjAyNSAyMzo0Nzo0MCAtMDMwMA==

www.pciconcursos.com.br

AGENTE DE TECNOLOGIA - Microrregião 16 DF-TI 22

BANCO DO BRASIL

GABARITO 1

57

Na etapa de preparação de dados em um ambiente de Big Data, pode ocorrer o aparecimento de dados ruidosos, que são dados fora de sentido no ambiente do Big Data e que não podem ser interpretados pelo sistema. Uma forma de reduzir esse efeito, e até eliminá-lo, é utilizar um método de suavização de dados, que minimiza os efeitos causados pelos dados ruidosos.

Esse método consiste em

(A) agrupar dados semelhantes em clusters, verificar os

dados que se apresentam como ruidosos e não os inserir no ambiente de Big Data, substituindo cada um desses dados ruidosos pelo valor NULL.

(B) criar um cubo de dados multidimensional para acelerar a identificação e a eliminação dos dados ruidosos encontrados, e, neste caso, os dados assumem valores predefinidos no momento do pré-processamento.

(C) dividir os valores dos dados originais em pequenos intervalos, denominados compartimentos, e, em seguida, substituí-los por um valor geral, ou genérico, calculado para cada compartimento específico.

(D) executar uma fusão de dados, tendo como base dados vizinhos, e obter novas variáveis que irão preencher os espaços incoerentes dentro do Big Data, eliminando o ruído.

(E) realizar uma ação de Data Mining com atributos que se pareçam com dados ruidosos, gerando novos atributos, classificados como atributos “fantasmas”, sem valor para o tratamento do Big Data.

58

Um programador de aplicativos para dispositivos Apple com iOS recebeu a seguinte parte de um código, escrito na linguagem swift:

```
var i:Int
var texto:String
var num:Int=0
var frase:String=""
for i in 1...3
{ num = num+1+i*2
switch num{
case 2...6: texto = "a "
case 7...9: texto = "casa "
case 10...13: texto = "carro "
case 14...16: texto = "eh "
case 17...20: texto = "o "
case 21...23: texto = "forte "
default:
texto = "não eh " }
frase = frase+texto }
print (frase)
```

A execução dessa parte do código produz como resposta

(A) o carro eh

(B) a casa eh

(C) o carro não eh

(D) a casa eh forte

(E) o carro eh forte

59

O desenvolvimento de um aplicativo bancário para dispositivos móveis exige que se dedique atenção à parte visual do mesmo, que vai depender, também, do modelo do aparelho e do tipo de sistema operacional (SO) que esteja rodando nele.

Caso um desenvolvedor de aplicativos para um iPhone com SO iOS precise fazer a programação visual das telas desses aplicativos, que ferramenta do Xcode 10 ele deve utilizar?

(A) Interface Builder

(B) Bundle Identifier

(C) Organizer Interface

(D) Apple LLVM

(E) Instruments

60

Em um ambiente Cluster, há um servidor Linux que tem o Ansible instalado.

Para construir um arquivo YAML, que informe os passos que o Ansible realizará automaticamente na conexão com os servidores que fazem parte do Cluster, além de executar um conjunto de tarefas solicitadas, o administrador desse servidor deve seguir o padrão

(A) Apache-Start

(B) Nodel

(C) Playbook

(D) Taskbook

(E) Tower

RASCUNHO