

AUDITOR FISCAL DA RECEITA ESTADUAL TECNOLOGIA DA INFORMAÇÃO



SUA PROVA

- Além deste caderno contendo **4 (quatro) questões discursivas**, você receberá do fiscal de prova as folhas de textos definitivos.



TEMPO

- Você dispõe de **4 (quatro) horas e 30 (trinta) minutos** para a realização da prova, já incluído o tempo para o preenchimento das folhas de textos definitivos;
- 3 (três) horas** após o início da prova é possível retirar-se da sala, sem levar o caderno de questões;
- A partir dos **30 (trinta) minutos** anteriores ao término da prova é possível retirar-se da sala **levando o caderno de questões**.



NÃO SERÁ PERMITIDO

- Qualquer tipo de comunicação entre os candidatos durante a aplicação da prova;
- Anotar informações relativas às respostas em qualquer outro meio que não seja o caderno de questões;
- Levantar da cadeira sem autorização do fiscal de sala;
- Usar o sanitário ao término da prova, após deixar a sala.



INFORMAÇÕES GERAIS

- Verifique se seu caderno de questões está completo, sem repetição de questões ou falhas. Caso contrário, notifique imediatamente o fiscal da sala, para que sejam tomadas as devidas providências.
- Confira seus dados pessoais, especialmente nome, número de inscrição e documento de identidade, e leia atentamente as instruções para preencher as folhas de textos definitivos.
- Use somente caneta esferográfica, fabricada em material transparente, com tinta preta ou azul.
- Assine seu nome apenas no(s) espaço(s) reservado(s).
- Confira seu cargo, cor e tipo do caderno de questões. Caso tenha recebido caderno de cargo **diferente** do impresso em suas folhas de textos definitivos, o fiscal deve ser **obrigatoriamente** informado para o devido registro na ata da sala.
- Reserve tempo suficiente para o preenchimento das suas folhas de textos definitivos. O preenchimento desses documentos é de sua responsabilidade e **não** será permitida a troca em caso de erro cometido pelo candidato.
- Para fins de avaliação, serão levadas em consideração apenas as marcações realizadas nas folhas de textos definitivos.
- A FGV coletará as impressões digitais dos candidatos na lista de presença.
- Os candidatos serão submetidos ao sistema de detecção de metais quando do ingresso e da saída de sanitários durante a realização das provas.

Boa sorte!

Questão 1

O termo *palíndromo* aplica-se a uma palavra que pode ser lida, indiferentemente, da esquerda para a direita ou vice-versa, como as palavras *sopapos*, *ama* e *radar*.

Nesse contexto, considere a especificação de uma função intitulada **palindromo**, que analisa se uma palavra é um palíndromo ou não. A função recebe como parâmetro uma palavra, na forma de uma string, e retorna uma string de acordo com a seguinte regra:

?, se a palavra tem menos de 3 caracteres;

S, se a palavra é um palíndromo;

N, se a palavra não é um palíndromo.

Exemplos dos valores retornados pela função.

palindromo("") → "?"

palindromo("anilina") → "S"

palindromo("mama") → "N"

palindromo("mm") → "?"

- a. Apresente um código Java Script que implemente a função **palindromo** como especificada.
- b. Apresente um código Python que implemente a função **palindromo** como especificada.

Observação:

As respostas apresentadas

- i) devem manter a assinatura original e a especificação original;
- ii) devem manter a sintaxe da linguagem de programação preconizada em cada item;
- iii) não devem conter comentários;
- iv) não devem conter trechos de texto não pertencentes ao código requisitado, pois serão considerados erros de sintaxe.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

Questão 2

Considere um banco de dados relacional que contém três tabelas, em que são registradas as vendas efetuadas por cada um dos vendedores de uma loja de móveis. A tabela **VENDEDOR** armazena os nomes dos vendedores, tendo a coluna **Nome** como chave primária. A tabela **PRODUTO** armazena os produtos e respectivos preços unitários de venda, tendo a coluna **Tipo** como chave primária. A tabela **VENDA** armazena a quantidade de cada tipo de produto que foi vendido por vendedor, tendo **Nome** e **Tipo** formando a chave primária. Na tabela **VENDA**, a coluna **Nome** é uma chave estrangeira que referencia a tabela **VENDEDOR**, e a coluna **Tipo** é uma chave estrangeira que referencia a tabela **PRODUTO**. A seguir, a ilustração de uma eventual instância válida para essas tabelas.

VENDEDOR		PRODUTO		VENDA		
Nome		Tipo	Valor	Nome	Tipo	Quantidade
Gabriela		Cadeira	800,00	Gabriela	Cadeira	8
João		Mesa	1400,00	Gabriela	Mesa	2
Maria		Sofá	2600,00	Gabriela	Sofá	3
Pedro				João	Cadeira	4
				João	Mesa	1
				João	Sofá	3
				Maria	Cadeira	2
				Maria	Mesa	1

- a. Apresente um comando **SELECT** do **SQL** que retorne o nome e o valor total das vendas de cada vendedor, incluindo todos os vendedores. As linhas do resultado devem estar ordenadas em ordem decrescente de valor total das vendas, e nome em ordem alfabética no caso de empate no valor total. A título de exemplo, veja a seguir o resultado desejado considerando a instância apresentada acima.

Nome	Total
Gabriela	17000,00
João	12400,00
Maria	3000,00
Pedro	0,00

- b. Apresente um comando **SELECT** do **SQL** que retorne os nomes dos vendedores que venderam móveis de todos os tipos, em ordem alfabética. Esse comando, e eventuais subconsultas, não podem conter funções agregadas como *count*, *sum*, etc. A título de exemplo, veja a seguir o resultado desejado considerando-se a instância apresentada acima.

Nome
Gabriela
João

Observação:

As respostas apresentadas:

- devem apresentar resultados corretos para qualquer instância válida das tabelas;
- não podem utilizar cursores, tabelas temporárias ou qualquer tipo de programação;
- não podem alterar as características das tabelas;
- devem evitar construções sintáticas incomuns do **SQL**;
- não devem conter limitadores de linhas, tais com **TOP**, **LIMIT**, dentre outros;
- não devem conter comentários, nem explicações, pois serão considerados erros de sintaxe.

Avaliação:

Nos dois itens, serão avaliadas

- a estrutura dos comandos e a correção dos resultados produzidos em cada item;
- a organização e a sintaxe dos *scripts*.

ATENÇÃO: a não observação de uma, ou mais, das restrições estabelecidas pode acarretar a atribuição da nota zero.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

Questão 3

José desenvolve um sistema de classificação de imagens para um *Pet Shop*. O sistema deve classificar, com boa acurácia, imagens de clientes que entram no *Pet Shop* com cães e gatos. No entanto, não é necessário identificar a raça do cão ou do gato, apenas se o cliente entra no *Pet Shop* com um cão ou um gato, o que caracteriza uma tarefa de classificação de duas classes.

Nesse contexto, considere o trecho de código Python a seguir.

```
1 from keras.models import Sequential
2 from keras.layers import Activation, Dropout, Flatten, Dense, Conv2D, MaxPooling2D
3
4 modelo = Sequential()
5
6 modelo.add(Conv2D(filters=32, kernel_size=(3,3),input_shape=(150,150,3), activation='relu',))
7 modelo.add(MaxPooling2D(pool_size=(2, 2)))
8
9 modelo.add(Conv2D(filters=64, kernel_size=(3,3),input_shape=(150,150,3), activation='relu',))
10 modelo.add(MaxPooling2D(pool_size=(2, 2)))
11
12 modelo.add(Conv2D(filters=64, kernel_size=(3,3),input_shape=(150,150,3), activation='relu',))
13 modelo.add(MaxPooling2D(pool_size=(2, 2)))
14
15 modelo.add(Flatten())
16
17 modelo.add(Dense(128))
18 modelo.add(Activation('relu'))
19
20 modelo.add(Dropout(0.5))
21
22 modelo.add(Dense(10))
23 modelo.add(Activation('softmax'))
24
25 modelo.compile(loss='binary_crossentropy',
26               optimizer='adam',
27               metrics=['accuracy'])
```

- Dado que se trata de uma tarefa de classificação para duas classes (cão e gato), há um equívoco no código que define a rede neural. Identifique e substitua e/ou reescreva e/ou remova a(s) linha(s) que devem ser modificadas. Use a numeração de linhas à esquerda como referência nas suas respostas quando necessário. As linhas que permanecem como estão não devem ser reescritas.
- Explique, sucintamente, a função do código na linha 20.
- Além da criação do modelo definido no código acima, José gerou um dataset de treinamento *'train'* e um dataset de validação *'val'* com o método *ImageGenerator* do framework Keras. Com base no dataset gerado por José, apresente um script para treinar o modelo definido no código acima, utilizando o método *fit_generator* com 50 épocas (epoch), 50 passos (steps) por época (epoch), e 12 passos de validação. O resultado do treinamento deve ser armazenado na variável *'results'*. Na sua resposta, tome como base a estrutura exibida a seguir:

```
modelo.fit_generator(..., epochs=...,
                    steps_per_epoch=...,
                    validation_data=...,
                    validation_steps=...)
```

Observação:

As respostas apresentadas

- devem manter a sintaxe adequada, quando aplicável;
- não devem conter comentários;
- não devem conter trechos com conteúdo alheio ao que é expressamente requisitado.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

Questão 4

Considere um Deployment para executar um servidor web Nginx em um cluster Kubernetes. O código YAML atual para o Deployment é o seguinte:

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: nginx-deploy
5  spec:
6    selector:
7      matchLabels:
8        app: nginx
9  template:
10   metadata
11   labels:
12     app: nginx
13   spec:
14     containers:
15       name: nginx
16       image:nginx:1.16
```

- Apresente um trecho de código que pode ser inserido no exemplo acima que define o número de réplicas para 3. Use a numeração de linhas original da coluna da esquerda como referência para indicar claramente onde o código deve ser inserido.
- Apresente um trecho de código que pode ser inserido no exemplo acima que define o limite do consumo de CPU para 250m e o consumo de memória para 250Mi para cada réplica. Use a numeração de linhas original da coluna da esquerda como referência para indicar claramente onde o código deve ser inserido.
- Apresente um trecho de código que pode ser inserido no exemplo acima que libere as portas HTTP e HTTPS para o servidor web. Use a numeração de linhas original da coluna da esquerda como referência para indicar claramente onde o código deve ser inserido.

Observação:

As respostas apresentadas

- devem manter a sintaxe do arquivo YAML para Kubernetes;
- não devem conter comentários;
- não devem conter trechos com conteúdo alheio ao que é expressamente requisitado.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

Realização

