

Trabalho Prático do Módulo 2

Entrega 31 jul em 19:30 **Pontos** 25 **Perguntas** 15
Disponível até 31 jul em 19:30 **Limite de tempo** Nenhum

Instruções

O Trabalho Prático do Módulo 2 está disponível!

1. Instruções para realizar o trabalho prático

Consulte a data de entrega no teste e em seu calendário.

Reserve um tempo para realizar a atividade, leia as orientações e enunciados com atenção. Em caso de dúvidas utilize o "Fórum de dúvidas do Trabalho Prático".

Para iniciá-lo clique em "Fazer teste". Você tem somente **uma** tentativa e não há limite de tempo definido para realizá-lo. Caso precise interromper a atividade, apenas deixe a página e, ao retornar, clique em "Retomar teste".

Clique em "Enviar teste" **somente** quando você concluí-lo. Antes de enviar confira todas as questões.

O gabarito será disponibilizado partir de sexta-feira, **31/07/2020**, às 21h.

Bons estudos!

2. O arquivo abaixo contém o enunciado do trabalho prático

[Enunciado do Trabalho Prático - Módulo 2 - Bootcamp Cientista de Dados.pdf](#) 

[Arquivos complementares - Módulo 2.zip](#)

[Tutorial de instalação do MySQL.pdf](#) 

Histórico de tentativas

	Tentativa	Tempo	Pontuação
MAIS RECENTE	Tentativa 1	157 minutos	25 de 25

⚠ As respostas corretas estarão disponíveis em 31 jul em 21:00.

Pontuação deste teste: **25** de 25

Enviado 29 jul em 2:12

Esta tentativa levou 157 minutos.

Pergunta 1	1,66 / 1,66 pts

Considerando a atividade 1, marque a alternativa **CORRETA**:



A API do Twitter é composta por um conjunto de aplicações que utilizam protocolos para troca de mensagens (request/response), permitindo a integração o entre o twitter e outros softwares ou aplicações.



A API do Twitter é composta por um conjunto de aplicações disponíveis por meio de Web Services, que permitem que qualquer usuário as utilize para criar integrações entre seu aplicativo e o twitter, sem a necessidade de qualquer autorização para ser utilizada.



A API do Twitter é uma linguagem comercial usada exclusivamente para integrar o Twitter com SGBDs relacionais.



Qualquer pessoa pode ter acesso à API do Twitter e às suas funcionalidades, independente de possuir uma conta no Twitter.

Pergunta 2

1,66 / 1,66 pts

A API do Twitter é uma linguagem comercial usada exclusivamente para integrar o Twitter com SGBDs relacionais.



Access token



Revoke Key



API key



API secret key

Pergunta 3

1,66 / 1,66 pts

Após receber a autorização de acesso à API do twitter, você criou sua aplicação. Todas as alternativas abaixo representam um campo obrigatório para o cadastro da aplicação, **EXCETO**.





☐ Website URL☐ App name☒ Callback URLs☐ Application description

Pergunta 4





1,66 / 1,66 pts









Após completar as atividades 3 e 4, nas quais você realizou a instalação do Framework Anaconda e a criação do ambiente de desenvolvimento Python/R denominado “dev”, marque a alternativa que apresenta apenas aplicativos disponíveis (instalados) neste ambiente.

☐

 CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated	 Notebook 6.0.3 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	 Qt Console 4.7.4 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.	 Glueviz 0.15.2 Multidimensional data visualization across files. Explore relationships within and among related datasets.
---	---	---	--

☒

 CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated	 Notebook 6.0.3 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	 RStudio 1.1.456 A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.	 Spyder 3.3.6 Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
---	---	---	---

 <p>CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated</p>	 <p>Jupyter Notebook 6.0.3 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.</p>	 <p>veusz 3.2 Veusz is a GUI scientific plotting and graphing package. It is designed to produce publication-ready Postscript or PDF output.</p>	 <p>Spyder 3.3.6 Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features</p>
 <p>Orange 3 3.26.0 Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.</p>	 <p>Jupyter Notebook 6.0.3 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.</p>	 <p>RStudio 1.1.456 A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.</p>	 <p>Spyder 3.3.6 Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features</p>

Pergunta 5

1,66 / 1,66 pts

Considerando a atividade 6, analise as alternativas abaixo e marque aquela que **NÃO** apresenta um comando válido ou para a criação do esquema *bootcamp* e de suas respectivas tabelas, ou para a carga de dados das tabelas.



```
INSERT TABLE `cidade` (
  `CodigoCompletoIBGE` varchar(45) NOT NULL,
  `CodigoCidadeIBGE` varchar(10) NOT NULL,
  `NomeCidade` varchar(150) NOT NULL,
  `CodEstadoIBGE` int NOT NULL,
  PRIMARY KEY (`CodigoCompletoIBGE`),
  KEY `fk_Cidade_Estado_idx` (`CodEstadoIBGE`),
  CONSTRAINT `fk_Cidade_Estado` FOREIGN KEY (`CodEstadoIBGE`)
  REFERENCES `estado` (`CodEstadoIBGE`)
);
```



```
CREATE TABLE `estado` (
  `CodEstadoIBGE` int NOT NULL,
  `NomeEstado` varchar(45) NOT NULL,
  `SiglaEstado` char(2) NOT NULL,
  `Regiao` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`CodEstadoIBGE`)
);
```



```
INSERT INTO `bootcamp`.`tipounidade` (`idTipoUnidade`, `dscTipoUnidade`) VALUES (1, 'Casa');
```



```
CREATE SCHEMA IF NOT EXISTS `bootcamp` DEFAULT CHARACTER SET utf8 ;
```

Pergunta 6

1,66 / 1,66 pts

Qual das opções abaixo **NÃO** pode ser utilizada para recuperar todos os dados da tabela *estado*?



```
query <- "SELECT CodEstadoIBGE, NomeEstado, SiglaEstado, Regiao FROM estado"
resultado <- dbSendQuery(con, query)
resultado
```



```
query <- "SELECT CodEstadoIBGE, NomeEstado, SiglaEstado, Regiao FROM estado"
resultado <- dbGetQuery(con, query)
resultado
```



```
query <- "SELECT * FROM estado"
resultado <- dbGetQuery(con, query)
resultado
```



```
resultado <- dbReadTable(con, "estado")
resultado
```

Pergunta 7

1,66 / 1,66 pts

Considerando o programa abaixo, escrito na linguagem R, marque a opção que represente o resultado esperado para execução do comando da linha 13.

```

1 #Importa pacote RMariaDB se ele ainda não foi carregado
2 if(!"RMariaDB" %in% (.packages())){require(RMariaDB)}
3
4 #Conecta ao SGBD MySQL --> Banco de dados bootcamp
5 con <- dbConnect(MariaDB(), user = "root", password = "igti",
6                 dbname = "bootcamp", host = "localhost", serverTimezone='UTC')
7
8 resultado <- dbListTables(con)
9 resultado
10
11 resultado <- dbReadTable(con, "tipounidade")
12 resultado <- dbReadTable(con, "caracteristicasgerais")
13 resultado
14
15 resultado <- dbSendQuery(con, "INSERT INTO tipounidade(idTipoUnidade, dscTipoUnidade) VALUES(9, 'Fazenda');")
16 resultado
17
18 query <- "SELECT * FROM caracteristicasgerais;"
19 resultado <- dbGetQuery(con, query)
20 resultado

```



'caracteristicageralimovel' 'caracteristicasgerais' 'cidade' 'estado' 'imovel' 'tipounidade'



idcaracteristicasGerais **dsccharacteristicasGerais**



```

<MariaDBResult>
SQL  INSERT INTO tipounidade(idTipoUnidade,dscTipoUnidade) VALUES(9,'Fazenda');
ROWS Fetched: 0 [complete]
Changed: 1

```



idcaracteristicasGerais	dsccharacteristicasGerais
1	Portaria 24 horas
2	Elevador
3	Piscina

Pergunta 8

1,66 / 1,66 pts

Considerando o programa abaixo, escrito na linguagem R, marque a opção **INCORRETA**.

```
1 #Importa pacote RMariaDB se ele ainda não foi carregado
2 if(!"RMariaDB" %in% (.packages())){require(RMariaDB)}
3
4 #Conecta ao SGBD MySQL --> Banco de dados bootcamp
5 con <- dbConnect(MariaDB(), user = "root", password = "igti",
6                 dbname = "bootcamp", host = "localhost", serverTimezone='UTC')
7
8 resultado <- dbListTables(con)
9 resultado
10
11 resultado <- dbReadTable(con, "tipounidade")
12 resultado <- dbReadTable(con, "caracteristicasgerais")
13 resultado
14
15 resultado <- dbSendQuery(con, "INSERT INTO tipounidade(idTipoUnidade, dscTipoUnidade) VALUES(9, 'Fazenda');")
16 resultado
17
18 query <- "SELECT * FROM caracteristicasgerais;"
19 resultado <- dbGetQuery(con, query)
20 resultado
```



O comando escrito nas linhas 5 e 6 serve para fazer a conexão do script R com o banco de dados MySQL.



Na linha 8, o método dbListTables(con) é usado para consultar todas as tabelas existentes no banco de dados à qual foi conectado. A linha 9 exibe o resultado obtido, que é:

'caracteristicageralimovel' 'caracteristicasgerais' 'cidade' 'estado' 'imovel' 'tipounidade'



Na linha 8, o método dbListTables(con) é usado para consultar todas as tabelas existentes no banco de dados à qual foi conectado. A linha 9 exibe o resultado obtido, que é:



Na linhas 11 e 12, o comando dbReadTable é usado para ler o script de criação (comando DDL) da tabela que é passada como parâmetro.

Pergunta 9

1,66 / 1,66 pts

Considerando as atividades 7 e 8, marque a alternativa que **NÃO** serve para incluir dados em uma tabela do seu banco de dados MySQL (como a tabela “caracteristicasgerais”) utilizando a linguagem R.



```
query <- "INSERT INTO caracteristicasGerais(idcaracteristicasGerais,dsccharacteristicasGerais)
VALUES(4,'Salão de festas');"

dbWriteTable(con, value = query, row.names = FALSE,
             name = "caracteristicasgerais", append = TRUE )
```



```
query <- "INSERT INTO caracteristicasGerais(idcaracteristicasGerais,dsccharacteristicasGerais)
VALUES(4,'Salão de festas');"
results <- dbSendQuery(con,query)
```



```
filename <- "C:/Bootcamp/Datasets/CSV/caracteristicasgerais.csv"
insertdata <- read.csv(file=filename, header=TRUE, sep=",")
insertdata

dbWriteTable(con, value = insertdata, row.names = FALSE,
             name = "caracteristicasgerais", append = TRUE )
```



```
id1 <- 5
desc1 <- 'Área gourmet'
id2 <- 6
desc2 <- 'Água individual'
id3 <- 7
desc3 <- 'Gás canalizado'
query <- paste("INSERT INTO caracteristicasGerais(idcaracteristicasGerais,dsccharacteristicasGerais)
VALUES(",id1,",",",desc1,"'),(",id2,",",",desc2,"'),(",id3,",",",desc3,"');",sep='')
results <- dbSendQuery(con,query)
```

Pergunta 10

1,66 / 1,66 pts

Considere a atividade 9. Usando o pacote `mysql.connector`, na linguagem Python, o comando usado para executar um comando no banco de dados MySQL é:

- ☐ `csv.reader(con, query)`
- ☒ `mycursor.execute("comando-desejado")`
- ☐ `mycursor.readlines("nome-da-tabela-desejado")`
- ☐ `print(query)`

Pergunta 11

1,66 / 1,66 pts

Considere a atividade 9. Na linguagem Python, usando o pacote `mysql.connector`, é possível inserir múltiplas linhas em uma tabela com um só comando. Qual comando é esse?

- ☐ `mycursor.execute(query, values)`
- ☒ `mycursor.executemany(query, values)`
- ☐ `dbWriteTable(query, values)`
- ☐ `insert.many(query, values)`

Pergunta 12

1,66 / 1,66 pts

Considere as atividades 9 e 10. Marque a alternativa que **NÃO** serve para incluir dados em uma tabela (por exemplo a tabela “*caracteristicageralimovel*”) do seu banco de dados MySQL utilizando a linguagem Python.

☐

```
1 query = "INSERT INTO caracteristicageralimovel (idcaracteristicasGerais,idImovel,temCaracteristica) VALUES (%s, %s, %s)"
2 values = [(4,22,1),(4,23,1),(4,26,0)]
3
4 mycursor.executemany(query, values)
5
6 mydb.commit()
```

☐

```
1 filename = 'C:\Bootcamp\Datasets\TXT\caracteristicaImovel.txt'
2
3 with open(filename, "r", encoding='utf-8') as fileobject:
4     for line in fileobject:
5         query = "INSERT INTO caracteristicageralimovel (idcaracteristicasGerais,idImovel,temCaracteristica)\
6             VALUES (%s)" % line
7         mycursor.execute(query)
8
9 mydb.commit()
```

☐

```
1 query = "INSERT INTO caracteristicageralimovel (idcaracteristicasGerais,idImovel,temCaracteristica) VALUES (%s, %s, %s)"
2 values = (4,22,1)
3
4 mycursor.execute(query, values)
5
6 mydb.commit()
```

☒

```
1 query = "INSERT INTO caracteristicageralimovel (idcaracteristicasGerais,idImovel,temCaracteristica) VALUES (%s, %s, %s)"
2 values = (4,22,1)
3
4 mycursor.sendQuery(query, values)
5
6 mydb.commit()
```

Pergunta 13**1,66 / 1,66 pts**

Considerando as linguagens R e Python, marque a alternativa **correta**.



Para instalar um pacote chamado json no ambiente Python, pode ser usado o comando "install.packages("json")" dentro do script Python.



Ao contrário da linguagem Python, a linguagem R exige que o pacote seja previamente importado antes de ser utilizado.



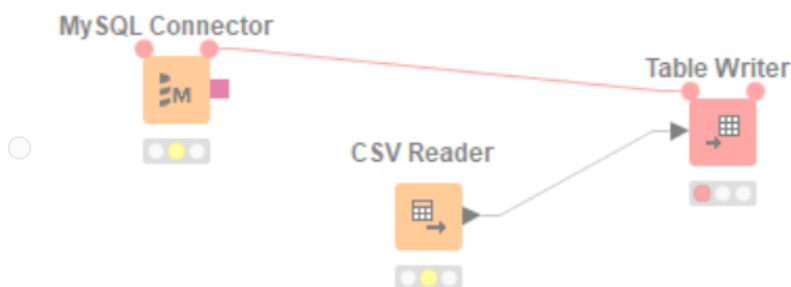
Tanto no Python quanto no R, uma vez que o pacote foi instalado, não é necessário importa-lo no seu script.

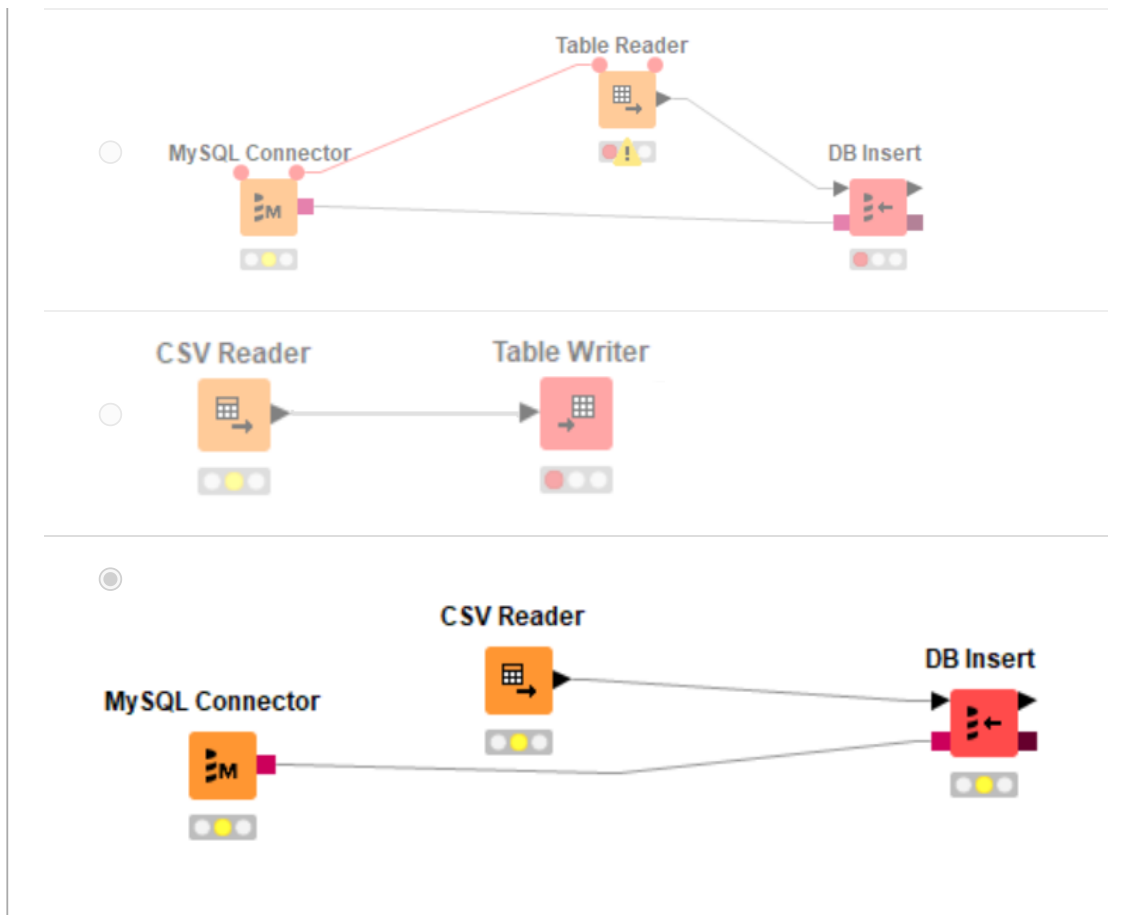


Antes de importar um pacote, tanto no Python quanto no R, este pacote deve ter sido previamente instalado.

Pergunta 14**1,66 / 1,66 pts**

Considere as atividades 12 e 13 sobre a plataforma Knime. Dos workflows apresentados abaixo, marque aquele que serve ao propósito de incluir dados existentes em um arquivo csv em uma tabela do banco de dados, por exemplo a tabela "imoveis".

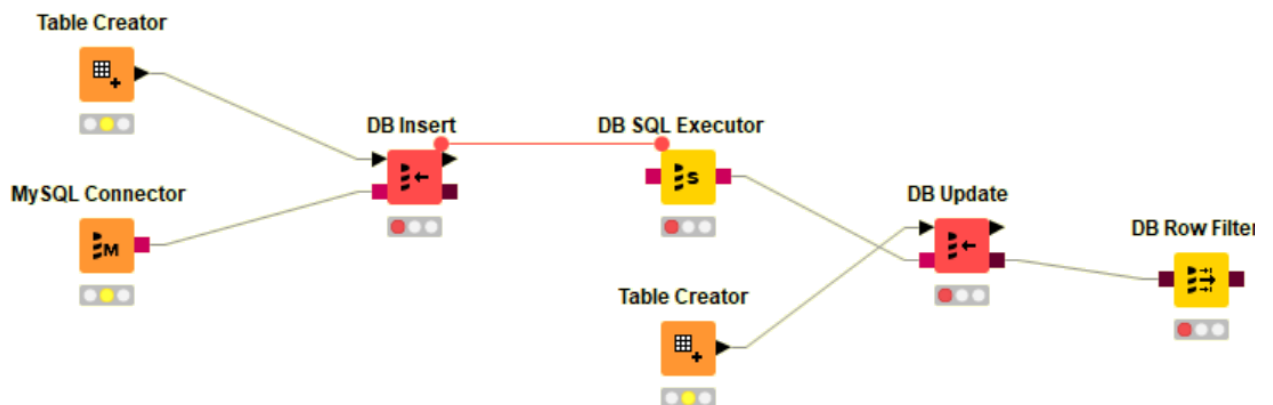




Pergunta 15

1,76 / 1,76 pts

Considere a atividade 11 e as do tópico “Coleta de dados estruturados: Exemplo utilizando a Plataforma Knime (parte 1 e 2)”. Marque a opção **INCORRETA** em relação a função de cada nó do workflow abaixo.



**DB Insert**

Pode ser usado para inserir dados em uma tabela do banco de dados.

**Table Creator**

Pode ser usado para criar uma tabela no banco de dados conectado.

**MySQL Connector**

Pode ser usado para realizar a conexão do workflow ao banco de dados MySQL.

**DB SQL Executor**

Pode ser usado para executar um comando SQL no banco de dados conectado.

Pontuação do teste: **25** de 25