#05 practical class

Docker exercises

COMPUTING SYSTEMS AND INFRASTRUCTURES

(SISTEMAS E INFRAESTRUTURAS DE COMPUTAÇÃO)

Docker exercise 1

- Create a RabbitMQ docker image with the MQTT plugin enabled
- Create a Python docker container with a custom script that generates random numbers from 0 to 100 every second and pushes the number to an MQTT queue
- Create a second Python docker container with a custom script that subscribes to the MQTT queue and prints the numbers received

Docker exercise 2

- Create a compose file to deploy the Docker exercise 1 services

Docker exercise 3

- Deploy one MariaDB container with a default table
- Deploy PHPMyAdmin
- Create a Python docker container with a custom script that connects to the MariaDB database and stores random numbers

Docker exercise 4

- Create a compose file to deploy the Docker exercise 2 services

Example of python MQTT connection (publish & subscribe)

```
import paho.mqtt.client as mqtt

def on_disconnect(client, userdata, rc):
    print("Disconnected with result code "+str(rc))

def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))

def on_publish(client, userdata, result):
    print("Message published: "+str(result))


if __name__ == '__main__':
    client= mqtt.Client("sic-pub")
    client.on_publish = on_publish
    client.on_connect = on_connect
    client.on_disconnect = on_disconnect

    client.connect("MQTT_SERVER",1883)

    client.publish("/sic", "MSG")
```

```
import paho.mqtt.client as mqtt

def on_disconnect(client, userdata, rc):
    print("Disconnected with result code "+str(rc))

def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("/sic/#")

def on_subscribe(client, userdata, mid, granted_qos):
    print("Subscribed")

def on_message(client, userdata, msg):
    print(msg.topic+" "+str(msg.payload))

if __name__ == '__main__':
    client= mqtt.Client("sic-sub")
    client.on_message = on_message
    client.on_subscribe = on_subscribe
    client.on_connect = on_connect
    client.on_disconnect = on_disconnect

    client.connect("MQTT_SERVER",1883)

    client.loop_forever()
```

Example of python MySQL connection

```
import mysql.connector
from mysql.connector import errorcode

if __name__ == '__main__':
    try:
        cnx = mysql.connector.connect(user='root',
password='my-secret-pw', host=MARIADB_SERVER',
database='sic')
        print("Connected to database")

        cursor = cnx.cursor()
    except mysql.connector.Error as err:
        print("Error connecting to database")
        exit(0)
```

```
try:
  TABLE = (
    "CREATE TABLE `sic` ("
    "  `id` int(11) NOT NULL AUTO_INCREMENT,"
    "  `date` TIMESTAMP NOT NULL,"
    "  `client` varchar(64) NOT NULL,"
    "  `value` int(11) NOT NULL,"
    "  PRIMARY KEY (`id`)"
    ") ENGINE=InnoDB")
  cursor.execute(TABLE)
except mysql.connector.Error as err:
  if err.errno == errorcode.ER_TABLE_EXISTS_ERROR:
    print("Table already exists")
  else:
    print("Error creating table")
```