



**UNIVERSIDADE FEDERAL DE RORAIMA CENTRO DE CIÊNCIA E
TECNOLOGIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
DCC 602 - SISTEMAS DISTRIBUÍDOS (2024.1)
Prof. LEANDRO N. BALICO**

**Nome: Fernando Souza Rodrigues 2019037493, Wandressa da Silva Reis
2020014698**

Relatório - Middleware com RabbitMQ

Implementação: Produtor que publica a temperatura da CPU em um tópico do RabbitMQ

1. Importações de bibliotecas:

- ``import pika``: Importa a biblioteca ``pika``, que permite a interação com o RabbitMQ para enviar e receber mensagens.
- ``import psutil``: Importa a biblioteca ``psutil``, que fornece uma interface para consultar informações do sistema, como a temperatura da CPU.

2. Definição de funções:

- ``obter_temperatura_cpu1()``: Esta função tenta abrir o arquivo que contém a temperatura da CPU no sistema de arquivos (``/sys/class/thermal/thermal_zone0/temp``). Se bem-sucedida, lê o valor, converte para Celsius e retorna. Caso contrário, imprime uma mensagem de erro e retorna ``None``.
- ``obter_temperatura_cpu2()``: Esta função utiliza a biblioteca ``psutil`` para obter a temperatura da CPU. Ela procura na lista de temperaturas por informações relacionadas à CPU (geralmente sob o nome ``coretemp``), e então busca pela temperatura do pacote (geralmente rotulada como ``Package id 0``). Se encontrar, retorna a temperatura atual; caso contrário, retorna ``None``.
- ``publicar_temperatura_cpu()``: Esta função é responsável por estabelecer uma conexão com o servidor RabbitMQ (localmente, no ``localhost``), criar um canal de comunicação, declarar uma fila chamada ``temperatura_cpu``, obter a temperatura da CPU chamando ``obter_temperatura_cpu1()``, publicar essa temperatura na fila e fechar a conexão.

3. Execução principal:

- Verifica se o script está sendo executado como o programa principal (``if __name__ == '__main__':``).
- Chama a função ``publicar_temperatura_cpu()`` para iniciar o processo de publicação da temperatura da CPU.

Implementação: Consumidor recebe a temperatura da CPU do RabbitMQ e verifica se ela está acima de um determinado limite (por exemplo, 70 graus Celsius). Caso a temperatura esteja acima do limite, o consumidor deve publicar uma mensagem indicando a detecção de um possível incêndio.

1. Importações de bibliotecas:

- ``import pika``: Importa a biblioteca ``pika`` para interagir com o RabbitMQ.

2. Definição de função de callback:

- ``callback_temperatura(body)``: Esta função é chamada quando uma nova mensagem é recebida na fila `'temperatura_cpu'`. Ela converte o corpo da mensagem (que deve conter a temperatura da CPU) em um número de ponto flutuante e verifica se é maior que 70 graus Celsius. Se for, estabelece uma nova conexão com o RabbitMQ, declara uma fila `'detecao_incendio'` e publica uma mensagem indicando a detecção de um incêndio. Por fim, fecha a conexão.

3. Função para consumir temperatura da CPU:

- ``consumir_temperatura_cpu()``: Esta função estabelece uma conexão com o RabbitMQ, cria um canal de comunicação e declara uma fila `'temperatura_cpu'` para consumir mensagens. Em seguida, configura um callback para processar as mensagens recebidas e inicia o consumo de mensagens.

4. Execução principal:

- Verifica se o script está sendo executado como o programa principal.
- Chama a função ``consumir_temperatura_cpu()`` para começar a consumir mensagens da fila `'temperatura_cpu'`.

Implementação: Alarme que complementa o sistema, adicionando a funcionalidade de detectar e reagir a mensagens que indicam a detecção de um incêndio.

1. Importações de bibliotecas:

- ``import pika``: Como antes, essa biblioteca é usada para interagir com o RabbitMQ.
- ``import os``: Importa o módulo ``os``, que fornece uma maneira de interagir com o sistema operacional, neste caso para disparar um comando para o sistema.

2. Definição de função de callback:

- ``callback_incendio()``: Esta função é chamada quando uma nova mensagem é recebida na fila `'detecao_incendio'`. Ela imprime uma mensagem indicando que um incêndio foi detectado e dispara um alarme sonoro utilizando o comando ``spd-say`` do sistema operacional. Em seguida, estabelece uma conexão com o RabbitMQ,

declara uma fila 'ativacao_prevencao_incendio' e publica uma mensagem indicando que o sistema de prevenção de incêndio deve ser ativado.

3. Função para consumir detecção de incêndio:

- ``consumir_detecao_incendio()``: Esta função é responsável por estabelecer uma conexão com o RabbitMQ, criar um canal de comunicação e declarar uma fila 'detecao_incendio' para consumir mensagens indicando a detecção de um incêndio. Em seguida, configura um callback para processar as mensagens recebidas e inicia o consumo de mensagens.

4. Execução principal:

- Verifica se o script está sendo executado como o programa principal.
- Chama a função ``consumir_detecao_incendio()`` para começar a consumir mensagens da fila 'detecao_incendio' e reagir quando um incêndio for detectado.