



EGM0017 (60H)

# **FLUXO E METODOLOGIAS DE PROJETO DE SISTEMAS EMBARCADOS**

PROF. JOSENALDE OLIVEIRA

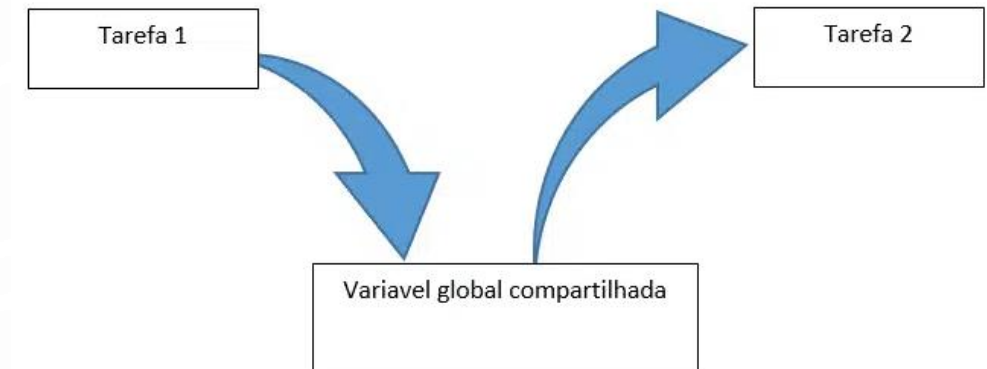
[josenalde.oliveira@ufrn.br](mailto:josenalde.oliveira@ufrn.br)

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - UFRN

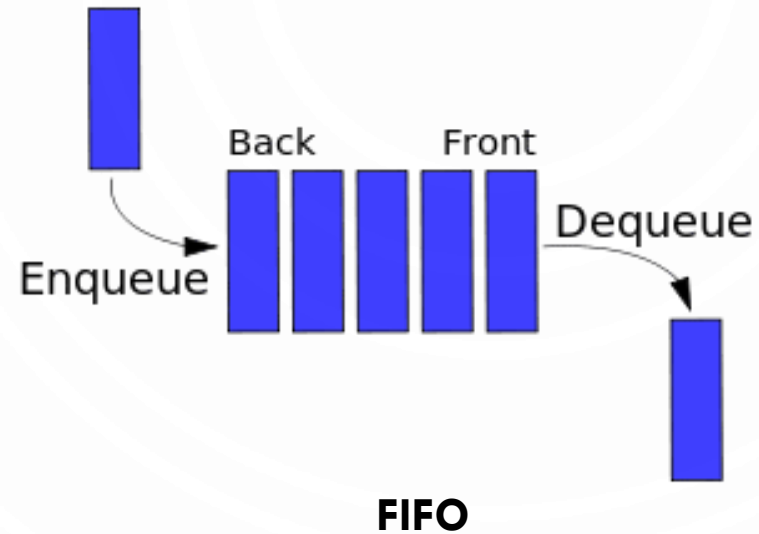
# FreeRTOS – trabalhando com FILAS (queue)

- Sincronização e comunicação entre tarefas e ISRs (em geral, envio de variáveis entre tasks)
- Queue é um BUFFER, uma fila de dados no formato **FIFO** (first in first out)
- Reduz necessidade de variáveis globais.

Mecanismos de comunicação entre tarefas denominados de *thread safe*. As tarefas envolvidas em comunicação *thread safe* poderão ser interrompidas ou executadas a qualquer momento sem que isso gere um estado inconsistente nos dados que são transferidos entre estas tarefas. (semáforos, mutexes, **filas**)

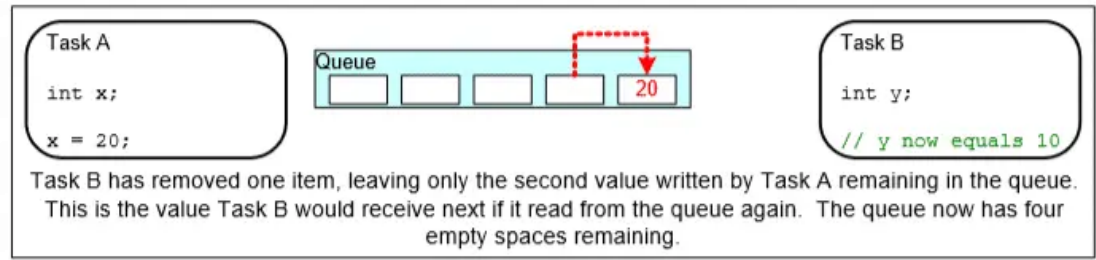
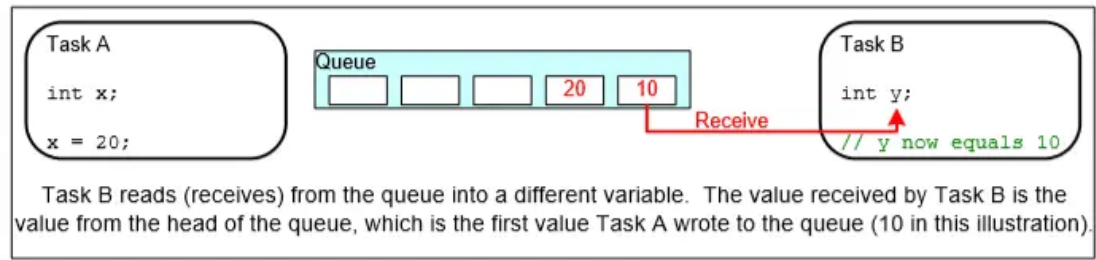
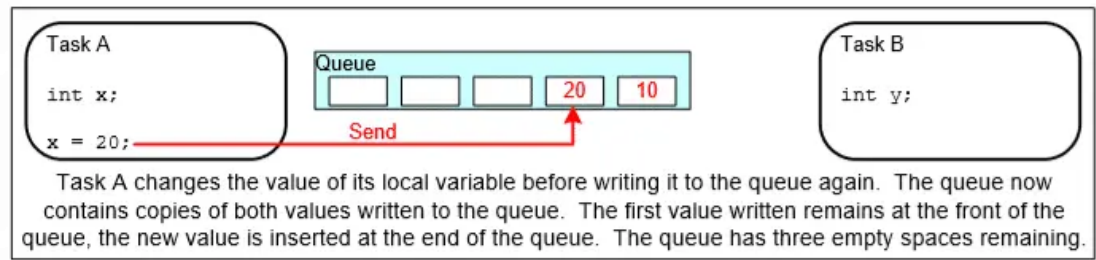
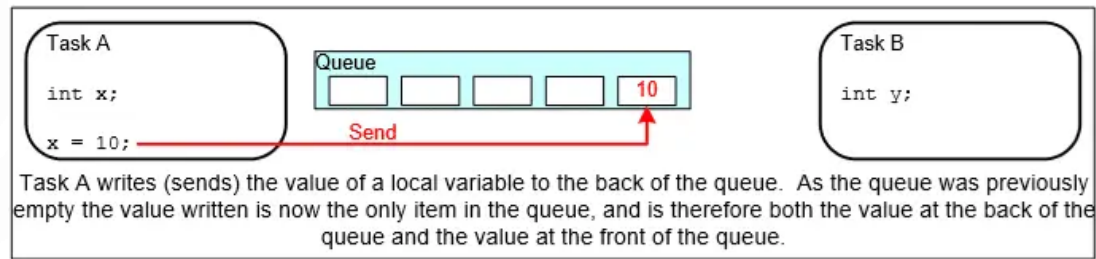
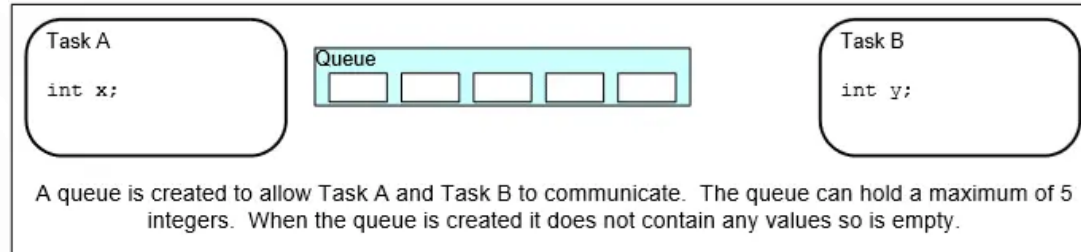


# FreeRTOS – trabalhando com FILAS (queue)



<https://embarcados.com.br/rtos-queue-sincronizacao-e-comunicacao/>

# FreeRTOS – trabalhando com FILAS (queue)



An example sequence of writes to, and reads from a queue

# FreeRTOS – trabalhando com FILAS (queue)

Nome da função	Finalidade	Observação
xQueueCreate	Cria uma fila nova	Antes de realizar qualquer operação com filas, esta função deve ser executada para criar a fila.
vQueueDelete	Apaga uma fila e libera toda a memória alocada para a fila	Em sistemas com pouca memória sempre é importante liberar memória não utilizada
xQueueSend	Enfileira um elemento no final da fila.	Esta função NÃO deve ser usada dentro de uma rotina de tratamento de interrupção.
xQueueSendFromISR	Enfileira um elemento no final da fila.	Esta função pode ser usada dentro de uma rotina de tratamento de interrupção.
xQueueReceive	Remove um elemento do início da uma fila.	Esta função NÃO deve ser usada dentro de uma rotina de tratamento de interrupção.
xQueueReceiveFromISR	Remove um elemento do início de uma fila.	Esta função pode ser usada dentro de uma rotina de tratamento de interrupção.

# FreeRTOS – trabalhando com FILAS (queue)

- Filas de tamanho fixo e tamanho definido por slot (bytes). Em geral usa-se `sizeof(...)`
- Exemplo, passagem de variáveis entre tasks

```
QueueHandle_t xQueueCreate( UBaseType_t uxQueueLength, UBaseType_t uxItemSize );
```

Se a fila for criada, a função retornará o identificador da fila. Se a fila não for criada, retornará 0.

```
BaseType_t xQueueSend( QueueHandle_t xQueue, const void * pvItemToQueue,  
TickType_t xTicksToWait ); // tempo de bloqueio até fila estar com slot  
disponível – se setado em 0, retorno imediato se a fila está cheia
```

Equivalente à `xQueueSendToBack()`. Não pode ser usado dentro de ISR

Se enviou retorna `pdTRUE`, senão `errQUEUE_FULL`

```
BaseType_t xQueueReceive( QueueHandle_t xQueue, void *pvBuffer, TickType_t  
xTicksToWait );
```

Se enviou retorna `pdTRUE`, senão `pdFALSE`. Recebe e remove da fila. Se quiser manter usar `xQueuePeek`

# FreeRTOS – trabalhando com FILAS (queue)

- Exemplo básico – framework Arduino  
`QueueHandle_t queue;`

```
void setup() {  
  
    Serial.begin(115200);  
    vTaskDelay(pd_MS_TO_TICKS(1000));  
  
    queue = xQueueCreate(10, sizeof(int));  
  
    if(queue == NULL){  
        Serial.println("Error creating the queue");  
        vTaskDelete(NULL); //não executará loop()  
    }  
}
```

```
void loop() {  
  
    for(int i = 0; i<10; i++){  
        xQueueSend(queue, &i, portMAX_DELAY);  
    }  
  
    int element;  
  
    for(int i = 0; i < 10; i++){  
        xQueueReceive(queue, &element, portMAX_DELAY);  
        Serial.print(element);  
        Serial.print("|");  
    }  
  
    Serial.println();  
    vTaskDelay(pd_MS_TO_TICKS(1000));  
}
```

## FreeRTOS – Exemplos([queue](#))

### FreeRTOS – produtor consumidor com filas

[https://github.com/josenalde/flux-embedded-design/blob/main/src/producer\\_consumer.ino](https://github.com/josenalde/flux-embedded-design/blob/main/src/producer_consumer.ino)

### FreeRTOS – filas para comunicação task - ISR

<https://blog.eletrogate.com/freertos-filas-trocando-informacao-entre-tarefas/>

### Interrupção e filas – esp-idf

<https://github.com/josenalde/intr-queue-espidf>

### Cálculo de média móvel

[https://github.com/josenalde/flux-embedded-design/blob/main/src/average\\_temperature\\_queue.c](https://github.com/josenalde/flux-embedded-design/blob/main/src/average_temperature_queue.c)

### Exemplo de aplicação multicore/task em automação

<https://blog.eletrogate.com/rtos-com-esp32-como-programar-multitarefas/>

### Exemplo de aplicação multicore/task IoT com MQTT

<https://www.makerhero.com/blog/faca-seu-rastreador-veicular-com-esp32-gps-e-freertos/>