

TRABAJO FINAL – RTOS 2 – CESE
GRUPO 1 v18-06-2019

TP2

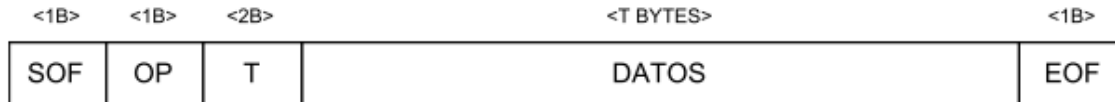
INTEGRANTES:

Julian Bustamante Narvaez
Jacobó Salvador
Gustavo Paredes D.
Rafael Oliva

Estructura TP2 / Grupo 1 RTOS2

Estructura y diagrama conceptual

Paquetes de datos: (en negrita los campos diferentes de la práctica 1)



Todos los caracteres son ASCII legibles.

Delimitación de paquete:

SOF: Carácter '{'

EOF: Carácter '}'.

Campos:

OP (Operación):

"0": Convertir los caracteres recibidos a mayúsculas. (CMD/RTA)

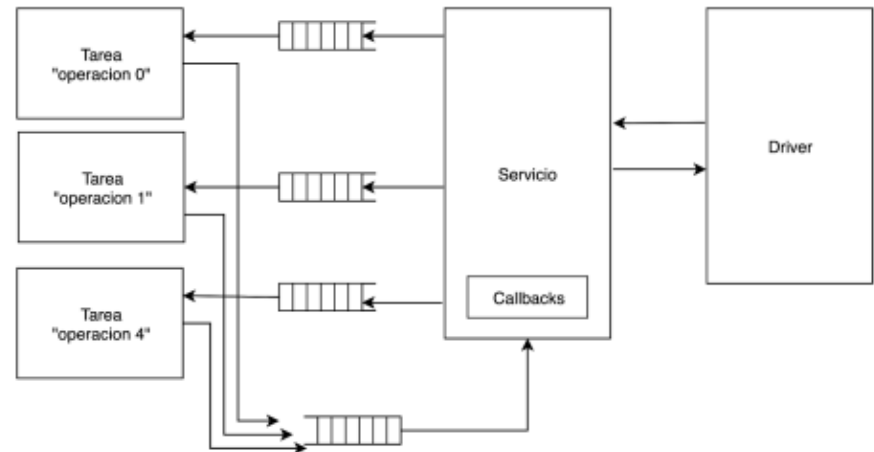
"1": Convertir los caracteres recibidos a minúsculas. (CMD/RTA)

"2": Reportar stack disponible (RTA)

"3": Reportar heap disponible. (RTA)

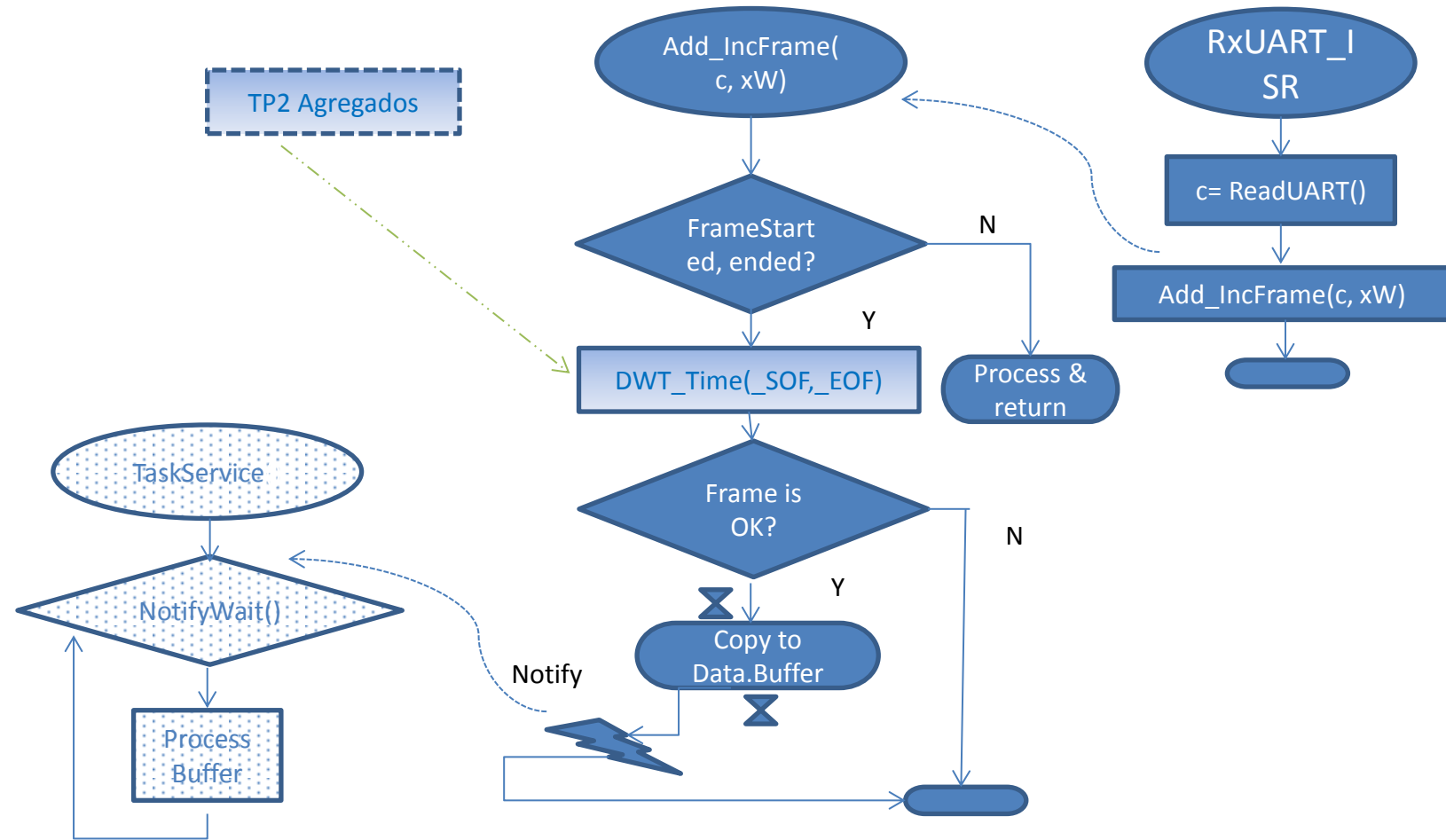
"4": Medir performance del sistema (CMD/RTA)

"5": Resultado de la medición performance del sistema (RTA)



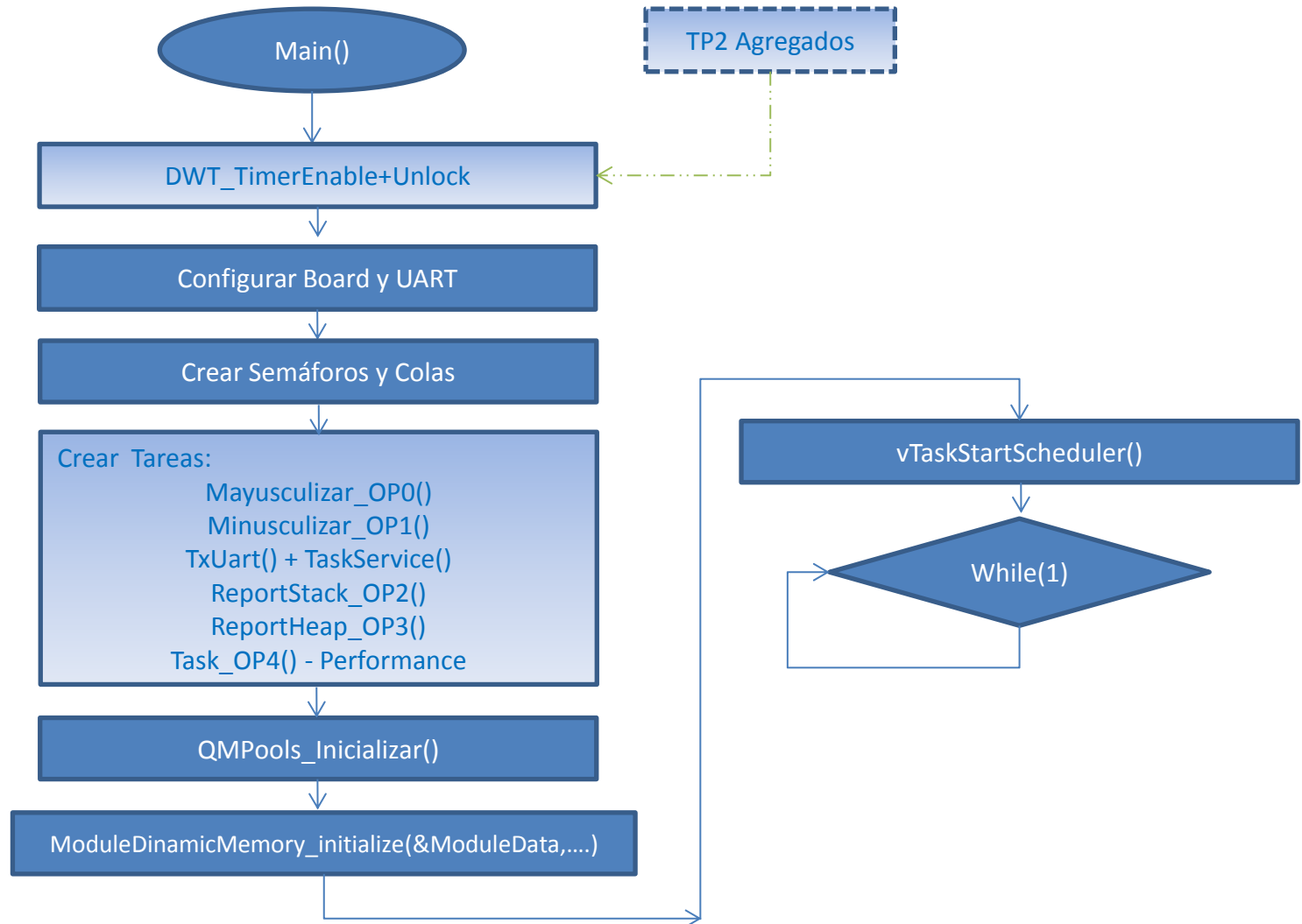
Estructura actual TP2 / Grupo 1 RTOS2

1.a) Ingreso caracteres dde UART, Add Time _SOF,_EOF v15.6.19



Estructura actual TP2 Main() / Grupo 1 RTOS2

1.b) TP2 Función main() Timer y Tsk_OP4 de Performance



Estructura actual Main() TP2 / Grupo 1 RTOS2

1.b.1) Función main() TP2

```
int main(void){

    boardConfig();

    QmPoolOrMalloc = eUseMalloc ;//eUseQMPool;

    // Timer para task_Medir_Performance
    * _DMCR = * _DMCR | 0x01000000;    // enable trace
    * _LAR = 0xC5ACCE55;                // <-- added unlock access to DWT (ITM, etc.)registers
    * _DWT_CTRL |= 1;

    /*=====Config Uart=====*/
    uartConfig(UART_USB, 115200);
    /*Callback interrupt*/
    uartCallbackSet(UART_USB, UART_RECEIVE, CallbackRx, NULL);
    /*Habilito todas las interrupciones de UART_USB*/
    uartInterrupt(UART_USB, true);

    semaphoreCreateAll();
    QueueCreateAll();
    TaskCreateAll();
    QMPools_inicializar();

    /*Inicializar Driver memoria dinamica*/
    ModuleDinamicMemory_initialize(&ModuleData,50,xQueueGenericSend,xQueueGenericSendFromISR,

    /* Iniciar scheduler*/
    vTaskStartScheduler();

    while( TRUE ) {
    }

    return 0;
}
```

Estructura de datos TP2 / Grupo 1 RTOS2

1.b.2) Versión TP2 v15.6.19

TP2 – Datos Performance

```
typedef struct {  
    State_perf State_Token;  
    uint32_t Id_de_paquete;  
    char * Payload;  
    uint32_t t_sof;  
    uint32_t t_eof;  
    uint32_t t_InitConvert;  
    uint32_t t_EndConvert;  
    uint32_t t_InitTx;  
    uint32_t t_EndTx;  
    uint16_t Length_Frame;  
    uint16_t Memory_Allocated;  
    void (*Completion_HandlerFCN)( void *T, BaseType_t * xHig );  
} Token_t;
```

```
/* Medir Performance */  
typedef enum State_med {  
    Time_LL = 0,  
    Time_R,  
    Time_I,  
    Time_F,  
    Time_S,  
    Time_T  
} State_perf;  
  
static uint32_t Id_Frame = 0;
```

```
/* ===== Datos para llenar buffer local ISR===== */  
typedef struct {  
    char Buffer[106];  
    uint32_t t_sof;  
    uint32_t t_eof;  
    uint32_t Id_Frame;  
    uint8_t Ready;  
    uint8_t Index;  
    uint8_t StartFrame;  
} DataFrame_t;  
extern volatile DataFrame_t Data;
```

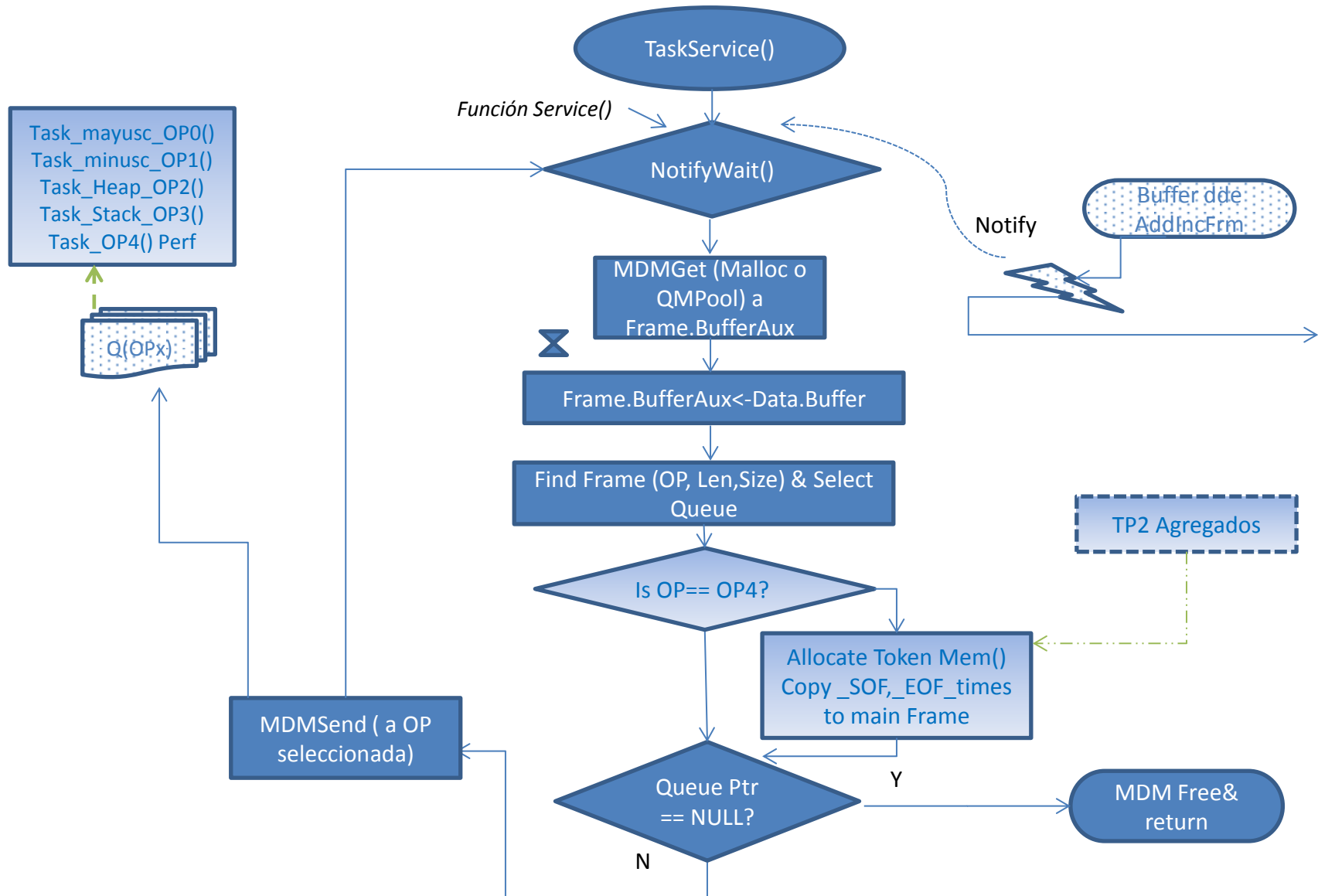
```
/* =====Parametros de la trama de llegada ===== */  
typedef struct {  
    Enum_Op_t Operation;  
    uint8_t T;  
    char* BufferAux;  
    Token_t *Token;  
} Frame_parameters_t;
```

TP2 – trama usada por ISR

TP2 – Nuevo frame c/Token procesamiento

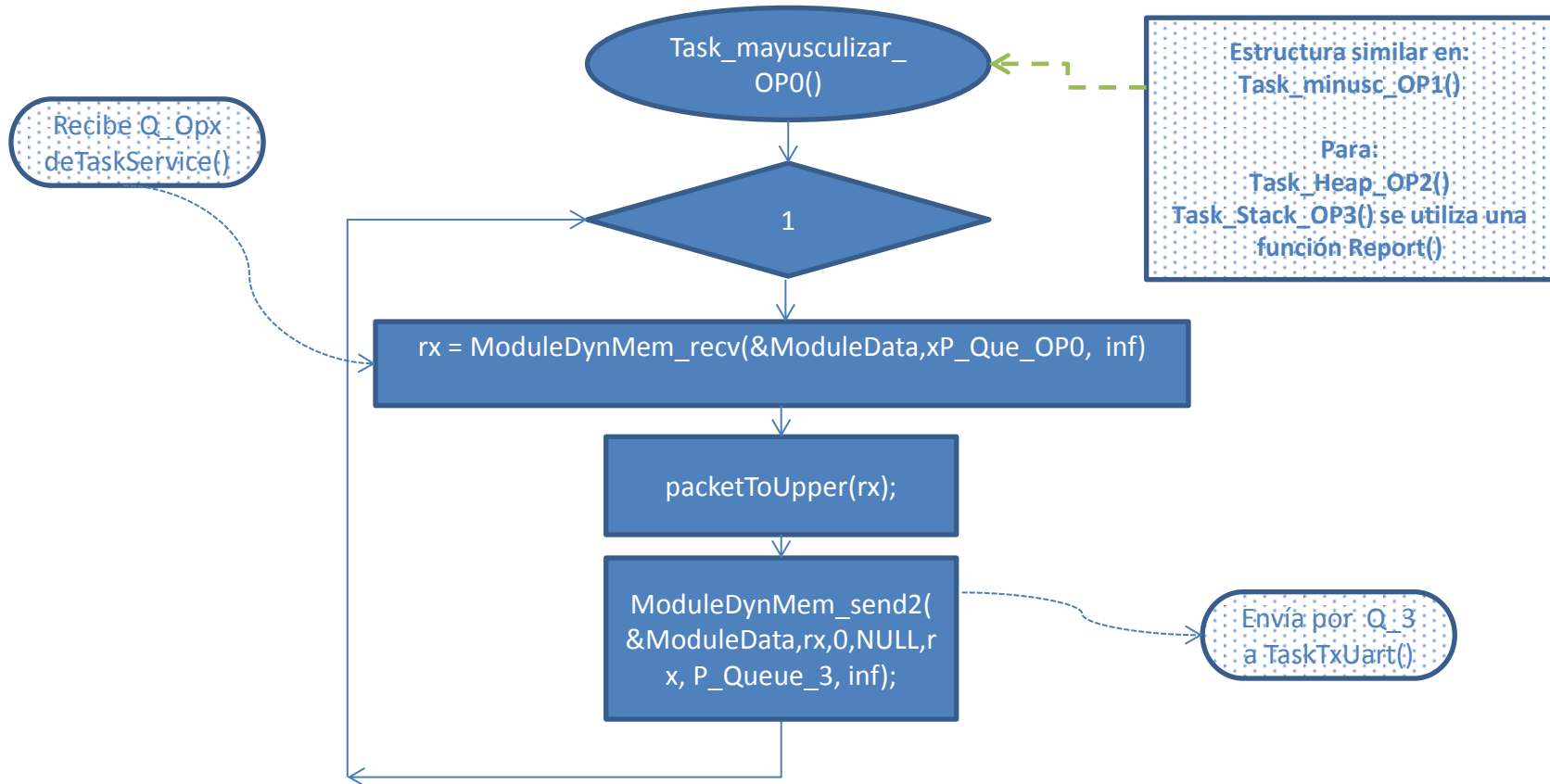
Estructura actual TP2 / Grupo 1 RTOS2

2) Proceso en TaskService v15.6.19



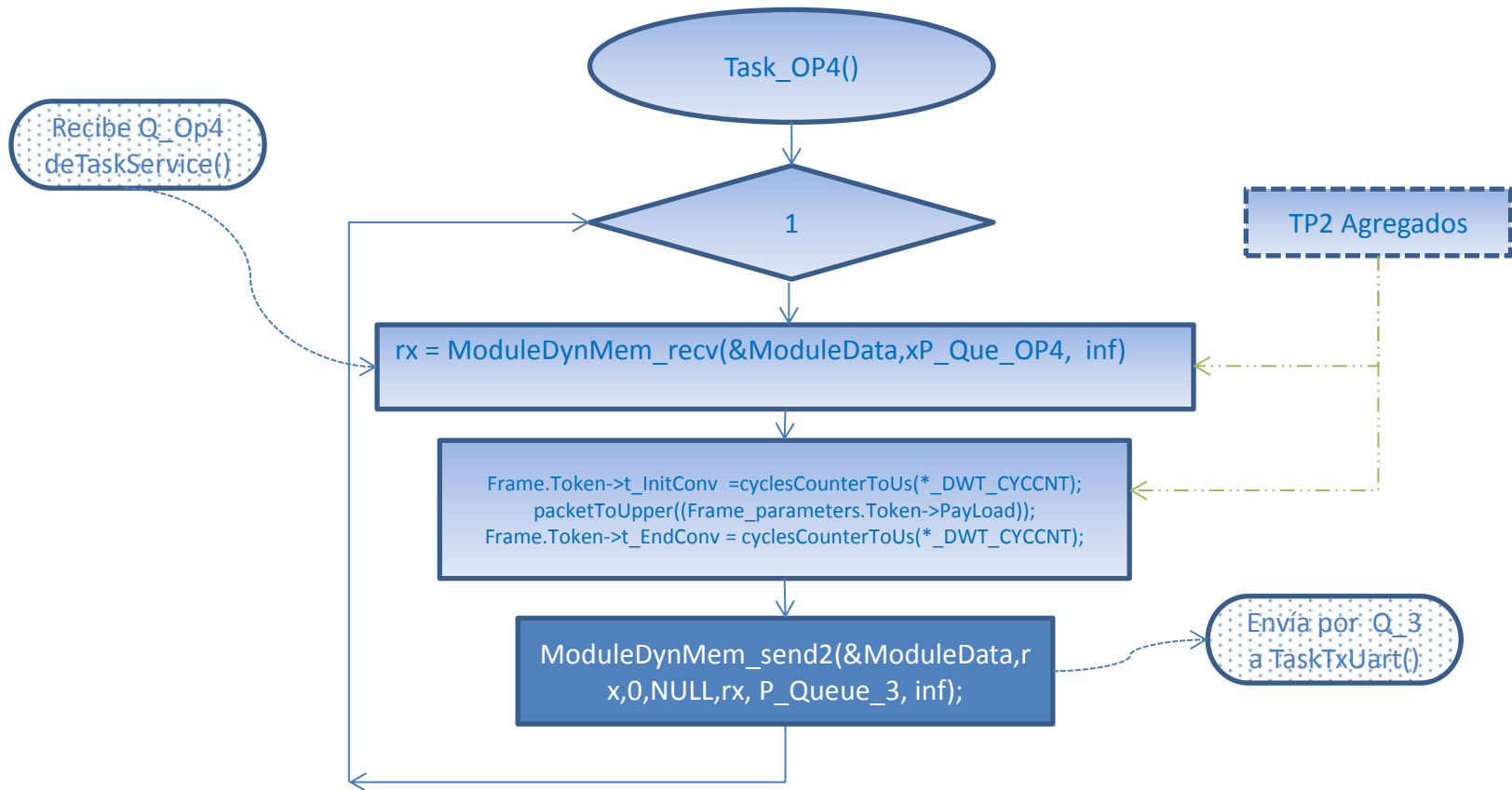
Estructura actual TP2 / Grupo 1 RTOS2

3) Formato de Tasks Mayusculizar, Minusc, Stack y Heap – Idem TP1



Estructura actual TP2 / Grupo 1 RTOS2

4) Formato de Task OP4 _ Mayusculizar con Medición performance



Estructura actual TP2 / Grupo 1 RTOS2

5) Salida de Tasks Mayusculizar, Minusc, Stack y Heap + especial para OP4

