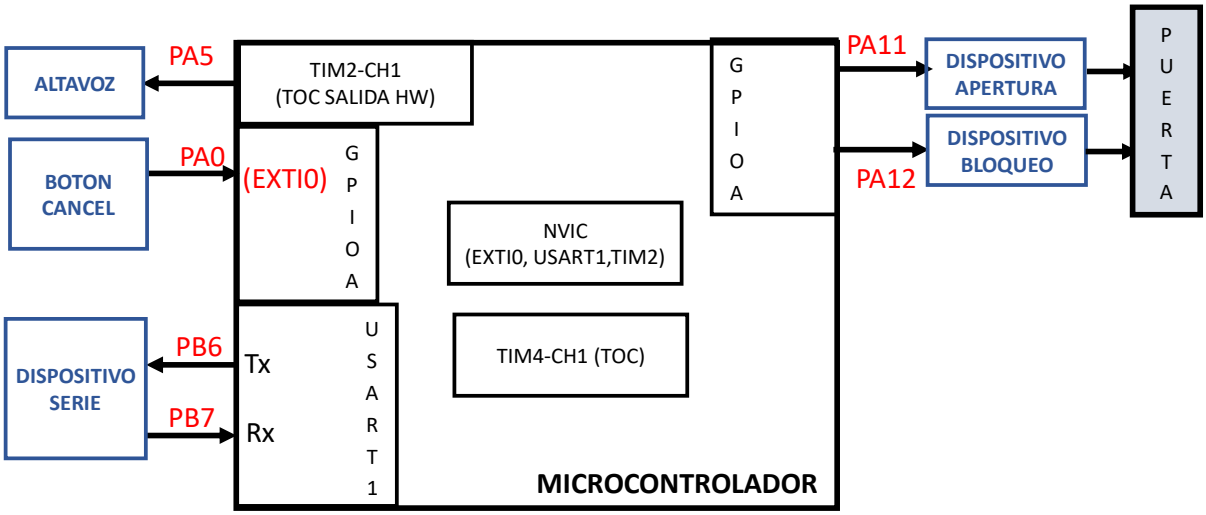
Tema 12: Diseño de Soluciones

SOLUCIÓN DE EJERCICIOS PROPUESTOS

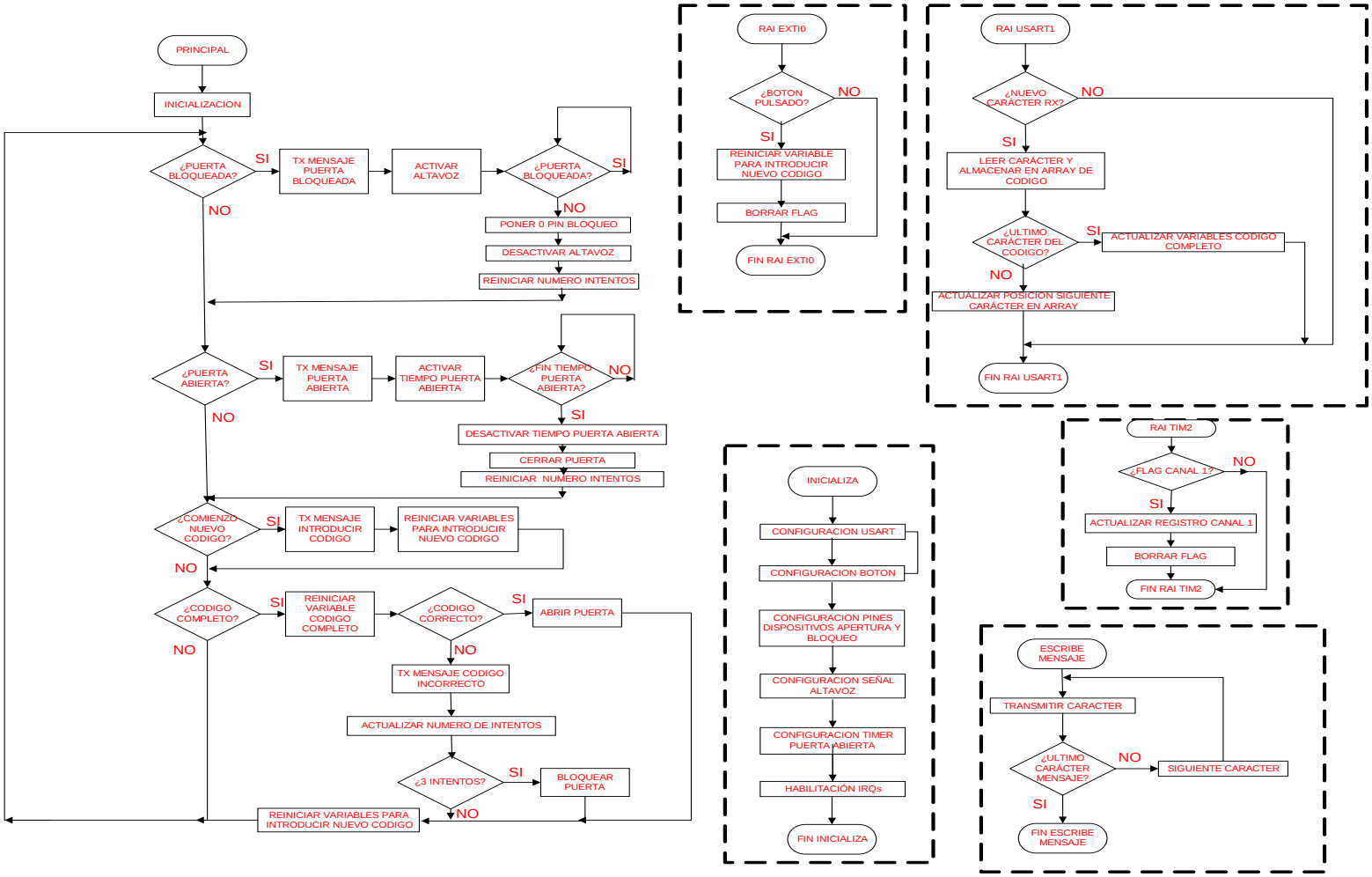
Ejercicio 1

1. Represente el esquema del hardware



PIN MICRO	HW CONECTADO	TIPO (E/S, DIGITAL/ ANALOGICO)	JUSTIFICACIÓN
PB6, PB7	DISPOSITIVO SERIE	TX/RX USART1	LÍNEAS 37-44 DEL CÓDIGO
PA0	BOTON CANCEL	ENTRADA DIGITAL (EXTI0)	LINEAS 46-51 DEL CODIGO
PA5	ALTAVOZ	SALIDA HW CH1-TIM2 MODO TOC	LINEAS 61-76 DEL CODIGO
PA11	DISPOSITIVO APERTURA	SALIDA DIGITAL	LINEAS 57-59 DEL CODIGO
PA12	DISPOSITIVO BLOQUEO	SALIDA DIGITAL	LINEAS 53-55 DEL CODIGO
ADEMÁS SE UTILIZA: - CH1-TIM4 EN MODO TOC SIN SALIDA HARDWARE (LINEAS 78-88 DEL CODIGO) - NVIC PARA EXTI0 E IRQ DE USART1 Y TIM2 (LINEAS 91-93 DEL CODIGO)			

2. Represente el diagrama de flujo del programa principal, función/es y rutina/s de atención a interrupción del código. **(30%)**



3. ¿Con qué recurso del microcontrolador se controla el tiempo que la puerta de acceso permanece abierta, y cuánto tiempo permanece la puerta abierta después de introducir el código de entrada correcto? Justifique su respuesta incluyendo todos los cálculos necesarios para obtenerla. **(15%)**

El tiempo que la puerta permanece abierta se controla con el canal 1 del TIM4 en modo TOC sin salida hardware. Este temporizador se configura en las líneas 78-88 del código. En esta configuración:

$$T_u = paso = T_{CLK} \cdot PSC = \frac{1}{32MHz} \cdot 32000 = 1ms$$

Como CCR1=10000 pasos, el temporizador (cuya cuenta se reinicia en el código cada vez que se abre puerta) cuenta 10s ⇒ Una vez introducido el código correcto la puerta permanece abierta 10s.

4. Indique la configuración completa de la comunicación serie utilizada para la introducción del código de acceso. Indique la/s modificación/es necesarias en el código para que los parámetros de esta comunicación serie pasen a ser 57600, 8, N, 1. **(15%)**

La conexión serie se configura en las líneas 41-43.

USART1->CR1 = 0x0000002C ⇒ Habilitación Tx, Rx; Rx por IRQ. Modo sobremuestro por 16 (OVER8=0), 8 bits de datos, sin bit de paridad
USART1->CR2 = 0x00000000 ⇒ 1 bit de parada
USART1->BRR = 0x00000115 ⇒ Div_Mantissa=0x11=17, Div_fraction=0x5=5 ⇒USARTDIV=17,5

$$BAUDRATE\,(bps) = \frac{f_{CLK}}{8x(2-OVER8)xUSARTDIV} = \frac{32MHz}{16x17{,}5} \cong 114{,}3kbps(NORMALIZADO: 115{,}2kbps)$$

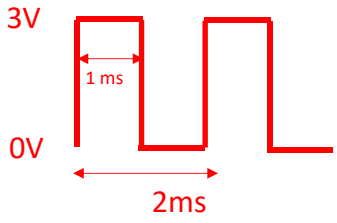
Por lo tanto, los parámetros de la comunicación serie configurada en el código son: 115200, 8, N, 1.

Para cambiar a la configuración solicitada (57600, 8, N, 1) manteniendo OVER8=0, solo habría que cambiar el registro BRR.

57600⇒USARTDIV=34,7⇒Div_Mantissa=34=0x22, Div_fraction=11=0xB
⇒ USART1->BRR = 0x0000022B

5. Represente dos períodos completos de la forma de onda de la señal que se aplica al altavoz (acotándola claramente en amplitud y tiempo). Indique la/s modificación/es necesarias en el código que esta señal sea una señal entre 0 y 3V con una frecuencia de 1Hz. **(15%).**

La señal que se aplica al altavoz se genera utilizando el canal 1 del TIM2 en modo TOC con salida HW toogle (configuración en líneas de código 61-76). En esta configuración: Tu=paso=1µs, Semiperíodo (registro CCR)=1000 pasos=1ms. Es una señal, por tanto de 500Hz.



Para cambiar la frecuencia a 1Hz, el semiperiodo debe ser 500ms, hay que cambiar:

- Preescala (con pasos de 1µs necesitaríamos 500000 pasos que no caben en un registro de 16bits). Cambiando por ejemplo la preescala a un valor para contar el mismo nº de pasos que en el código original en cada semiperiodo (1000 pasos) solo sería necesario modificar este registro: `TIM2->PSC= 16000;`

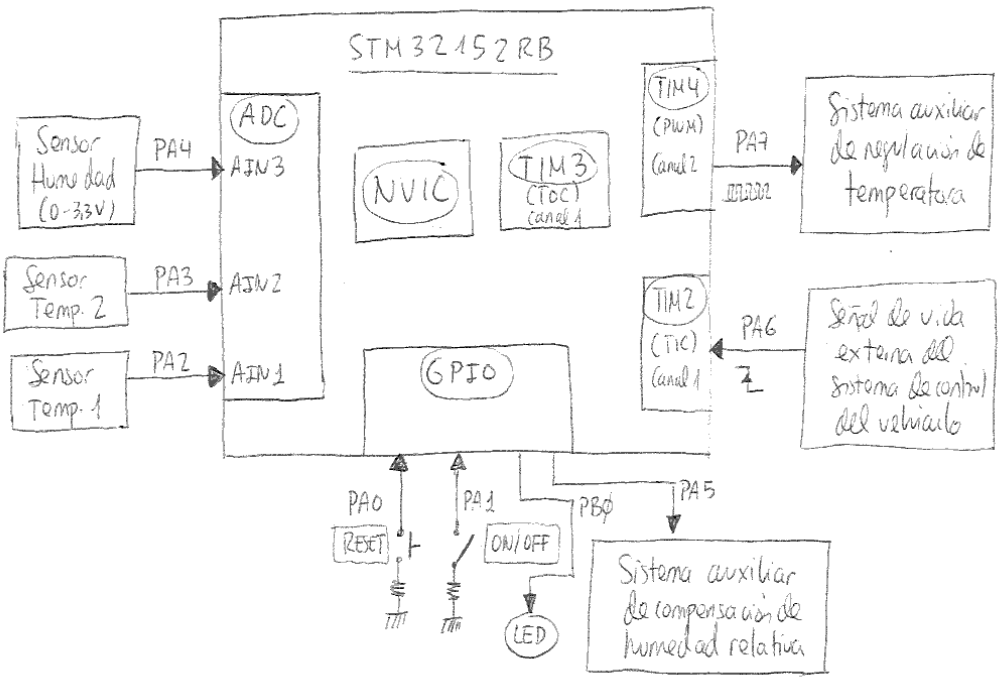
Ejercicio 2

a) Diagrama de bloques del sistema. (20%)

Un diagrama de bloques posible sería el siguiente. Consideraciones:

- Las señales de RESET y ON/OFF son simplemente señales de entrada digital del bloque GPIO
- Los 3 sensores analógicos necesarios para el sistema se conectan a 3 canales del bloque ADC
- El bloque NVIC se añade en diagrama de bloques porque se supone que alguna hará falta en el apartado b).
- Hay que contar segundos, generar una señal PWM y contar la señal de vida en modo TIC. Como se dice que no se pueden mezclar modos en los canales de cada temporizador, es necesario usar 3 diferentes. El canal 1 del TIM2 para contar los 10 segundos de la señal de vida en modo TIC, el canal 1 del TIM3 se utiliza para contar 5 segundos y un minuto en modo TOC y el canal 2 del TIM4 se utiliza para generar la señal PWM necesaria para el sistema auxiliar de control de temperatura, pero se pueden usar otros canales.

Con estas consideraciones y teniendo en cuenta que se pueden variar algunos de los pines del microcontrolador, no tienen por qué ser exactamente los propuestos, un posible diagrama de bloques sería el siguiente.



b) Indique si utilizará interrupciones, y en caso afirmativo, cuáles y para qué. (20%)

Los periféricos utilizados permiten la utilización de interrupciones, pero:

- La consulta de la señal de vida es cada 10 segundos y habría que usar el modo TIC del TIM 2 para hacerlo, que preferentemente exige el uso de interrupciones en el canal 1 del TIM 2.

- Los 5 segundos y el minuto de los sensores analógicos no son críticos en el tiempo pero son múltiplos entre sí por lo que con un sólo canal del TIM 3 sería suficiente para medir esos tiempos usando interrupciones. La interrupción debería saltar cada 5 segundos para trabajar con el sensor de temperatura y cada 12 saltos también trabajar con sensor de humedad relativa al detectar el minuto transcurrido.
- Para la señal PWM no sería necesario utilizar interrupciones en el TIM 4 porque la frecuencia es fija de 100Hz y el DC se puede cambiar en el programa principal en función de la señal media de las medidas de los sensores analógicos.

- El pulsador de RESET del PA0 hay que programarlo obligatoriamente por interrupción externa EXTI porque nunca se sabe cuándo podría aparecer y además se dice que debe actuar inmediatamente.
- Para el botón ON/OFF del PA1 no se dice nada de inmediatez y se puede elegir hacerlo por interrupciones o no. En la solución propuesta se hace por una entrada digital normal sin interrupciones.

c) Indique qué periféricos utiliza y cómo los configuraría. No escriba el código, sino el valor que tendría que tener cada uno de los bits afectados de cada uno de los registros del periférico y por qué. (40%)

Los distintos periféricos con su configuración son los siguientes:

- **GPIO:** Los pines PA0 (para el botón de RESET) y PA1 (para el botón de ON/OFF) como entradas digitales, y PA5 (para la salida digital del sistema auxiliar de compensación de humedad relativa) como salida digital. Los pines PA2 y PA3 como entradas analógicas de los canales 1 y 2 (para conectar los dos sensores de humedad de 0-3,3V) y el pin PA4 como entrada analógica del canal 3 (para conectar el sensor de humedad relativa de 0-3,3V). El pin PA6 se configura como función especial para la señal TIC asociada al TIM2.El pin PA7 se configura como función especial para la señal PWM asociada al TIM4.

```
GPIOA->MODER &= ~(1 << (0*2 +1));       // PA0 como entrada digital (00)
GPIOA->MODER &= ~(1 << (0*2 +1));
GPIOA->MODER &= ~(1 << (1*2 +1));       // PA1 como entrada digital (00)
GPIOA->MODER &= ~(1 << (1*2 +1));
GPIOA->MODER |= 0x000003F0;            // PA2, 3 y 4 como señales analógicas (11)
GPIOA->MODER &= ~(1 << (5*2 +1));      // PA5 como salida digital (01))
GPIOA->MODER |= (1 << (5*2));
GPIOA->MODER|=0x00000001 << (2*6 +1); // PA6 como función especial (10)
GPIOA->MODER&=~(0x00000001 << (2*6));
GPIOA->AFR[0]|=(0x02 << (6*4));        // AFR[0] = 0x00010000000000 para que
                                       // el PA6 tenga la función especial 0001 = TIM2
GPIOA->MODER|=0x00000001 << (2*7 +1); // PA7 como función especial (10)
GPIOA->MODER&=~(0x00000001 << (2*7));
```

```
GPIOA->AFR[0]|=(0x02 << (7*4));        // AFR[0] = 0x02000000000000 para que
                                       // el PA7 tenga la función especial 0010 = TIM4
GPIOB->MODER &= ~(1 << (0*2 +1));      // PB0 como salida digital
GPIOB->MODER |= (1 << (0*2));
```

- **ADC:** Por sencillez se configuran para 12 bits, modo continuo, modo scan y sin interrupciones

```
ADC1->CR2 &= ~(0x00000001);     // ADON = 0 (ADC apagado)
ADC1->CR1 = 0x00000100;         // Configuración del registro CR1
                                // OVRIE = 0 (sin la habilitación por interrupción)
                                // RES = 00 (resolución = 12 bits)
                                // SCAN = 1 (modo scan para leer los 3 canales
                                // de manera secuencial)
                                 // EOCIE = 0 (deshabilitada la interupción por EOC)
                                 // OVRIE = 0 (deshabilitada la habilitación por
                                // interrupción
ADC1->CR2 = 0x00000412;         // EOCS = 1 (activado bit EOC al acabar conversión)
                                 // DELS = 000 (sin retardo de la conversión)
                                // CONT = 1 (conversión continua)
ADC1->SMPR1 = 0;                 // Sin sampling time (4 cycles)
ADC1->SMPR2 = 0;
ADC1->SMPR3 = 0;
ADC1->SQR1 = 0x00020000;         // 3 elementos  en la secuencia (0010 en los bits 23-20)
ADC1->SQR5 = 0x00000C41;         // Los elementos son los canales AIN1, AIN2 y AIN3 por
                                // este orden (0-00011-00010-00001)
ADC1->CR2 |= 0x00000001;         // ADON = 1 (ADC activado)
```

- **TIMER 2**: En modo TIC para contar lo 10 segundos de la señal de vida con el PA6 asociado a esa señal en su canal 1.. PCS = 32000 para contar 1000 pasos por segundo. En la INT se comprueban los 10 segundos de la señal de vida

```
TIM2->CR1 = 0x0000;             // ARPE = 0 -> No es PWM, es TIC; CEN = 0; Contador
                                 // apagado
TIM2->CR2 = 0x0000;             // CCyIE = 0 -> No se provoca interrupción con el TIMER2
TIM2->SMCR = 0x0000;             // Siempre "0" en este curso
TIM2->PSC = 32000;              // Preescalado=32000 -> F_contador=32000000/32000 = 1000
                                 // pasos/seg
TIM2->CNT = 0;                  // Inicializo el valor del contador a cero
TIM2->ARR = 0xFFFF;             // Valor recomendado si no es PWM
TIM2->DIER = 0x0002;            // Se genera INT al terminar de contar -> CCyIE = 1
TIM2->CCMR1 = 0x0001;            // CCyS = 1 (TIC); OCyM = 000 y OCyPE = 0 (siempre en
                                 // TIC)
TIM2->CCER = 0x0001;            // CCyNP:CCyP = 00 (activo a flanco de subida)
                                // CCyE = 1 (captura habilitada para TIC)
TIM2->EGR |= 0x0001;            // UG = 1 -> Se genera evento de actualización
TIM2->SR = 0;                   // Limpio los flags del contador
TIM2->CR1 |= 0x0001;            // CEN = 1 -> Arranco el contador
```

- **TIMER 3:** En modo TOC para contar 5 segundos y un minuto con interrupciones y sin HW asociado. PCS = 32000 para contar 1000 pasos por segundo y CCR1 = 5000, para que la interrupción salte cada 5 segundos. Luego en la INT se comprueban los 5 segundos y el minuto

```
TIM3->CR1 = 0x0000;        // ARPE = 0 -> No es PWM, es TOC
                           // CEN = 0; Contador apagado
TIM3->CR2 = 0x0000;        // CCyIE = 1 -> No se provoca interrupción con el TIMER3
TIM3->SMCR = 0x0000;       // Siempre "0" en este curso
TIM3->PSC = 32000;         // Preescalado = 32000 -> Frecuencia del contador =
                           // 32000000/32000 = 1000 pasos por segundo
TIM3->CNT = 0;             // Inicializo el valor del contador a cero
TIM3->ARR = 0xFFFF;        // Valor recomendado = FFFF
TIM3->CCR1 = 5000;         // Registro donde se guarda el valor que marca la
                           // comparación existosa en TOC.
                           // Inicializo al valor que quiero llegar: 5000 pasos = 5 sg
TIM3->DIER = 0x0002;       // Se genera INT al terminar de contar -> CCyIE = 1
```

- **TIMER 4:** En modo PWM para generar por el PA7 una señal PWM de 100Hz y un DC de 5-90% sin ninguna interrupción asociada. Se configura inicialmente un DC del 10%

```
TIM4->CR1 = 0x0080;    // ARPE = 1 -> Es PWM
                       // CEN = 0; Contador apagado
TIM4->CR2 = 0x0000;    // CCyIE = 0 -> No se provoca interrupción con el
                       // TIMER4
TIM4->SMCR = 0x0000;   // Siempre "0" en este curso
TIM4->PSC = 32000;     // Preescalado = 32000 -> Frecuencia del contador =
                       // 32000000/32000 = 1000 pasos por segundo
TIM4->CNT = 0;         // Inicializo el valor del contador a cero
TIM4->ARR = 9;         // Pongo una frecuencia PWM de 100 Hz, sólo cuento 10
                       // pasos (de 0 a 9) y empiezo de nuevo
TIM4->CCR2 = 1;        // El Duty cycle se pone a 1 paso = 10% de 100Hz
TIM4->DIER = 0x0000;   // No se genera INT al terminar de contar -> CCyIE = 0
TIM4->CCMR1 = 0x6800;  // CCyS = 0 (TOC, PWM)
                       // OCyM = 110 (PWM con el primer semiciclo a 1)
                       // OCyPE = 1 (con precarga)
TIM4->CCER = 0x0010;   // CCyP = 1 (siempre en PWM)
                       // CCyE = 0 (captura deshabilitada)
TIM4->CR1 |= 0x0001;   // CEN = 1 -> Arranco el contador
TIM4->EGR |= 0x0001;   // UG = 1 -> Se genera evento de actualización
TIM4->SR = 0;          // Limpio los flags del contador
```

- **EXTI para la EXTI0** asociada al PA0 para el botón de RESET

```
EXTI->FTSR |= 0x01;      // '1' habilita el evento por flanco de bajada en EXTI0
EXTI->RTSR &= ~(0x01);   // '0' inhabilita el evento por flanco de subida en EXTI0
SYSCFG->EXTICR[0] = 0;   // La EXTI0 la provoca el bit 0 del GPIOA (= PA0)
EXTI->IMR |= 0x01;       // Un '1' habilita la EXTI0, no la enmascara
```

- **NVIC:** Se activan interrupciones para la EXTI0, el TIM2 y el TIM 4 (0,25 puntos).

```
NVIC->ISER[0] |= (1 << 6);  // Habilito la EXTI0 en el NVIC para la entrada de reset.
NVIC->ISER[0] |= (1 << 28); // Habilitación de la INT TIM2_IRQ en NVIC (pos. 28) para
                            // la señal de vida .
NVIC->ISER[0] |= (1 << 29); // Habilitación de la INT TIM3_IRQ en NVIC (pos. 29) para
                            // la señal PWM del sistema auxiliar de regulación de
                            // temperatura.
```

d) Dibuje el diagrama de flujo del programa de control y las posibles interrupciones. (20%)

Esta sería una posible solución según lo supuesto en los apartados anteriores.