

Microcontroladores – Período 24.2 (dezembro/2024 a maio/2025)

Controle de atividades

Alunos	Atividades																																			
	1			2			3			4			5			6			7			8			9			10								
	a	b	c	A	b	c	A	b	c	a	b	c	a	b	c	a	B	c	a	b	c	a	b	c	a	b	c	a	b	c						
Ana Luisa Miranda Pessoa	1	3	6	0	3	2	1	3	4	1	3	1	1	3	5	1	3	6																		
Anabelle Carreiro de Sousa	-	-	-	-	-	-	-	-	-	1	1	0	1	3	6	0	3	4																		
Carlos Eduardo Cavalcanti de S. Filho	1	3	6	1	3	6	1	3	6	1	3	3	1	3	5	1	3	4																		
Deivid Gabriel da Silva Lopes Procopio	1	3	6	1	3	5	1	3	6	Projeto Integrado																										
Emanuel Thadeu dos Santos Conrado	1	3	6	1	3	6	1	3	5	0	3	4	1	3	6	1	3	5																		
Enzo Fernandes Andrade	1	3	6	1	3	6	1	3	6	1	3	5	1	3	6	1	3	6																		
Ezequiel Teotonio Jo	1	3	6	1	3	6	1	3	6	1	3	6	1	3	6	1	3	6																		
Frankley Kaiky Maia Silva	1	3	6	1	3	1	1	3	6	1	3	4	1	3	6	1	3	5																		
Gabriel Soares Santos do Nascimento	1	3	6	1	3	6	1	1	6	1	3	3	1	3	1	1	3	1																		
Gabriela Gomes Cavalcanti A. Monteiro	0	3	6	0	3	5	1	3	1	1	3	1	1	3	5	1	3	4																		
Guilherme Rocha Cruz	1	3	6	-	-	-	-	-	-	-	-	-	-	-	-	0	1	4																		
Gustavo Gomes Souta da Silva	1	3	6	1	3	6	1	3	6	1	3	6	1	3	6	1	3	5																		
Gustavo Jose Leite Cordeiro	1	3	6	1	3	6	1	3	1	1	3	1	1	3	6	0	3	6																		
Joao Marcelo Candido Borges	1	3	6	1	3	3	1	3	6	1	3	4	1	3	5	1	3	6																		
Joaquim Breno Brito Cavalcante	1	3	6	1	3	0	1	3	6	1	3	2	1	3	6	0	3	1																		
Jose Alves de Oliveira Neto	1	3	6	1	3	6	1	3	6	1	3	5	1	3	1	1	3	5																		
Kaio Cesar de Oliveira Barreto	0	3	6	0	3	6	0	3	4	0	3	1	1	3	6	1	3	6																		
Kelvin Soares Oliveira	1	3	6	1	3	6	1	3	2	1	3	2	1	3	1	0	3	1																		
Lucas Freitas Bitu Satiro	1	3	6	1	3	5	1	3	6	-	-	-	1	3	6	1	3	6																		
Luiz Augusto Gomes Landim	1	3	6	1	0	0	1	3	6	0	3	6	1	0	6	1	3	6																		
Marcelo de Freitas Cavalcante	1	3	6	1	3	6	1	3	6	1	3	5	1	3	6	1	3	6																		
Matheus Felipe Lima Santos	1	3	6	1	0	6	1	3	5	1	3	4	1	3	5	1	3	6																		
Petrus George Leal Ismael da C. Neves	1	3	6	1	3	1	1	3	6	1	3	6	1	3	6	1	3	6																		
Rafael Gomes Sousa Abreu	0	3	6	1	3	5	-	-	-	0	3	5	1	3	5	0	3	6																		
Raquel Patricio Moraes	1	3	6	1	3	6	0	3	6	1	3	5	1	3	6	1	3	6																		
Rodrigo Araujo de Souza	1	3	6	1	3	6	1	3	6	1	3	5	1	3	5	1	3	6																		
Rodrigo Lanes Meneses	1	3	6	1	3	0	0	3	3	-	-	-	1	3	5	0	2	4																		
Samuel Almeida Barros	1	3	6	1	3	6	1	3	6	1	3	5	1	3	6	1	3	6																		
Victoria Monteiro Pontes	1	3	6	1	3	3	-	-	-	1	3	2	1	3	1	1	3	1																		
Virgilio Schettini de Oliveira Neto	1	3	6	0	3	6	1	3	4	1	1	0	1	3	1	0	3	1																		

Legenda:

- a) Entrega na data, **até às 09h** (1,0). **Penalização de 3,0 pontos** se entregue no dia seguinte.
- b) Com comentários suficientes e esclarecedores. Até 3,0 pontos.
- c) Atende as especificações. Até 6,0 pontos.

Para a atividade 1, item "c"

Critérios de correção (pontos somados):

Até 2 – Português sem erro de sintaxe

Até 3 – Hipóteses testadas

1 – Apenas um teste por estrutura

Vejam as páginas seguintes com as atividades.

Descrição

Atividade 10 - Data de entrega: 22/04/25

Tema: Sistema (simples) tolerante a falhas

Não deixe para tirar dúvidas na véspera da entrega!

Objetivo: Controlar, a partir de 2 microcontroladores, um atuador para impedir que pare de funcionar em caso de problema em um dos controladores.

Especificações:

- A implementação desta aplicação deve ser feita com dois sistemas microcontrolados, utilizando o microcontrolador **PIC12F675**.
- Entenda-se como componente a ser controlado um LED (externo às placas) que deverá acender e permanecer aceso (indicando ALARME) enquanto uma tensão externa for inferior a 4V;
- O LED deve ser controlado por 2 microcontroladores. Cada microcontrolador assumirá, alternadamente, o controle sobre o atuador a cada 5 segundos;
- O microcontrolador que não estiver no controle deve permanecer em estado de economia de energia;
- A estratégia/criatividade a ser adotada para controle da alternância fazem parte da solução do problema;
- A comunicação entre os módulos do sistema deve ser feita através de uma porta comum mantida em *pull-up* através de um resistor externo;
- O sistema implementado deverá ser capaz de manter o atuador em funcionamento em caso de pane/defeito/problema/ausência em um dos microcontroladores, passando o seu homólogo a assumir o controle do atuador;
- Para indicar qual o microcontrolador que está atuando no controle, um outro LED (**GP5**) deve permanecer aceso.

Atividade 9 - Data de entrega: 15/04/25

Tema: Comunicação SERIAL

Objetivo: Exercício de aplicação de microcontrolador com Implementação do protocolo de comunicação SERIAL (UART).

Especificações:

- O protocolo SERIAL deve ser implementado no PIC (12F675) com baud rate de 9600;
- O PIC deve receber um byte pela porta serial e, se este byte corresponder ao valor identificado na tabela abaixo, o PIC deve:
 - Sinalizar sua identificação através de um LED, mantendo-o aceso por **50 ms**;
 - Efetuar uma conversão A/D (8 bits) e transmitir o valor da conversão pela porta serial;

Para padronizar a utilização das portas, deve ser adotado:

GP0 - RX - porta para receber os bits;

GP1 - TX - porta para transmitir os bits;

GP5 - Led

Os valores individuais que serão recebidos pelo PIC são:

Nome	Valor	
Ana Luisa Miranda Pessoa	07 _d	07 _h
Anabelle Carreiro de Sousa	13 _d	0D _h
Carlos Eduardo Cavalcanti de S. Filho	23 _d	17 _h
Deivid Gabriel da Silva Lopes Procopio	31 _d	1F _h
Emanuel Thadeu dos Santos Conrado	37 _d	25 _h
Enzo Fernandes Andrade	41 _d	29 _h
Ezequiel Teotonio Jo	43 _d	2B _h
Frankley Kaiky Maia Silva	51 _d	33 _h
Gabriel Soares Santos do Nascimento	09 _d	09 _h
Gabriela Gomes Cavalcanti A. Monteiro	15 _d	0F _h

Guilherme Rocha Cruz	25 _d	19 _h
Gustavo Gomes Souta da Silva	33 _d	21 _h
Gustavo Jose Leite Cordeiro	39 _d	27 _h
Joao Marcelo Candido Borges	42 _d	2A _h
Joaquim Breno Brito Cavalcante	45 _d	2D _h
Jose Alves de Oliveira Neto	53 _d	35 _h
Kaio Cesar de Oliveira Barreto	62 _d	3E _h
Kelvin Soares Oliveira	75 _d	4B _h
Lucas Freitas Bitu Satiro	83 _d	53 _h
Luiz Augusto Gomes Landim	91 _d	5B _h
Marcelo de Freitas Cavalcante	94 _d	5E _h
Matheus Felipe Lima Santos	97 _d	61 _h
Petrus George Leal Ismael da C. Neves	103 _d	67 _h
Rafael Gomes Sousa Abreu	109 _d	6D _h
Raquel Patricio Moraes	111 _d	6F _h
Rodrigo Araujo de Souza	51 _d	33 _h
Rodrigo Lanes Meneses	122 _d	7A _h
Samuel Almeida Barros	18 _d	12 _h
Victoria Monteiro Pontes	115 _d	73 _h
Virgilio Schettini de Oliveira Neto	70 _d	46 _h

Atividade 8 - Data de entrega: 08/04/25

Tema: Economizando Energia

Objetivo: Otimizar o consumo de energia, colocando o microcontrolador em modo “SLEEP”.

Contexto: Implementar medidor “econômico” de bateria para que indique o valor da tensão em 4 níveis.

Especificações:

- O microcontrolador deve permanecer em modo “adormecido” devendo ser “acordado” a cada, aproximadamente, 2s;
- Para garantir o menor consumo de energia, **TODAS** as atividades desnecessárias **DEVEM** permanecer desligadas durante o modo SLEEP;
- Ao ser “acordado”, utilizando o *Watchdog*, o microcontrolador deve medir a tensão de uma bateria (através da porta GP0) e indicar o nível da tensão, segundo a tabela:

Tensão	GP1	GP2	GP4	GP5
Menor que 2	ON	OFF	OFF	OFF
$2 \leq V < 3$	OFF	ON	OFF	OFF
$3 \leq V \leq 4$	OFF	OFF	ON	OFF
Maior que 4V	OFF	OFF	OFF	ON

Os valores limite ficam a cargo do projetista.

- Após a atualização do nível de tensão, conforme a tabela acima, o microcontrolador deve voltar ao modo “adormecido”. Lembre que, mesmo em estado adormecido, um LED deve permanecer para indicar o nível da bateria;
- Estime (e demonstre seu cálculo), com base no tempo de execução do seu programa e nas informações do consumo estimado para o sistema, qual é a energia consumida em μWh .

Atividade 7 - Data de entrega: 27/03/25

Tema: Determinando o valor máximo de um sinal e gravá-lo na EEPROM

Objetivo: Medir o valor **máximo** de um sinal, gravar este valor na EEPROM e sinalizar sua ocorrência.

Contexto: O sinal mostrado na Figura 1 corresponde ao registro da altitude de um foguete durante seu lançamento. A duração deste evento é estimada em 60 segundos, a partir do início do acionamento do motor do foguete. Cinco segundos **após** o foguete atingir a altura máxima (apogeu) o sistema de paraquedas deve ser acionado para permitir que o foguete reduza sua velocidade de queda e seja recuperado sem danos. Para

certificar seu desempenho, o valor apogeu deve ser registrado para conferência após a recuperação do foguete.

Especificações:

- Considere que o altímetro fornece um sinal analógico cuja proporção é linear em relação à altitude, fornecendo **1 V** para cada **100 m** e sendo **0 V** a altitude correspondente ao nível do solo;
- Considere que o foguete está projetado para que o apogeu não ultrapasse **420 m**;
- Considere que, instantes antes do lançamento do foguete, um botão (configurado em **pull up**) deve ser pressionado para acionar o início da aquisição de dados;
- Um sistema para aquisição e registro da altitude deve ser implementado utilizando o microcontrolador **PIC12F675**, programado em **Assembly**;
- O botão enviará nível lógico **LOW** à porta **GP0** quando pressionado, indicando que a aquisição deve ser iniciada;
- O valor do apogeu deve ser armazenado na posição **13_h** da **EEPROM**;
- A porta **GP1** deve ser utilizada para acionar o paraquedas;
- A porta **GP2** deve ser utilizada para **conversão A/D** em **8 bits**;
- A conversão A/D deve ser tão rápida quanto possível (limitada pela velocidade do microcontrolador);
- **Cada medida da altura** deve corresponder à **média aritmética de 32 conversões A/D** do sinal analógico fornecido pelo altímetro;
- Os procedimentos a seguir **só devem iniciar** após o microcontrolador identificar nível **LOW** na porta **GP0**;
 - O procedimento de medição da altitude deve ficar em **loop** enquanto o altímetro indicar um valor inferior a **10 m** de altura;
 - Quando o valor da **altura for maior** que o equivalente a **10 m** de altitude (nível de trigger), um outro **loop** de medida de altitude deve ser iniciado e permanecer até que o sistema identifique o apogeu do foguete;
 - Cada novo valor de altitude deve ser comparado para buscar a **MAIOR** altitude no evento;
 - Quando o apogeu for identificado, seu valor deve ser armazenado na **EEPROM** na posição **13_h**;
 - Após a identificação do apogeu, **um TIMER deve ser inicializado para acionar o paraquedas 5 s** após o foguete atingir seu apogeu;
 - A abertura do paraquedas será efetuada com um pulso de **2 s** de duração em nível lógico **HIGH** na porta **GP1**;
 - Após o pulso de abertura do paraquedas, o sistema deve permanecer em **loop** sem atividade (**FIM GOTO FIM**).

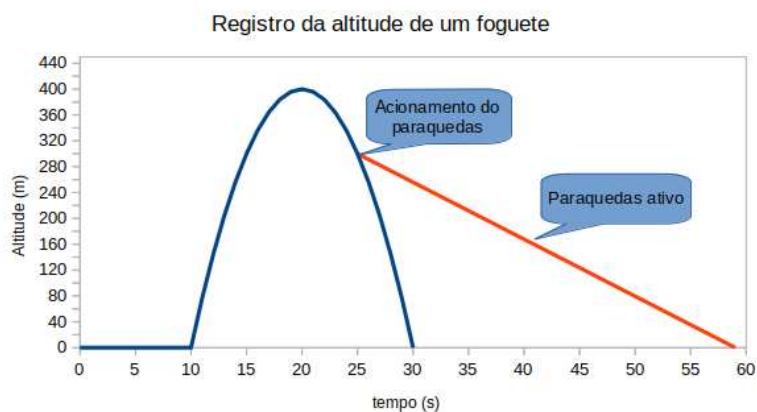


Figura 1. Registro da altitude a partir do altímetro.

Atividade 6 - Data de entrega: 15/03/25

Tema: Controlador de LED RGB

Objetivo: Exercícios para implementação de portas com PWM.

Contexto: Controle da cor e da intensidade do brilho de um LED RGB. Como resultado, o LED pode fornecer mais de 15 milhões de combinações de cores!

Especificações:

- Após o RESET, os LEDs deverão estar apagados;
- Duas chaves serão utilizadas para selecionar o LED que será ajustado, segundo a tabela:

Chaves	Cor do LED
00	Desligados
01	Red
10	Green
11	Blue

- Um canal será utilizado para conversão A/D. O valor da conversão será utilizado para controlar a intensidade (*duty cycle*) do brilho do LED selecionado;
- O percentual do *duty cycle* será dado a partir de uma **conversão A/D** (8 bits), em que valores da conversão maiores de 249 corresponde a 100%. Para valores da conversão menores de 250, o *duty cycle* deverá ser proporcional ao valor da conversão (semelhante à atividade 05);
- A intensidade do LED ajustado deverá ser mantida após a alteração da seleção para outro LED;
- Quando houver *duty cycle* diferente de 100%, a frequência do sinal deve ser de 500Hz;
- GP0, GP1 e GP2 deverão ser utilizados para produzir os sinais PWM, respectivamente, para os LED R, G e B;
- GP3 e GP5 deverão ser utilizados para efetuar a seleção do LED que será ajustado;
- GP4 deverá ser utilizado para efetuar a conversão A/D;

Atividade 5 - Data de entrega: 08/03/25

Tema: Controlador do brilho de um LED (segunda parte)

Objetivo: Exercício com **conversor A/D**.

Contexto: Dimmer para um LED.

Especificações:

- O LED será acionado pela porta GP0;
- O sinal enviado pela porta GP0 será modulado pela largura do pulso, alterando o *duty cycle*;
- O controle do brilho do LED será, portanto, efetuado pela alteração do *duty cycle* (de 0 a 100%);
- Após o RESET, o LED deve iniciar apagado;
- A indicação do percentual do *duty cycle* será dada a partir de uma **conversão A/D** (8 bits), em que valores da conversão maiores de 249 corresponde a 100%. Para valores da conversão menores de 250, o *duty cycle* deverá ser proporcional ao valor da conversão;
- Quando houver *duty cycle* diferente de 100%, a frequência do sinal deve ser de 500Hz;
- GP4 deverá ser utilizado para efetuar a **conversão A/D**.

Atividade 4 - Data de entrega: 25/02/25

Tema: Controlador do brilho de um LED

Objetivo: Exercício com comparador.

Contexto: Dimmer para um LED.

Especificações:

- O LED será acionado pela porta GP4;
- O sinal enviado pela porta GP4 será modulado pela largura do pulso, alterando o *duty cycle*;
- O controle do brilho do LED será, portanto, efetuado pela alteração do *duty cycle* (de 0 a 100%);
- Após o RESET, o LED deve iniciar apagado;
- A indicação do percentual do *duty cycle* será dada a partir de uma comparação, em que 3,5V corresponde a 100%;
- Quando houver *duty cycle* diferente de 100%, a frequência do sinal deve ser de 500Hz;
- GP1 deverá ser utilizado para a entrada do sinal de tensão a ser comparado.

Atividade 3 - Data de entrega: 11/02/25

Tema: Semáforo de trânsito

Objetivo: Exercício de aplicação da linguagem Assembly, com controle de portas.

Contexto: Implemente um aplicativo em Assembly (PIC12F675) para controlar o semáforo de um cruzamento simples de rua: fecha para um lado e abre para o outro, utilizando 1 dígito BCD e 1 porta (LEDs).

Especificações:

- Utilize dois LEDs (verde e vermelho) aplicados a uma **única porta** para funcionar em oposição, segundo a notação:
 - quando a porta é HIGH → LED1 é ON e LED2 é OFF;
 - quando a porta é LOW → LED1 é OFF e LED2 é ON;
- A transição de estado (verde → vermelho ou vermelho → verde), ocorrerá após uma contagem decrescente e cíclica, de 9 até 0;
- Por se tratar de um semáforo didático, cada transição da contagem (indicada no display) deve ocorrer a cada 500 ms.
- A contagem deve ser indicada em um display de 7 segmentos, utilizando a codificação BCD;
- Para que todos tenham a mesma conectividade, siga a seguinte configuração:
 - **GP0** → para conectividade dos LEDs (verde/vermelho)
 - **GP1** → **b₀** (MENOS significativo) do BCD
 - **GP2** → **b₁** do BCD
 - **GP4** → **b₂** do BCD
 - **GP5** → **b₃** (MAIS significativo) do BCD

Atividade 2 - Data de entrega: 19/12/24

Tema: Produto de dois números em assembly

Objetivo: Exercícios de familiarização com o conjunto de instruções do PIC.

Especificações:

- Dado dois valores em **hexadecimal**, armazenados nas variáveis X1 e X2 (1 byte cada), implemente um programa na linguagem assembly (PIC12F675) para efetuar o produto desses números;
- Como o resultado do produto pode ser um valor maior que FF, considera-se que os resultados devem ser apresentados em dois bytes (variáveis R1 e R2);
- A variável R2 será utilizada como byte mais significativo e a variável R1 como byte menos significativo;
- Se o resultado do produto for menor ou igual a 00FF, a variável R2 deverá conter zero.
- Veja os exemplos:

Operandos		Resultado	
X1	X2	R2	R1
25	4A	0A	B2
FA	51	4F	1A
12	1A	01	D4
0E	0A	00	8C

Valores utilizados nos teste:

X1	X2	R2	R1
5	9	00	2D
3B	0	00	00
13	5C	06	D4
C9	77	5D	6F

Atividade 1 - Data de entrega: 10/12/24

Tema: Divisão de 2 números e resultado com uma casa decimal

Objetivo: Exercício com algoritmo para posterior implementação com microcontrolador.

Contexto: Dados dois valores, proponha um ALGORITMO que descreva as instruções necessárias para efetuar a divisão desses valores, fornecendo o resultado com precisão de uma casa decimal.

Especificações:

$$\frac{A}{B} = X, Y$$

- Numerador (**A**) e denominador (**B**) são números inteiros, com valores entre 0 e 255 (um byte);
- Supor que o denominador (**B**) não será zero;
- O resultado deve ser armazenado em duas variáveis inteiras, sendo uma para armazenar a parte inteira (**X**) da divisão e a outra para armazenar a parte decimal (**Y**), conforme alguns exemplos abaixo:

A	B	Resultado	Arredondado	X	Y
9	5	1,8	1,8	1	8
15	4	3,75	3,8	3	8
8	3	2,66667	2,7	2	7
241	40	6,025	6,0	6	0

- As regras de arredondamento devem ser utilizadas;
- O **ALGORITMO** deve ser escrito em Portugol (pseudocódigo escrito em português). Como referência bibliográfica, sugiro o livro: Lógica de Programação: A Construção de Algoritmos e Estruturas de Dados, dos autores Andre Luis Forbellone e Henri Eberspacher;
- O algoritmo **DEVE SER MANUSCRITO** e a entrega será o arquivo gerado por scanner ou fotografia do manuscrito;
- A finalidade da elaboração de um algoritmo é FORÇAR você a PENSAR A SOLUÇÃO, antes de utilizar qualquer linguagem de programação. Portanto, **NÃO É PERMITIDO** a utilização de ambientes de desenvolvimento para fazer o algoritmo em português;
- Pela própria definição, o ALGORITMO deve ter uma sequência de passos descritivos, ordenados e sem ambiguidade;
- Os passos descritivos não podem conter ações abstratas ou que impliquem na utilização de outras operações diferentes das 3 operações aritméticas básicas: **soma**, **subtração** e **multiplicação** (+, - e *);
- A utilização de estruturas condicionais **NÃO PODE CONTER** mais de um teste lógico (por estrutura);
- Apresente exemplos, mostrando que seu algoritmo funciona!
- Não esqueçam de colocar comentários no seu algoritmo.

Projetos Integrados para quem se interessar

Projetos Integrados são opções de atividades voluntárias que podem substituir uma ou mais das atividades regulares da disciplina. O objetivo destes projetos é propor atividades que permitam ao aluno voluntariamente desenvolver habilidades e conteúdos antecipadamente ao andamento da disciplina e/ou conteúdos não contemplados/previstos.

Projeto Integrado - Deivid

Um foguete experimental projetado para atingir 500 m de apogeu precisa ser monitorado em sua trajetória, com transmissão de dados em tempo real. A monitoração consiste nas informações de altura e de aceleração. Os dados transmitidos a partir do foguete deverão ser enviados em uma taxa de amostragem elevada para permitir o registro adequado da sua trajetória.

Os dados de altura e acelerometria serão obtidos a partir dos sensores BMP280 e MPU-6050, em que ambos utilizam o protocolo de comunicação I²C. O sensor BMP280 fornecerá os valores de pressão e de temperatura, que permitem o cálculo da altura. O sensor MPU-6050 fornecerá os valores de aceleração para cada eixo (x, y, z).

Os dados obtidos dos sensores deverão ser transmitidos via rádio, utilizando o módulo LORA. Este módulo utiliza o protocolo de comunicação serial/UART.

Para fazer a integração destes módulos (BMP280, MPU-6050 e LORA), dois microcontroladores PIC12F675 serão utilizados: um dentro do foguete para coleta e transmissão dos dados e o outro para recepção, registro e visualização dos dados.

A comunicação deve ser bi-direcional, mas não simultânea, obedecendo o procedimento do ciclo do evento:

- 1) Pre-lançamento: calibração dos sensores e início da transmissão/recepção dos dados;
- 2) Durante o vôo: coleta e transmissão de dados; Identificação do apogeu e acionamento do para-quedas; Recepção, registro e visualização dos dados recebidos.

Cronograma:

Até a data da At 04: Estudo e compreensão do das características de funcionamento, configuração e operação do módulo LORA;

Até a data da At 05: Estudo e compreensão do das características de funcionamento e operação do sensor BMP280;

Até a data da At 06: Implementação da comunicação serial/UART entre PIC/LORA e LORA/PIC, com exemplos iniciais de transmissão de dados. A taxa de amostragem já deve ser compatível com as características necessárias ao registro adequado da trajetória do foguete;

Até a data da At 07: Fluxograma completo e implementação preliminar, contemplando a integração com o sensor BMP280. Visualização dos dados transmitidos;

Até a data da At 08: Testes de validação de atendimento das especificações com a integração dos sensores e visualização dos dados transmitidos;

Até a data da At 09: Testes finais e entrega da documentação completa.