

UNIVERSIDADE FEDERAL DO CEARÁ



DOCUMENTO DE ELICITAÇÃO DE REQUISITOS DO SISTEMA DE PISTA AUTOMATIZADA PARA COMPETIÇÃO DE ROBÓTICA

Equipe:

Allisson Jardel Alves de Oliveira - 378561
Anny Caroline da Silva Cruz - 378563
Fernando Cavalcanti Lima Salviano de Medeiros - 378567
Gilgleison Paulino Lima - 378571
Mateus Souza Araújo - 374858

Professora: Jéssyka Flavianne Ferreira Vilela

**MAIO
2018**

Sumário

1. INTRODUÇÃO	5
2. CONVENÇÕES ADOTADAS	5
2.1. Convenção para identificação de requisitos	5
2.1.1. Prioridades dos Requisitos Funcionais	5
3. CONTEXTO	6
3.1. Propósito e as metas organizacionais do produto frente ao mercado	6
3.2. Características gerais do produto	6
3.3. Impactos organizacionais com o desenvolvimento do produto	6
3.4. Impactos negativos com o não desenvolvimento do produto	6
3.5. Expectativas de tempo total de desenvolvimento do produto	6
3.6. Público a ser atingido	6
3.7. Projetos de sistemas legados	7
4. STAKEHOLDERS	7
4.1. Principais stakeholders	7
4.2. Stakeholder especialistas de domínio	7
4.3. Stakeholders contrário ao projeto	7
4.4. Perfil do usuário	8
5. REQUISITOS DE ALTO NÍVEL	8
5.1. Funções do produto (Requisitos Funcionais)	8
5.2. Restrições do produto (Requisitos não funcionais)	8
5.3. Restrições físicas do ambiente	8
5.4. Características de consumo de potência	9
5.5. Características físicas e mecânicas	9
5.6. Interface	9
5.7. Situações críticas	9
5.8. Grau de confiabilidade	9
5.9. Solução encontrada	10
5.10. Estimativa de custos	10
6. HARDWARE	10

6.1. Sensores	10
6.2. Atuadores	10
6.3. Interação com o usuário	11
6.4. interrupções de Hardware	11
6.5. Botões	11
6.6. Memórias	11
6.7. Portas de comunicação externa	11
6.8. Requisitos de componentes	11
6.9. Requisitos de layout da placa controladora	11
6.10. Parâmetros de HW legados	11
6.11. Parâmetros de COTS especiais.	12
6.12. Microcontroladores	12
7. SOFTWARE	12
7.1. Variáveis de ambiente	12
7.2. Funções de Software	13
7.3. Exceções	13
7.4. Funções de interrupções	13
7.5. Requisitos de idioma	13
7.6. Interface de comunicação (software)	13
7.7. Funções de monitoramento	13
7.8. Funções de armazenamento de dados	13
8. MÉTRICAS DE QUALIDADE	14
8.1. Grau de segurança	14
8.2. Desempenho	14
8.3. Métricas de manutenção	14
9. Métricas de linha de produção	14
9.1. Aspectos de produção	14
9.2. Embalagem	14

10.	Conclusão	15
11.	Relatório da Equipe	16
12.	Apêndice	17
13.	Referências	20

1. INTRODUÇÃO

Será feito uma elicitação de componentes para uma pista de robô seguidor de linha. O intuito é de cronometrar o tempo que um determinado carrinho leva para completar o circuito da competição. O foco do competidor é criar um carrinho que consiga fazer uma leitura rápida da pista e executar a ação de segui-la o mais rápido possível para concluir a volta com êxito. Os competidores terão que desenvolver o seu próprio protótipo, desse modo, irá estimular o aprendizado dos competidores e uma melhor assimilação de um determinado conteúdo.

2. CONVENÇÕES ADOTADAS

2.1. Convenção para identificação de requisitos

Para facilitar a busca e referência de requisitos ao longo do documento será adotada uma convenção para cada tipo de requisito. Os requisitos funcionais serão representados no formato [RFxx] e os requisitos não funcionais no formato [NFRxx], no qual “RF” e “NFR” são as siglas para cada tipo e o “xx” representa o número do requisito.

2.2. Prioridades dos Requisitos Funcionais

Os Requisitos Funcionais foram classificados como:

.Essencial: É o requisito funcional indispensável ao funcionamento do sistema. Esse tipo de requisito deve ser implementado impreterivelmente, caso contrário, o projeto perderá sua utilidade. Desenho da pista em um banner, com o circuito completo para a competição.

.Importante: Sem este requisito funcional, o sistema ainda é capaz de ser utilizado. Contudo, essa utilização ocorre de forma não satisfatória pelo cliente. Cronometragem do tempo que cada competidor levou para completar o circuito da competição.

.Desejável: Esse tipo de requisito poderá ser implementado em versões posteriores do sistema, visto que, mesmo sem a sua implementação, o sistema atende as suas funcionalidades básicas. Marcação do tempo exato para a conclusão de cada volta, com diferenças de tempo de um para o outro.

3. CONTEXTO

3.1. Propósito e as metas organizacionais do produto frente ao mercado

O propósito é o desenvolvimento de um circuito de corrida para competições de robôs seguidores de linha, que são financeiramente viáveis para o público de competidores e amantes de competições de robôs seguidores de linha. A meta é alcançar o público alvo que consiste em estudantes universitários até competidores amadores.

3.2. Características gerais do produto

O circuito da competição será feito em um banner, com um sensor de presença para marcar a largada e chegada de cada carro, um display LCD para marcação do tempo dos competidores, um sensor bluetooth para comunicação entre aplicativo do celular e a pista e um sensor de infravermelho que iniciará a corrida do carrinho com ativado pelo aplicativo.

3.3. Impactos organizacionais com o desenvolvimento do produto

Contribui para o desenvolvimento de sistemas mais eficientes, no caso do projeto, o robô seguidor de linha.

3.4. Impactos negativos com o não desenvolvimento do produto

Não possui impactos negativos com o não desenvolvimento.

3.5. Expectativas de tempo total de desenvolvimento do produto

4 meses.

3.6. Público a ser atingido

Competidores e organizadores da competição de robôs seguidores de linha.

3.7. Projetos de sistemas legados

Trata-se de um projeto que será realizado pela primeira vez, não sendo utilizado nenhum sistema legado.

4. STAKEHOLDERS

4.1. Principais stakeholders

Os principais *Stakeholders* que se encaixam no perfil do projeto são os interessados em competições de robótica ou aqueles que desejam automatizar o perfil de produção de determinado segmento comercial, já que o produto pode ser modificado com a necessidade de cada finalidade. Para os competidores, se enquadram estudantes, *hobbystas* e profissionais da área de computação, professores e organizadores de eventos voltados para esse tipo de competição. Para o caso interno do Campus da UFC em Quixadá no período de 2018.1, o público principal serão os estudantes que participarão de uma corrida de robôs seguidores de linha. Esse projeto levará em consideração as especificações internas da disciplina de microcontroladores.

4.2. Stakeholder especialistas de domínio

Os *Stakeholders* que terão uma familiaridade maior com o produto serão professores ou pessoas que se relacionam diretamente com competições robóticas, como organizadores, compra de materiais relacionados a esse tipo de área. Dependendo da necessidade do cliente, modificações podem ser feitas no produto para direcionar isso a finalidades de produção comercial em indústrias ou comércios em geral. Gerentes ou consultores de empresas, que necessitam de recursos de automação para viabilizar e agilizar suas linhas de produção, podem ser citados.

4.3. Stakeholders contrário ao projeto

Os *stakeholders* contrários ao projeto pelo menos em uma visão comercial minimalista seriam aqueles que desconfiavam do produto que se tratar de algo de implementação com finalidade a projetos internos de uma instituição de ensino. Como seus interesses são comerciais, impossibilitando um acordo ou pedido de flexibilização maior de suas necessidades à equipe que produziu o produto. Além disso, temos também a visão de alguns profissionais que visualizaram o projeto de forma crítica, e dessa forma, exigindo talvez mais

requisitos ou sendo até mesmos contrários a alguns adeptos ou especificações internas exigidas no momento do levantamento para implementação.

4.4. Perfil do usuário

Usuários de tecnologia em geral que implementam seus dispositivos como *hobby*, para competições de robôs seguidores de linha se enquadram muito bem no tipo de usuário a qual o produto se destina. Citações de fatores comerciais de produção serão feitas a fim de possibilitar a procura desse tipo de produto por gestores ou consultores. Nesse último caso, encontra-se a necessidade de uma maior produção em escala, uma vez que a parte competitiva destinada primeiramente para estudantes se torna lucro na inserção de tal produto nas suas linhas comerciais.

5. REQUISITOS DE AUTO NÍVEL

5.1. Funções do produto(Requisitos Funcionais)

Identificação:	Pista para robôs seguidores de linha.
Descrição:	O sistema deve notificar o aplicativo sobre a detecção do robô seguidor de linha pelos sensores de presença, a fim de estimar o tempo de conclusão do circuito pelo robô.
Prioridade:	<input checked="" type="checkbox"/> Essencial <input type="checkbox"/> Importante <input type="checkbox"/> Desejável

5.2. Restrições do produto (Requisitos não funcionais)

A pista pode conter apenas um robô seguidor de linha.

5.3. Restrições físicas do ambiente

A pista será feita em um banner, logo não é aconselhável que a pista permaneça em lugares úmidos ou seja molhada. Além disso, é interessante que a competição ocorra em lugares bem iluminados a fim de possibilitar um melhor desempenho dos robôs que participarão da competição. Todavia, a pista terá

obstáculos em seus desenhos propondo que o competidor ache soluções para aqueles pequenos desvios.

5.4. Características de consumo de potência

Robô: Pilhas AA - 1,5 V e Pilha 9V
Pista: Pilha 9V

5.5. Características físicas e mecânicas

- .A pista não possui restrições de peso para o robô.
- .A pista será de 1 x 2 metros
- .O robô não poderá exceder 25 cm de largura.
- .Uma pista será feita em um banner que terá sensores de presença para notificarem o sistema sobre a presença do robô.

5.6. Interface

Será através de um aplicativo, onde o usuário iniciará o robô através do botão presente na interface do aplicativo. Ao botão ser pressionado, o aplicativo guardará o tempo do sistema e enviará uma mensagem para a pista a fim de ligar o robô. Ao robô ser detectado pelos sensores de presença, a pista notificará o aplicativo que guardará o tempo do sistema e mostrará o tempo que o robô levou para chegar a determinado ponto da pista, esse processo continuará até que o robô conclua todo o percurso da pista.

5.7. Situações críticas

- .O Sistema desligar por falta de energia.
- .As conexões bluetooth não serem realizadas com sucesso.
- .O robô não ser detectado pelos sensores de presença.

5.8. Grau de confiabilidade

Alto.

5.9. Solução Encontrada

1 - A pista será conectada a uma pilha de 5 Volts, a fim de manter uma fonte de energia estável para o sistema.

2 - O sistema contará com uma boa conexão entre os módulos bluetooths anexados tanto na pista quanto no robô.

3 - Os sensores de presença serão dispostos ao decorrer da pista a fim de detectar o robô nos seus respectivos pontos. Os sensores serão ajustados de modo que possam notificar a pista sobre a detecção do robô.

5.10. Estimativa de custos

Valor Total: R\$ 233,10

Equipamento	Quantidade	Valor(Total) - Reais(R\$)
Sensor de Presença	3	44,70
Jumpers	65	14,90
Resistores	20	1,40
Capacitores	10	0,80
Protoboard	1	14,90
Arduino	1	52,90
Buzzer	3	5,10
Banner - Pista	1	130,0

6. HARDWARE

6.1. Sensores

São usados três sensores de presença.

6.2. Atuadores

Atuadores não são utilizados.

6.3. Interação com o usuário

Interação com o usuário será feita a partir do aplicativo.

6.4. Interrupções de Hardware

Somente a partir dos sensores de presença.

6.5. Botões

Botões não são utilizados.

6.6. Memórias

32KBytes de In-System Self-Programmable Flash program Memory
1KBytes EEPROM – 2KBytes Internal SRAM

6.7. Portas de comunicação externa

Dois Módulos Bluetooth, um para comunicação Robô - Pista e outro para comunicação Pista - Aplicativo.

6.8. Requisitos de componentes

Sensor de Presença: Distância detectável: 3~7m (Ajustável);

6.9. Requisitos de layout da placa controladora

- .Arduino: Uno Atmega 328 .
- .Tensão de operação :5V.
- .Tensão de entrada (recomendada) :7-12V.
- .Tensão de entrada (limite) :6-20V.
- .Pinos de E / S :digitais 54 (dos quais 15 fornecem saída PWM).

6.10. Parâmetros de HW legados

Não há sistemas legados nesse projeto.

6.11. Parâmetros de COTS especiais.

Não existem sistemas prontos a serem utilizados nesse projeto.

6.12. Microcontroladores

Microcontrolador da Atmega328P:

Tensão de operação : 5V TTL.

Entradas e saídas digitais: 23.

Memória Flash: 32 KB.

Velocidade do Clock: 0 – 20 MHz.

7. SOFTWARE

7.1. Variáveis de ambiente

Variável	Função
VarSensor1	<i>VarSensor1 inicia com o valor 0. VarSensor1 recebe 1 quando o robô for detectado pelo Sensor de Presença 1 pela primeira vez.</i>
VarSensor2	<i>VarSensor2 inicia com o valor 0. VarSensor2 recebe 1 quando o robô for detectado pelo Sensor de Presença pela primeira vez.</i>
VarSensor3	<i>VarSensor3 inicia com o valor 0. VarSensor3 recebe 1 quando o robô for detectado pelo Sensor de Presença pela primeira vez.</i>
VarBluetooth1	<i>VarBluetooth2 conterá o bit que será enviado para o celular(Aplicativo).</i>
VarBluetooth2	<i>VarBluetooth2 conterá o bit que será enviado para o celular(Aplicativo).</i>

7.2. Funções de Software

1 - Inicia o robô.

2 - Mostra o tempo que o robô levou para percorrer respectivas distâncias da pista, marcadas pelo sensor de presença.

7.3. Exceções

1 - Aplicativo inicia apenas um robô por vez.

2 - A pista suporta apenas um robô por vez.

7.4. Funções de interrupções

1 - Iniciar o robô.

2 - Notificar que o robô foi detectado por um dos sensores de presença.

3 - Parar o robô.

7.5. Requisitos de idioma

Idioma de comunicação: Português.

7.6. Interface de comunicação (software)

Interface em java.

7.7. Funções de monitoramento

Monitorar a detecção do robô por um dos sensores de presença.

7.8. Funções de armazenamento de dados

1 - Guardar o tempo do sistema(horário) que o usuário inicia o robô.

2 - Guardar os tempos do sistema(horário) que o robô foi detectado pelos respectivos sensores

8. MÉTRICAS DE QUALIDADE

8.1. Grau de segurança

O sistema necessita de um grau elevado de confiabilidade para a execução exata do instante em que o circuito da pista for finalizado.

8.2. Desempenho

O desempenho do sistema tende a um grau elevado de precisão no momento de captura das ações do senso para armazenar o tempo de cada volta no circuito.

8.3. Métricas de manutenção

O sistema deve ter todos os componentes analisados no decorrer de cada montagem da pista. É necessária uma calibragem periódica sobre o dispositivo de medição de tempo, evitando dados fantasmas durante a competição.

9. Métricas de linha de produção

9.1. Aspectos de produção

Para tornar mais flexível a montagem da pista foi utilizado um banner com o desenho da pista. O material permite fácil montagem e alteração no percurso. Os demais componentes são produzidos a base de acrílico tornando mais leve e maleável os competidores poderem fazer alterações.

9.2. Embalagem

Para o transporte do material será usado uma caixa contendo os componentes eletrônicos que serão concatenados na pista no momento da corrida. O transporte da pista será feito por um cone para projeto evitando qualquer tipo de dano durante o transporte.

10. Conclusão

O desenvolvimento da pista para robôs seguidores de linha tem como objetivo obter o tempo que um robô seguidor de linha demora para percorrer um determinado percurso e, de certa forma, encontrar o robô projetado da forma mais eficiente possível através do tempo que o mesmo levou para completar o trajeto. Esse documento contém o conjunto de condições e restrições para o desenvolvimento e funcionamento do sistema. As informações desse documento vão desde os dispositivos que serão utilizados no hardware, como os sensores de presença, e buzzers, até a declaração das variáveis de ambiente e suas respectivas funções no sistema. Além disso, o documento mostra, através de diagramas, o comportamento do sistema em durante o ciclo de execução do mesmo, além dos possíveis casos que suas respectivas partes (Aplicativo + Pista) podem assumir durante o ciclo de execução.

11. Relatório da Equipe

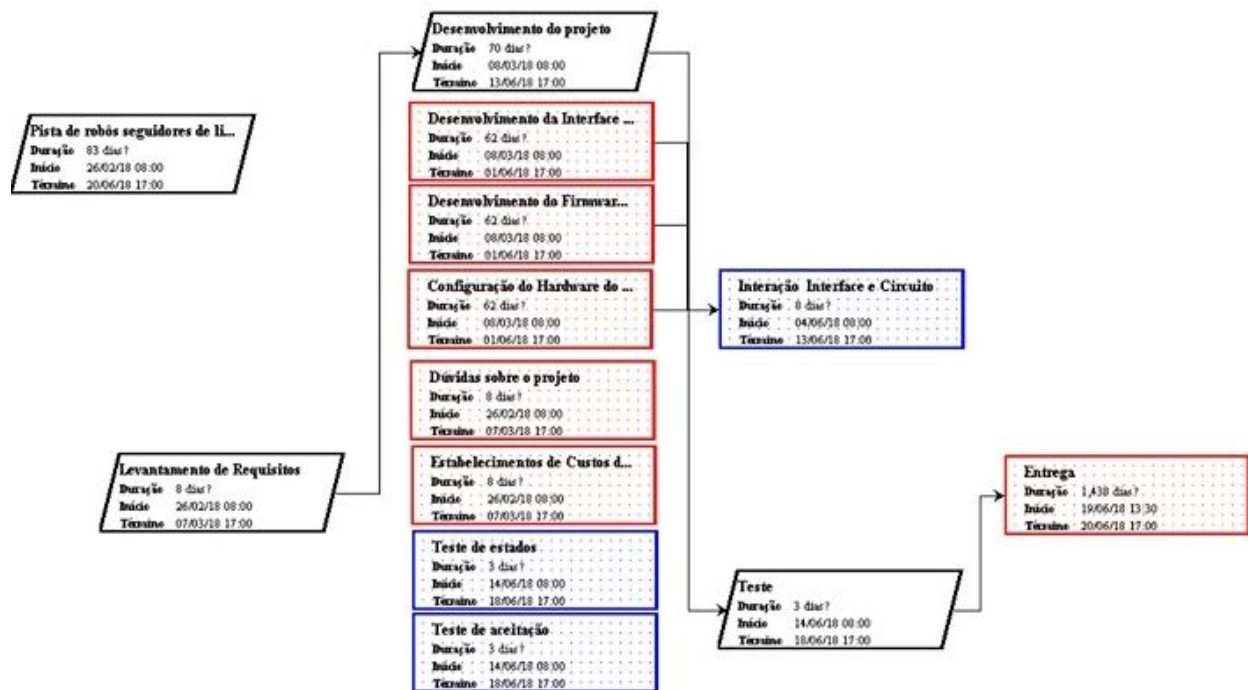
Nome	Esforço da Equipe
Allisson Jardel Alves de Oliveira	20%
Mateus Sousa de Araújo	20%
Fernando Cavalcanti Lima Salviano de Medeiros	20%
Gilgleison Paulino Lima	20%
Anny Caroline da Silva Cruz	20%

Apêndice

Atmega328P : ATmega328 / P é um CMOS de 8 bits de baixa potência. O ATmega328P é baseado na arquitetura RISC aprimorada do AVR.

TERASE: Modelo(Template) usado para especificação de requisitos em sistemas embarcados.

Fases do Projeto



		Nome	Duração	Início	Término	Predecessoras	Nome do Recurso
1		[-] Pista de robôs seguidores de linha	83 dias?	26/02/18 08:00	20/06/18 17:00		Fernando Cavalcanti; Mateus Sousa; Allisson...
2		[-] Desenvolvimento do projeto	70 dias?	08/03/18 08:00	13/06/18 17:00	7	
3		Desenvolvimento da Interface para Android	62 dias?	08/03/18 08:00	01/06/18 17:00		Allisson Jardel; Fernando Cavalcanti
4		Desenvolvimento do Firmware do Projeto	62 dias?	08/03/18 08:00	01/06/18 17:00		Mateus Sousa; Fernando Cavalcanti
5		Configuração do Hardware do projeto	62 dias?	08/03/18 08:00	01/06/18 17:00		Fernando Cavalcanti; Anny Caroline; Gilgelson Pau...
6		Interação Interface e Circuito	8 dias?	04/06/18 08:00	13/06/18 17:00	3;4;5	Mateus Sousa; Allisson Jardel; Fernando Cavalcant...
7		[-] Levantamento de Requisitos	8 dias?	26/02/18 08:00	07/03/18 17:00		Mateus Sousa; Allisson Jardel; Fernando Cav...
8		Dúvidas sobre o projeto	8 dias?	26/02/18 08:00	07/03/18 17:00		Mateus Sousa; Allisson Jardel; Fernando Cavalcant...
9		Estabelecimentos de Custos do Projeto	8 dias?	26/02/18 08:00	07/03/18 17:00		Mateus Sousa; Allisson Jardel; Fernando Cavalcant...
10		[-] Teste	3 dias?	14/06/18 08:00	18/06/18 17:00	2	Mateus Sousa; Allisson Jardel; Fernando Cav...
11		Teste de estados	3 dias?	14/06/18 08:00	18/06/18 17:00		Mateus Sousa; Allisson Jardel; Fernando Cavalcant...
12		Teste de aceitação	3 dias?	14/06/18 08:00	18/06/18 17:00		Mateus Sousa; Allisson Jardel; Fernando Cavalcant...
13		Entrega	1,438 dias?	19/06/18 13:30	20/06/18 17:00	10	Mateus Sousa; Allisson Jardel; Fernando Cavalcant...

Esboço do código do aplicativo

```

1 private void lblT1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String hora, minuto, segundo, total;
    int hora_atual, minuto_atual, segundo_atual;
    int tempo_sec_atual;
    int tempo_levado;
    Calendar Relogio1 = Calendar.getInstance();
    int h_levado, m_levado, s_levado;
    total = dt.format(Relogio1.getTime());
    hora = hr.format(Relogio1.getTime());
    minuto = mn.format(Relogio1.getTime());
    segundo = sg.format(Relogio1.getTime());

    hora_atual = Integer.parseInt(hora);
    minuto_atual = Integer.parseInt(minuto);
    segundo_atual = Integer.parseInt(segundo);
    tempo_sec_atual = 3600*hora_atual + 60*minuto_atual + segundo_atual;
    tempo_levado = tempo_sec_atual - tempo_sec;
    h_levado = tempo_levado / 3600;
    m_levado = (tempo_levado % 3600) / 60;
    s_levado = (tempo_levado % 3600) % 60;

    list.addElement("Horário : "+total+"--"+h_levado+" : "+m_levado+" : "+s_levado);
}

```

```

int vetor_Tempo[] = new int[10];
int horas, minutos, segundos;
int tempo_sec;
int contador = 0;

Calendar Relogio = Calendar.getInstance();
SimpleDateFormat hr = new SimpleDateFormat("HH");
SimpleDateFormat mn = new SimpleDateFormat("mm");
SimpleDateFormat sg = new SimpleDateFormat("ss");
SimpleDateFormat dt = new SimpleDateFormat("HH:mm:ss");

String hora_inicial, minuto_inicial, segundo_inicial;
DefaultListModel list = new DefaultListModel();

/**
 * Creates new form tela
 */
public tela() {
    initComponents();
    hora_inicial = hr.format(Relogio.getTime());
    minuto_inicial = mn.format(Relogio.getTime());
    segundo_inicial = sg.format(Relogio.getTime());
    lblTempo.setText(dt.format(Relogio.getTime()));
    horas = Integer.parseInt(hora_inicial);
    minutos = Integer.parseInt(minuto_inicial);
    segundos = Integer.parseInt(segundo_inicial);
    tempo_sec = 3600*horas + 60*minutos + segundos;
}

```

Referências

- 1. Ossada, J. C., Martins, L. E. G., Ranieri, B. S., & Belgamo, A. GERSE: Guia de Elicitação de Requisitos para Sistemas Embarcados**
- 2. GREENE, Jennifer; STELLMAN, Andrew, NOÇÕES DE GERENCIAMENTO DE PROJETOS**
- 3. Martins, L. E. G., de Souza Júnior, R., de Oliveira Jr, H. P., & Peixoto, C. S. A. TERASE: Template para Especificação de Requisitos de Ambiente em Sistemas Embarcados.**

