UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

LUIS GUILHERME MACHADO CAMARGO
PEDRO ALBERTO DE BORBA
RICARDO FARAH
STEFAN CAMPANA FUCHS
TELMO FRIESEN

MAPEAMENTO DE AMBIENTES COM O ROBÔ BELLATOR

PRIMEIRO ENTREGÁVEL

CURITIBA

2013

LUIS GUILHERME MACHADO CAMARGO
PEDRO ALBERTO DE BORBA
RICARDO FARAH
STEFAN CAMPANA FUCHS
TELMO FRIESEN

MAPEAMENTO DE AMBIENTES COM O ROBÔ BELLATOR

Primeiro entregável apresentada à Unidade Curricular de Oficina de Integração 3 do Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná como requisito parcial para aprovação.

CURITIBA

2013

SUMÁRIO

1 PRIMEIRO ENTREGÁVEL - 13/03/2013 3
2 MODELAGEM UML 4
2.1 REQUISITOS FUNCIONAIS 4
2.2 REQUISITOS NÃO FUNCIONAIS 4
2.3 CASOS DE USO IDENTIFICADOS 5
2.4 DIAGRAMA DE CLASSES 6
2.5 DESCRIÇÃO DAS CLASSES 7
2.5.1 Pacote *visual* 7
2.5.2 Pacote *controle* 7
2.5.3 Pacote *comunicacao* 7
2.5.4 Pacote *gui* 8
3 DIAGRAMA DE BLOCOS DO HARDWARE 10

1 PRIMEIRO ENTREGÁVEL - 13/03/2013

Conforme estabelecido na relação de *deliverables* do documento de análise tecnológica, este primeiro entregável consiste nos seguintes itens:

1. Versões iniciais dos diagramas de casos de uso e de classes (estação base).
2. Versão inicial do diagrama em blocos (hardware).
3. Explicação inicial de cada bloco (hardware).

2 MODELAGEM UML

2.1 REQUISITOS FUNCIONAIS

1. A estação base deve mostrar na interface gráfica um mapa 2D (atualizado automaticamente) representando o robô e os obstáculos detectados por ele. Representado pelo requisito funcional: **"Estação base mostra mapa 2D do robô e dos obstáculos detectados – RF1"**.
2. O usuátio pode salvar o mapa 2D no disco rígido. Representado pelo requisito funcional: **"O usuário pode salvar o mapa – RF2"**.
3. O usuário pode carregar o mapa 2D do disco rígido. Representado pelo requisito funcional: **"O usuário pode carregar o mapa – RF3"**.
4. A estação base deve mostrar na interface gráfica a imagem captada pela *webcam* do robô. Representado pelo requisito funcional: **"Estação base mostra a imagem captada pela webcam – RF4"**.
5. O usuário pode movimentar o robô, controlando a velocidade de suas rodas remotamente pelo teclado da estação base. Representado pelo requisito funcional: **"O usuário pode movimentar o robô – RF5"**.
6. O usuário pode parar o robô remotamente através de comandos do teclado da estação base. Representado pelo requisito funcional: **"O usuário pode parar o robô – RF6"**.
7. A estação base deve ser capaz de estabelecer conexão com o robô, informando o usuário caso a conexão ocorra com sucesso ou não. Representado pelo requisito funcional **"O usuário pode estabelecer a conexão entre o robô e a estação base – RF7"**.

2.2 REQUISITOS NÃO FUNCIONAIS

1. A imagem transmitida pela câmera do robô deve ser colorida. Representado pelo requisito não funcional: **"O robô deve enviar vídeo em imagem colorida para a estação base - RNF1"**.

2. O robô deve transmitir as imagens de sua câmera em tempo real. Representado pelo requisito não funcional: **"O robô deve transmitir os dados de vídeo captados pela câmera em tempo real - RNF2"**.

2.3 CASOS DE USO IDENTIFICADOS

1. Visualização de mapa 2D na interface gráfica segundo os dados lidos dos sensores do robô. Representado pelo caso de uso: **"Mostrar mapa - UC1"**.
2. Gravação do mapa em um arquivo no disco rígido. Representado pelo caso de uso: **"Salvar mapa - UC2"**.
3. Leitura do mapa de um arquivo do disco rígido. Representado pelo caso de uso: **"Carregar mapa - UC3"**.
4. Leitura de informações dos sensores do robô. Representado pelo caso de uso: **"Leitura de sensores - UC4"**.
5. Captura de imagens da *webcam* do robô. Representado pelo caso de uso: **"Capturar imagens da câmera - UC5"**.
6. Visualização de imagens da *webcam* do robô. Representado pelo caso de uso: **"Visualizar imagens da câmera - UC6"**.
7. Alteração pelo usuário da velocidade das rodas do robô. Representado pelo caso de uso: **"Movimentar o robô - UC7"**.
8. Parada do robô solicitada pelo usuário. Representado pelo caso de uso: **"Parar o robô - UC8"**.
9. Solicitação de estabelecimento de conexão com o robô. **"Estabelecer conexão - UC9"**.
10. Consulta da documentação do robô solicitada pelo usuário. Representado pelo caso de uso: **"Consultar documentação do robô - UC10"**.

Na Figura 1 está presente o diagrama de casos de uso, que foi produzido tomando-se como base as especificações apresentadas anteriormente. Vale ressaltar que as flechas pontilhadas (que denotam dependência entre casos de uso), apesar de deixarem o aspecto visual mais complexo, foram inseridas para reforçar a relação entre as funcionalidades da estação base e do robô.

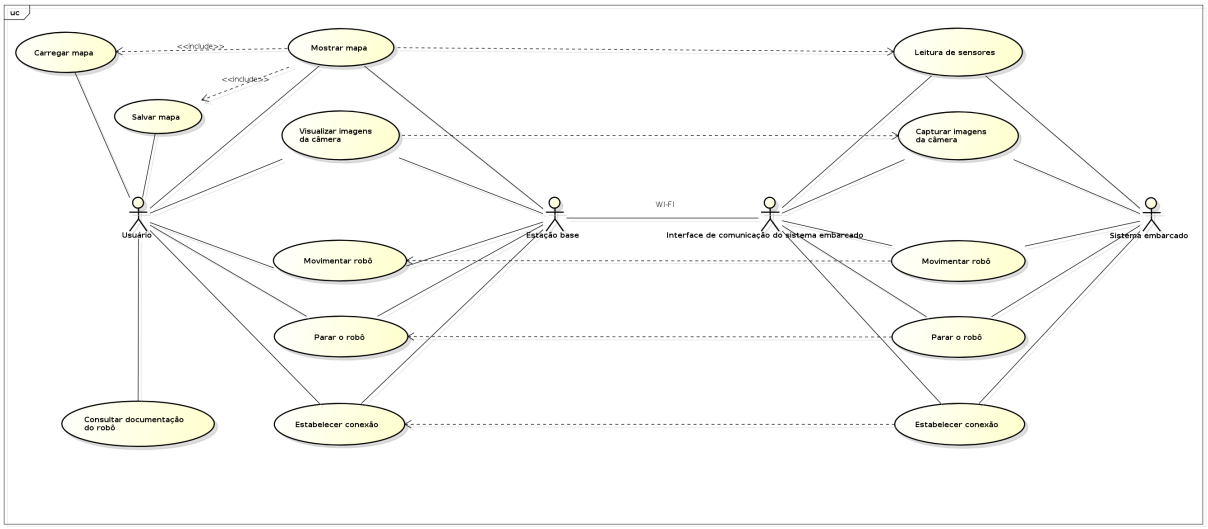

Figura 1: Diagrama de casos de uso.

2.4 DIAGRAMA DE CLASSES

A Figura mostra o diagrama de classes, tanto da estação base quanto do *software* embarcado. A maior parte das classes pertence ao *software* da estação base. As classes do *software* do sistema embarcado que existem até o momento são as com prefixo *Server* (no pacote *comunicacao*).

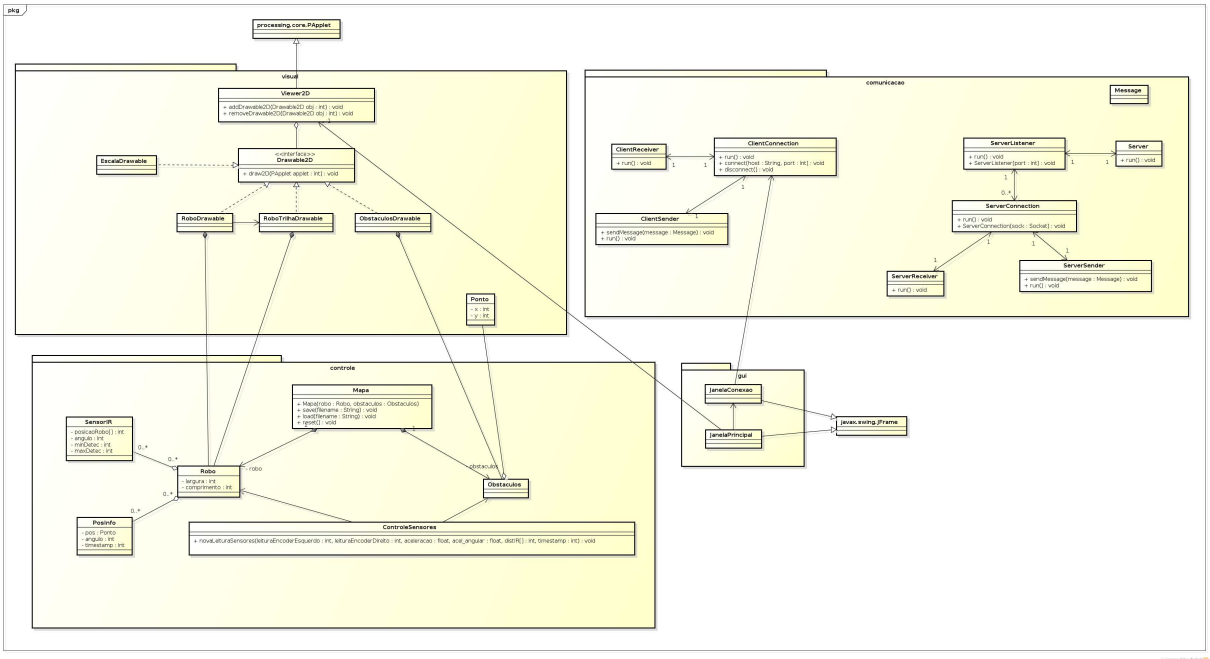

Figura 2: Diagrama de classes

2.5 DESCRIÇÃO DAS CLASSES

O software da estação base do robô foi dividido em cinco pacotes: *visual*, *controle*, *comunicao*, e *gui*. A seguir há uma descrição de cada pacote e das suas respectivas classes.

2.5.1 Pacote *visual*

Este pacote consiste de toda a parte visual da estação base e conta com as seguintes classes: Viewer2D, Drawable2D, EscalaDrawable, RoboDrawable, RoboTrilhaDrawable, ObstaculosDrawable, e Ponto. Na Tabela 1 estão descritas as classes deste pacote.

Tabela 1: Pacote *visual*

Classe	**Descrição**
Viewer2D	Responsável por exibir os objetos Drawable2D. Possui recursos de pan, zoom e rotate.
Drawable2D	Representa genericamente objetos 2D que podem ser desenhados em um Viewer2D.
EscalaDrawable	Responsável por desenhar uma escala gráfica no mapa.
RoboDrawable	Responsável por desenhar o robô no mapa.
RoboTrilhaDrawable	Responsável por desenhar a trilha percorrida pelo robô no mapa.
ObstaculosDrawable	Responsável por desenhar os pontos de cada obstáculo no mapa.
Ponto	Representa um ponto de cordenadas cartesianas (x,y).

2.5.2 Pacote *controle*

Este pacote consiste de toda a parte da estação base que controla as informações essenciais do robô. Conta com as seguintes classes: Mapa, Obstaculos, Robo, ControleSensores, Posinfo, SensorIR e ControleCamera. Na Tabela 2 estão descritas as classes deste pacote.

2.5.3 Pacote *comunicacao*

Este pacote consiste em toda a parte de comunicação da estação base com o robô e conta com as seguintes classes: ClientConnection, ClientReceiver, ClientSender, Server, Ser-

Tabela 2: Pacote *controle*

Classe	**Descrição**
Mapa	Responsável por representar o mapa. Armazena as informações essenciais do robô e dos obstáculos detectados.
Obstaculos	Responsável por conter os obstáculos detectados pelo robô.
Robo	Responsável por representar o robô, este contêm largura, comprimento e centro de movimento (ponto central entre as duas rodas).
ControleSensores	Responsável em atualizar a posição do robô e dos pontos que representão os obstáculos, de acordo com as leituras feitas pelos sensores.
Posinfo	Responsável por conter as informações de uma posição do robô.
SensorIR	Responsável por representar um sensor IR do robô.

verListener, ServerSender, ServerReceiver e Message. Na Tabela 3 estão descritas as classes deste pacote.

2.5.4 Pacote *gui*

Este pacote consiste em toda a interface gráfica do sistema e conta com as seguintes classes: JanelaConexao e JanelaPrincipal. Na Tabela 4 estão descritas as classes deste pacote.

Tabela 3: Pacote *comunicacao*

Classe	**Descrição**
ClientConnection	Responsável por efetuar a gerência da conexão do cliente (estação base) com o servidor (robô).
ClientReceiver	Responsável por receber mensagens de um host de uma conexão.
ClientSender	Responsável por enviar mensagens ao host de uma conexão.
Server	Responsável gerenciar o servidor (robô).
ServerListener	Responsável por escutar as novas conexões de clientes.
ServerSender	Responsável por enviar mensagens ao host de uma conexão.
ServerReceiver	Responsável por receber mensagens de um host de uma conexão.
Message	Contém uma mensagem a ser enviada por um Sender.

Tabela 4: Pacote *gui*

Classe	**Descrição**
JanelaConexao	Janela com as informações e configurações da conexão com o Bellator.
JanelaPrincipal	Janela principal da interface gráfica da estação base.

3 DIAGRAMA DE BLOCOS DO HARDWARE

Na figura 3 mostra-se o diagrama de blocos do sistema embarcado e suas conexões com o restante do robô. A seguir está também uma descrição para cada um dos blocos da placa de circuito impresso do sistema embarcado.

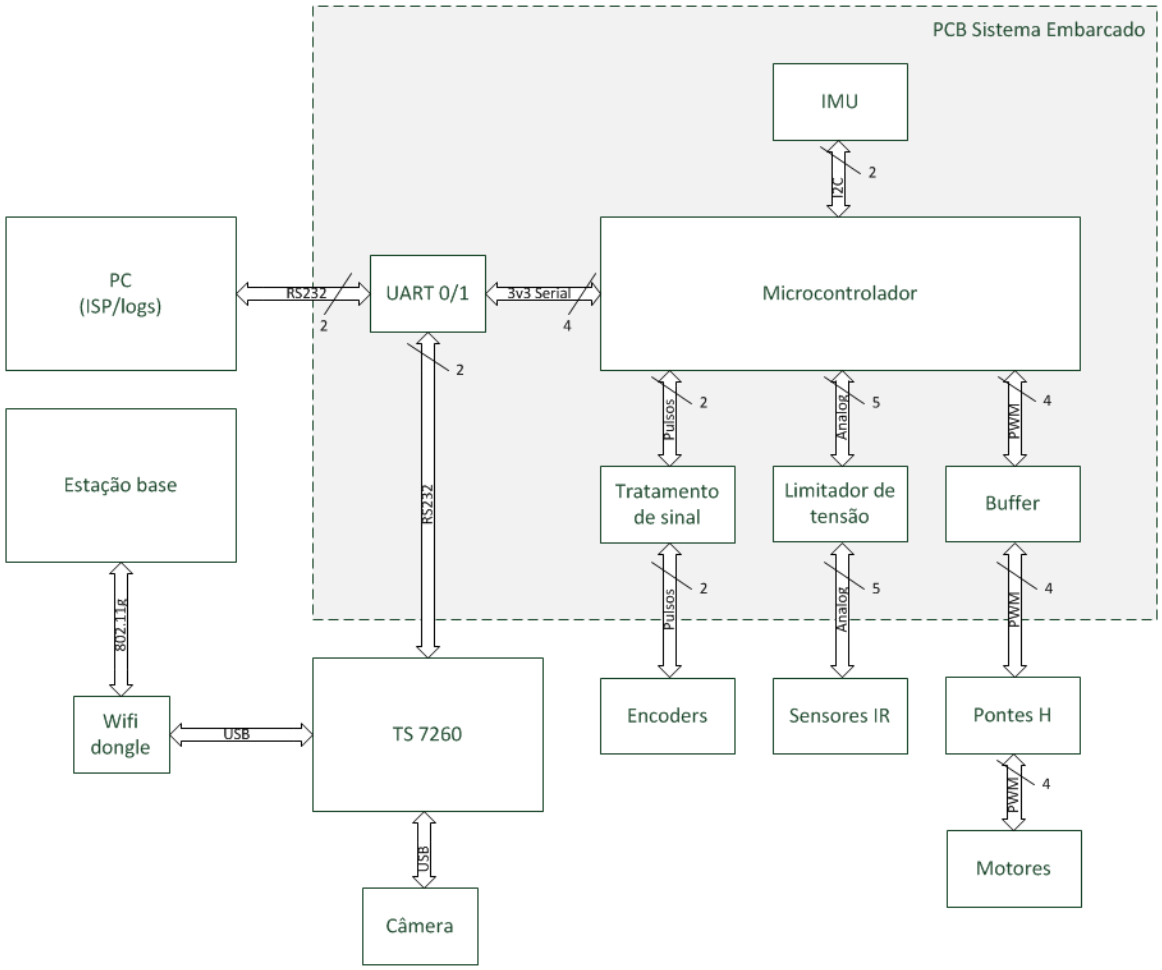

Figura 3: Diagrama de blocos do hardware

1. Microcontrolador: Este bloco fará a leitura dos sensores: encoders, infra-vermelhos, acelerômetro e giroscópios. Além disso possui a implementação do protocolo de comunicação para interação com o linux embarcado da placa TS-7260.

2. UART 0/1: Responsável por ajustar os níveis de tensão para comunicação serial no padrão RS-232 com a placa TS-7260.
3. Buffer: Responsável por fornecer corrente e elevar os níveis de tensão de saída do microcontrolador de 3,3V para 5,0V. Esse buffer é conectado às pontes H já existentes no robô.
4. IMU: possui o acelerômetro e o giroscópio e se comunicará com o microcontrolador por meio do protocolo I2C.
5. Limitador de tensão: Necessário pois os sinais de saída dos sensores de infravermelho que já existem no robô não estão limitados em 5V, podendo a saída ultrapassar 5,0V e danificar o microcontrolador.
6. Tratamento de sinal: Composto por um filtro RC passa baixas e um schmitt trigger para remover qualquer falha que possa ocorrer na geração dos pulsos no encoder. A frequência de corte do filtro pode ser obtida pela velocidade máxima que o robô pode atingir, que foi suposta em 1 m/s (como apresentado nos requisitos de hardware).