

MAPEAMENTO DE AMBIENTES COM O ROBO BELLATOR

Primeiro entregável

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

Requisitos funcionais e não funcionais

Requisitos funcionais

Estação base mostra mapa 2D do robô e dos obstáculos detectados - RF1

O usuário pode salvar o mapa - RF2

O usuário pode carregar o mapa - RF3

Estação base mostra a imagem captada pela webcam - RF4

O usuário pode movimentar o robô - RF5

O usuário pode parar o robô - RF6

O usuário pode estabelecer conexão entre o robô e a estação base - RF7

Requisitos não funcionais

O robô deve enviar vídeo em imagem colorida para a estação base - RNF1

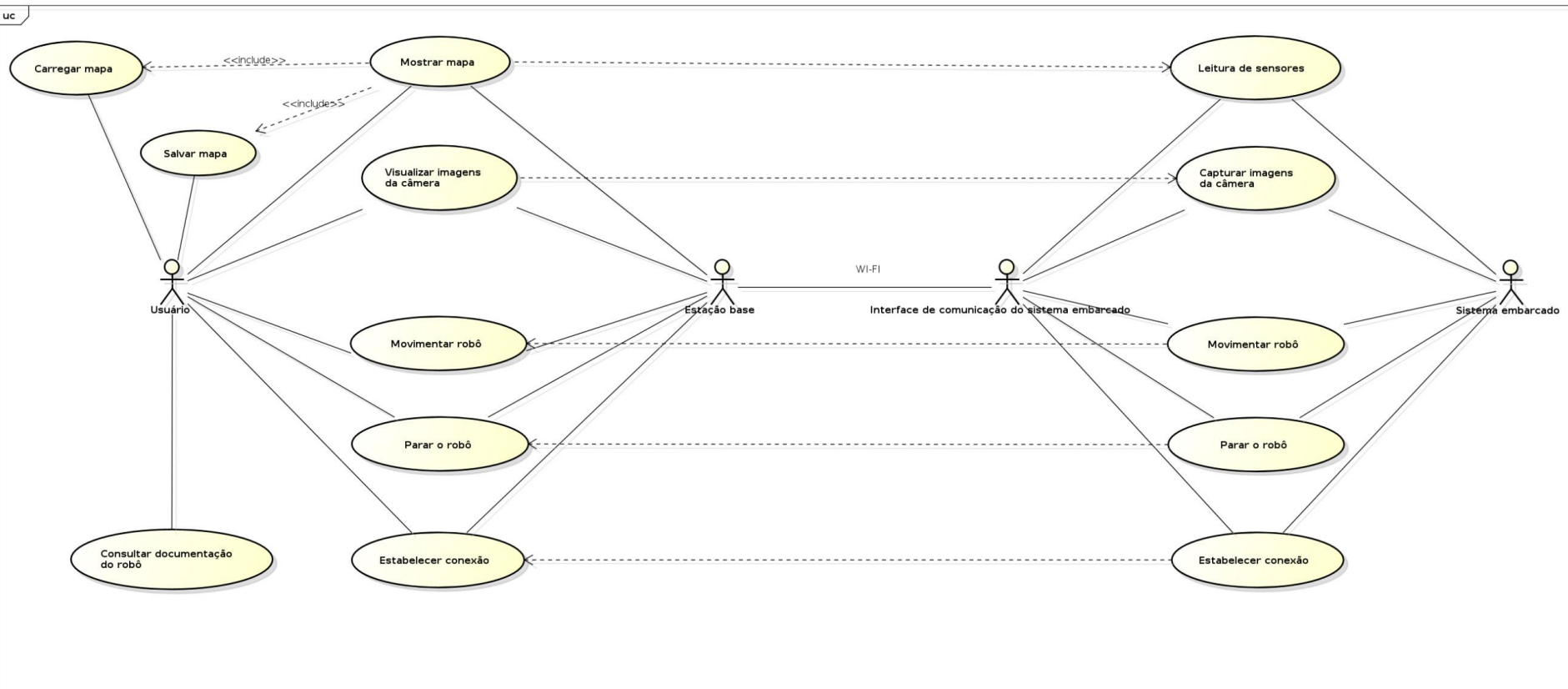
O robô deve transmitir os dados de vídeo captados pela câmera em tempo real - RNF2

Casos de uso

- Mostrar mapa - UC1
- Salvar mapa - UC2
- Carregar mapa - UC3
- Leitura de sensores - UC4
- Capturar imagens da câmera - UC5
- Visualizar imagens da câmera - UC6
- Movimentar robô - UC7
- Parar o robô - UC8
- Estabelecer conexão - UC9
- Consultar documentação do robô - UC10

Diagrama de casos de uso

uc

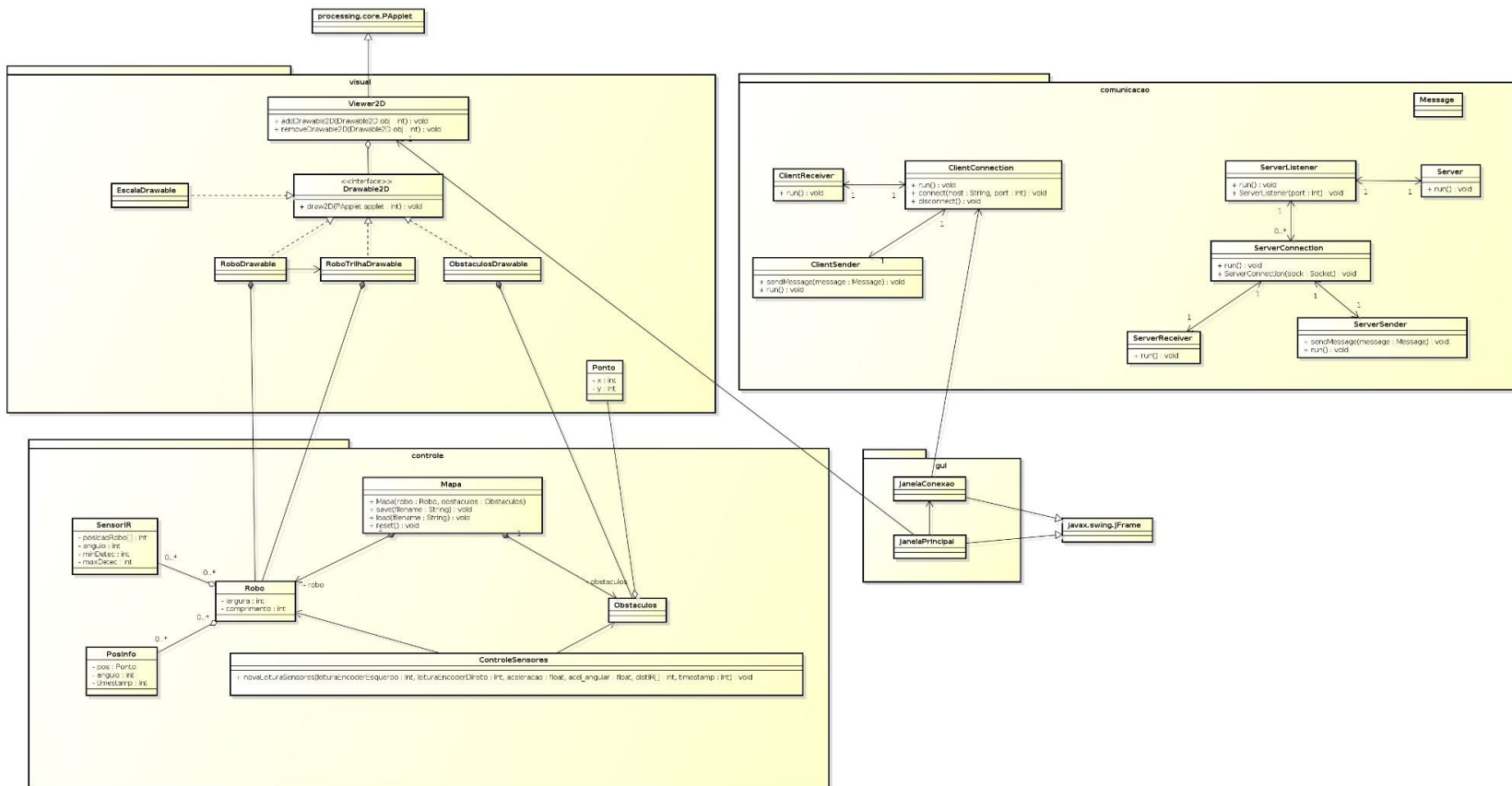


Classes do sistema

As classes foram divididas em 4 pacotes principais:

- ***visual***: desenho do mapa 2D
- ***gui***: interface gráfica
- ***controle***: gerencia as informações do robô
- ***comunicacao***: gerencia a comunicação (estação base - robô)

Diagrama de classes



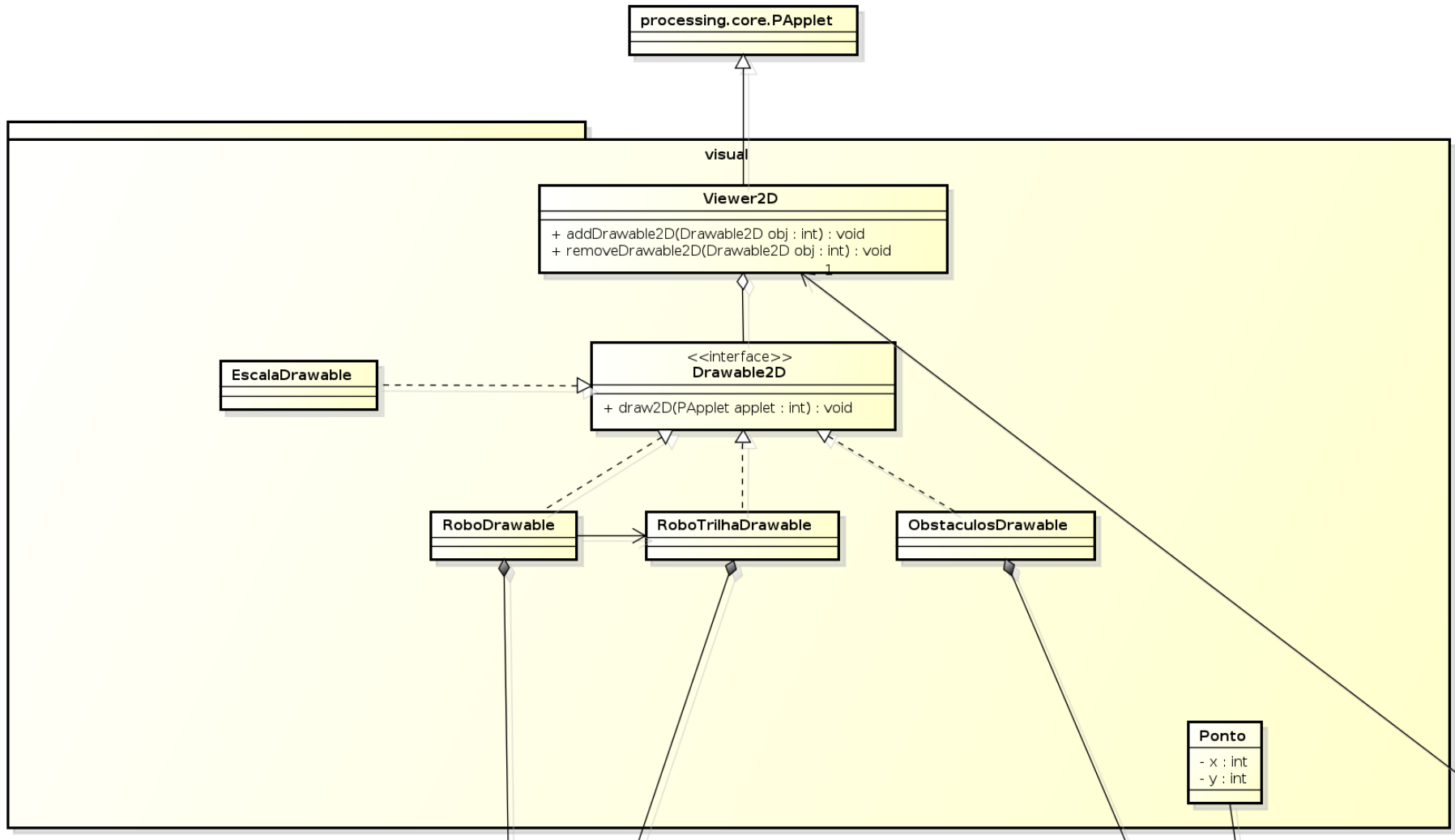
Pacote *visual*

- **Viewer2D**: visualizador de objetos 2D.
- **Drawable2D**: representa objetos que podem ser desenhados pelo visualizador 2D.
- **RoboDrawable**: desenha o robô.
- **RoboTrilhaDrawable**: desenha a trilha percorrida pelo robô.
- **ObstaculosDrawable**: desenha os obstaculos detectados pelo robô.

Pacote *visual*

- **EscalaDrawable**: desenha a escala do mapa.
- **Ponto**: representa um ponto cartesiano (x,y)

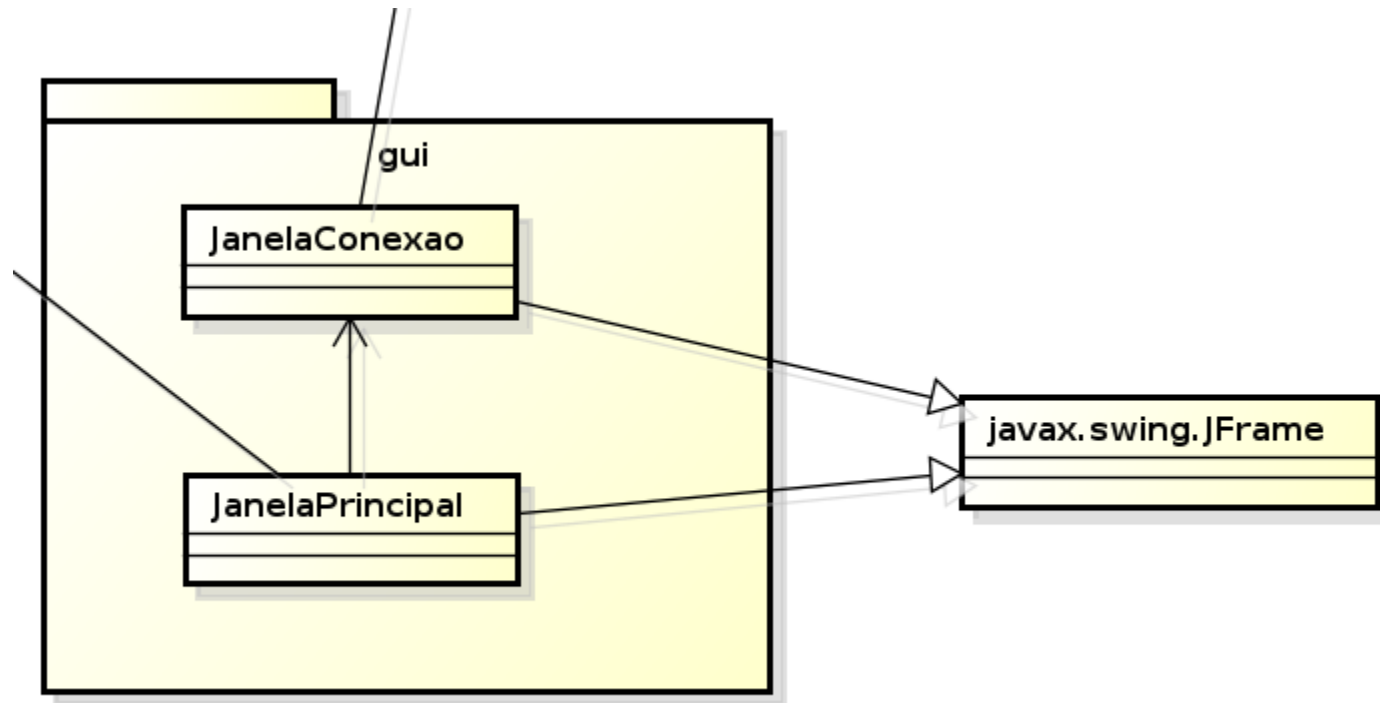
Pacote *visual*



Pacote *gui*

- **JanelaPrincipal**: janela base do programa.
- **JanelaConexao**: mostra configurações da conexão.

Pacote *gui*



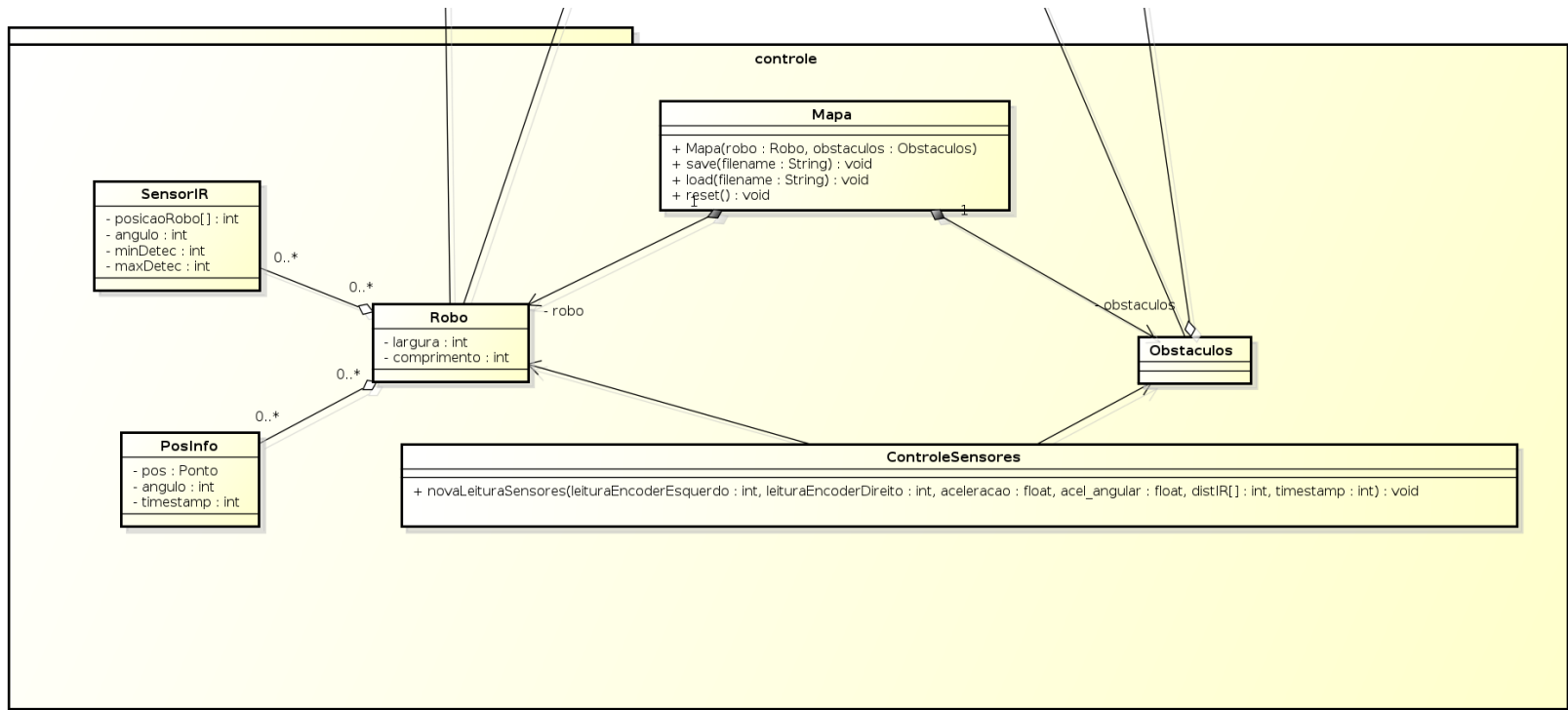
Pacote *controle*

- **Robo:** contém informações do robô.
- **SensorIR:** representa um sensor IR do robô.
- **PosInfo:** representa uma posição pelo qual o robô passou.
- **Obstaculos:** contém as informações de obstáculos detectados pelo robô.

Pacote *controle*

- **Mapa:** contém as informações essenciais do mapa; É capaz de salvá-las e carregá-las.
- **ControleSensores:** recebe leituras dos sensores do robô e atualiza informações do mapa.

Pacote controle



Pacote *comunicacao*

- **ClientConnector**: thread responsável por realizar e gerenciar uma conexão com o robô.
- **ClientSender**: thread responsável por enviar mensagens ao robô.
- **ClientReceiver**: thread responsável por receber mensagens do robô.
- **Message**: mensagem que pode ser enviada e recebida.

Pacote *comunicacao*

- **Server**: classe principal de gerenciamento do servidor (robô).
- **ServerListener**: thread responsável por escutar novas conexões.
- **ServerConnection**: thread responsável por gerenciar uma conexão com a estação base.
- **ServerSender**: thread responsável por enviar mensagens à estação base.
- **ServerReceiver**: thread responsável por receber mensagens da estação base.

Pacote comunicacao

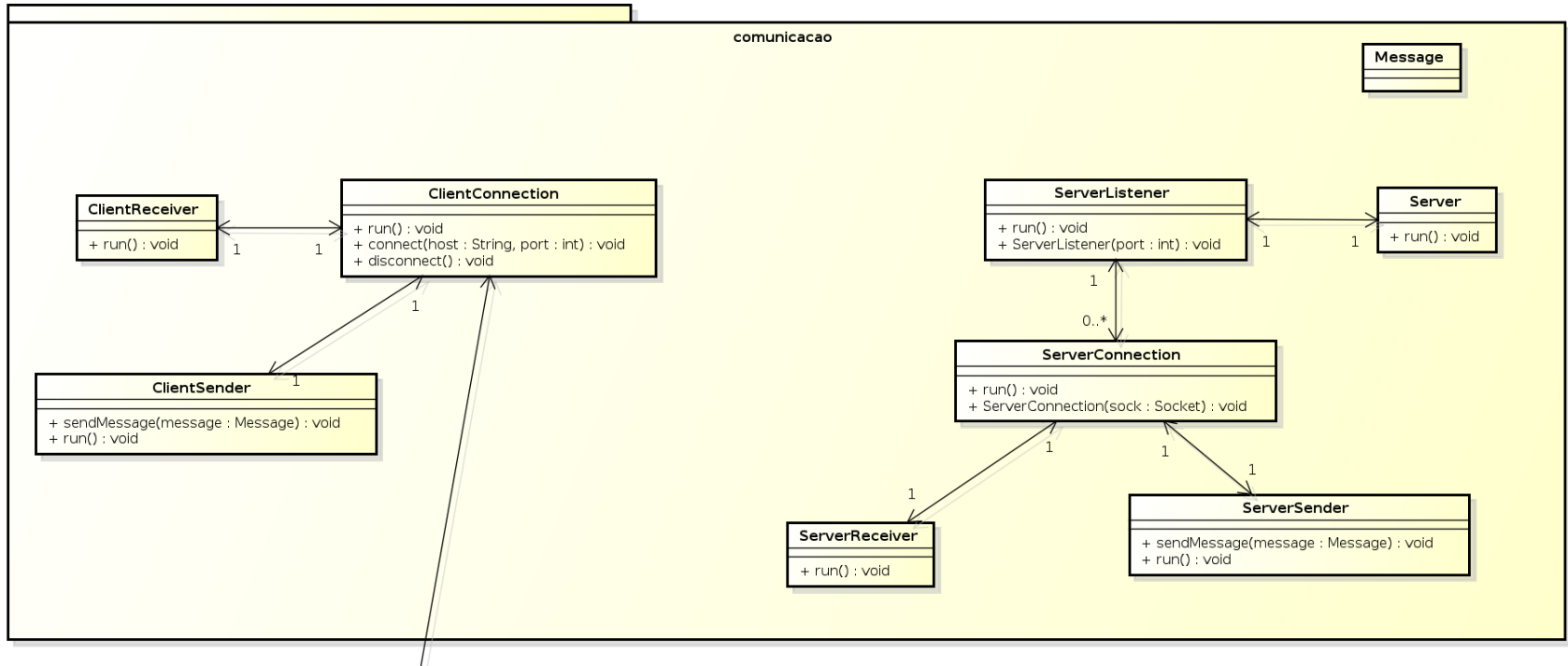


Diagrama de classes

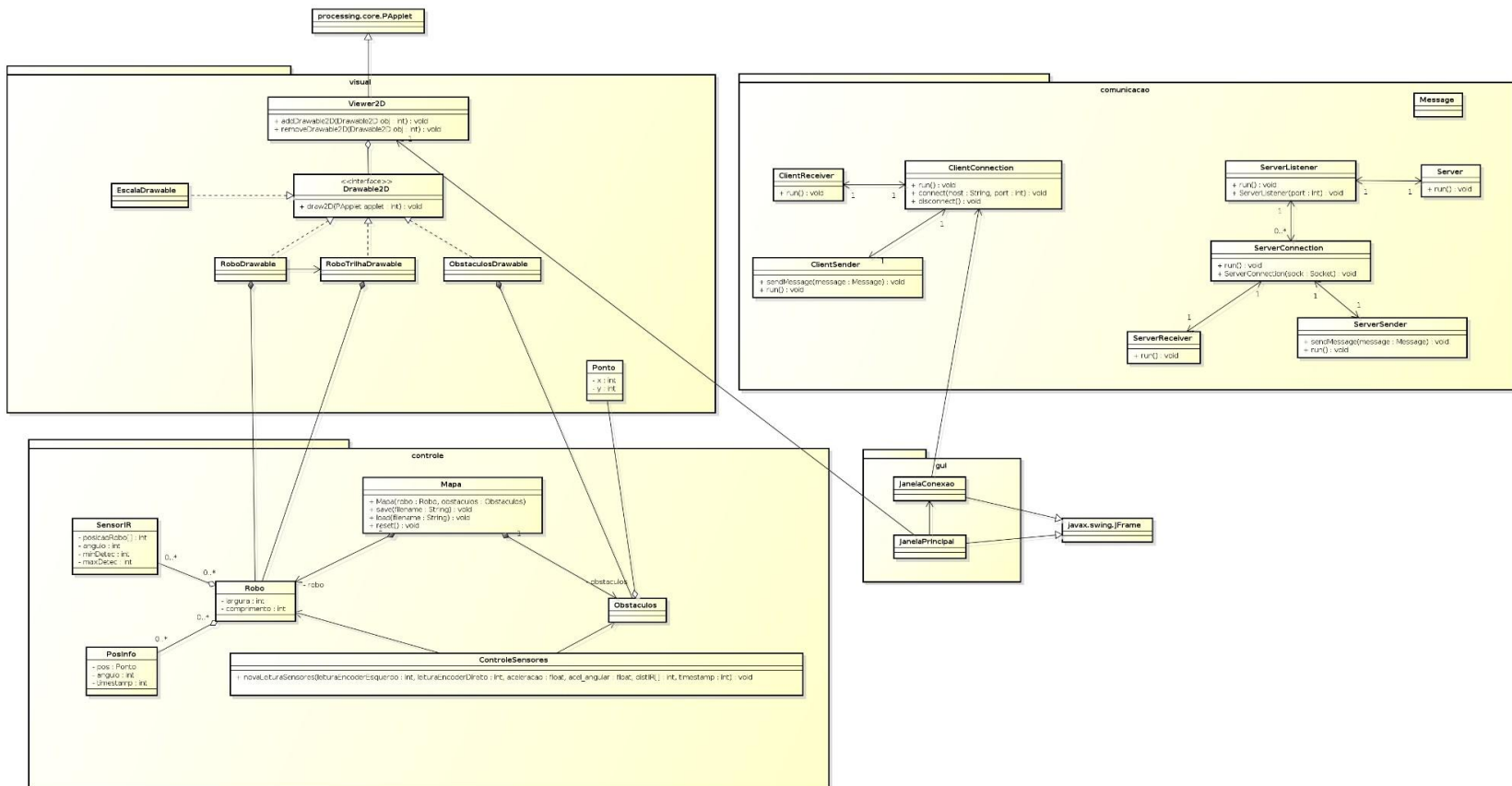


Diagrama de blocos do hardware

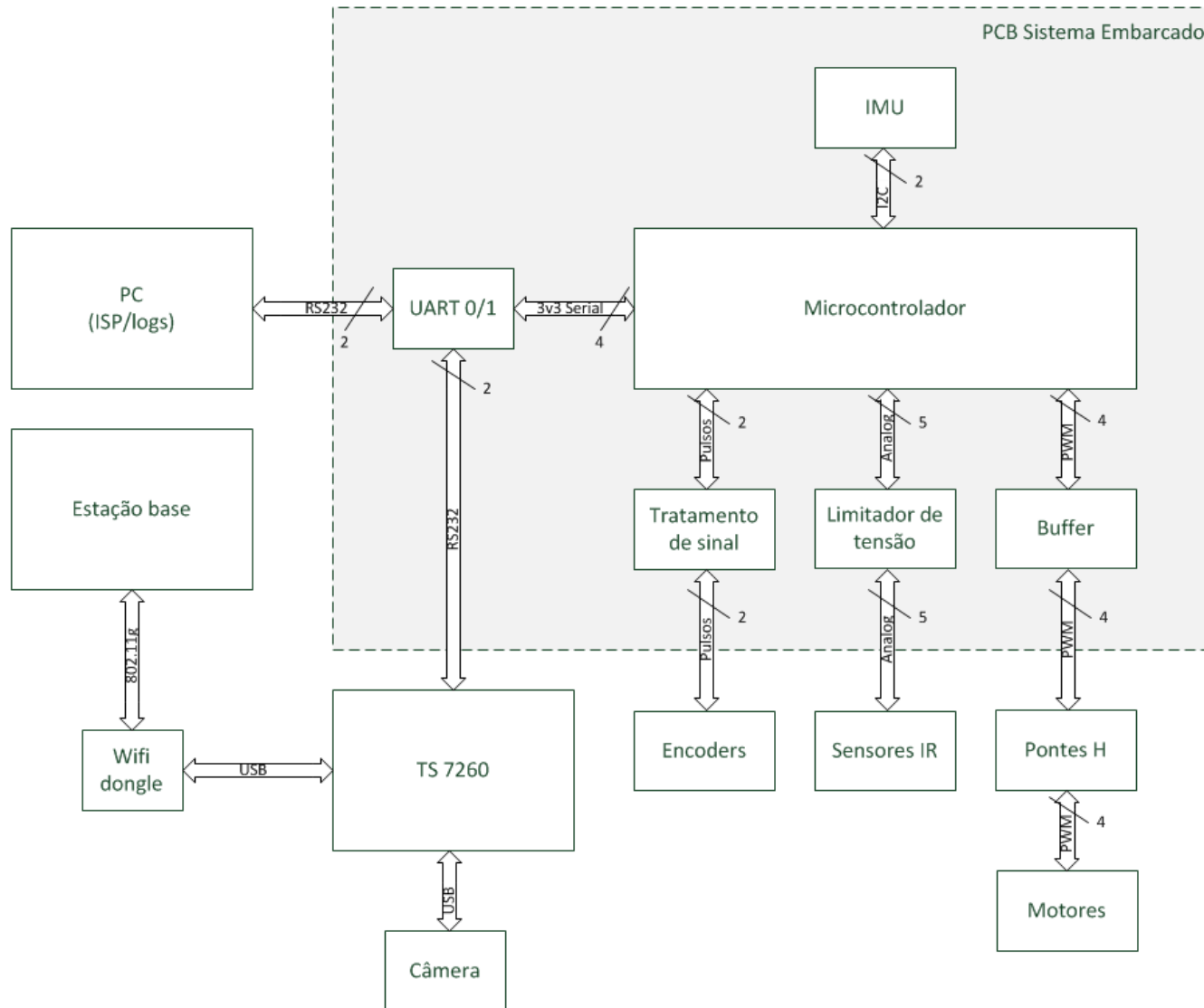


Diagrama de blocos do hardware

- Microcontrolador:
 - Leitura de sensores;
 - Comunicação com TS-7260;
- UART 0/1:
 - Elevação dos níveis de tensão para o padrão RS232
- Buffer:
 - Elevar níveis de tensão para 5V;
 - Fornecer corrente;

Diagrama de blocos do hardware

- IMU:
 - Acelerômetro;
 - Giroscópio;
- Limitador de tensão:
 - Limitar saída dos sensores IR em 5V;
- Tratamento de sinal:
 - Filtro RC;
 - Schmitt trigger.

Calendário do Projeto

- Calendário Atualizado com os Deliverables e suas datas
- Mais dias para atividades como implementações entre hardware/software e testes necessários
- Calendário e repositório de documentos já atualizados com documentos já feitos para primeiro deliverable
- Caminho crítico contém maior margem de erro para atrasos

