

RADAR ULTRASÔNICO MILITAR - MARINHA DO BRASIL (F LIBERAL / F-43)

Adriano Carvalho

Luiz Carlos

Thiarlleson

Cristiano Roberto

Marllom Moraes

Victor Correa

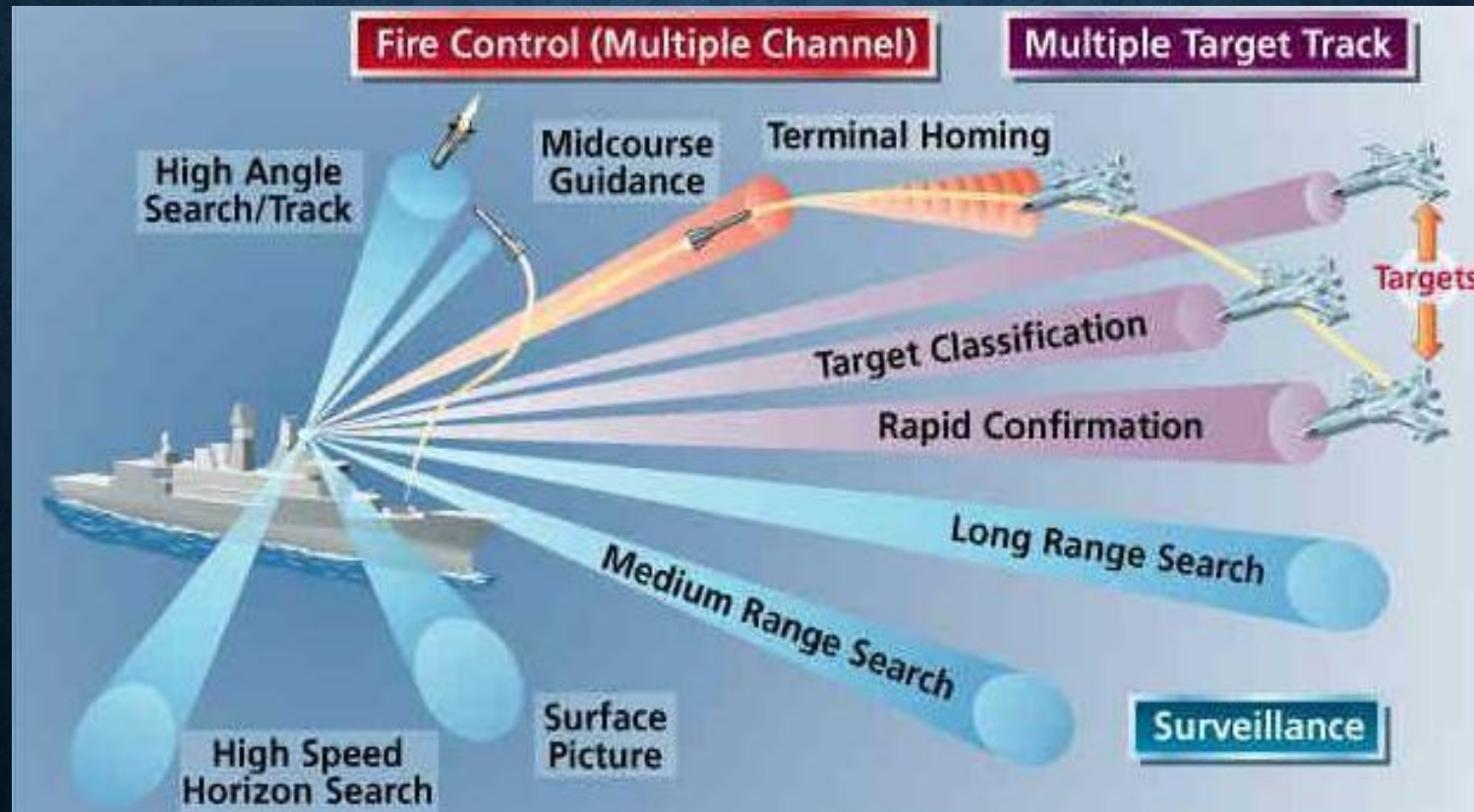
1 INTRODUÇÃO

- A Fragata Liberal (F-43) é uma fragata da Classe Niterói, da Marinha do Brasil. Fruto do "Programa de Renovação e Ampliação de Meios Flutuantes" da Marinha, concebido na década de 1970, que previa a construção de seis fragatas da Classe Niterói, foi a quarta a ser iniciada.



2 PROBLEMA

- Os radares do tipo Selex Orion RTN-30X estão desatualizados.
- Atualização para um moderno sistema de detecção de mísseis com RADARES ULTRASSÔNICOS.



3 OBJETIVO DO PROJETO

- I. Rastreamento de objetos em uma cobertura de 180 graus contínua.
- II. A rotação com cerca de 60 rpm.
- III. Menor razão de alarmes falsos e maior precisão de acompanhamento.
- IV. Capacidade de acompanhar alvos a mais de 40 cm; acompanhar 1 a 5 alvos.
- V. O alcance máximo de 2 m.
- VI. Interface gráfica de fácil entendimento e usabilidade para usuários com um mínimo de suficiência em informática básica.

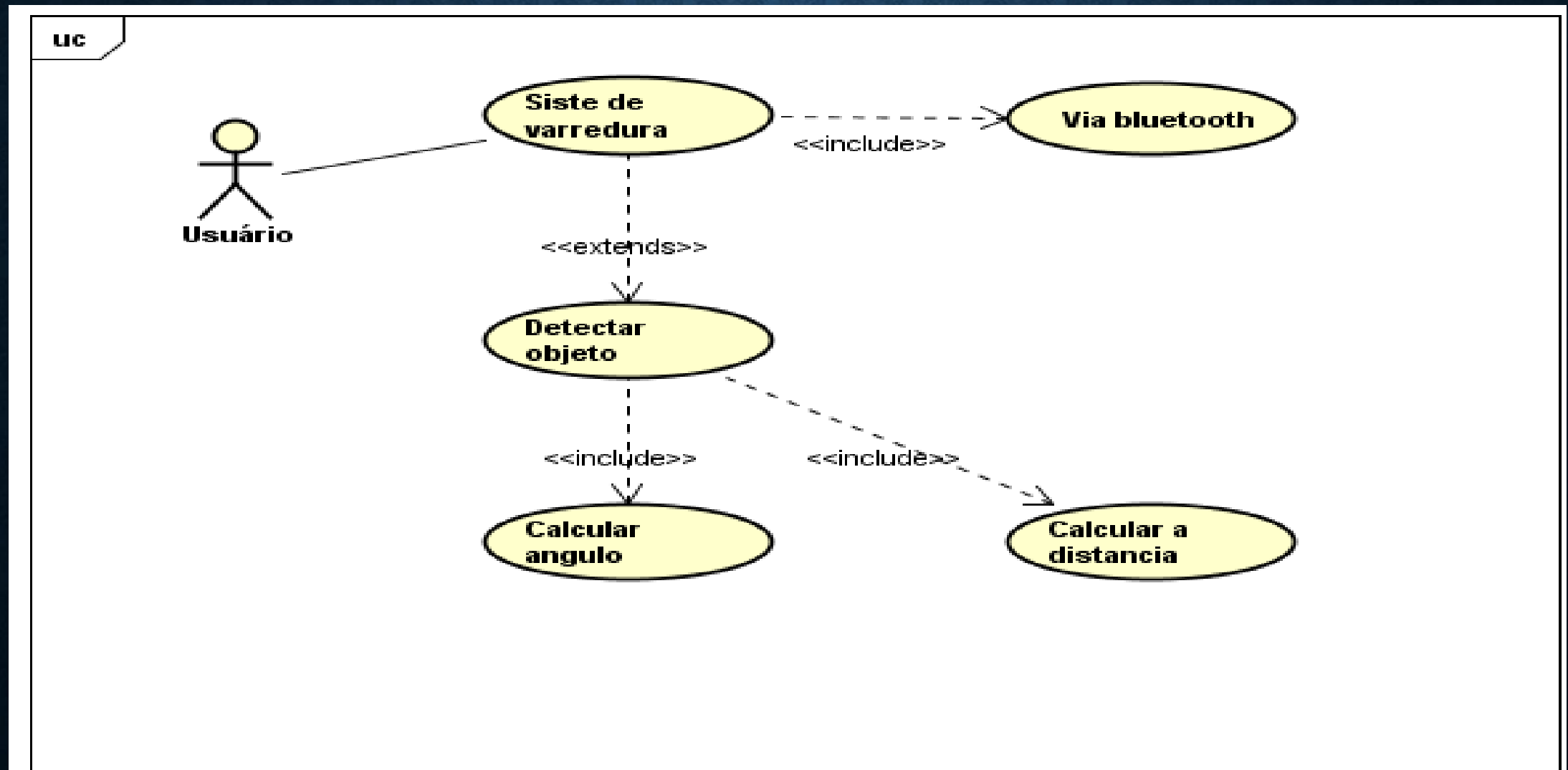
4 FUNDAMENTAÇÃO TEÓRICA

- Os radares de varredura eletrônica navais estão disponíveis a mais de 20 anos com o AEGIS americano.
- A parte mais crítica de um sistema de guerra antiaérea (AAW) naval é conter ataques de saturação.
- Estes ataques incluem aeronaves e mísseis antinavio vindo de várias direções coordenadamente, para vencer as defesas.

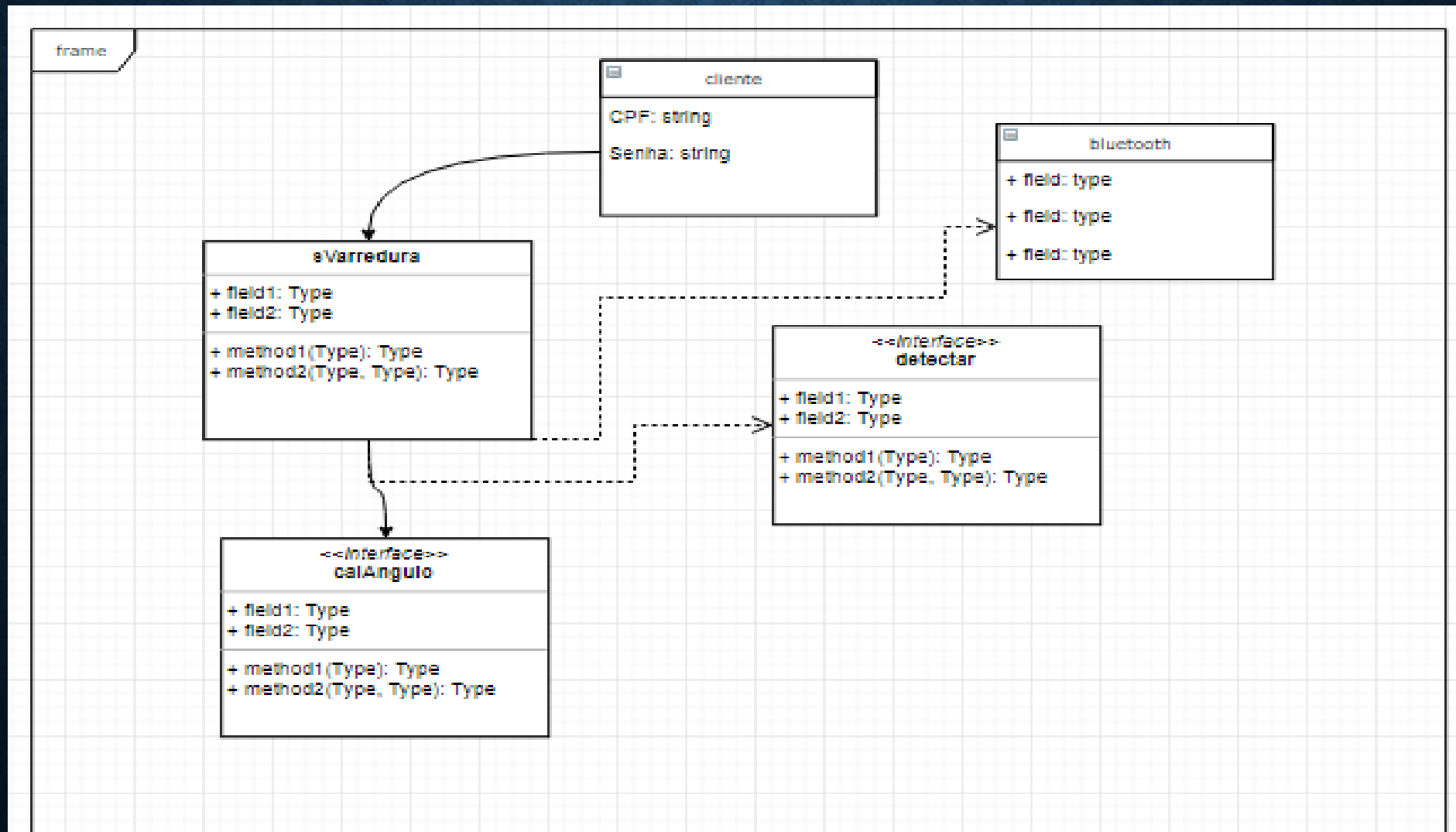
4 FUNDAMENTAÇÃO TEÓRICA

- As antenas convencionais de radar fazem este acompanhamento relacionando os ecos sucessivos de cada alvo.
- Esta informação é feita no modo TWS "track-while-scan" e é feita com quantos alvos o computador do sistema permitir.
- Quanto maior a razão de varredura do radar, maior a quantidade de informação do alvo. A razão de dados é proporcional a rotação da antena.

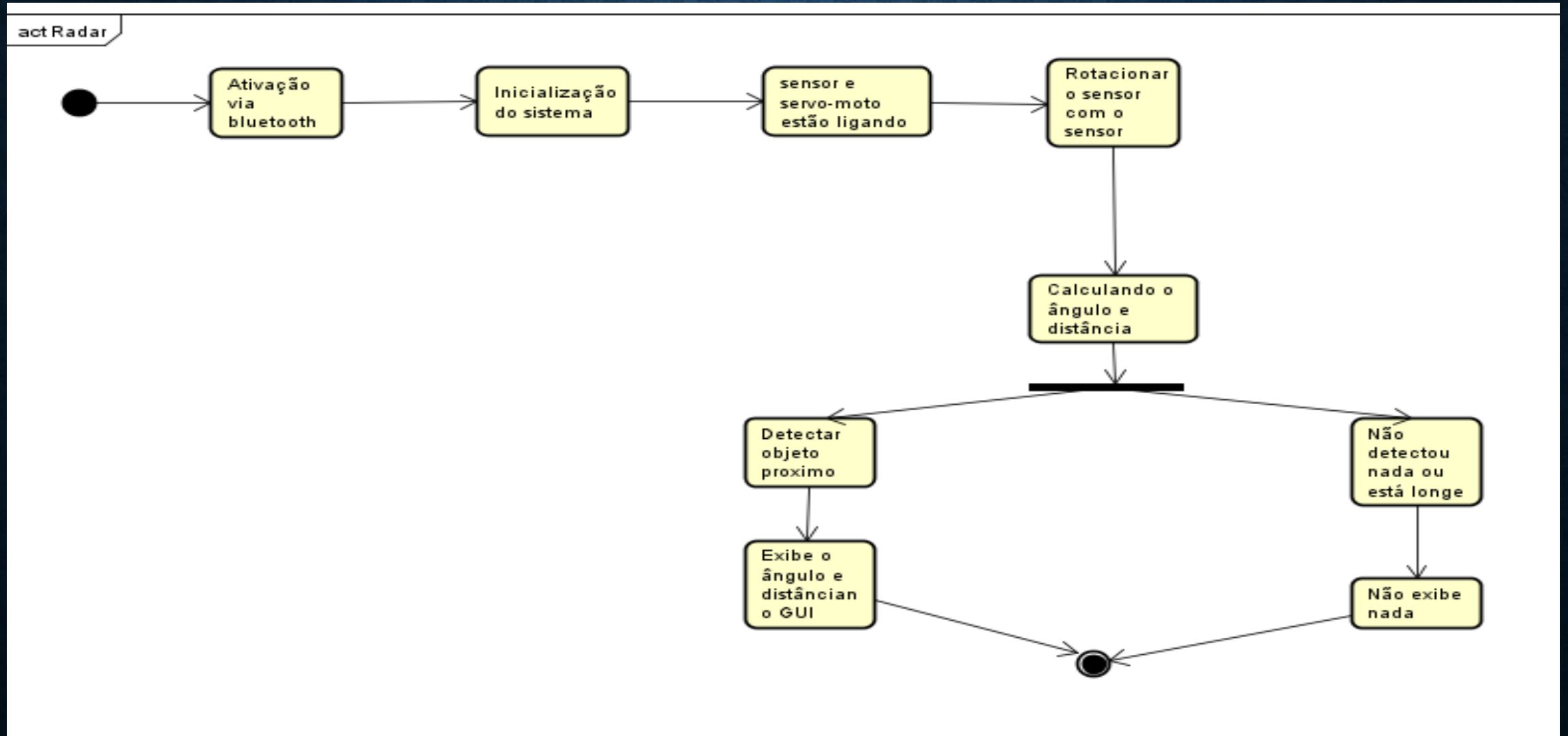
5 CASO DE USO



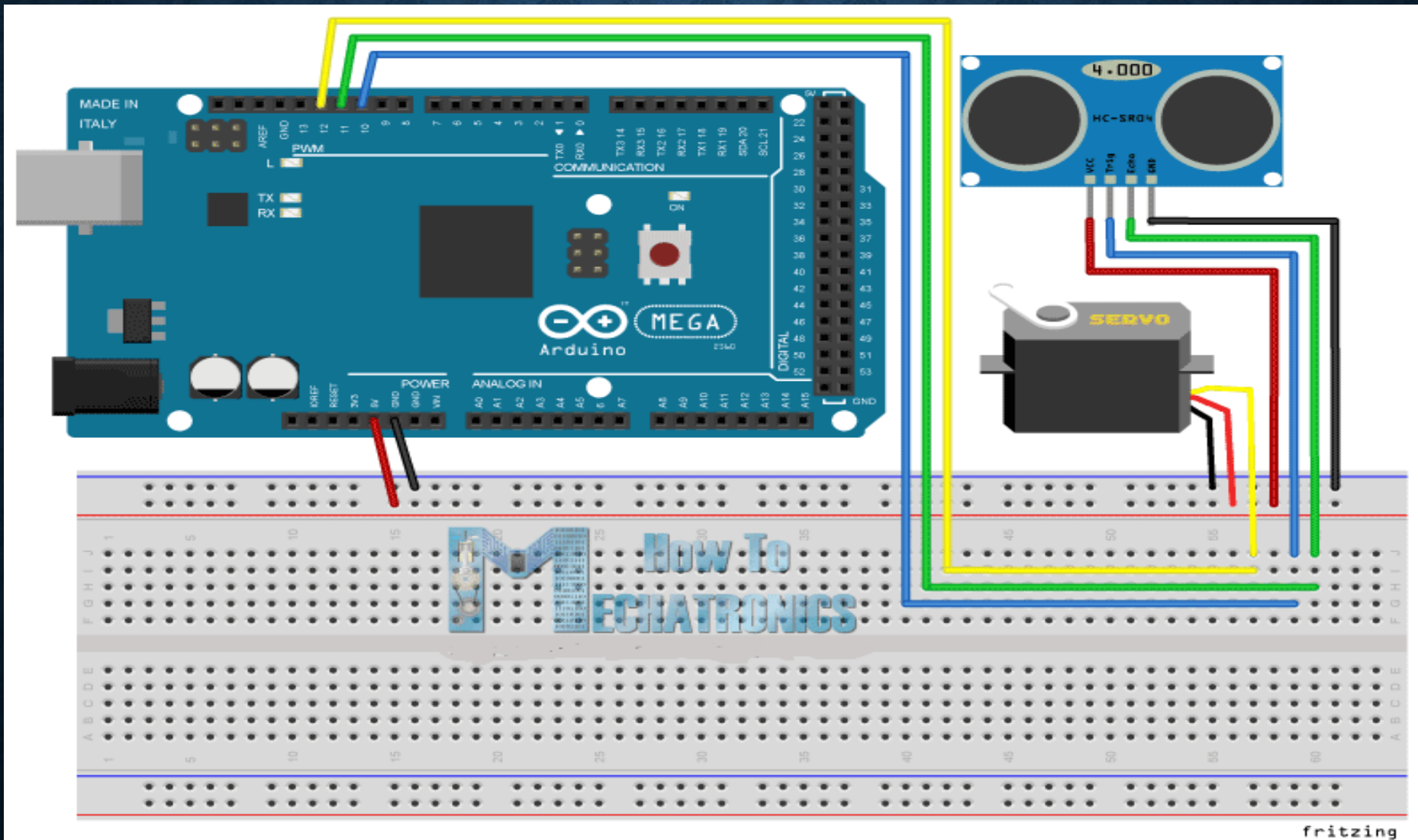
6 DIAGRAMA DE CLASSES



7 DIAGRAMA DE ATIVIDADES



8 PROJETO SIMULAÇÃO NO FRITZING



8 PROJETO

ARDUINO INTEGRADO COM PROCESSING

```
Arduino_Radar_Project | Arduino 1.8.5
File Edit Sketch Tools Help
Arduino_Radar_Project
#include <Servo.h>

const int trigPin = 10;
const int echoPin = 11;

long duration;
int distance;

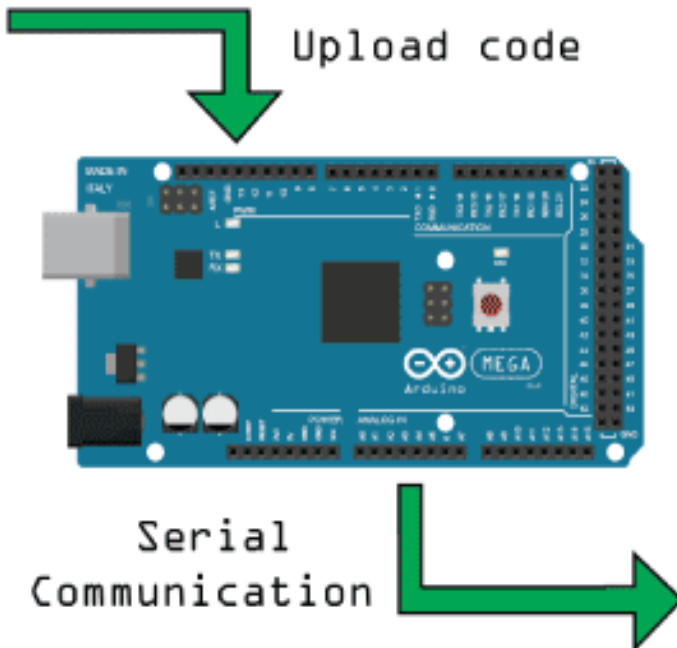
Servo myServo;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
  myServo.attach(12);
}

void loop() {

  for(int i=15;i<=165;i++){
    myServo.write(i);
    delay(30);

    distance = calculateDistance();
  }
}
```



```
Arduino_Radar_Project | Processing 2.2.1
File Edit Sketch Tools Help
Arduino_Radar_Project
import processing.serial.*; // imports library for serial communication
import java.awt.event.*; // imports library for reading the data
import java.io.IOException;

Serial myPort; // defines Object Serial

String angle="";
String distance="";
String data="";
String noObject;
float pixoDistance;
int iAngle, iDistance;
int index1=0;
int index2=0;
PFont arcFont;

void setup() {

  size(1920, 1080);
  smooth();
  myPort = new Serial(this, "COM4", 9600); // starts the serial communication
  myPort.bufferUntil('.'); // reads the data from the serial port up to the
  arcFont = loadFont("OCR_A48Extended-39.uxb");
}

void draw() {

  fill(0, 245, 31);
  textFont(arcFont);
  noStroke();
  fill(0, 4); // semi-transparent white
  rect(0, 0, width, height);
  fill(0, 245, 31);
  drawRadar();
  drawLine();
  drawObject();
  drawText();
}
```

8 PROJETO

INTERFACE GRÁFICA DO APLICATIVO



8 PROJETO

PROGRAMAÇÃO EM BLOCO DO APLICATIVO

The image displays a block-based programming interface with two main panels: **Blocks** and **Viewer**.

Blocks Panel:

- Built-in:** Control, Logic, Math, Text, Lists, Colors, Variables, Procedures.
- Screen1:**
 - OrganizaçãoHorizontal
 - Pintura1
 - Legenda1
 - OrganizaçãoHorizontal
 - Imagem1
 - OrganizaçãoHorizontal
 - Botão1

Viewer Panel:

The code is organized into three event-driven blocks:

- do set EscolheLista1 . Elements to ClienteBluetooth1 . AddressesAndNames**
- when EscolheLista1 . AfterPicking**
 - do
 - if call ClienteBluetooth1 . Connect address EscolheLista1 . Selection
 - then call Notificador1 . ShowAlert notice Tentando conectar!
 - if ClienteBluetooth1 . IsConnected
 - then call Notificador1 . ShowAlert notice Conectado com sucesso!
 - else call Notificador1 . ShowAlert notice Erro ao conectar!
- when Botão1 . Click**
 - do
 - if ClienteBluetooth1 . IsConnected
 - then call ClienteBluetooth1 . SendText text A
 - else call Notificador1 . ShowAlert notice Conecte ao dispositivo primeiro!
- when Botão2 . Click**
 - do
 - if ClienteBluetooth1 . IsConnected
 - then call ClienteBluetooth1 . SendText text B
 - else call Notificador1 . ShowAlert notice Conecte ao dispositivo primeiro!

At the bottom of the Viewer panel, there is a status bar showing 0 warnings and 0 errors, with a **Show Warnings** button.

9 PROTÓTIPO

