```
/*    FASES DE PROCESSO                         Quando entrar?
                          Quando Sair?

      0 = tudo desligado                        Quando bliga for precionado e o
sistema estiver ligado        Quando bliga for precionado e o sistema estiver
desligado
      1 = Bombeamento para TQ002                Quando sair da fase 0
                          Quando o TQ2 estiver cheio
      2 = Valvula  Aberta                       Quando sair da fase 1
                          Apos 5 segundos
      3 = Misturador ligado                     Quando sair da fase 2
                          Apos 10 segundos
      4 = Decantação                            Quando sair da fase 3
                          Apos 10 segundos
      5 = Bombeamento para TQ001                Quando sair da fase 4
                          Quando TQ1 estiver cheio

*/

//inclui bibliotecas
#include <Servo.h>

//declara contantes
Servo VF001;

//declara portas
#define bliga 10
#define misturador 12
#define b1 13
#define b2 14


//configurações
int TempoDeAberturaDaValvula = 5;            //segundos
int TempoDeFloculacao = 10;                  //segundos
int TempoDeDecantacao = 10;                  //segundos
int TempoPulsosB2 = 2;                       //segundos
int SensorMaisAlto = 4;

int TempoDeAberturaDaValvulaAUX = TempoDeAberturaDaValvula;
//segundos
int TempoDeFloculacaoAUX = TempoDeFloculacao;
//segundos
int TempoDeDecantacaoAUX = TempoDeDecantacao;
//segundos
int TempoPulsosB2AUX = TempoPulsosB2;
//segundos
int SensorMaisAltoAUX = SensorMaisAlto;

#define fechado 50                             //valvula VF001
#define aberto 0                              //valvula VF001
#define LIGADO LOW
#define DESLIGADO HIGH
#define QuantidadeDeTanques 2
```

```
#define QuantidadeDeSensores 4

//declara variaveis
int on = 0;
int fase = 0;
bool stB2 = DESLIGADO;
int TQ[QuantidadeDeTanques];
int nivel[QuantidadeDeTanques][QuantidadeDeSensores] = {
  //Minimo >> Maximo
  {2, 3, 4, 5}, //Tanque 1
  {6, 7, 8, 9} //Tanque 2
};
long int temp = 0;

void setup() {
  //Inicia serial
  Serial.begin(9600);
  delay (500);
  //declara entradas
  pinMode(bliga, INPUT_PULLUP);
  for (int a = 0; a < QuantidadeDeTanques; a++) {
    for (int b = 0; b < QuantidadeDeSensores; b++) {
      pinMode(nivel[a][b], INPUT);
    }
  }
  //declara saidas
  pinMode(13,OUTPUT);
  pinMode(b1, OUTPUT);
  pinMode(b2, OUTPUT);
  pinMode(misturador, OUTPUT);
  VF001.attach(11);
  //Regula saidas para posição inicial
  digitalWrite(b1, DESLIGADO);
  digitalWrite(b2, DESLIGADO);
  digitalWrite(misturador, DESLIGADO);
  VF001.write(fechado);
}

void loop() {
  LeNivel();
  RecebeParametros();
  EnviaInformacoes();
  //liga e desliga o PETA
  if (digitalRead(bliga) == LOW) {
    if(on==0){on=1;}
    if(on==1){on=0;}
  }
  if (on == 0) {
      fase = 0;
  }
  //Verifica as fases
  switch (fase) {
    case 0:                              //Tudo Desligado
      if (TQ[0] >= SensorMaisAlto && on == 1) {
```

```
        fase = 1;
      }
      break;
    case 1:                               //Bombeamento para TQ002
      if (TQ[1] >= SensorMaisAlto) {
        fase = 2;
        temp = millis() + (TempoDeAberturaDaValvula * 1000) ; // SETOU TEMPO
      }
      break;
    case 2:                               //Abre Valvula
      if (millis() >= temp) {
        fase = 3;
        temp = millis() + ( TempoDeFloculacao * 1000) ; // SETOU TEMPO
      }
      break;
    case 3:                               //Floculação
      if (millis() >= temp) {
        fase = 4;
        temp = millis() + (TempoDeDecantacao * 1000) ; // SETOU TEMPO
      }
      break;
    case 4:                              //Decantação
      if (millis() >= temp) {
        fase = 5;
        temp = millis(); // SETOU TEMPO
      }
      break;
    case 5:                               //Bombeamento para TQ001
      if (TQ[0] >= SensorMaisAlto) {

        //Escolha o modo: (Deixe um comentado e o outro não)
        // Ciclo apos ciclo
        //fase = 0;

        // Ciclo e pausa
        on = 0;
      }
      if (millis() >= temp) {
        stB2 = !stB2;
        temp = millis() + (TempoPulsosB2 * 1000) ; // SETOU TEMPO
      }
      break;
  }
  AtualizaSaidas();
}

int LeNivel() {
  for (int a = 0; a < QuantidadeDeTanques; a++) {
    TQ[a] = 0;
    for (int b = 0; b < QuantidadeDeSensores; b++) {
      if (digitalRead(nivel[a][b]) == HIGH) {
        TQ[a] = (b + 1); // b + 1 representa de 1 a QuantidadeDeSensores
      }
    }
```

```
  }
}

int AtualizaSaidas() {
  switch (fase) {
    case 0:                             //Tudo Desligado
      digitalWrite(b1, DESLIGADO);
      digitalWrite(b2, DESLIGADO);
      digitalWrite(misturador, DESLIGADO);
      VF001.write(fechado);
      break;
    case 1:                             //Bombeamento para TQ002
      digitalWrite(b1, LIGADO);
      digitalWrite(b2, DESLIGADO);
      digitalWrite(misturador, DESLIGADO);
      VF001.write(fechado);
      break;
    case 2:                             //Abre Valvula
      digitalWrite(b1, DESLIGADO);
      digitalWrite(b2, DESLIGADO);
      digitalWrite(misturador, DESLIGADO);
      VF001.write(aberto);
      break;
    case 3:                             //Floculação
      digitalWrite(b1, DESLIGADO);
      digitalWrite(b2, DESLIGADO);
      digitalWrite(misturador, LIGADO);
      VF001.write(fechado);
      break;
    case 4:                            //Decantação
      digitalWrite(b1, DESLIGADO);
      digitalWrite(b2, DESLIGADO);
      digitalWrite(misturador, DESLIGADO);
      VF001.write(fechado);
      break;
    case 5:                             //Bombeamento para TQ001
      digitalWrite(b1, DESLIGADO);
      digitalWrite(b2, stB2);
      digitalWrite(misturador, DESLIGADO);
      VF001.write(fechado);
      break;
  }
}

  int EnviaInformacoes() {
  //envia os valores pro supervisorio
  Serial.print((String)"page0.ON.val=" + on);
  envia();
  Serial.print((String)"page0.TQ001.val=" + TQ[0]);
  envia();
  Serial.print((String)"page0.TQ002.val=" + TQ[1]);
  envia();
  Serial.print((String)"page0.fase.val=" + fase);
  envia();
```

```
  Serial.print((String)"page2.TValvula.val=" + TempoDeAberturaDaValvulaAUX);
  envia();
  Serial.print((String)"page2.TDecantacao.val=" + TempoDeDecantacaoAUX);
  envia();
  Serial.print((String)"page2.TFloculacao.val=" + TempoDeFloculacaoAUX);
  envia();
  Serial.print((String)"page2.TPB2.val=" + TempoPulsosB2AUX);
  envia();
  Serial.print((String)"page2.NivelMax.val=" + SensorMaisAltoAUX);
  envia();
  if(digitalRead(b2)==0){
    Serial.print("page0.p1.pic=7");
    envia();
  }else{
    Serial.print("page0.p1.pic=6");
    envia();
  }
}

int RecebeParametros() {
  //Recebe Paremetros do Supervisório
  if(Serial.available()){
   delay(500);
   int parametro, valor;
   parametro = Serial.read();
   valor = Serial.read();
    switch (parametro) {
      case 0:
        on = valor;
      break;
      case 1:
        TempoDeAberturaDaValvulaAUX = valor;
      break;
      case 2:
        TempoDeDecantacaoAUX = valor;
      break;
      case 3:
        TempoDeFloculacaoAUX = valor;
      break;
      case 4:
        TempoPulsosB2AUX = valor;
      break;
      case 5:
        SensorMaisAltoAUX = valor;
      break;
    }
  }
    if(fase<=1)                                                    // Se
quiser atualizar os parametros apenas em um novo ciclo, caso contrario, comente
essa linha
    {
      TempoDeAberturaDaValvula = TempoDeAberturaDaValvulaAUX;         //segundos
      TempoDeFloculacao = TempoDeFloculacaoAUX;                       //segundos
      TempoDeDecantacao = TempoDeDecantacaoAUX;                       //segundos
```

```
      TempoPulsosB2 = TempoPulsosB2AUX;                                    //segundos
      SensorMaisAlto = SensorMaisAltoAUX;
    }
}

int envia()
{
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);
}
```