```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

#define pinPOWER 0
#define pinCURRENT 1

const float sensCURRENT = 0.133;
const int numSAMPLES = 1200;
const int rawBASIC = 101;
const int maxVOLTS = 5.;
const int digitalSLICES = 1023.;

byte txStatus = false;

LiquidCrystal_I2C lcd(0x27,16,2);
/*
 * la función siguiente devuelve la relación entre la lectura digital de la
tensión
 * de salida directa del puente bidireccional de RF y la potencia detectada en
el
 * medidor de potencia en los tramos de excitación (5W=>60W)
 */
float ratioRawWatt(int rw) {
  float rt;
  if (rw <= 500) {                      //Con 5W - raw 499 PA 36W
    rt = 13.86;
  } else if (rw > 501 && rw <= 562) {   //Con 10W - raw 562 PA 58W
    rt = 9.69;
  } else if (rw > 563 && rw <= 594) {   //Con 15W - raw 594 PA 81W
    rt = 7.33;
  } else if (rw > 595 && rw <= 614) {   //Con 20W - raw 614 PA 112W
    rt = 5.48;
  } else if (rw > 615 && rw <= 628) {   //Con 25W - raw 628 PA 139W
    rt = 4.52;
  } else if (rw > 629 && rw <= 645) {   //Con 30W - raw 645 PA 174W
    rt = 3.71;
  } else if (rw > 646 && rw <= 658) {   //Con 35W - raw 658 PA 200W
    rt = 3.29;
  } else if (rw > 659 && rw <= 670) {   //Con 40W - raw 670 PA 230W
    rt = 2.91;
  } else if (rw > 671 && rw <= 683) {   //Con 45W - raw 683 PA 270W
    rt = 2.53;
  } else if (rw > 684 && rw <= 695) {    //Con 50W - raw 695 PA 310W
    rt = 2.24;
  } else if (rw > 696 && rw <= 702) {   //Con 55W - raw 702 PA 330W
    rt = 2.13;
  } else if (rw > 502) {                //Con 60W - raw 708 PA 350
    rt = 2.02;
  }
  return rt;
}

int readSamples(int pin) {
  int maxRaw = 0;
  int raw;

  for (int i = 1; i <= numSAMPLES; i++) {
    raw = analogRead(pin);
    if (raw > maxRaw) {
      maxRaw = raw;
    }
  }
  return maxRaw;
}

float calcVolts(float raw) {
  return (raw * maxVOLTS) / digitalSLICES;
}
```

```
float calcCurrent(float volts) {
  return volts / sensCURRENT;
}

float calcPower(int raw) {
  return raw / ratioRawWatt(raw);
}

void displayCurrent(float amps) {
  lcd.setCursor(0, 0);
  lcd.print((String) amps + " Amperios  ");
}

void displayPower(float watts) {
  lcd.setCursor(0, 1);
  lcd.print((String) int(watts) + " Vatios      ");
}

void clearDisplay() {
  lcd.setCursor(0, 0);
  lcd.print("                ");

  lcd.setCursor(0, 1);
  lcd.print("                ");
}

void setup() {
  Serial.begin(9600);
  Serial.flush();

  lcd.init();
  lcd.backlight();
}

void loop() {
  int rawCurrent = readSamples(pinCURRENT);
  int rawPower = readSamples(pinPOWER);

  Serial.print("RAW Básica= ");
  Serial.println(rawCurrent);
  delay(1000);

  if (rawPower > 0) {
    float voltsCurrent = calcVolts(rawCurrent - rawBASIC);
    float ampsCurrent = calcCurrent(voltsCurrent);
    float wattsPower = calcPower(rawPower);

    displayCurrent(ampsCurrent);
    displayPower(wattsPower);
  } else {
    clearDisplay();
  }

  delay(1000);


}
```