```
/*
 ------ARDUINO WATER PUMP POWER CONTROL-------

 ARDUINO MICRCONTROLLERS COURSE 2022
 UTN - EDUTECH

 LUCAS ARGENTO
 Lucasargento7@gmail.com

 ------------------------------
*/
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Define constant values
#define OFF false;
#define ON true;

// Ultrasonic sensor
const int pingPin = 7;
const int echoPin = 6;

// Water pump relay
const int relePin = 8;

// As a user you want to tweek this variable acording
to your water tank and instalation capabilities
const long criticalLevel = 5;

// LCD display
LiquidCrystal_I2C lcd(0x27,20,4);

// Arduino Logic
void setup() {
```

```
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  digitalWrite(relePin,LOW);
}

void loop() {
   checkWaterLevel();
}

void informUser(long waterLevel, bool status){
  /*
   Prints water level and pump status to an LCD display
  */
  lcd.setCursor(0, 0);
  lcd.print("Nivel:");
  lcd.print(waterLevel);
  lcd.print("cm         ");
  lcd.setCursor(0,1);
  lcd.print("Bomba:");
  lcd.setCursor(6,1);

  if(status){
    lcd.print("on          ");
  } else {
    lcd.print("off!Critico");
  }

}

void checkWaterLevel(){
  /*
  Measures water level (distance from the top to the
edge of the liquid) using an ultrasonic Hc-sr04
```

```
  sensor
  */
  long duration, level;
  bool pumpStatus;
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(pingPin, LOW);
  pinMode(echoPin, INPUT);

  duration = pulseIn(echoPin, HIGH);
  level = microsecondsToCentimeters(duration);
  pumpStatus = analizeWaterLevel(level);
  informUser(level, pumpStatus);
  delay(300);
}

bool analizeWaterLevel(long level){
  /*
  Compares actual level with critical level. If wlvl
is < critical, return Off status.
  */
  if(level >= criticalLevel) {
    digitalWrite(relePin,LOW);
    return OFF;
  } else{
    digitalWrite(relePin,HIGH);
    return ON;
  }
}

long microsecondsToCentimeters(long microseconds) {
```

```
  /*
  Auxiliary function to calculate distance from time
estimated from the Hc-sr04 sensor
  */
  return microseconds / 29 / 2;
}
```