**Código fuente de: Programación para Arduino CubeSat**

**Lenguaje: Arduino**

**Fecha: 18/03/2018**

**Institución educativa: CBTis 168**

**Autor: Leonel Iván Fernández Carrillo**

static void smartdelay(unsigned long ms);

void TempHum\_satelite();

void Infrarojo\_satelite();

void Corriente\_satelite();

void gy\_87();

void gps\_satelite();

void Calidad\_aire();

void Rayos\_UV();

void FotoRes();

void Panel\_solar();

void BMP180\_satelite();

//GPS

#include <SoftwareSerial.h>

#include <TinyGPS.h>

TinyGPS gps;

int year;

byte month, day, hour, minute, second, hundredths;

unsigned long chars;

unsigned short sentences, failed\_checksum;

//Sensor de humedad y temperatura

#include <DHT.h>

#include "DHT.h"

#define DHTPIN 2

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

//Sensor de corriente

float Sensibilidad=0.139;

float offset=0.100;

//Tarjeta SD

#include <SD.h>

File SDFILE;

//GY-87

#include "I2Cdev.h"

#include "MPU6050.h"

#include "HMC5883L.h"

static const char LED = 6;

static const float ACCEL\_SENS = 16384.0;

static const float GYRO\_SENS = 131.0;

#include <SFE\_BMP180.h>

SFE\_BMP180 bmp180;

double PresionNivelMar=1013.25;

void setup() {

Serial.begin(115200);

Serial1.begin(9600);

Serial2.begin(57600);

//temp hum

dht.begin();

Serial.print("Iniciando SD ...");

if (!SD.begin(4)) {

Serial.println("No se pudo inicializar");

return;

}

Serial.println("inicializacion exitosa");

//GY-87

boolean state = HIGH;

unsigned int count = 0;

while (!Serial && (count < 30) )

{

delay(200);

state = !state;

count++;

}

Wire.begin();

accelgyro.initialize();

Serial.print("Testing Accel/Gyro... ");

Serial.println(accelgyro.test () ? "MPU6050 connection successful" : "MPU6050 connection failed");

accelgyro.setI2CBypassEnabled(true); // set bypass mode

mag.initialize();

Serial.print("Testing Mag... ");

Serial.println(mag.test () ? "HMC5883L connection successful" : "HMC5883L connection failed");

}

void loop() {

SDFILE = SD.open("Data.cvc", FILE\_WRITE);

if (SDFILE) {

} else {

}

gps\_satelite();

TempHum\_satelite();

Infrarojo\_satelite();

gy\_87();

BMP180\_satelite();

Calidad\_aire();

Rayos\_UV();

FotoRes();

Panel\_solar();

Corriente\_satelite();

SDFILE.close();

}

void gps\_satelite(){

smartdelay(500);

{

float latitude, longitude;

gps.f\_get\_position(&latitude, &longitude);

Serial.print("0;");

Serial.print(latitude,5);

Serial.print(",1;");

Serial.print(longitude,5);

Serial.print(",2;");

Serial.print(gps.f\_altitude());

Serial.print(",3;");

Serial2.print("0;");

Serial2.print(latitude,5);

Serial2.print(",1;");

Serial2.print(longitude,5);

Serial2.print(",2;");

Serial2.print(gps.f\_altitude());

Serial2.print(",3;");

SDFILE.print("0;");

SDFILE.print(latitude,5);

SDFILE.print(",1;");

SDFILE.print(longitude,5);

SDFILE.print(",2;");

SDFILE.print(gps.f\_altitude());

SDFILE.print(",3;");

gps.crack\_datetime(&year,&month,&day,&hour,&minute,&second,&hundredths);

Serial.print(gps.f\_course());

Serial.print(",4;");

Serial.print(gps.f\_speed\_kmph());

Serial.print(",5;");

Serial.print(gps.satellites());

Serial.print(",6;");

Serial2.print(gps.f\_course());

Serial2.print(",4;");

Serial2.print(gps.f\_speed\_kmph());

Serial2.print(",5;");

Serial2.print(gps.satellites());

Serial2.print(",6;");

SDFILE.print(gps.f\_course());

SDFILE.print(",4;");

SDFILE.print(gps.f\_speed\_kmph());

SDFILE.print(",5;");

SDFILE.print(gps.satellites());

SDFILE.print(",6;");

gps.stats(&chars, &sentences, &failed\_checksum);

}

}

static void smartdela(unsigned long ms){

unsigned long start = millis();

do

{

while (Serial1.available())

gps.encode(Serial1.read());

} while (millis() - start < ms);

}

void TempHum\_satelite(){

float h = dht.readHumidity();

float t = dht.readTemperature();

Serial.print(h);

Serial.print(",7;");

Serial.print(t);

Serial.print(",8;");

Serial2.print(h);

Serial2.print(",7;");

Serial2.print(t);

Serial2.print(",8;");

SDFILE.print(h);

SDFILE.print(",7;");

SDFILE.print(t);

SDFILE.print(",8;");

}

void Infrarojo\_satelite(){

int measure = analogRead(0);

Serial.print(measure);

Serial.print(",9;");

Serial2.print(measure);

Serial2.print(",9;");

SDFILE.print(measure);

SDFILE.print(",9;");

}

void Corriente\_satelite(){

float I=get\_corriente(200);

Serial.print(",22;");

Serial.print(I,3);

Serial.println();

Serial2.print(",22;");

Serial2.print(I,3);

Serial2.println();

SDFILE.print(",22;");

SDFILE.print(I,3);

SDFILE.println();

}

float get\_corriente(int n\_muestras){

float voltajeSensor;

float corriente=0;

for(int i=0;i<n\_muestras;i++)

{

voltajeSensor = analogRead(A1) \* (5.0 / 1023.0);

corriente=corriente+(voltajeSensor-2.5)/Sensibilidad;

}

corriente=corriente/n\_muestras;

return(corriente);

}

void Calidad\_aire(){

int aire = analogRead(2);

Serial.print(",20;");

Serial.print(aire);

Serial2.print(",20;");

Serial2.print(aire);

SDFILE.print(",20;");

SDFILE.print(aire);

}

void Rayos\_UV(){

Serial.print(",24;");

SDFILE.print(",24;");

Serial2.print(",24;");

int UV\_Val\_RAMBAL;

int UV;

UV\_Val\_RAMBAL = analogRead(3);

if(UV\_Val\_RAMBAL < 10) {

Serial2.print("0");

SDFILE.print("0");

Serial.print("0"); }

else { if(UV\_Val\_RAMBAL < 46) {

Serial2.print("1");

SDFILE.print("1");

Serial.print("1"); }

else { if(UV\_Val\_RAMBAL < 65) {

Serial2.print("2");

SDFILE.print("2");

Serial.print("2"); }

else { if(UV\_Val\_RAMBAL < 83) {

Serial2.print("3");

SDFILE.print("3");

Serial.print("3"); }

else { if(UV\_Val\_RAMBAL < 103) {

Serial2.print("4");

SDFILE.print("4");

Serial.print("4"); }

else { if(UV\_Val\_RAMBAL < 124) {

Serial2.print("5");

SDFILE.print("5");

Serial.print("5"); }

else { if(UV\_Val\_RAMBAL < 142) {

Serial2.print("6");

SDFILE.print("6");

Serial.print("6"); }

else { if(UV\_Val\_RAMBAL < 163) {

Serial2.print("7");

SDFILE.print("7");

Serial.print("7"); }

else { if(UV\_Val\_RAMBAL < 180) {

Serial2.print("8");

SDFILE.print("8");

Serial.print("8"); }

else { if(UV\_Val\_RAMBAL < 200) {

Serial2.print("9");

SDFILE.print("9");

Serial.print("9"); }

else { if(UV\_Val\_RAMBAL < 221) {

Serial2.print("10");

SDFILE.print("10");

Serial.print("10"); }

else { if(UV\_Val\_RAMBAL < 239) {

Serial2.print("11");

SDFILE.print("11");

Serial.print("11"); }

else {

Serial2.print("12");

SDFILE.print("12");

Serial.print("12"); }

}}}}}}}}}}}

}

void FotoRes(){

int fres = analogRead(4);

Serial.print(",21;");

Serial.print(fres);

Serial2.print(",21;");

Serial2.print(fres);

SDFILE.print(",21;");

SDFILE.print(fres);

}

void Panel\_solar(){

int solar = analogRead(5);

Serial.print(",22;");

Serial.print(solar);

Serial2.print(",22;");

Serial2.print(solar);

SDFILE.print(",22;");

SDFILE.print(solar);

}

void gy\_87(){

static unsigned long ms = 0;

static boolean state = HIGH;

if (millis() - ms > 100)

{

accelgyro.getMotion(&ax, &ay, &az, &gx, &gy, &gz);

Serial.print(ax/ACCEL\_SENS);

Serial.print(",10;");

Serial.print(ay/ACCEL\_SENS);

Serial.print(",11;");

Serial.print(az/ACCEL\_SENS);

Serial.print(",12;");

Serial.print(gx/GYRO\_SENS);

Serial.print(",13;");

Serial.print(gy/GYRO\_SENS);

Serial.print(",14;");

Serial.print(gz/GYRO\_SENS);

Serial.print(",15;");

SDFILE.print(ax/ACCEL\_SENS);

SDFILE.print(",10;");

SDFILE.print(ay/ACCEL\_SENS);

SDFILE.print(",11;");

SDFILE.print(az/ACCEL\_SENS);

SDFILE.print(",12;");

SDFILE.print(gx/GYRO\_SENS);

SDFILE.print(",13;");

SDFILE.print(gy/GYRO\_SENS);

SDFILE.print(",14;");

SDFILE.print(gz/GYRO\_SENS);

SDFILE.print(",15;");

Serial2.print(ax/ACCEL\_SENS);

Serial2.print(",10;");

Serial2.print(ay/ACCEL\_SENS);

Serial2.print(",11;");

Serial2.print(az/ACCEL\_SENS);

Serial2.print(",12;");

Serial2.print(gx/GYRO\_SENS);

Serial2.print(",13;");

Serial2.print(gy/GYRO\_SENS);

Serial2.print(",14;");

Serial2.print(gz/GYRO\_SENS);

Serial2.print(",15;");

mag.getHeading(&mx, &my, &mz);

Serial.print(mx);

Serial.print(",16;");

Serial.print(my);

Serial.print(",17;");

Serial.print(mz);

Serial.print(",18;");

SDFILE.print(mx);

SDFILE.print(",16;");

SDFILE.print(my);

SDFILE.print(",17;");

SDFILE.print(mz);

SDFILE.print(",18;");

Serial2.print(mx);

Serial2.print(",16;");

Serial2.print(my);

Serial2.print(",17;");

Serial2.print(mz);

Serial2.print(",18;");

float heading = atan2(my, mx);

}

}

void BMP180\_satelite(){

Serial.print(",18;");

Serial2.print(",18;");

SDFILE.print(",18;");

char status;

double T,P,A;

status = bmp.startTemperature();

if (status != 0)

{

delay(status);

status = bmp.getTemperature(T);

if (status != 0)

{

status = bmp.startPressure(3);

if (status != 0)

{

delay(status);

status = bmp.getPressure(P,T);

if (status != 0)

{

Serial.print(T);

Serial.print(",19;");

Serial.print(P);

A= bmp180.altitude(P,PresionNivelMar);

Serial.print(",23;");

Serial.print(A);

Serial2.print(T);

Serial2.print(",19;");

Serial2.print(P);

A= bmp180.altitude(P,PresionNivelMar);

Serial2.print(",23;");

Serial2.print(A);

SDFILE.print(T);

SDFILE.print(",19;");

SDFILE.print(P);

A= bmp180.altitude(P,PresionNivelMar);

SDFILE.print(",23;");

SDFILE.print(A);

}

}

}

}

}

**Código fuente de: Aplicación de rastreo satelital**

**Lenguaje: C Sharp**

**Fecha: 13/03/2018**

**Institución educativa: CBTis 168**

**Autor: Axel Iván Valdez Reyes**

**Google.cs**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class Google : MonoBehaviour {

string url2 = "localhost/cubesat/gps\_datos.php";

public RawImage img;

public float lat;

public float lon;

public int zoom;

int h;

int w;

public enum mapType

{

roadmap, satellite, hybrid, terrain

}

public mapType mapSelected;

public int scale;

IEnumerator map() {

WWW ww = new WWW (url2);

yield return ww;

string id = ww.text;

Debug.Log (id);

if (id != "") {

int ide = id.IndexOf ("/");

string dato = id.Substring (ide + 1);

dato = id.Substring (0,ide);

float.TryParse (dato, out lat);

url = "https://maps.googleapis.com/maps/api/staticmap?center=" + lat + "," + lon +

"&zoom=" + zoom + "&size=" + w + "x" + h + "&scale=" + scale

+"&maptype=" + mapSelected +

"&markers=color:blue%7Clabel:S%7C40.702147,-74.015794&markers=color:green%7Clabel:G%7C40.711614,-74.012318&markers=color:red%7Clabel:C%7C40.718217,-73.998284&key=AIzaSyDlFrZdnHsh2efepBB8OiAkAdephH4Av4o";

WWW fotm = new WWW (url);

yield return fotm;

img.texture = fotm.texture;

img.SetNativeSize();

}

StartCoroutine ("wait");

}

IEnumerator wait() {

yield return new WaitForSeconds (0.05f);

StartCoroutine ("map");

}

// Use this for initialization

void Start () {

url2 = Ruta.texto;

img = gameObject.GetComponent <RawImage>();

h = Screen.height;

StartCoroutine ("map");

}

// Update is called once per frame

void Update () {

}

}

**Ruta.cs**

**Código fuente de: Depuración de datos**

**Lenguaje: Python**

**Fecha: 01/03/2018**

**Institución educativa: CBTis 168**

**Autor: Luis Fernando Caro Reyna**

# CubeSat 2018

# FIM168 II

import numpy as np

import mysql.connector

import serial

import time

import os

import math

class con:

def \_\_init\_\_(self, user="root", password="", database="", host="127.0.0.1"):

self.acc = {'user': user, 'password': password,

'database': database, 'host': host}

def est\_con(self):

try:

self.cnx = mysql.connector.connect(\*\* self.acc)

self.cursor = self.cnx.cursor()

print "Conexion establecida con ", self.acc["host"]

return True

except:

print "ERROR al conectar con ", self.acc["host"]

def des\_con(self):

self.cnx.close()

def ex\_ins(self, query):

try:

self.cursor.execute(query)

self.cnx.commit()

return True

except Exception as e:

print "ERROR: ", e

return False

def pros(d=[]):

val = []

for y in d[2]:

for z in range(0, len(d[2][y])):

if d[2][y][z] == 0 or d[2][y][z] == None or d[2][y][z] == "nan" or d[2][y][z] == d[1][y][z]:

if y != "Ace" and y != "Mag" and y != "Giro":

d[2][y][z] = 0

print "ERROR ENCONTRADO "

for i in range(0, len(val)):

A1 = d[val[i][0] - 1][val[i][1]][val[i][2]]

A2 = d[val[i][0] - 2][val[i][1]][val[i][2]]

vv = A1 + (A1 - A2)

print "Correcion en: ", val[i], " -> ", d[val[i][0]][val[i][1]][val[i][2]], " -> ", A1, "+ (", A1, " - ", A2, ") = ", vv

d[val[i][0]][val[i][1]][val[i][2]] = vv

print "Numero de correcciones: ", len(val)

return d

def valdatf1(a=""):

lon = len(d)

val = np.zeros(27)

valo = np.zeros(27)

for i in range(0, lon):

d[i] = d[i].split(";")

for i in range(0, lon):

if len(d[i]) == 2:

try:

if val[int(d[i][0])] != 1:

val[int(d[i][0])] = 1

else:

val[int(d[i][0])] = 0

val[int(d[i][0])] = 1

except:

print "ERROR en el ingreso de datos en la validacion"

cont = 0

print a

print val

for i in range(0, len(valo)):

if(val[i] == 1):

try:

valo[i] = float(d[cont][1])

except:

print "ERROR en cast a float"

cont += 1

else:

valo[i] = 0

print valo

dt = {

"Gps": [valo[0], valo[1], valo[2]],

"Curso": [valo[3]],

"Vel": [valo[4]],

"Sat": [valo[5]],

"Hum": [valo[6]],

"Temp": [valo[7], valo[18]],

"Infra": [valo[8]],

"Ace": [valo[9], valo[10], valo[11]],

"Giro": [valo[12], valo[13], valo[14]],

"Mag": [valo[15], valo[16], valo[17]],

"Pres": [valo[19]],

"CA": [valo[20]],

"IntLuz": [valo[23]],

"Corr": [valo[22]],

"Alt": [valo[21]]

}

print dt

def dift(t1, t2):

return t2 - t1

def varAltura(alt1, alt2):

return alt2 - alt1

def joules(alt):

return float((alt / 1000) \* float(9.81 \* 11))

def altBar(p):

# P=0.885^h en kilometros

p = p \* 0.0009869

h = math.log(p) / math.log(0.885)

return h

def \_\_main\_\_():

bd = con(database="cubesat", user="cubesat",

password="cubesat", host="192.168.0.100")

bdcon = bd.est\_con()

conser = False

try:

ser = serial.Serial('COM4', 57600)

print "Conexion con serial establecida "

except Exception as e:

print "ERROR al conectarse al puerto serial"

time.sleep(1)

clcn = 0

d = []

for i in range(0, 3):

a = ser.readline()

print a

d.append(valdatf1(a))

altini = d[0]["Alt"][0]

while bdcon:

inicio = time.time()

d[0] = d[1]

d[1] = d[2]

a = ser.readline()

d[2] = (valdatf1(a))

d = pros(d)

for i in range(0, 3):

print i, ": "

for j in d[i]:

print "\t", j, ": ", d[i][j]

joul = joules(d[2]["Alt"][0])

varalt = varAltura(altini, d[2]["Alt"][0])

vartemp = dift(d[2]["Temp"][0], d[2]["Temp"][1])

altPres = altBar(d[2]["Pres"][0])

print "\tJoules: ", joul

print "\tVarAlt: ", varalt

print "\tVarTemp: ", vartemp

print "\tPresBar: ", altPres

clcn += 1

print a, "\n"

query = "CALL InsertarDatos(" + str(d[2]["Gps"][0]) + "," + str(d[2]["Gps"][1]) + "," + str(d[2]["Alt"][0]) + "," + str(d[2]["Ace"][0]) + "," + str(d[2]["Ace"][1]) + "," + str(d[2]["Ace"][2]) + "," + str(d[2]["Giro"][0]) + "," + str(d[2]["Giro"][

1]) + "," + str(d[2]["Giro"][2]) + "," + str(d[2]["Mag"][0]) + "," + str(d[2]["Mag"][1]) + "," + str(d[2]["Mag"][2]) + "," + str(d[2]["Temp"][0]) + "," + str(d[2]["Temp"][1]) + "," + str(d[2]["CA"][0]) + "," + str(d[2]["IntLuz"][0]) + "," + str(d[2]["Hum"][0]) + "," + str(d[2]["Infra"][0]) + "," + str(d[2]["Pres"][0]) + "," + str(joul) + "," + str(vartemp) + "," + str(varalt) + "," + str(altPres) + "," + str(d[2]["Pres"][0]) + "," + str(d[2]["Corr"][0]) + ");"

print query

if bd.ex\_ins(query):

print "Datos insertados"

if bd.ex\_ins("INSERT INTO serial(Hra\_evento, Cadena) VALUES (NOW(), '" + a + "');"):

print "Serial isnertado"

fin = time.time()

tot = fin - inicio

print "Tiempo de ejecucion: ", tot

if clcn == 25:

os.system("cls")

clcn = 0

time.sleep(0)

if \_\_name\_\_ == '\_\_main\_\_':

\_\_main\_\_()

# made in 168

**Código fuente de: Sistema de graficacion**

**Lenguajes: PHP, javascript**

**Fecha: 4/03/2018**

**Institución educativa: CBTis 168**

**Autor: Eduardo Dávila Campos**

HTML

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="keyword" content="">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Fim-168</title>

<!-- start: Css -->

<link rel="stylesheet" type="text/css" href="asset/css/bootstrap.min.css">

<!-- plugins -->

<link rel="stylesheet" type="text/css" href="asset/css/plugins/font-awesome.min.css"/>

<link rel="stylesheet" type="text/css" href="asset/css/plugins/simple-line-icons.css"/>

<link rel="stylesheet" type="text/css" href="asset/css/plugins/animate.min.css"/>

<link href="asset/css/style.css" rel="stylesheet">

<!-- end: Css -->

<link rel="shortcut icon" href="Imagenes/LogoSat.png">

<!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body id="mimin" class="dashboard">

<!-- start: Header -->

<nav class="navbar navbar-default header navbar-fixed-top">

<div class="col-md-12 nav-wrapper">

<div class="navbar-header" style="width:100%;">

<div class="opener-left-menu is-open">

<span class="top"></span>

<span class="middle"></span>

<span class="bottom"></span>

</div>

<a href="index.html" class="navbar-brand">

<b>CubeSAT 168</b>

</a>

</div>

</div>

</nav>

<!-- end: Header -->

<div class="container-fluid mimin-wrapper">

<!-- start:Left Menu -->

<div id="left-menu">

<div class="sub-left-menu scroll">

<ul class="nav nav-list">

<center><IMG src="imagenes/LogoSat.png" width=230px height=190px></IMG></center>

<li class="active ripple">

<a href="index.html"><span class="fa-home fa"></span> Inicio </a>

<li class="ripple">

<a class="tree-toggle nav-header">

<span class="fa-area-chart fa"></span> CubeSAT

<span class="fa-angle-right fa right-arrow text-right"></span>

</a>

<ul class="nav nav-list tree">

<li><a href="TempVSTiem.html"> Temperatura vs Tiempo</a></li>

<li><a href="TempVSAlt.html"> Temperatura vs Altitud</a></li>

<li><a href="PreVSTiem.html">Presión vs Tiempo </a></li>

<li><a href="PreVSAlt.html">Presión vs Altitud</a></li>

<li><a href="HumVSTiem.html"> Humedad vs Tiempo</a></li>

<li><a href="HumVSAlt.html"> Humedad vs Altitud </a></li>

<li><a href="FuerzaVsTiempo.html"> Fuerza vs Tiempo </a></li>

<li><a href="FuerzaVsAltitud.html"> Fuerza vs Altitud </a></li>

<li><a href="CalAireVsTiempo.html"> Calidad del Aire vs Tiempo</a></li>

<li><a href="CalAireVsAlt.html"> Calidad del Aire vs Altitud </a></li>

<li><a href="Ubicacion.php"> Ubicación en Tiempo Real </a></li>

</ul>

</li>

<li class="ripple"><a class="tree-toggle nav-header"><span class="fa fa-table"></span> Base de Datos <span class="fa-angle-right fa right-arrow text-right"></span> </a>

<ul class="nav nav-list tree">

<li><a href="RegistrosHistoricos.html">Registros Históricos</a></li>

<li><a href="Estadisticas.html">Estadísticas</a></li>

</ul>

</li>

</div>

<div id="content">

<div class="panel">

<div class="panel-body">

<div class="col-md-6 col-sm-12">

<h3 class="animated fadeInLeft">Lanzamiento del Cube SAT FIM 168</h3>

<p class="animated fadeInDown"><span class="fa fa-map-marker"></span> Aguascalientes, México</p>

<ul class="nav navbar-nav">

<li><a href=""> Documentación</a></li>

<li><a href="Patrocinadores.html">Patrocinadores</a></li>

<li><a href="http://www.cbtis168.edu.mx/">CBTis 168</a></li>

</ul>

</div>

</div>

</div>

<div class="col-md-12">

<div class="panel">

<div class="panel-body">

<center><img src="Imagenes/cubesatblack.png" width=300px height=210px></center>

<h3 style="color:#165E26;" font-size= > Introducción</h3>

<p>El CubeSat es un Pico Satélite desarrollado por los alumnos del CBTis 168 de las especilidades

de Laboratorio Clínico, Mecatrónica, Ofimática y Programación. La finalidad de éste proyecto es promover

el interés de los alumnos de educación media superior a las ciencias y tecnologías.

Esta simulación de un satélite real funciona de la siguiente manera: Captura datos que percibe a través de

sensores y los almacena en una base de datos para que puedan ser graficados e interpretados.

</p>

<div class="panel-body">

<video width="300" autoplay muted loop id="myVideo" controls>

<source src="Videos/Video1.mp4" type="video/mp4">

</video>

</div>

<!-- start: Javascript -->

<script src="asset/js/jquery.min.js"></script>

<script src="asset/js/jquery.ui.min.js"></script>

<script src="asset/js/bootstrap.min.js"></script>

<!-- plugins -->

<script src="asset/js/plugins/moment.min.js"></script>

<script src="asset/js/plugins/jquery.nicescroll.js"></script>

<!-- custom -->

<script src="asset/js/main.js"></script>

<!-- end: Javascript -->

</body>

</html>

HTML Y JAVASCRIPT Gráfica Fuerza Vs Altitud

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="description" content="Miminium Admin Template v.1">

<meta name="author" content="Isna Nur Azis">

<meta name="keyword" content="">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Fim-168 Fuerza vs Altitud </title>

<!-- start: Css -->

<link rel="stylesheet" type="text/css" href="asset/css/bootstrap.min.css">

<!-- plugins -->

<link rel="stylesheet" type="text/css" href="asset/css/plugins/font-awesome.min.css"/>

<link rel="stylesheet" type="text/css" href="asset/css/plugins/simple-line-icons.css"/>

<link rel="stylesheet" type="text/css" href="asset/css/plugins/animate.min.css"/>

<link href="asset/css/style.css" rel="stylesheet">

<!-- end: Css -->

<link rel="shortcut icon" href="Imagenes/LogoSat.png">

<!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body id="mimin" class="dashboard">

<!-- start: Header -->

<nav class="navbar navbar-default header navbar-fixed-top">

<div class="col-md-12 nav-wrapper">

<div class="navbar-header" style="width:100%;">

<div class="opener-left-menu is-open">

<span class="top"></span>

<span class="middle"></span>

<span class="bottom"></span>

</div>

<a href="index.html" class="navbar-brand">

<b>CubeSAT 168</b>

</a>

</div>

</div>

</nav>

<!-- end: Header -->

<div class="container-fluid mimin-wrapper">

<!-- start:Left Menu -->

<div id="left-menu">

<div class="sub-left-menu scroll">

<ul class="nav nav-list">

<center><IMG src="imagenes/LogoSat.png" width=230px height=190px></IMG></center>

<li class="active ripple">

<a href="index.html"><span class="fa-home fa"></span> Inicio </a>

<li class="ripple">

<a class="tree-toggle nav-header">

<span class="fa-area-chart fa"></span> CubeSAT

<span class="fa-angle-right fa right-arrow text-right"></span>

</a>

<ul class="nav nav-list tree">

<li><a href="TempVSTiem.html"> Temperatura vs Tiempo</a></li>

<li><a href="TempVSAlt.html"> Temperatura vs Altitud</a></li>

<li><a href="PreVSTiem.html">Presión vs Tiempo </a></li>

<li><a href="PreVSAlt.html">Presión vs Altitud</a></li>

<li><a href="HumVSTiem.html"> Humedad vs Tiempo</a></li>

<li><a href="HumVSAlt.html"> Humedad vs Altitud </a></li>

<li><a href="FuerzaVsTiempo.html"> Fuerza vs Tiempo </a></li>

<li><a href="FuerzaVsAltitud.html"> Fuerza vs Altitud </a></li>

<li><a href="CalAireVsTiempo.html"> Calidad del Aire vs Tiempo</a></li>

<li><a href="CalAireVsAlt.html"> Calidad del Aire vs Altitud </a></li>

<li><a href="Ubicacion.php"> Ubicación en Tiempo Real </a></li>

</ul>

</li>

<li class="ripple"><a class="tree-toggle nav-header"><span class="fa fa-table"></span> Base de Datos <span class="fa-angle-right fa right-arrow text-right"></span> </a>

<ul class="nav nav-list tree">

<li><a href="RegistrosHistoricos.html">Registros Históricos</a></li>

<li><a href="Estadisticas.html">Estadísticas</a></li>

</ul>

</li>

</div>

<div id="content">

<div class="panel">

<div class="panel-body">

<div class="col-md-6 col-sm-12">

<h3 class="animated fadeInLeft">Fuerza vs Altitud</h3>

<p class="animated fadeInDown"><span class="fa fa-bar-chart-o"></span> Gráfica en Tiempo Real</p>

</div>

</div>

</div>

<div class="col-md-12">

<div class="panel">

<div class="panel-body">

</div>

</div>

<!-- start: Javascript -->

<div style="width:90%;">

<canvas id="canvas1"></canvas>

</div>

<button id="llamada"> Update </button>

<div class="panel-body">

<div class="col-md-6">

<a href="historialFuerzaVsAltitud.html" class=" btn btn-3d btn-primary"> Ir a registros Históricos </a>

</div>

<script src="dist/Chart.bundle.js"></script>

<script src="scripts/jquery-3.3.1.min.js"></script>

<script src="scripts/utils.js"></script>

<script> src="dist/Chart.bundle.js"

src="scripts/jquery-3.3.1.min.js"

src="scripts/utils.js"

function generarGrafica(datos){

var ctx = document.getElementById('canvas1').getContext('2d');

window.myBar = new Chart(ctx, {

type: 'line',

data: datos,

options: {

title: {

display: true,

text: 'Fuerza (J) Vs Altitud (m s. n. m.)'

},

tooltips: {

mode: 'index',

intersect: false

},

responsive: true

}

});

}

window.onload = function() {

//generarGrafica(barChartData);

};

$( document ).ready(function() {

$("#llamada").on("click",function (){

var colores=['#cc0000','#3366ff'];

$.ajax( "querysFuerzaA.php" )

.done(function(response) {

var data=$.parseJSON(response);

console.log(data);

var datasets = [];

for (var i = 0; i < data.grafica.length; i++) {

var obj = {

label: 'Dataset',

borderColor: colores[i],

data: [],

fill: false

};

obj.label = data.grafica[i].name;

obj.data = data.grafica[i].data;

datasets.push(obj);

}

var labels = [];

for (var i = 0; i < data.label.length; i++) {

labels.push(data.label[i]);

}

var barChartData = {

labels: labels,

datasets: datasets

};

console.log(datasets);

generarGrafica(barChartData);

})

.fail(function() {

alert( "error" );

})

.always(function() {

});

});

});

</script>

<script src="asset/js/jquery.min.js"></script>

<script src="asset/js/jquery.ui.min.js"></script>

<script src="asset/js/bootstrap.min.js"></script>

<!-- plugins -->

<script src="asset/js/plugins/moment.min.js"></script>

<script src="asset/js/plugins/morris.min.js"></script>

<script src="asset/js/plugins/raphael.min.js"></script>

<script src="asset/js/plugins/jquery.nicescroll.js"></script>

<script src="asset/js/main.js"></script>

</body>

</html>

PHP Querys de Fuerza Vs Altitud

<?php

require('conexion.php');

$Json=Array();

$i=0;

$sql = "SELECT G.Altitud, V.Joules FROM gps G INNER JOIN Variable\_proce V ON V.ID\_Event=G.ID\_Event order by altitud desc limit 0,150;";

$result = mysqli\_query($conn, $sql);

if (mysqli\_num\_rows($result) > 0) {

$Json['grafica'][0]['name'] = 'Fuerza (J)';

$Json['grafica'][0]['data'] = array();

$Json['label'] = array();

$i = 0;

while($row = mysqli\_fetch\_assoc($result)) {

array\_push($Json['grafica'][0]['data'], $row["Joules"]);

array\_push($Json['label'], $row["Altitud"]);

}

}

echo json\_encode($Json);

mysqli\_close($conn);

?>

HTML Registros Históricos

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="description" content="Miminium Admin Template v.1">

<meta name="author" content="Isna Nur Azis">

<meta name="keyword" content="">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Fim-168</title>

<!-- start: Css -->

<link rel="stylesheet" type="text/css" href="asset/css/bootstrap.min.css">

<!-- plugins -->

<link rel="stylesheet" type="text/css" href="asset/css/plugins/font-awesome.min.css"/>

<link rel="stylesheet" type="text/css" href="asset/css/plugins/handsontable.full.min.css"/>

<link rel="stylesheet" type="text/css" href="asset/css/plugins/animate.min.css"/>

<link href="asset/css/style.css" rel="stylesheet">

<!-- end: Css -->

<link rel="shortcut icon" href="Imagenes/LogoSat.png">

<!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body id="mimin" class="dashboard">

<!-- start: Header -->

<nav class="navbar navbar-default header navbar-fixed-top">

<div class="col-md-12 nav-wrapper">

<div class="navbar-header" style="width:100%;">

<div class="opener-left-menu is-open">

<span class="top"></span>

<span class="middle"></span>

<span class="bottom"></span>

</div>

<a href="index.html" class="navbar-brand">

<b>CubeSAT 168</b>

</a>

</div>

</div>

</nav>

<!-- end: Header -->

<div class="container-fluid mimin-wrapper">

<!-- start:Left Menu -->

<div id="left-menu">

<div class="sub-left-menu scroll">

<ul class="nav nav-list">

<center><IMG src="imagenes/LogoSat.png" width=230px height=190px></IMG></center>

<li class="active ripple">

<a href="index.html"><span class="fa-home fa"></span> Inicio </a>

<li class="ripple">

<a class="tree-toggle nav-header">

<span class="fa-area-chart fa"></span> CubeSAT

<span class="fa-angle-right fa right-arrow text-right"></span>

</a>

<ul class="nav nav-list tree">

<li><a href="TempVSTiem.html"> Temperatura vs Tiempo</a></li>

<li><a href="TempVSAlt.html"> Temperatura vs Altitud</a></li>

<li><a href="PreVSTiem.html">Presión vs Tiempo </a></li>

<li><a href="PreVSAlt.html">Presión vs Altitud</a></li>

<li><a href="HumVSTiem.html"> Humedad vs Tiempo</a></li>

<li><a href="HumVSAlt.html"> Humedad vs Altitud </a></li>

<li><a href="FuerzaVsTiempo.html"> Fuerza vs Tiempo </a></li>

<li><a href="FuerzaVsAltitud.html"> Fuerza vs Altitud </a></li>

<li><a href="CalAireVsTiempo.html"> Calidad del Aire vs Tiempo</a></li>

<li><a href="CalAireVsAlt.html"> Calidad del Aire vs Altitud </a></li>

<li><a href="Ubicacion.php"> Ubicación en Tiempo Real </a></li>

</ul>

</li>

<li class="ripple"><a class="tree-toggle nav-header"><span class="fa fa-table"></span> Base de Datos <span class="fa-angle-right fa right-arrow text-right"></span> </a>

<ul class="nav nav-list tree">

<li><a href="RegistrosHistoricos.html">Registros Históricos</a></li>

<li><a href="Estadisticas.html">Estadísticas</a></li>

</ul>

</li>

</div>

<div id="content">

<div class="panel">

<div class="panel-body">

<div class="col-md-6 col-sm-12">

<h3 class="animated fadeInLeft">Registros Históricos</h3>

<p class="animated fadeInDown"><span class="fa fa-history"></span> Datos en Tiempo Real</p>

</div>

</div>

</div>

<div class="col-md-12">

<div class="panel">

<div class="panel-body">

<center><iframe src="registrosh.php" width=1400px height=450px frameborder="0"></iframe></iframe> </center>

</div>

</div>

</div>

</div>

<!-- start: Javascript -->

<script src="asset/js/jquery.min.js"></script>

<script src="asset/js/jquery.ui.min.js"></script>

<script src="asset/js/bootstrap.min.js"></script>

<!-- plugins -->

<script src="asset/js/plugins/moment.min.js"></script>

<script src="asset/js/plugins/handsontable.full.min.js"></script>

<script src="asset/js/plugins/jquery.nicescroll.js"></script>

<!-- custom -->

<script src="asset/js/main.js"></script>

</script>

<!-- end: Javascript -->

</body>

</html>

**Código fuente de: Base de datos**

**Lenguajes: SQL**

**Fecha: 18/03/2018**

**Institución educativa: CBTis 168**

**Autor: Rogelio Trejo Estrada**

-- phpMyAdmin SQL Dump

-- version 4.7.7

-- https://www.phpmyadmin.net/

--

-- Servidor: 127.0.0.1

-- Tiempo de generación: 23-03-2018 a las 03:39:13

-- Versión del servidor: 10.1.30-MariaDB

-- Versión de PHP: 7.1.14

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time\_zone = "+00:00";

--

-- Base de datos: `cubesat`

--

DELIMITER $$

--

-- Procedimientos

--

CREATE DEFINER=`root`@`localhost` PROCEDURE `ConsultarAcele` () BEGIN

SELECT H.HoraEvent,A.ax,A.ay,A.az FROM hra\_evento H INNER JOIN acelerometro A ON A.ID\_Event=H.ID\_Event;

END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `ConsultarExtra` () BEGIN

SELECT H.HoraEvent,E.Temp\_1,E.Temp\_2,E.Humedad,E.Presion,E.Calidad\_Aire,E.Infrarrojo,E.Canti\_Luz

FROM hra\_evento H INNER JOIN extra E ON E.ID\_Event=H.ID\_Event

ORDER BY horaevent DESC LIMIT 0,19;

END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `ConsultarGiro` () BEGIN

SELECT H.HoraEvent,G.gx,G.gy,G.gz FROM hra\_evento H INNER JOIN giroscopio G ON G.ID\_Event=H.ID\_Event;

END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `ConsultarGPS` () BEGIN

SELECT H.HoraEvent,G.Latitud,G.Longitud,G.Altitud FROM hra\_evento H INNER JOIN gps G ON G.ID\_Event=H.ID\_Event;

END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `ConsultarMagne` () BEGIN

SELECT H.HoraEvent,M.mx,M.my,M.mz FROM hra\_evento H INNER JOIN magnetometro M ON M.ID\_Event=H.ID\_Event;

END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `InsertarDatos` (IN `VLatitud` FLOAT, IN `VLongitud` FLOAT, IN `VALtitud` FLOAT, IN `Vax` FLOAT, IN `Vay` FLOAT, IN `Vaz` FLOAT, IN `Vgx` FLOAT, IN `Vgy` FLOAT, IN `Vgz` FLOAT, IN `Vmx` FLOAT, IN `Vmy` FLOAT, IN `Vmz` FLOAT, IN `VTemp\_1` FLOAT, IN `VTemp\_2` FLOAT, IN `VCalidad\_Aire` FLOAT, IN `VCanti\_Luz` FLOAT, IN `VHumedad` FLOAT, IN `VInfrarrojo` FLOAT, IN `VPrecion` FLOAT, IN `VJoules` FLOAT, IN `VVariacion\_Temp` FLOAT, IN `VVariacion\_Altura` FLOAT, IN `VPrecion\_Barometrica` FLOAT, IN `VAltPress` INT, IN `VCanti\_Volta` INT) BEGIN

INSERT INTO hra\_evento(HoraEvent) VALUES(NOW());

INSERT INTO acelerometro(ax,ay,az) VALUES(Vax,Vay,Vaz);

INSERT INTO giroscopio(gx,gy,gz) VALUES(Vgx,Vgy,Vgz );

INSERT INTO gps(Latitud,Longitud,Altitud) VALUES(VLatitud,VLongitud,VAltitud);

INSERT INTO magnetometro(mx,my,mz) VALUES(Vmx,Vmy,Vmz);

INSERT INTO extra(Temp\_1,Temp\_2,Humedad,Presion,Infrarrojo,Calidad\_Aire,Canti\_Luz,AltPress,Canti\_Volta) VALUES(VTemp\_1,VTemp\_2,VHumedad,VPrecion,VInfrarrojo,VCalidad\_Aire,VCanti\_Luz,VAltPress,VCanti\_Volta);

INSERT INTO variable\_proce(Joules,Variacion\_Temp,Variacion\_Altura,Presion\_Barometrica) VALUES(VJoules,VVariacion\_Temp,VVariacion\_Altura,VPrecion\_Barometrica);

END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `Reiniciar` () BEGIN

TRUNCATE acelerometro;

TRUNCATE giroscopio;

TRUNCATE gps;

TRUNCATE extra;

TRUNCATE hra\_evento;

TRUNCATE magnetometro;

TRUNCATE variable\_proce;

TRUNCATE serial;

END$$

DELIMITER ;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `acelerometro`

--

CREATE TABLE `acelerometro` (

`ID\_Event` int(11) NOT NULL,

`ax` float NOT NULL,

`ay` float NOT NULL,

`az` float NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `extra`

--

CREATE TABLE `extra` (

`ID\_Event` int(11) NOT NULL,

`Temp\_1` float NOT NULL,

`Temp\_2` float NOT NULL,

`Humedad` float NOT NULL,

`Presion` float NOT NULL,

`Infrarrojo` float NOT NULL,

`Calidad\_Aire` float NOT NULL,

`Canti\_Luz` float NOT NULL,

`AltPress` float NOT NULL,

`Canti\_Volta` float NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `giroscopio`

--

CREATE TABLE `giroscopio` (

`ID\_Event` int(11) NOT NULL,

`gx` float NOT NULL,

`gy` float NOT NULL,

`gz` float NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `gps`

--

CREATE TABLE `gps` (

`ID\_Event` int(11) NOT NULL,

`Latitud` float NOT NULL,

`Longitud` float NOT NULL,

`Altitud` float NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `hra\_evento`

--

CREATE TABLE `hra\_evento` (

`ID\_Event` int(11) NOT NULL,

`HoraEvent` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `magnetometro`

--

CREATE TABLE `magnetometro` (

`ID\_Event` int(11) NOT NULL,

`mx` float NOT NULL,

`my` float NOT NULL,

`mz` float NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `variable\_proce`

--

CREATE TABLE `variable\_proce` (

`ID\_Event` int(11) NOT NULL,

`Joules` float NOT NULL,

`Variacion\_Temp` float NOT NULL,

`Variacion\_Altura` float NOT NULL,

`Presion\_Barometrica` float NOT NULL,

`Velocidad\_Acent` float NOT NULL,

`Velocidad\_Desent` float NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Índices para tablas volcadas

--

--

-- Indices de la tabla `acelerometro`

--

ALTER TABLE `acelerometro`

ADD PRIMARY KEY (`ID\_Event`);

--

-- Indices de la tabla `extra`

--

ALTER TABLE `extra`

ADD PRIMARY KEY (`ID\_Event`);

--

-- Indices de la tabla `giroscopio`

--

ALTER TABLE `giroscopio`

ADD PRIMARY KEY (`ID\_Event`);

--

-- Indices de la tabla `gps`

--

ALTER TABLE `gps`

ADD PRIMARY KEY (`ID\_Event`);

--

-- Indices de la tabla `hra\_evento`

--

ALTER TABLE `hra\_evento`

ADD PRIMARY KEY (`ID\_Event`);

--

-- Indices de la tabla `magnetometro`

--

ALTER TABLE `magnetometro`

ADD PRIMARY KEY (`ID\_Event`);

--

-- Indices de la tabla `variable\_proce`

--

ALTER TABLE `variable\_proce`

ADD PRIMARY KEY (`ID\_Event`);

--

-- AUTO\_INCREMENT de las tablas volcadas

--

--

-- AUTO\_INCREMENT de la tabla `acelerometro`

--

ALTER TABLE `acelerometro`

MODIFY `ID\_Event` int(11) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `extra`

--

ALTER TABLE `extra`

MODIFY `ID\_Event` int(11) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `giroscopio`

--

ALTER TABLE `giroscopio`

MODIFY `ID\_Event` int(11) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `gps`

--

ALTER TABLE `gps`

MODIFY `ID\_Event` int(11) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `hra\_evento`

--

ALTER TABLE `hra\_evento`

MODIFY `ID\_Event` int(11) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `magnetometro`

--

ALTER TABLE `magnetometro`

MODIFY `ID\_Event` int(11) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `variable\_proce`

--

ALTER TABLE `variable\_proce`

MODIFY `ID\_Event` int(11) NOT NULL AUTO\_INCREMENT;

COMMIT;