

Instituto Tecnológico de Estudios Superiores de Monterrey

*Campus Ciudad de México*

*-Escuela de ingeniería y ciencias-*

***Administración del Reto - Etapa #4***

Implementación de internet de las cosas

*TC1004B*- *Grupo 532*

**Profesor:**

Paloma Vilchis

**Integrantes de equipo:**

Rodrigo Fernando Rivera Olea - A01664716

Katia Geraldine Vidals Estrever - A01657587

Filiberto Lucio Alvarez - A01661125

Noviembre 2023

**Introducción**  
  
En este trabajo se presenta de manera detallada la integración de las diversas etapas de la administración del proyecto en el desafio de llevar a cabo la implementacion de Iot.Se ofrece una visión exhaustiva de la implementación llevada a cabo durante el proyecto del rover , el cual ha sido el foco de nuestra trabajo durante las últimas 5 semanas.Este documento aborda la planificación , ejecución y seguimiento del proyecto , destacamos aspectos clave que han contribuido al éxito de la implementación de Iot en el proyecto

Asimismo analizamos las posibles áreas de mejora ,proporcionando un perspectiva integral sobre el proceso de administración de proyectos en el ámbito de las tecnologías emergentes

El internet de las cosas (IoT) es una red de dispositivos físicos interconectados , que se encuentran integrados con sensores , software entre otros elementos tecnológicos , permitiéndonos recopilar y compartir datos vía internet .Estos dispositivos pueden ser electrodomésticos y wearables hasta maquinaria industrial , estos se encuentran conectados para recopilar información , tomar decisiones automatizadas .La interconexión de estos dispositivos impulsa la comodidad y la inteligencia en múltiples sectores transformando al manera en la que interactuamos con la tecnología y el entorno que nos rodea

**Objetivo**

En el marco de este proyecto , nos centramos en el desarrollo de un EX-rover con un propósito dual , se buscó la creación de un vehículo que pudiera desplazarse de manera controlada de forma remota.En segundo lugar se orientó hacia la capacidad de recopilar datos de su entorno mediante la integración de sensores , tales como los de temperatura, presión y los ultrasonidos  
  
La movilidad remota del X-rover ofrece un enfoque práctico para la exploración y operación en entornos variados , desde espacios confinados hasta áreas de difícil acceso.Por otro lado la integración de los sensores ultrasónicos , presión y temperatura representan un componente crucial en este proyecto , ya que permiten al X-rover interactuar con su entorno , recopilando datos , estos datos son recopilados para posteriormente canalizarlos y almacenarlos en una base de datos , facilitando el análisis y la monitorización   
  
**Implementación**

En la ejecución de este proyecto del X-rover nos enfrentamos al desafio de asegurar sus correcto funcionamiento mediante la implementación de hardware y software

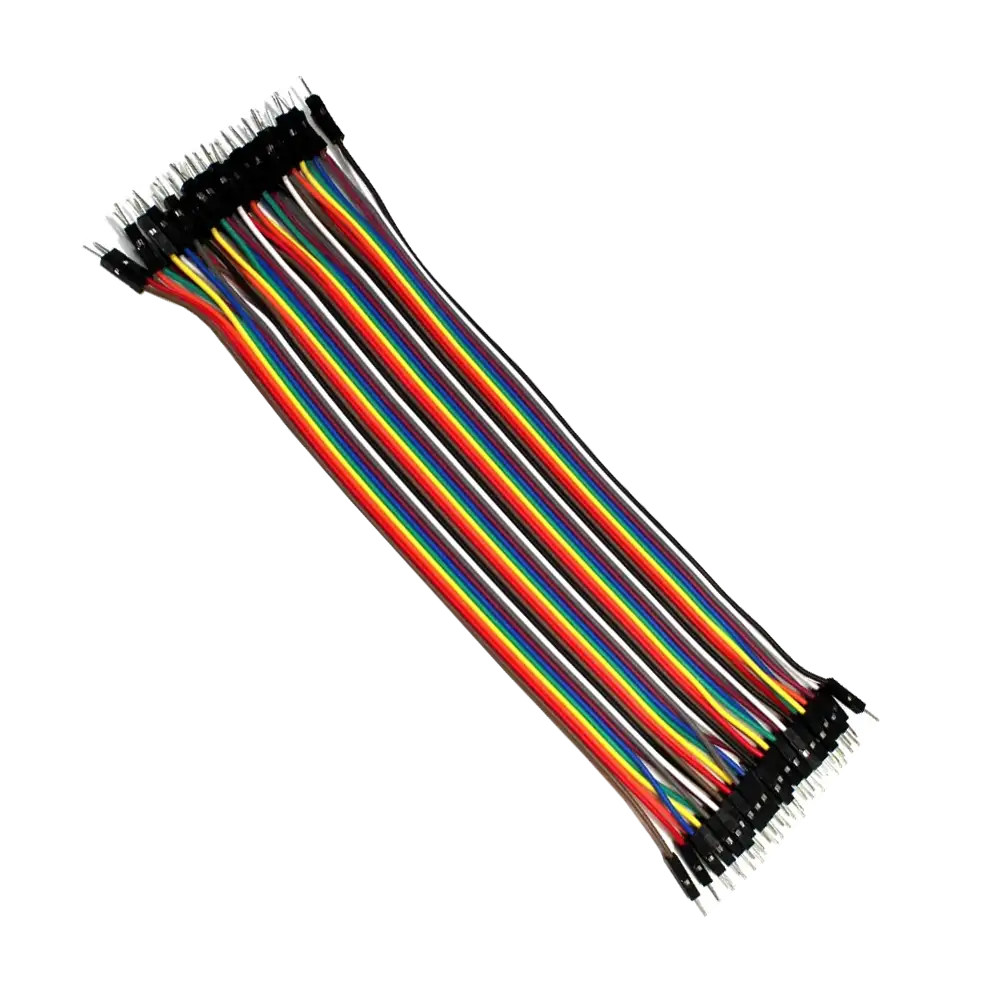
La primera fase del proyecto proyecto se dedicó a la integración de hardware necesario , esto incluye la configuración de los componentes físicos esenciales para el vehículo como los motores , chasis ,baterias y pos supuesto los sensores especializados( ultrasonico,temperatura, presión)

La segunda fase se centró en el desarrollo de software que permite la gestión eficiente y coordinada de todos los componentes de hardware. Se crearon programas específicos para el control , la interpretación de datos de los sensores y la comunicación con la base de datos

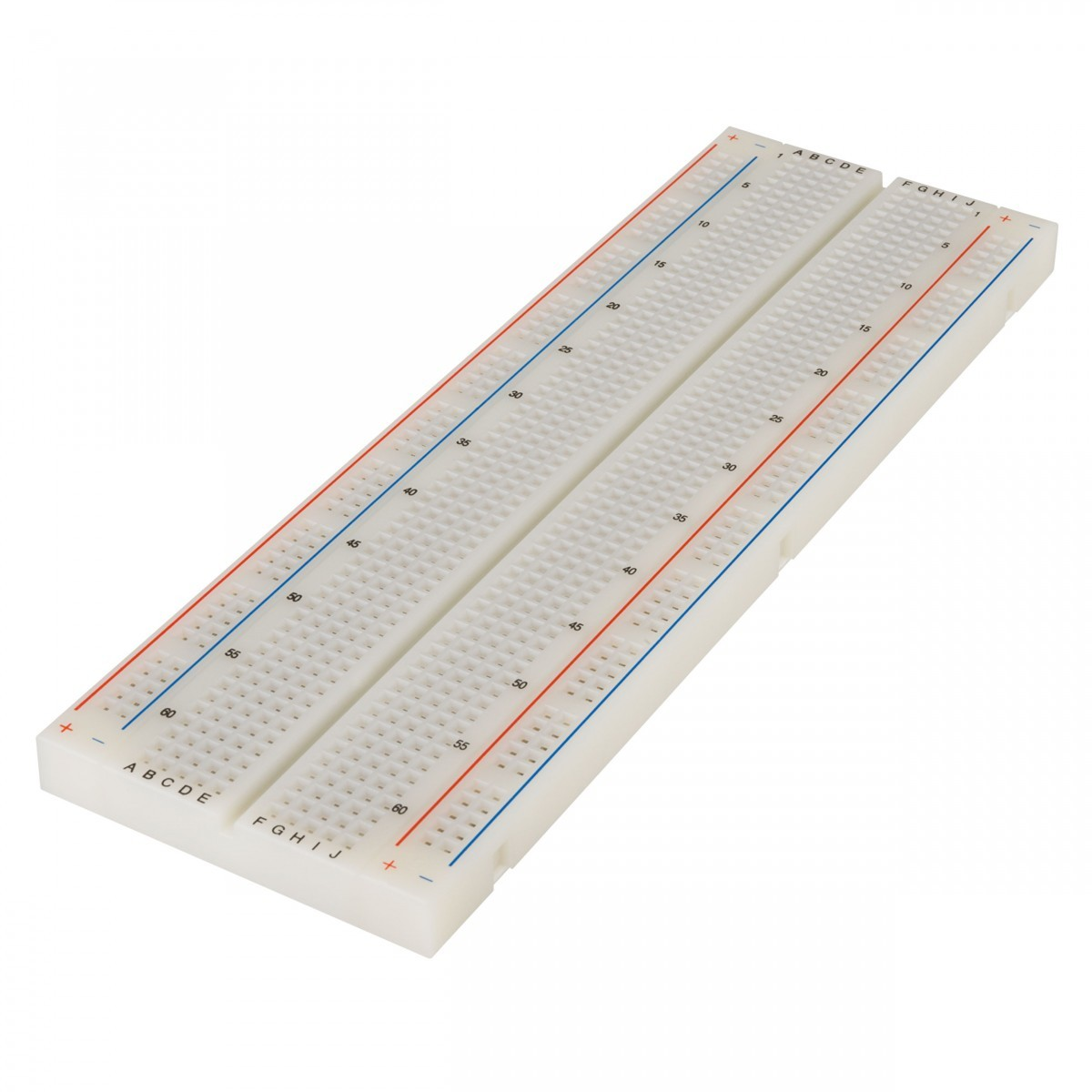
La combinación de estos dos elementos , hardware y software es fundamental para el éxito del proyecto

**Materiales**  
**Hardware**

Cables jumper



2 protoboards



batería de 10000mA



raspberry pi3



resistencias  


sensor ultrasónico



sensor temperatura  


2 ruedas



**Software:**  
En el ambito del software , la implementación de este proyecto del x-rover se beneficio de la aplicación de diversas herramientas especializadas.Cada una de estas herramientas desempeña un papel crucial en garantizar la eficiencia y conectividad del sistema

Node-red:  
Para la gestión de flujo de datos incorporamos Node-red.Esta plataforma de codigo facilita la creación de flujos visuales permitiendo una integración sin problemas entre diferentes elementos , fue esencial para la ejecución de comandos de control remoto   
Mysql:  
En el ámbito de la gestión de base de datos, la elección de MySQL proporcionar una infraestructura sólida para almacenar y recuperar datos de manera eficiente , este sistema se aprovechó para garantizar la disponibilidad de información recopilada  
React  
Para la interfaz de usuario y la visualización de datos, optamos por React. Este marco de desarrollo de interfaz de usuario ofrece una experiencia interactiva y dinámica, fundamental para presentar de manera clara y comprensible la información recopilada por el X-rover.  
Google Cloud  
  
La inclusión de Google Cloud en nuestro proyecto tecnológico garantizó la escalabilidad, seguridad y disponibilidad de los servicios en la nube utilizados en el proyecto.  
  
  
Lenguajes:

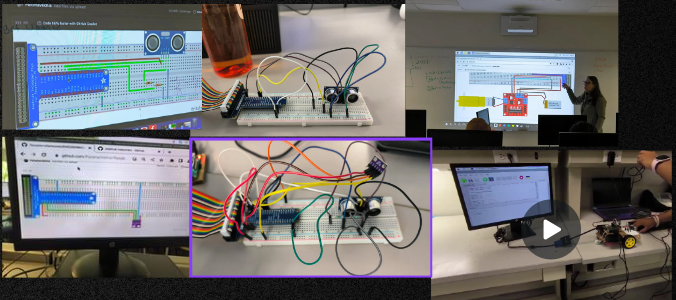
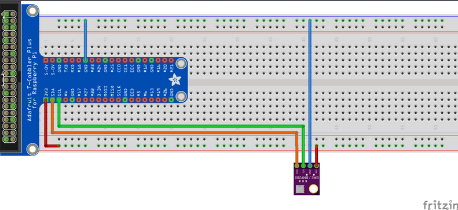
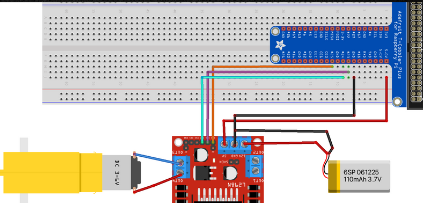
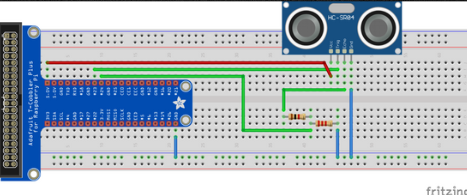
Se emplearon diversos lenguajes para potenciar la funcionalidad y la interfaz del proyecto X-rover.La combinación de Python, JS (JavaScript), HTML y CSS desempeñó roles específicos y complementarios para lograr un sistema coherente y eficaz.  
  
Python:

Se utilizó en la lógica principal y el desarrollo de operaciones del X-rover.Python proporciona flexibilidad y eficiencia en la ejecución de tareas críticas.  
  
JS (JavaScript):

Desempeñando un papel esencial en la creación de interactividad y dinamismo, JavaScript se empleó para mejorar la experiencia del usuario y facilitar la comunicación fluida con Node-Red.  
  
HTML y CSS:

Se utilizaron para la estructuración y el diseño de la interfaz de usuario.HTML proporcionó el esqueleto de la página, mientras que CSS se encargó de la presentación y el estilo, contribuyendo a una interfaz visualmente atractiva y funcional.  
  
  
**Pasos a seguir**  
  
1) soldar sensores y motores



2) construir el X-rover  
  
  
  
  
diagramas  
  
Sensor presion y temperatura  
  
  
Motores  
  
  
Sensor de distancia  


**Código implementado**

# Referencias a a las librerias

from bmp\_280 import BMP280

from time import sleep

import RPi.GPIO as GPIO

import time

import mysql.connector

#configuracion base de datos

mydb=mysql.connector.connect(

host="34.27.235.22"

user="root"

password="hola"

database="iot\_proyect"

)

## SENSOR PRESION

bmp = BMP280(port=1, mode=BMP280.FORCED\_MODE, oversampling\_p=BMP280.OVERSAMPLING\_P\_x16, oversampling\_t=BMP280.OVERSAMPLING\_T\_x1,

filter=BMP280.IIR\_FILTER\_OFF, standby=BMP280.T\_STANDBY\_1000)

# MOTOR

GPIO.setmode(GPIO.BOARD)

Motor1 = 16 # Entrada

Motor2 = 18 # Entrada

Motor3 = 22 # Habilitar

Motor4 = 17 # Entrada

Motor5 = 19 # Entrada

Motor6 = 23 # Habilitar

#Establecemos el set up del motor como salida

GPIO.setup(Motor1, GPIO.OUT)

GPIO.setup(Motor2, GPIO.OUT)

GPIO.setup(Motor3, GPIO.OUT)

GPIO.setup(Motor4, GPIO.OUT)

GPIO.setup(Motor5, GPIO.OUT)

GPIO.setup(Motor6, GPIO.OUT)

# CONFIGURACION Y SENSOR DE DISTANCIA

GPIO\_TRIGGER = 23

GPIO\_ECHO = 24

GPIO.setup(GPIO\_TRIGGER, GPIO.OUT)

GPIO.setup(GPIO\_ECHO, GPIO.IN)

GPIO.output(GPIO\_TRIGGER, False)

# Configuración para guardar datos

sFileStamp = time.strftime('%Y%m%d%H')

sFileName = 'out/' + 'out' + sFileStamp + '.txt'

f = open(sFileName, 'a')

f.write('TimeStamp,Value' + '\n')

print("Inicia la toma de datos")

print("Inicia la toma de datos")

try:

while True:

print("Acerque el objeto para medir la distancia")

# Medir distancia

GPIO.output(GPIO\_TRIGGER, True)

time.sleep(0.00001)

GPIO.output(GPIO\_TRIGGER, False)

start = time.time()

while GPIO.input(GPIO\_ECHO) == 0:

start = time.time()

while GPIO.input(GPIO\_ECHO) == 1:

stop = time.time()

elapsed = stop - start

distance = (elapsed \* 34300) / 2

# Leer presión

pressure = bmp.read\_pressure()

# Leer temperatura

temp=bmp.read\_temperature()

TimeStamp = time.strftime('%Y%m%d%H%M%S')

f.write(sTimeStamp + ',' + str(distance)'\n')

#Imprimir valores

print("Presión(hPa): " + str(pressure))

print("Temperatura (°C): " + str(temp))

print(" Distancia: " + str(distance))

#Abrir base de datos mysql e insercion

cursor=mydb.cursor()

sql="INSERT INTO temperatura(temperatura,fecha) VALUES (%s,CURRENT\_TIMESTAMP)"

val=(temp,)

cursor.execute(sql,val)

mydb.commit()

print("Dato de temperatura insertado")

sql1="INSERT INTO presion(presion,fecha) VALUES (%s,CURRENT\_TIMESTAMP)"

val1=(pressure,)

cursor.execute(sql1,val1)

mydb.commit()

print("Dato de presion insertado")

sql2="INSERT INTO distancia(distancia,fecha) VALUES(%s,CURRENT\_TIMESTAMP)"

val2(distance,)

cursor.execute(sql2,val2)

mydb.commit()

print("Dato de distancia insertado")

#ambos motores se accionan

GPIO.output(Motor1, GPIO.HIGH)

GPIO.output(Motor2, GPIO.LOW)

GPIO.output(Motor3, GPIO.HIGH)

GPIO.output(Motor4, GPIO.HIGH)

GPIO.output(Motor5, GPIO.LOW)

GPIO.output(Motor6, GPIO.HIGH)

# Avanzar hasta cierta distancia

if distance < 10:

print("Obstáculo detectado. Deteniendo motores y girando.")

# Detener motores

GPIO.output(Motor1, GPIO.LOW)

GPIO.output(Motor2, GPIO.LOW)

GPIO.output(Motor3, GPIO.LOW)

GPIO.output(Motor4, GPIO.LOW)

GPIO.output(Motor5, GPIO.LOW)

GPIO.output(Motor6, GPIO.LOW)

# Girar

GPIO.output(Motor1, GPIO.HIGH)

GPIO.output(Motor2, GPIO.LOW)

GPIO.output(Motor3, GPIO.HIGH)

GPIO.output(Motor4, GPIO.LOW)

GPIO.output(Motor5, GPIO.LOW)

GPIO.output(Motor6, GPIO.LOW)

sleep(3)

except KeyboardInterrupt:

print('\n' + 'Termina la captura de datos.' + '\n')

f.close()

GPIO.cleanup()

**Link drive videos funcionamiento Xrover:**

[Xrover bloque - Google Drive](https://drive.google.com/drive/u/0/folders/1PHmPvH6jkCuWagaE8L-TrzjS-lIlcrsn)

**Problemas de implementación:**  
  
La creación del X-rover no estuvo exenta de desafíos , tanto en la esfera del hardware como en la de software .A pesar de estos contratiempos nuestro equipo encontró soluciones efectivas para el exito del proyecto  
  
Problemas en el Hardware  
  
En la fase inicial nos enfrentamos a diversos inconvenientes en el hardware especialmente relacionados con la conectividad de los sensores , la comunicación irregular y las interrupciones en la conexion con los sensores plantearon desafios considerables,.Ademas de esto nuestro sensor de distancia experimentó fallas lo que nos llevó a tomar la decisión de reemplazarlos para garantizar mediciones precisas y un rendimiento óptimo   
  
En el ámbito del software, nos encontramos con obstáculos durante la instalación de la librería bmp280. Esta dificultad nos llevó a reconsiderar y reiniciar la instalación de programas desde cero, asegurando una base sólida para el correcto funcionamiento del sistema. Asimismo, enfrentamos desafíos al desarrollar tanto el front-end como el back-end  
  
A pesar de estos desafíos nuestro equipo logros superar estos obstáculos . Se implementaron estrategias específicas para abordar los problemas de conectividad del hardware, y se llevaron a cabo ajustes precisos en el software para solventar las dificultades en la instalación y el desarrollo de la interfaz. **Referencias:**PKI for IoT Buyer’s Guide – Intertrust Technologies. (s. f.). Intertrust Technologies. <https://www.intertrust.com/resources/pki-for-iot-buyers-guide/?utm_source=bing&utm_medium=cpc&utm_campaign=pki-for-iot-buyers-guide&utm_agid=1230354502751461&creative=&device=c&placement=&utm_term=iot&utm_campaign=&utm_source=bing&utm_medium=ppc&hsa_acc=7383551549&hsa_cam=18725666578&hsa_grp=1230354502751461&hsa_ad=&hsa_src=o&hsa_tgt=kwd-76897403080169:loc-119&hsa_kw=iot&hsa_mt=e&hsa_net=adwords&hsa_ver=3&msclkid=d0993396ecf216945b4356fe7a38c789>

http://gsii.der.usal.es/bisite\_new/en/group/team/corchado. (s. f.). ¿Cómo surgió el Internet de las cosas? Grupo de investigación BISITE | University of Salamanca. <https://bisite.usal.es/es/blog/formacion/21/04/21/como-surgio-el-internet-de-las-cosas-BISITE>