
Universidad Mariano Gálvez de Guatemala
Facultad de Ingeniería de Sistemas de Información y ciencias de la computación
Ing. Allan Gonzáles



Proyecto Sistema Solar STEM

Jose Miguel Argueta Reyes 9941-25-843
Rodrigo Adrián Barrios Monterroso 9941-25-841
Moisés Abinadí Farfan Gonzáles 9941-25-86
Jose Daniel Real García 9941-25-837
Ariel Yuam Vides Ordoñez 9941-25-20300

Plan Fin de Semana

Sección "A"

Índice

Contenido

Perfil de la Empresa	1
Misión.....	1
Visión	1
Valores empresariales.....	1
Estructura del equipo (organigrama)	2
Áreas de especialización	2
Redes sociales y medios de contacto	2
Breve portafolio o proyectos previos	3
Objetivo general de la empresa.....	3
Propuesta de Proyecto, Sistema Solar Interactivo Tecnológico para Educación STEM	4
Objetivo General:	4
Introducción	7
Justificación	8
Planteamiento del problema	9
Objetivos.....	10
Objetivo general	10
Objetivos específicos	10
Metodología del trabajo	11
Arquitectura del Proyecto	13
Conceptos básicos de la electrónica.	15
¿Qué es una LED?	15
¿Qué es un Arduino?	15
Sistema Solar	16
El sistema solar se compone de la siguiente manera:	16
¿Qué otros elementos tienen el Sistema Solar?	17
Descubrimientos y Exploración	17

¿En qué galaxia se ubica el sistema solar?	18
¿Como se formó el Sistema Solar?	19
La dimensión astronómica de las distancias en el espacio	19
Que es la electrónica.....	20
Diagrama de flujo del sistema general.	21
Diagrama de flujo del funcionamiento de maqueta	22
Proyección presupuestal.	23
Presupuesto.....	23
Proyección presupuestal.	24
Presupuesto	25
Maqueta.....	26
Funcionamiento de la Maqueta	27
Motor Nema17	28
Sistema de refrigeración: disipador de calor	33
Información de material utilizado.....	39
Códigos y configuraciones	41
Arduino Código	41
Configuración WIFI	41
Configuración de LEDS.....	41
Nombres y descripciones cortas de los modos LED para la página web.....	42
Configuración del motor.....	42
Funciones de inicio	43
Configuración inicial de la tira de LEDs.....	43
Configuración del motor.....	43
Lool principal	43
Funciones de soporte	44
Herramientas y software.....	53
Propuestas de Mejora	55
Referencias Bibliográficas.	56

Agradecimientos	57
-----------------------	----

Perfil de la Empresa

Nombre y logotipo de la empresa

Nombre de la empresa: DTS (Dream Team Systems)

Logotipo:



Misión

Nuestra misión es desarrollar proyectos innovadores y funcionales que reflejen los conocimientos adquiridos durante el trayecto de la carrera de ingeniería en sistemas, aplicando la teoría a la práctica para resolver necesidades reales o mejorar procesos existentes.

Visión

Aspiramos a que los trabajos desarrollados durante la carrera sean la base para futuras innovaciones, impulsando una mentalidad de mejora continua, investigación y compromiso con la excelencia. Queremos ser reconocidos como un equipo que trasciende lo académico, dejando huella por su pasión, disciplina y capacidad de transformar ideas en resultados.

Valores empresariales

- Innovación
- Transparencia
- Compromiso
- Ética profesional
- Creatividad
- Aprendizaje continuo

Estructura del equipo (organigrama)

Socios Estratégicos

|

|— Área de Desarrollo de Software y Desarrollo Web – **José Daniel Real García**
/ Frontend Lead Developer

|— Área de IoT y Electrónica – **Moisés Abinadí Farfan González / Hardware Engineering Manager**

|— Área de Diseño y Experiencia de Usuario – **Rodrigo Adrián Barrios Monterroso / UX/UI Design Manager**

|— Área de Análisis y Gestión Documental – **José Miguel Argueta Ortiz / Business Analyst**

|— Área de Dirección de Tecnología y Ciberseguridad – **Ariel Yuam Vides Ordoñez / Chief Information Security Officer (CISO)**

Áreas de especialización

- Desarrollo de Software y Desarrollo web
- Internet de las Cosas (IoT) y Electrónica Aplicada
- Diseño y Experiencia de Usuario
- Análisis y Gestión Documental
- Área de Dirección de Tecnología y Ciberseguridad

Redes sociales y medios de contacto

Sitio web: <http://dreamteamstem.infinityfree.me/>

Instagram: [@dreamteamsystems](https://www.instagram.com/dreamteamsystems)

Github: https://github.com/yuamvord/Sistema_Solar_STEM

Breve portafolio o proyectos previos

- Maqueta del Sistema Solar STEM

Objetivo general de la empresa

Impulsar el desarrollo de proyectos tecnológicos con enfoque educativo, fomentando la creatividad, el aprendizaje práctico y la aplicación de los conocimientos de ingeniería para contribuir a la formación integral de futuros profesionales.

Propuesta de Proyecto, Sistema Solar Interactivo Tecnológico para Educación STEM

Objetivo General:

Desarrollar un sistema solar interactivo usando tecnologías modernas que combine automatización, visualización de datos y una maqueta interactiva, con fines educativos y de simulación.

Sistema solar en maqueta con:

1. Motores que hacen girar los planetas
2. Luces LED que indican las características (Luna, Sol, Planetas, Asteroides, etc.)
3. Panel de control web para interactuar con los planetas (Encender luces, hacer girar, etc.)
4. Arduino UNO o ESP32
5. Servomotores
6. Node.js + Express
7. Html + Css + Js
8. Puerto COM para conectar Arduino y node.js

Distribución del trabajo

Miembro	Rol	Actividades principales
Yuam	Backend Developer	Node.js, Express, Comunicación con Arduino. API Rest para manejar comandos y sensores
Goob	Frontend Developer	HTML, CSS, JavaScript, diseño de interfaz web con botones, feedback visual, conexión a API
Moisés	Electrónico	Cableado, conexión de LEDs y motores, codificación del Arduino, pruebas de circuito
Adrián	Diseñador y maquetador	Construcción física, planetas, sistema giratorio, estructura estable, organización de componentes físicos
José	Documentador y Analista	Documentación técnica, algoritmos, diagrama de flujo, manual del usuario, presentación gráfica, defensa teórica del proyecto

Cronograma del trabajo

Semana	Actividades	Responsable
Semana 1	Investigación, planificación, lista de materiales, diagrama de flujo y algoritmo	Todos, liderado por documentador(José)
	Boceto de maqueta, diseño visual del Frontend, arquitectura del sistema (Node.js – Arduino)	Yuam, Goob, Adrián
Semana 2	Cableado y ensamblado básico: luces, motores	Moisés, Adrián
	Backend: Servidor Node.js + Conexión serial con Arduino	Yuam
	Frontend: Estructura de la página + prueba de botones	Goob
Semana 3	Integración Backend - Arduino	Yuam, Moisés
	Integración Frontend – API Rest	Yuam, Goob
	Finalización de maqueta y ajustes físicos	Maquetador
Semana 4	Pruebas finales, solución de bugs, presentación en PowerPoint	Todos
	Documentación final: algoritmo, diagrama de flujo y manual de usuario	José
	Video demostrativo y grabación de funcionamiento	José, Goob

Funcionalidades del sistema

Función	Tecnología	Descripción
Movimiento de planetas	Arduino + Servomotores	Cada planeta puede girar individualmente
Encendido de LEDs	Arduino + LEDs	Un LED por planeta indica condiciones como atmosfera
Control Web	Node.js + HTML/CSS/JS	WebApp con botones por planeta
API Rest	Node.js + Express	Ruta para enviar comandos a Arduino
Comunicación Arduino <-> Web	Comunicación Serial (USB)	Arduino escucha comandos y los ejecuta

OPCIONAL EN DADO CASO FUESE NECESARIO

- **Base de datos ligera** con la info de cada planeta que se consulta desde la web.
- **Dashboard educativo** con gráficas simples (ej. tamaño relativo de planetas).

Sensores y componentes:

Componente	Uso	Aprox. Precio
LEDs	Luz por planeta	10-20 Q
Resistencias 220	Limitar la corriente del LED	Q5
Servomotores SG90	Girar planetas	15-25 Q cada uno
Protoboard + Jumpers	Conexión sin soldadura	20Q
Fuente 9V o Powerbank	Alimentación	20-40 Q

Softwares de trabajo:

Documento	Herramienta	Justificación
Informe técnico, manual	Microsoft Word	Más presentable y fácil exportación a PDF
Algoritmo y Diagrama de flujo	Draw.io	Gratis, online, Exportación a PNG/SVG/PDF
Código y control de versiones	Github + Git + Visual Studio Code	Profesional, backups, colaboración
Presentación	PowerPoint, Canva o Google slides	Accesible, buen diseño visual
Notas y tareas del grupo	Notion	Organización de tareas, fechas y pendientes

Herramientas y software:

Área	Herramienta	Listo	Usos
Programación Arduino	Arduino IDE	<input type="checkbox"/>	Subir código al Arduino
Backend/API Rest	Node.js + Express	<input type="checkbox"/>	Crear servidor y comunicar con Arduino
Frontend	HTML/CSS/JS + VsCode	<input type="checkbox"/>	Página de control del sistema solar
IDE principal	Visual Studio Code	<input type="checkbox"/>	Código
Repositorio	Github	<input type="checkbox"/>	Guardar código, documentación e historial
Diagramas	Draw.io	<input type="checkbox"/>	Algoritmo, flujo y arquitectura
Documentos	Word	<input type="checkbox"/>	Informe, documentación y manual de usuario
Presentación	Canva / PowerPoint	<input type="checkbox"/>	Para defender el proyecto
Gestión del proyecto	Notion	<input type="checkbox"/>	Asignar tareas y organización

Introducción

El presente proyecto tiene como objetivo el diseño y desarrollo de un Sistema Solar interactivo, utilizando una placa Arduino R4uno y un motor eléctrico como elementos principales de control y movimiento. Este sistema busca representar de manera didáctica y visual el movimiento de los planetas alrededor del Sol, permitiendo comprender mejor los principios básicos de la astronomía y la automatización electrónica.

A través de la programación del Arduino, se controla el motor que simula el movimiento orbital de los planetas, integrando además componentes como luces LED para representar el brillo natural espacial y el movimiento de la maqueta en general. El proyecto combina conocimientos de física, electrónica, programación y diseño para crear una maqueta educativa que facilita el aprendizaje.

Este tipo de implementación no solo fomenta la creatividad y la comprensión de los sistemas automatizados, sino que también impulsa el interés por las STEM (Ciencia, Tecnología, Ingeniería y Matemáticas), al demostrar cómo la tecnología puede emplearse para representar fenómenos naturales de manera innovadora y accesible.

Justificación

Aprender sobre el sistema solar solo con textos e imágenes puede ser aburrido y difícil de entender. Usando programación y algoritmos podemos crear maquetas y simulaciones que muestren cómo se mueven los planetas, sus tamaños y distancias de una forma más práctica y entretenida. Esto no solo hace que las clases sean más fáciles de entender, sino que también ayuda a que los estudiantes aprendan a programar y usar tecnología mientras obtienen más conocimientos de los temas y a su vez aprenden de nuevas tecnologías.

Planteamiento del problema

Aprender sobre el sistema solar casi siempre es de una forma básica o normal, usando libros o videos que no muestran todo de manera dinámica o fácil de entender. Esto puede hacer que los estudiantes se aburran o no entiendan bien las distancias, movimientos y tamaños de los planetas. Con la programación y la automatización se podrían crear simuladores y maquetas interactivas que hagan el aprendizaje más visual y entretenido, pero estas herramientas casi no se usan en clase. Por eso, es importante buscar formas de aplicar la tecnología para que conocer el sistema solar sea más interesante y fácil de entender para los alumnos y poder tener una mejor comprensión sobre el tema.

Objetivos

Objetivo general

Crear una forma divertida e interactiva de aprender usando programación y automatización de sistemas.

Objetivos específicos

1. Hacer una maqueta que muestre el movimiento del sistema solar.
2. Crear un programa donde se pueda ver información y animaciones de la maqueta
3. Usar un motor de Arduino para simular fenómenos del espacio.
4. Enseñar nociones básicas de programación mientras se estudia los planetas.

Metodología del trabajo

Se realizó un análisis de la metodología de trabajo ideal para el equipo de trabajo Dream Team, se tomaron en cuenta dos en específico las cuales son:

1. **SCRUM:** Es una metodología de trabajo que generalmente se usa para equipos de trabajo grandes y empresas de gran tamaño, también para proyectos grandes y extensos, algunas de sus características son:
 - a. Trabaja en iteraciones fijas o también conocidas como sprints
 - b. Cuenta con roles definidos para el equipo de trabajo
 - c. Reuniones clave y de seguimiento constante
 - d. Entregables incrementables al final de cada sprint
 - e. Backlog o lista de pendientes priorizada

PROS:

- Claridad en los roles de cada integrante
- Ritmo predecible con plazos definidos
- Buena gestión en proyectos grandes
- Comunicación constante y mejora continua
- Asegura entregables frecuentes

CONTRAS:

- Puede llegar a sentirse rígido por los sprints
- Muchas reuniones
- Requiere compromiso y disciplina de todo el equipo
- Difícil de aplicar en equipos muy pequeños

2. **KANBAN:** Es una metodología que se basa más en el trabajo en conjunto de todos los integrantes por igual, algunas de sus características son:
 - a. Flujo continuo del trabajo, no hay sprints
 - b. Uso de un tablero visual
 - c. Se basa en WIP (Work in Progress)
 - d. No hay roles fijos

PROS:

- Simple y con fácil capacidad de adopción
- Es flexible y se adapta a cambios inmediatos
- Transparencia en el estado del trabajo en todo momento
- Menos reuniones

CONTRAS:

- Puede generar falta de enfoque si no hay límites claros
- No garantiza entregables en tiempos definidos

- Menos estructura en responsabilidades
- Riesgo de que se acumulen tareas si no se controla

Sin embargo, a la hora de querer adoptar alguna de las dos era difícil aplicar exactamente alguna de las dos metodologías, por lo que se buscó optar por una fusión de ambas la cual es llamada:

SCRUMBAN:

- a. Usa iteraciones cortas, pero con un tablero visual
- b. Tablero con limites WIP
- c. Roles flexibles
- d. Se reducen reuniones y se adapta a las necesidades
- e. Priorización continua del backlog

PROS:

- Combina la disciplina de una con la flexibilidad de la otra
- Menos reuniones que scrum
- Permite respuestas rápido a cambios
- Ideal para equipos en transición con proyectos mixtos
- Evita rigidez y caos

CONTRAS:

- Si no se define bien puede terminar en desorden
- Requiere madurez del equipo
- No es tan popularmente estandarizado
- Puede confundir si no se entienden ambas metodologías

Por lo que se optó trabajar bajo esta metodología ya que se hizo una definición clara de roles que habría dentro del equipo de desarrollo, sin embargo, no hubo una gran cantidad de sprints a la hora del trabajo ya que era un trabajo más flexible, contamos con reuniones ocasionales y permitía adaptarnos a cualquier eventualidad que pudiese surgir durante el proyecto.

Arquitectura del Proyecto

Para realizar la arquitectura del proyecto se pensó en todo el funcionamiento que debía tener el proyecto, como se iba a comportar durante las pruebas y como se iba a comportar de cara al usuario, también se pensó en todos los componentes y herramientas de software que nos iban a permitir que se realizaran todas las tareas que se programaron al proyecto, y se realizó una arquitectura por capas, en este proyecto contamos con lo que son 6 capas que son primordiales para el funcionamiento del proyecto.

Capa 1 – Capa de Dispositivos Hardware

Se utilizaron componentes físicos como lo son el Arduino UNO R4 WiFi para poder realizar la programación, pero también se utilizaron componentes como Protoboard, drivers, motores, jumpers, resistencias, leds y fuentes de alimentación, esto nos permitió interactuar con el lado físico del proyecto y que se pudiesen hacer los movimientos del sistema solar.

Capa 2 – Capa del Servidor/Backend

Esta capa nos permitió realizar dos funciones primordiales del sistema, como lo es el registro de nuevos usuarios y el inicio de sesión, también nos permitió hacer la creación de la base de datos en MySQL para almacenar los usuarios que se registraban en el sistema, esto es primordial para poder regular la cantidad de personas que tienen acceso a la web lo que nos lleva a obtener un factor de seguridad bueno, nos permitió también poder realizar la conexión de la base de datos de Firebase con la página web y con el Arduino, lo que nos permitía la comunicación entre el servidor y el hardware.

Capa 3 – Capa de Datos

Se realizó un análisis del monto que se proyectaba gastar para la elaboración del proyecto para así poder brindar un presupuesto que se adecuara al nivel del proyecto, se realizó un análisis presupuestal y después se realizó un análisis de lo que se gastó realmente, se realizó un Dashboard para poder ver gráficamente el monto que abarcó el proyecto, de igual manera dentro de la capa de datos podemos agregar la creación de la base de datos en MySQL que nos brindó la oportunidad de crear un inicio de sesión que almacenaría los datos de los usuarios.

Capa 4 – Capa de Presentación

Dentro de esta capa entra el desarrollo de la página web, todo el apartado visual que nos permite poder utilizar la maqueta y todas sus funciones, también nos permite desarrollar una página web intuitiva para que sea sencillo para el usuario navegar en ella sin complicaciones.

Capa 5 – Capa de Comunicación IoT

Esto nos permitió realizar la correcta comunicación entre el Arduino, la base de datos y la página web, el problema radicaba en el soporte que tenía el Arduino de los protocolos, ya que no era compatible con un protocolo HTTPS, por lo que se tuvo que utilizar protocolo HTTP para la comunicación con la base de datos, sin embargo la comunicación de la base de datos con la página web si se realiza por medio de HTTPS lo que permite tener seguridad de los datos de los usuarios que se registran en la web, ya que lo único que se transmite por HTTP son las acciones que debe seguir el Arduino.

Capa 6 – Capa de Ciberseguridad

Esta capa nos permitió realizar un sistema que almacenara los datos de forma segura, encriptando contraseñas a la hora de enviar los datos a la base de datos, también nos permitió realizar un sistema de accesos ya que a la página web se puede acceder solamente si el usuario cuenta con los accesos disponibles, de lo contrario no podrá ingresar, lo que nos ayudó a tener un grado de seguridad para el Arduino y evitar la sobrecarga de información y acciones que el Arduino puede ejecutar.

Conceptos básicos de la electrónica.

Antes de empezar algún de algo que esté relacionado con la electrónica hay que conocer los conceptos básicos que la componen, así que a continuación se presentan algunos conceptos básicos de electrónica de los materiales que se utilizaron en el proyecto:

¿Qué es una LED?

Una led es también conocida como un diodo emisor de luz. Dentro de la led hay un semiconductor que al ser atravesado por una tensión continua emite luz, existen diferente tipos de luz en función de las tecnologías que se utilicen para su creación o para los circuitos que se desean elaborar.

Las tecnologías de las LEDS es algo muy utilizado, uno de los primeros usos que se le puede dar a nivel comercial son los diodos infrarrojos que se usan para mandos a distancia de todo tipo de aparatos.

También se le puede utilizar en sistemas de iluminación basados en LED ya que son de bajo consumo y suelen tener mayor durabilidad.

Otra área donde se pueden ver mucho es a la hora de crear pantallas LED de gran tamaño, generalmente se suelen utilizar para publicidad.

¿Qué es un Arduino?

Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra.

Cuando se habla de Arduino no se habla de solamente un placa ya que cuenta con diferentes tipos de modelos, cada uno pensado para diferentes tipos de propósitos ya que cuentan con características variadas, algunos de ellos son:

- Arduino NANO
- Arduino MEGA
- Arduino LEONARDO
- Arduino UNO
- Arduino YUN

El uso de Arduino puede ser beneficioso ya que es un sistema libre y extensible, lo cual indica que cualquiera puede ampliar y mejorar el diseño hardware de las tablas y el entorno de desarrollo, esto permite que existan una variedad de placas

para diferentes propósitos y diferentes librerías para poder usar en nuestros proyectos.

Sistema Solar

El sistema solar es un sistema planetario constituido por una estrella que ejerce atracción gravitacional sobre los cuerpos celestes que girara su alrededor. Según la NASA “se cree que nuestro sistema solar se formó a partir de una sola nube plana de gas”. O la otra teoría que se dice es que se formó a través de un objeto de un gran tamaño que paso cerca del Sol, y que empujo una corriente de gas lejos de él.

La mayor parte del sistema solar, al igual que sucede con los demás sistemas planetarios es un espacio vacío ya que sin embargo alrededor de todo ese espacio existe multitud de objetos influenciados por la gravedad del sol.

El sistema solar se compone de la siguiente manera:

El Sol es la parte más importante del sistema solar ya que se encuentra en el centro y todos los objetos del sistema solar están influenciados por su gravedad. Se dice que es una estrella de Tipo G o que también son conocidas como enanas amarillas, que se encuentran aproximadamente en la mitad de su vida hasta el día de hoy tienen unos “4,600 millones de años”. El sol también está formado por tres cuartas partes de hidrógeno y una de helio, y gira sobre su propio eje alrededor del cual tarda “25 días” en dar una vuelta y por sí mismo representa aproximadamente del 99.86% de la masa total del sistema solar.

Los objetos más importantes del sistema solar son Los planetas, los planetas se dividen en dos clases diferentes que son:

Ocupando las órbitas internas del sistema solar se encuentra Mercurio, Venus, Tierra y Marte. Estos primeros planetas son de menor tamaño y por su posición en el sistema solar conocidos como los planetas interiores y por su naturaleza sólida de roca y de metal también denominados planetas rocosos.

En las órbitas más externas del sistema solar encontramos los planetas exteriores, mucho más grandes y compuestos por gas, ya que el motivo es porque son denominados Gigantes gaseoso y Gigantes de hielo, así que en orden a la distancia del sol encontramos a Júpiter, Saturno, Urano y Neptuno.

Y también existen los planetas más pequeños del sistema solar que son 5 denominados los planetas enanos, la característica es que poseen una gravedad suficiente como para haber adquirido una forma esférica, sin embargo, no la suficiente como para haber limpiado la vecindad de sus órbitas de otros objetos lo

que son diferencia a los demás planetas. Y estos planetas son Ceres ubicado en el cinturón de asteroides entre Marte y Júpiter los planetas enanos son Plutón, Haumea, Makemake y Eris, también denominados plutoides y localizados en el llamado cinturón de Kuiper.

¿Qué otros elementos tienen el Sistema Solar?

Entre los demás objetos que integran el sistema solar se encuentra los Asteroides (objetos espaciales rocosos que pueden tener kilómetros de diámetro) y los Meteoroides (pequeños fragmentos de materia, similares a la piedra o al metal). Estos elementos se pueden encontrar, como el cinturón de asteroides, una región entre las orbitas de Marte y Júpiter.

Los Cometas, cuerpos de hielo, polvo y gas que viajan a miles de kilómetros por hora por el espacio, también mantienen su propia órbita alrededor del Sol.

Y la NASA estima que existen más de 100 mil millones de cometas en el sistema solar.

El Cinturón de Asteroides: es una región del dolor sistema solar que se sitúa entre las orbitas de Marte y Júpiter que alberga una gran cantidad de pequeños objetos formados por rocas y hielo, en su mayoría asteroides. Los cuales se cree que son los restos de un planeta que nunca llego a formarse debido a la influencia Gravitatoria de Júpiter.

El Cinturón de Kuiper: es una región del sistema solar situada más allá de la órbita de Neptuno. Es similar al cinturón de asteroides, pero es mucho más grande: 20 veces más ancho y hasta 200 veces más masivo.

La Nube de Oort: Es una nube esférica de objetos que se encuentran más allá de la órbita de Neptuno, hasta a un año de luz de distancia del Sol. Según las estimaciones esta nube podría albergar entre 1,000 y 100,000 millones de objetos formados por hielo, metano y amoníaco que podría sumar una masa de 5 veces la del planeta Tierra.

Descubrimientos y Exploración

Se dice que algunas de las más antiguas civilizaciones conocieron al universo desde una perspectiva geocéntrica, como en babilonia en donde su visión del mundo estuvo representada de esta forma. Anaximandro, un filósofo griego presocrático, consideró a la Tierra como el centro del universo, imaginándola como un tambor equilibrado. Pitágoras introdujo la idea de la Tierra como esfera, basándose en eclipses. Platón y Aristóteles combinaron las ideas de Anaximandro y Pitágoras en un modelo geocéntrico. Sin embargo, fue Claudio Ptolomeo, en el

siglo II d.C., quien consolidó este modelo en su obra *Almagesto*, que dominó la astronomía durante casi 1300 años.

Aunque Aristarco y Aryabhata propusieron la teoría heliocéntrica en siglos anteriores, el modelo geocéntrico no fue realmente desafiado hasta que Nicolás Copérnico, en el siglo XVI, lo hizo. Su obra provocó una revolución en la astronomía,

y lo convirtió en el padre de la astronomía moderna. A diferencia de sus predecesores, su trabajo se difundió ampliamente, a pesar de que inicialmente fue concebido para circulación privada. El Papa Clemente VII mostró interés en 1533, mientras que Lutero lo descalificó en 1539. Copérnico describió dos movimientos de la Tierra: uno de rotación sobre su eje cada 24 horas y otro de traslación alrededor del Sol cada año, aunque consideraba que este movimiento era circular y no elíptico. En el siglo XVII, el trabajo de Copérnico fue potenciado por científicos como Galileo Galilei, quien, con el telescopio, descubrió satélites naturales orbitando Júpiter, lo que desafió la teoría geocéntrica. Esto generó un conflicto con la Iglesia, resultando en la condena de Galileo por herejía. Su contemporáneo Johannes Kepler, tras intentar explicar la traslación planetaria con órbitas circulares sin éxito, formuló las leyes de Kepler en 1609, estableciendo órbitas elípticas. Estas fueron confirmadas con su predicción del tránsito de Venus en 1631. Isaac Newton, por su parte, proporcionó una explicación del movimiento planetario a través de sus leyes y el concepto de gravedad.

¿En qué galaxia se ubica el sistema solar?

El sistema solar forma parte de nuestra galaxia, la Vía Láctea, una galaxia del tipo espiral barrada que tiene un diámetro aproximadamente 105,000 años luz entre sus extremos más distantes. En su estructura de la Vía Láctea está conformada por dos brazos secundarios, los de norma y sagitario.

Nuestro sistema solar se encuentra en el brazo de Orión o Local, el cual forma parte del brazo espiral de Sagitario. Es decir que el sol es la estrella alrededor de la que gira todo el sistema solar a su vez se mueve a los 210 kilómetros por segundo dentro de la Vía Láctea y tarda 225 millones de años en completar una vuelta alrededor del centro de la galaxia. Es lo que los científicos conocen como un año galáctico.

La pregunta que se hacen todos ¿Dónde empieza y acaba el sistema solar?

Sin duda se trata de una de las preguntas que no tiene una respuesta fácil ya que 4 de noviembre del año 2019, la NASA anuncia que la sonda espacial Voyager 2,

había abandonado el sistema solar tras 40 años de su partida desde la tierra. Su sonda gemela es la Voyager 1 ya lo había conseguido 7 años antes en marzo de 2012. Según la agencia espacial americana, las sondas amabas los objetos creados por el ser humano que más se han aleja do de la tierra jamás, había salido de la Heliosfera, es decir la región espacial que se encuentra bajo la influencia del viento solar y su campo magnético, y atravesado la Helio pausa, una línea imaginaria que constituye el límite de la heliosfera y en la que el viento solar se une al medio interestelar e interactúa con el viento estelar procedente de otras estrellas.

¿Como se formó el Sistema Solar?

Según los científicos tiene múltiples teorías que tratan y explican cómo se formó el sistema solar, sin embargo, una de las más aceptadas propone que antes de existiera el sistema solar, su lugar lo ocupaba una enorme nube de gas molecular que se acumulaba cada vez en mayores cantidades y densidad debido a las bajas temperaturas que imperan en la mayor parte del Universo. La teoría parece indicar que llegado cierto momento, bien debido al colapso gravitatorio de este gas interestelar (acreción por gravedad), o bien motivado por un aporte de energía procedente de la explosión de una estrella o supernova cercana, tuvo lugar el nacimiento de una protoestrella.

Esta protoestrella, es decir, nuestro Sol en estado gestación, continuó atrayendo gas y materia formando un disco de material alrededor de la misma y a partir del cual se formarían los planetas. Posteriormente, la protoestrella alcanzaría una densidad y presión suficiente como para que en su interior se iniciasen los procesos de fusión nuclear que caracterizan a estos astros, convirtiendo en su seno el hidrógeno en helio, y dando lugar a su vez al origen del viento interestelar que limpió de escombros las órbitas de los planetas actuales.

La dimensión astronómica de las distancias en el espacio

Para tener una noción de la dimensión astronómica de las distancias en el espacio, es interesante hacer un modelo a escala ya que permita tener una percepción más clara del mismo. Tan así que un ejemplo sería el modelo reducido en el que el Sol está representado por una pelota de 220 mm de diámetro. A esa escala a 23,6 m de una distancia y sería una esfera con apenas 2 mm de diámetro (la Luna estaría a unos 5cm de la tierra y tendría un diámetro de unos 0,5 mm).

Júpiter y Saturno sería bolitas con cerca de 2 cm de diámetro a 123 y a 226 m del Sol, respectivamente. Plutón estaría a 931 m del Sol, con cerca de 0,3 mm de diámetro. En cuanto a la estrella más próxima Centauri estaría a 6,332 km del Sol, y la estrella siria a 13,150 km.

Que es la electrónica.

Se llama electrónica a una disciplina técnica y científica, considerada como una rama de la física y como una especialización de la ingeniería, que se dedica al estudio y la producción de sistemas físicos basados en la conducción y el control de un flujo de electrones o de partículas cargadas eléctricamente.

Para ello, la electrónica se sirve no solo de ciertos principios teóricos básicos como el electromagnetismo, sino también de la ciencia de los materiales y otras formas de aplicación práctica del conocimiento científico. Sus resultados son de especial interés para otros campos del saber especializado, como la informática o la ingeniería de sistemas.

Diagrama de flujo del sistema general.

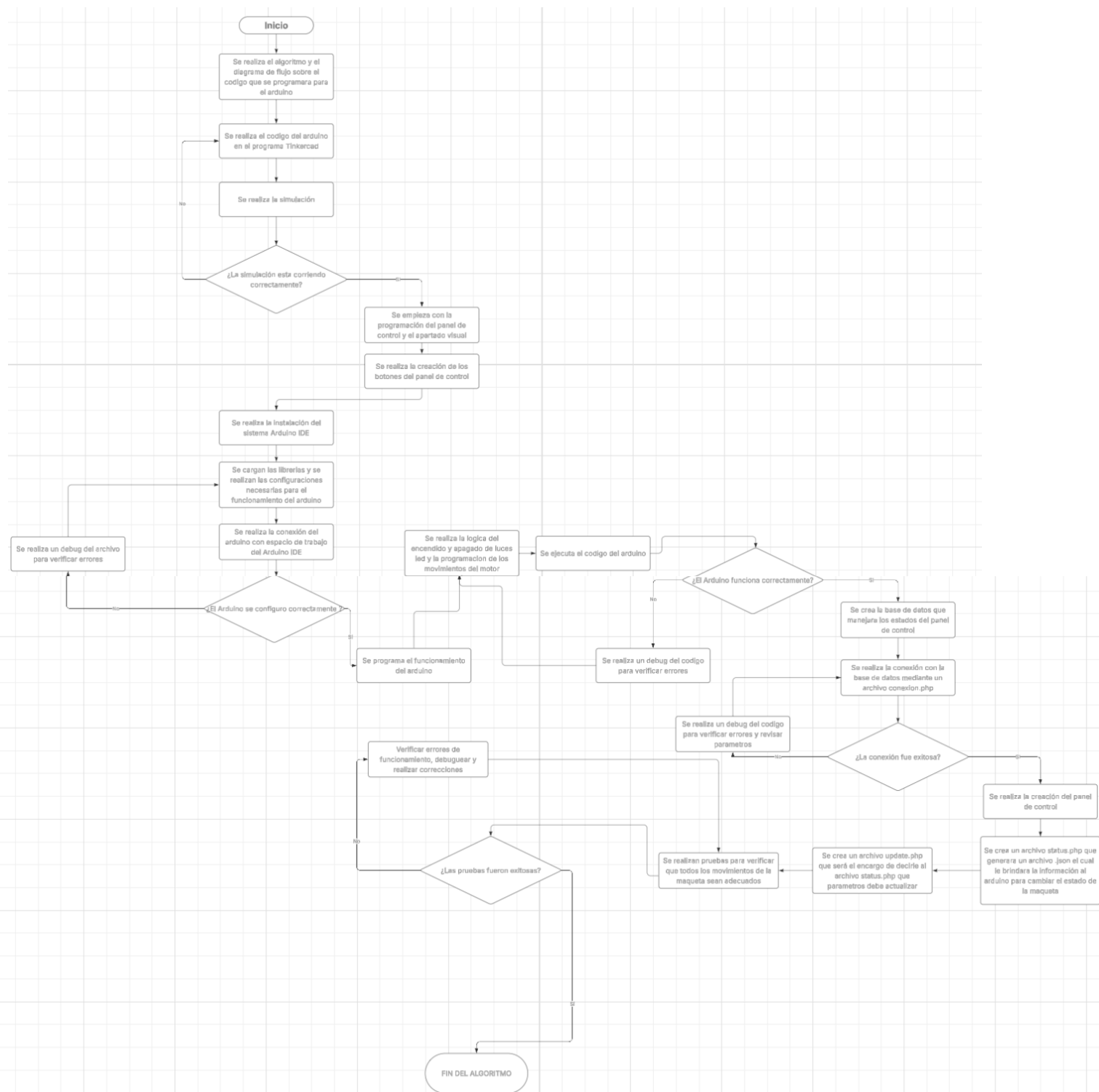
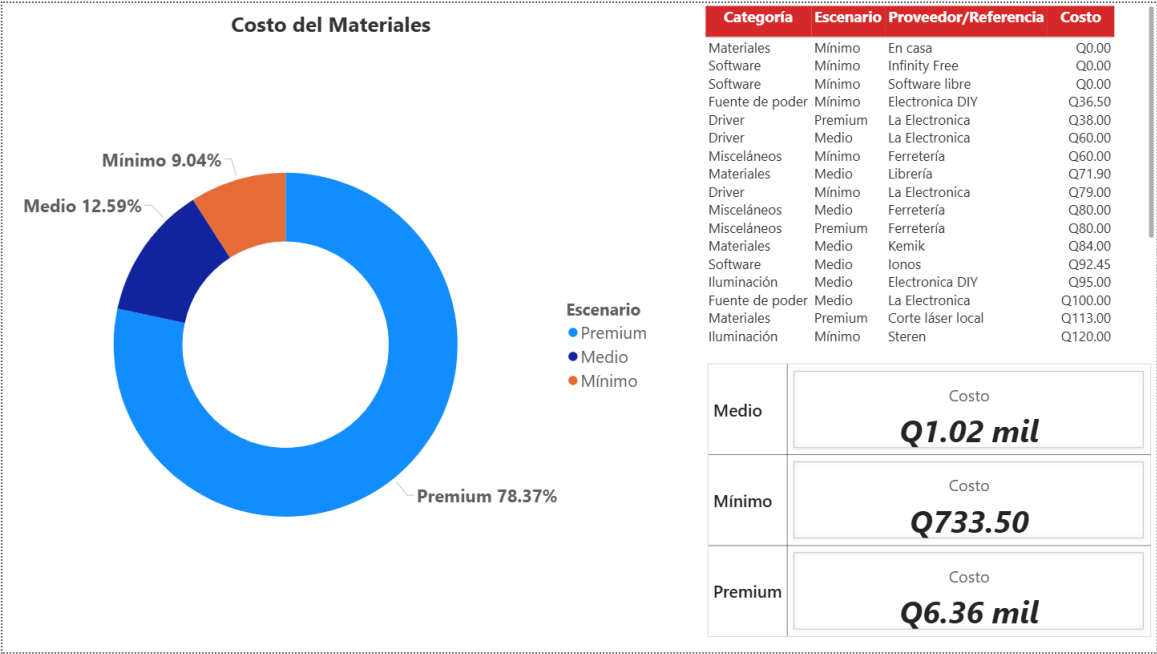
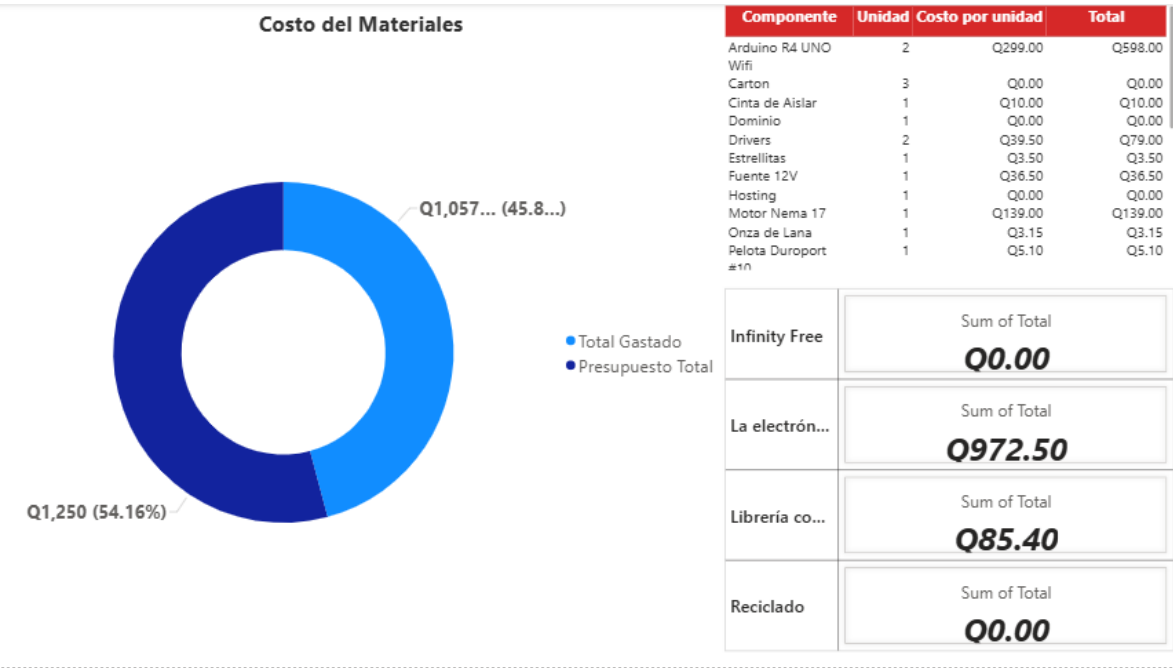


Diagrama de flujo del funcionamiento de maqueta

Proyección presupuestal.



Presupuesto.



Esce nario	Categoría	Componente	Costo	Proveedor/Referencia
Mínimo	Control	Arduino Uno R4 Wifi	Q 299.00	La Electronica
Mínimo	Motor	Motor Nema 17	Q 139.00	La Electronica
Mínimo	Iluminación	Tira LED	Q 120.00	Steren
Mínimo	Materiales	Cartón	Q -	En casa
Mínimo	Fuente de poder	Fuente 12V 2A económica	Q 36.50	Electronica DIY
Mínimo	Driver	Dos drivers	Q 79.00	La Electronica
Mínimo	Misceláneos	Cables, pegamento, tornillos	Q 60.00	Ferretería
Mínimo	Software	Hosting y Dominio	Q -	Infinity Free
Mínimo	Software	Software gratuito	Q -	Software libre
Medio	Control	Arduino Uno R4 Wifi	Q 299.00	La Electronica
Medio	Motor	Motor NEMA17	Q 139.00	La Electronica
Medio	Driver	Driver A4988	Q 60.00	La Electronica
Medio	Iluminación	Tira LED WS2812B 1m	Q 95.00	Electronica DIY
Medio	Fuente de poder	Fuente 12V 2A	Q 100.00	La Electronica
Medio	Materiales	Pinturas acrilicas 6 botes	Q 84.00	Kemik
Medio	Materiales	Otros	Q 71.90	Librería
Medio	Software	Hosting y Dominio	Q 92.45	Ionos
Medio	Misceláneos	Cables y materiales varios	Q 80.00	Ferretería
Premium	Control	Arduino Uno original	Q 230.00	Nexel electronics
Premium	Motor	Motor NEMA23	Q 319.00	La Electronica
Premium	Driver	Driver DRV8825	Q 38.00	La Electronica
Premium	Materiales	Acrílico y engranajes metálicos	Q 113.00	Corte láser local
Premium	Software	Autodesk Fusion (1 año)	Q 5,210.16	Autodesk
Premium	Misceláneos	Cables y accesorios	Q 80.00	Ferretería
Premium	Software	Hosting y Dominio	Q 369.81	Hostinger

Proyección presupuestal.

Presupuesto

Componente	Unidad	Costo	Proveedor/Referencia	Total	Presupuesto
		Q		Q	Q
Arduino R4 UNO Wifi	1	299.00	La electrónica	299.00	951.00
		Q		Q	Q
Motor Nema 17	1	139.00	La electrónica	139.00	812.00
		Q		Q	Q
Tira LED	1	120.00	La electrónica	120.00	692.00
		Q		Q	Q
Carton	3	Q -	Reciclado	Q -	692.00
		Q		Q	Q
Fuente 12V	1	36.50	La electrónica	Q 36.50	655.50
		Q		Q	Q
Drivers	2	39.50	La electrónica	Q 79.00	576.50
		Q		Q	Q
Hosting	1	Q -	Infinity Free	Q -	576.50
		Q		Q	Q
Dominio	1	Q -	Infinity Free	Q -	576.50
		Q		Q	Q
Pelota Duroport #3	1	0.60	Librería convencional	0.60	575.90
		Q		Q	Q
Pelota Duroport #4	2	1.00	Librería convencional	2.00	573.90
		Q		Q	Q
Pelota Duroport #5	2	1.15	Librería convencional	2.30	571.60
		Q		Q	Q
Pelota Duroport #6	3	1.60	Librería convencional	4.80	566.80
		Q		Q	Q
Pelota Duroport #9	1	3.25	Librería convencional	3.25	563.55
		Q		Q	Q
Pelota Duroport #10	1	5.10	Librería convencional	5.10	558.45
		Q		Q	Q
Pelota Duroport #14	1	7.60	Librería convencional	7.60	550.85
		Q		Q	Q
Onza de Lana	1	3.15	Librería convencional	3.15	547.70
		Q		Q	Q
Tempera Tucan Blanca	1	3.35	Librería convencional	3.35	544.35
		Q		Q	Q
Tempera Tucan Negra	1	11.75	Librería convencional	Q 11.75	532.60
		Q		Q	Q
Silicon Liquido	1	18.45	Librería convencional	Q 18.45	514.15
		Q		Q	Q
Pinchos de Madera	1	9.55	Librería convencional	9.55	504.60
		Q		Q	Q
Cinta de Aislar	1	10.00	Librería convencional	10.00	195.60
		Q		Q	Q
Estrellitas	1	3.50	Librería convencional	3.50	192.10

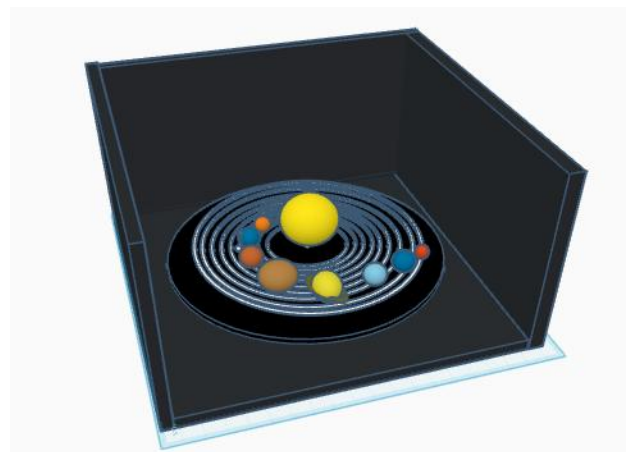
Maqueta

Materiales Para la Maqueta

- Cajas de Catón
- Pelotas de Duroport
- Pinchos de Madera
- Lana
- Temperas y Pinceles
- Silicon Frio / Caliente

Armado de Maqueta

- Se investigo sobre el sistema solar y el orden de sus planetas.
- Se comento en el grupo el tamaño que se realizaría la maqueta.
- Se realizo un boceto en **Tinkercad** de como iba a quedar la maqueta en físico.
- Se pinto cada esfera según el color y las características de cada planeta.
- Se recorto un círculo de cartón de tal manera que quepan todos los planetas en el círculo para disimular el sistema solar.
- También se cortó un cuadrado donde quede centrado el círculo de catón para hacer la base de la maqueta donde ira el **Motor** para que el círculo gire dependiendo los funcionamientos que se le pedirán y aparte se aparte se cortaron otros rectángulos de cartón que servirán para las paredes de la maqueta.
- Se pintó el círculo, el cuadrado y los rectángulos de cartón de color negro ya que disimulara el color del espacio.
- Se le coloco lana al círculo de cartón recreando cada **orbita** de cada Planeta del sistema solar.
- Se le coloco pinchos de madera a cada bolita de duroport (Planeta), y se colocó en el orden del sistema solar.
- Se le coloco también el nombre de cada planeta a las bolitas de duroport.



Funcionamiento de la Maqueta

- La maqueta cuenta con una base que sostiene los planetas asegurando la estabilidad.
- Se utilizar un **Motor** que estará ubicado en el centro de la maqueta, esto nos servirá para guiar el sistema solar con diferentes comandos que los cuales serían los siguientes:
 - ✓ Para Encender el Motor
 - ✓ Para Apagar el Motor
 - ✓ Hacer que el Motor Gire para el Lado Izquierdo
 - ✓ Hacer que el Motor Gire para el Lado Derecho
- El Motor estará conectado a la protoboard y Arduino, estos nos servirán para mandarle las señales al motor por medio de las funciones que se realice.
- También se le instalará una tira de **Luces LED** esto nos servirá para simular el resplandor de las estrellas.
- Las Luces LED también llevaran diferentes comandos que serían las siguientes:
 - ✓ Para encender las Luces LED
 - ✓ Para apagar las Luces LED
 - ✓ Para subirle la opacidad del Brillo
 - ✓ Para Bajarle la opacidad del Brillo
 - ✓ Y otro para cambiarle de colores.

Motor Nema17

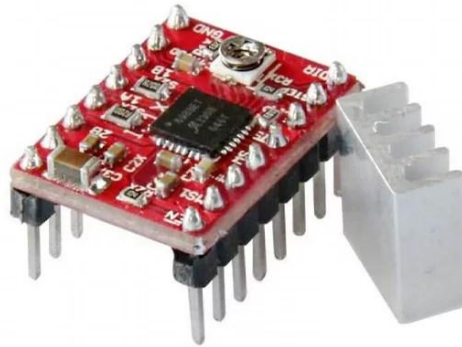
- Modelo: 17HS4401S
- Tipo de motor: bipolar
- Número de conductores: 4
- Fases: 2
- Resistencia del devanado: 1.5Ω
- Tensión de alimentación: 2.6-24 VDC
- Pasos por revolución: 200 (paso nominal)
- Ángulo de paso: 1.8°
- Consumo de corriente: 1.7 A / fase
- Torque: 0.43 Newton/metro o 4.3 kg/cm

Driver para motores paso a paso A4988

El driver A4988, es un módulo controlador de micropasos que simplifica el control de motores paso a paso bipolares. Lleva muchos años en el mercado, es uno de los más usados por su fiabilidad y bajo costo.

A los módulos que llevan el circuito integrado A4988 se les acostumbra a llamar Pololu, que es una marca comercial que popularizó el uso de este circuito integrado.

En la imagen inferior observamos al driver A4988. Es un módulo que sirve de soporte al circuito integrado del mismo nombre A4988; este integrado o chip está cubierto por un disipador de calor. En el módulo hay otros componentes necesarios para su funcionamiento, el más notorio es el potenciómetro para ajustar la intensidad de corriente máxima que se entregará al motor paso a paso, y se indica con una flecha.



Especificaciones técnicas A4988

Listemos las características técnicas más importantes del driver **A4988** extraídas de su *datasheet*.

- Tensión de trabajo: 3 a 5.5 VDC para la lógica y 8 a 35 VDC para el motor paso a paso
- Intensidad máxima: 2 A
- Color característico: verde o rojo
- Ajuste de intensidad de corriente mediante un potenciómetro de ajuste en el módulo
- Resolución: paso completo, 1/2 de paso, 1/4 de paso, 1/8 de paso y 1/16 de paso
- Pines de control de paso (*STEP*) y dirección (*DIRECTION*)
- Protección por alta temperatura, cortocircuito, sobre tensión y sobre intensidad
- Generan ruido en el motor, pero es muy económico

- Configuración de pines del driver A4988



Primero analicemos los terminales del lado izquierdo de la imagen, coloreados en azul, que son los pines de control.

- **ENABLE**

Esta es la habilitación del módulo y la línea en la parte superior nos indica que la función es activa en bajo, es decir, se debe mantener a nivel bajo (LOW) para que el módulo se encuentre habilitado o en modo de funcionamiento normal. Este pin tiene resistencia de pull-down que mantiene habilitado el módulo.

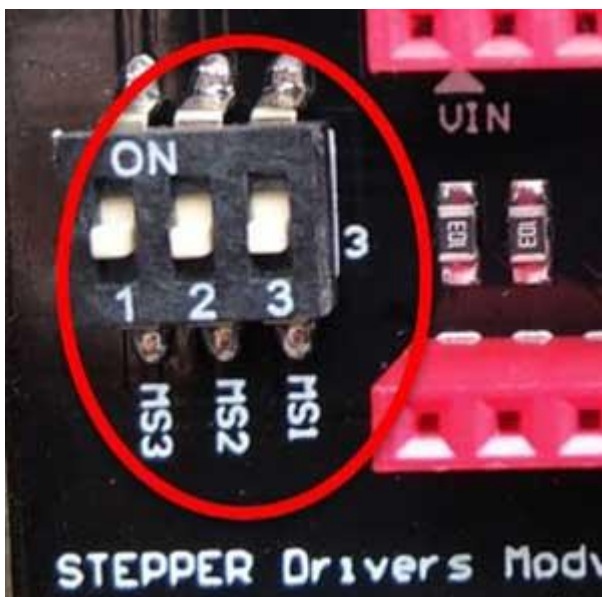
- **MS**

Los 3 pines MS o micropasos (*microstepping*), nos permite configurar los pasos de motor, según la siguiente tabla.

MS1	MS2	MS3	Resolución - Micropasos
Low	Low	Low	Paso completo
High	Low	Low	Medio paso
Low	High	Low	1/4 de paso
High	High	Low	1/8 de paso
High	High	High	1/16 de paso

Estas entradas MS, tienen resistencias internas de *pull-down* y por tanto están a nivel bajo (LOW) y si las dejamos «al aire» o sin conexión, estaremos en el primer caso de la tabla y por tanto el motor paso a paso trabajará en paso completo. Recordemos que en esta configuración se obtiene aproximadamente el 71% de la intensidad total de la corriente que entrega el controlador, a diferencia de los micropasos que obtienen el 100%, en algún momento de su funcionamiento, pero no de forma permanente.

Con el uso de la tarjeta de expansión para el driver **A4988** se facilita la configuración de los micropasos por medio de tres pequeños interruptores que están montados en un solo paquete (*DIP switch* de 3 posiciones) que se muestra en la imagen inferior.



Si utilizas un escudo RAMPS puedes controlar los micropasos mediante puentes (*jumpers*) que trae para cada uno de los cinco controladores que puede contener.

Hay que considerar que a medida que aumentamos la resolución del motor, el torque disminuye.

- **RESET**

Permite restablecer o reiniciar el módulo **A4988** y es activo en bajo. Este pin no tiene resistencia de *pull-up*.

- **SLEEP**

Coloca al circuito integrado en un modo de bajo consumo, también activo en bajo. Este pin sí tiene resistencia de *pull-up*.

- **STEP**

Arduino utiliza este pin para enviar un pulso y que el motor realice un paso.

- **DIR**

Arduino enviará un alto (HIGH) o un bajo (LOW) a este pin y lograremos que el motor gire en una dirección u otra.

Ahora los pines del lado derecho, que son los pines de alimentación y del motor paso a paso.

- **Vmotor y GND**

Conexión para una fuente de alimentación externa para la etapa de potencia del driver **A4988** y las bobinas del motor paso a paso, está comprendida entre 8 y 35 VDC. El valor de tensión y la capacidad de corriente de esta fuente de alimentación dependerán del modelo del motor paso a paso que utilicemos.

- **2B y 2A**

Conexión para la bobina 2 del motor paso a paso.

- **1A y 1B**

Conexión para la bobina 1 del motor paso a paso.

- **VDD y GND**

Es la alimentación para la parte lógica del circuito integrado **A4988**, puede estar comprendida entre 3 y 5.5 VDC. El consumo de corriente de esta sección es baja y por tanto podemos tomar la alimentación directamente desde Arduino.

Sistema de refrigeración: disipador de calor

Si la corriente en el motor paso a paso es inferior a 1 A, el driver **A4988** puede trabajar sin disipador de calor de aluminio; para corrientes superiores a 1 A y hasta máximo 2 A, se debe utilizar un disipador de calor y de ser posible ventilación.



En la imagen superior vemos un disipador de calor para el driver **A4988**. Para adherirse al chip, el disipador trae en la parte inferior una cinta con un protector para el adhesivo, la cinta debe retirarse al momento de colocarse en el chip.

¿Cómo limitar la corriente?

El driver **A4988** posee un potenciómetro para ajustar la intensidad de corriente máxima que entregará a las bobinas del motor paso a paso. Esto lo hacemos para ajustarnos al motor específico para nuestros proyectos y se debe reajustar si cambiamos de motor por otro de diferentes características, por ejemplo de un **NEMA 17** a un **NEMA 23**.

El fabricante del circuito integrado nos ofrece una fórmula matemática para limitar la corriente del módulo.

$$I_{max} = \frac{V_{ref}}{8 * RS}$$

Despejemos a V_{ref} , que es lo que necesitamos calcular:

$$V_{ref} = \frac{I_{max}}{8 * RS}$$

Donde V_{ref} , es el voltaje o tensión de referencia que debemos ajustar por medio del potenciómetro colocado en el módulo **A4988**.

I_{max} , es la máxima corriente que entregará el módulo **A4988**.

RS (*Resistor Sense*) es una de las dos resistencias en derivación (*shunt*) o resistencias de detección, una para cada fase o bobina del motor paso a paso. Son llamadas S1 y S2 (*Sense 1 y Sense 2*), otras veces R4 y R5, cuyos valores pueden estar comprendidos entre 0.01 hasta 0.2 Ω . Debes encontrar el valor de esas resistencias en tu driver **A4988**; un valor típico de RS es de 0.1 Ω , pero hay que verificarlo.



En la imagen superior vemos que el valor de RS es R100, según el código de Dispositivos de Montaje Superficial, SMD (*Surface Mount Device*) cambiamos la R por un punto decimal, entonces $RS = 0.1 \Omega$.

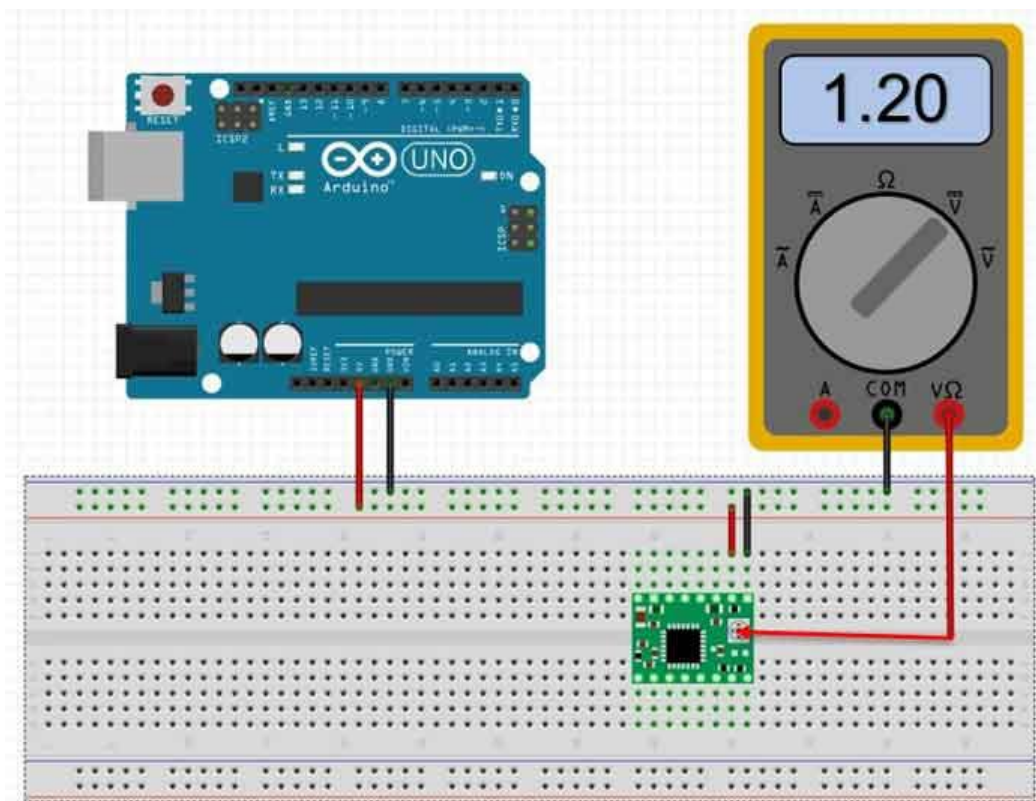
Aunque el *datasheet* del motor NEMA 17 de referencia 17HS4401S, que usaremos para estos ejemplos, nos indica una corriente máxima de 1.7 A por fase o bobina, usaremos una corriente de 1.5 A para no trabajar al límite.

Reemplazamos en la ecuación:

$$V_{ref} = 1.5 * 8 * 0.1$$

$$V_{ref} = 1.2V$$

Para hacer este ajuste de corriente, solo debemos conectar los 5 VDC y GND de nuestro Arduino a la parte lógica del driver **A4988**, VDD y GND.

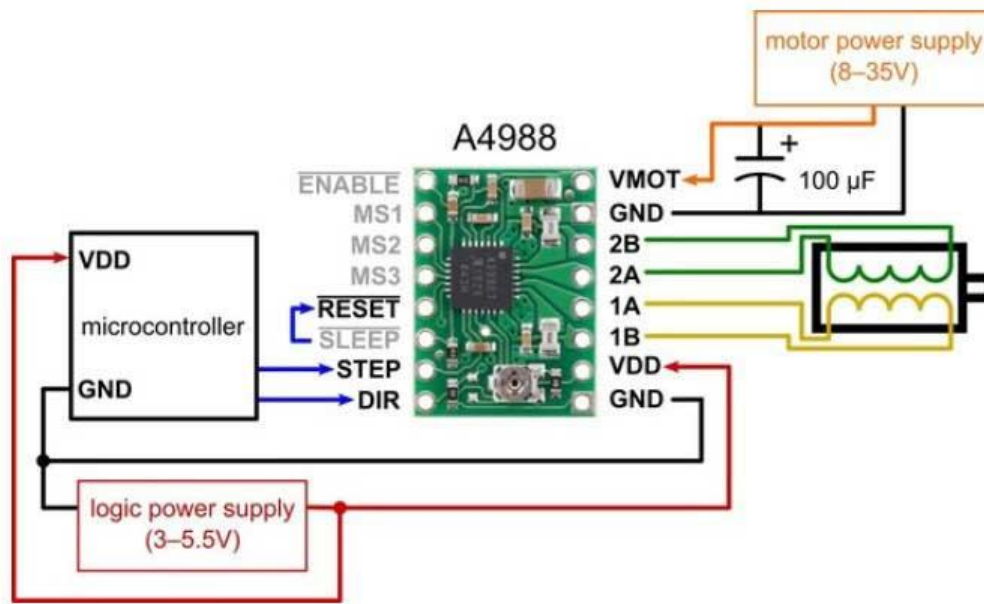


Con ayuda de un «cocodrilo» o «caimán» unimos la punta roja o positiva del polímetro o multímetro, que deberá estar ajustado como voltímetro en una escala de 2VDC, a la punta metálica de un destornillador pequeño y la punta negra o negativa a cualquier GND; luego giramos lentamente y con cuidado la perilla del potenciómetro midiendo la tensión en el cursor o parte móvil del potenciómetro hasta llegar al valor de tensión hallado en la fórmula, en tu caso puede ser un valor diferente ya que depende del valor de corriente que desees limitar, de la R_S de tu controlador **A4988** y de la corriente del motor paso a paso que utilices.

De esta forma, sin importar los micropasos que utilicemos, el controlador **A4988**, no proporcionará una intensidad de corriente superior a 1.5A; si por algún motivo el motor paso a paso exige una corriente mayor, el driver **A4988** quitará la alimentación de las bobinas del motor, protegiéndolas de un posible daño.

Esquema de conexiones Arduino, A4988 y motor paso a paso

Usaremos para el ejemplo básico el diagrama de Pololu para conectar nuestro controlador Arduino con el driver **A4988** y el motor paso a paso NEMA17, de referencia 17HS4401S, este esquema es el mismo que recomienda el fabricante del chip **A4988** Allegro MicroSystems.

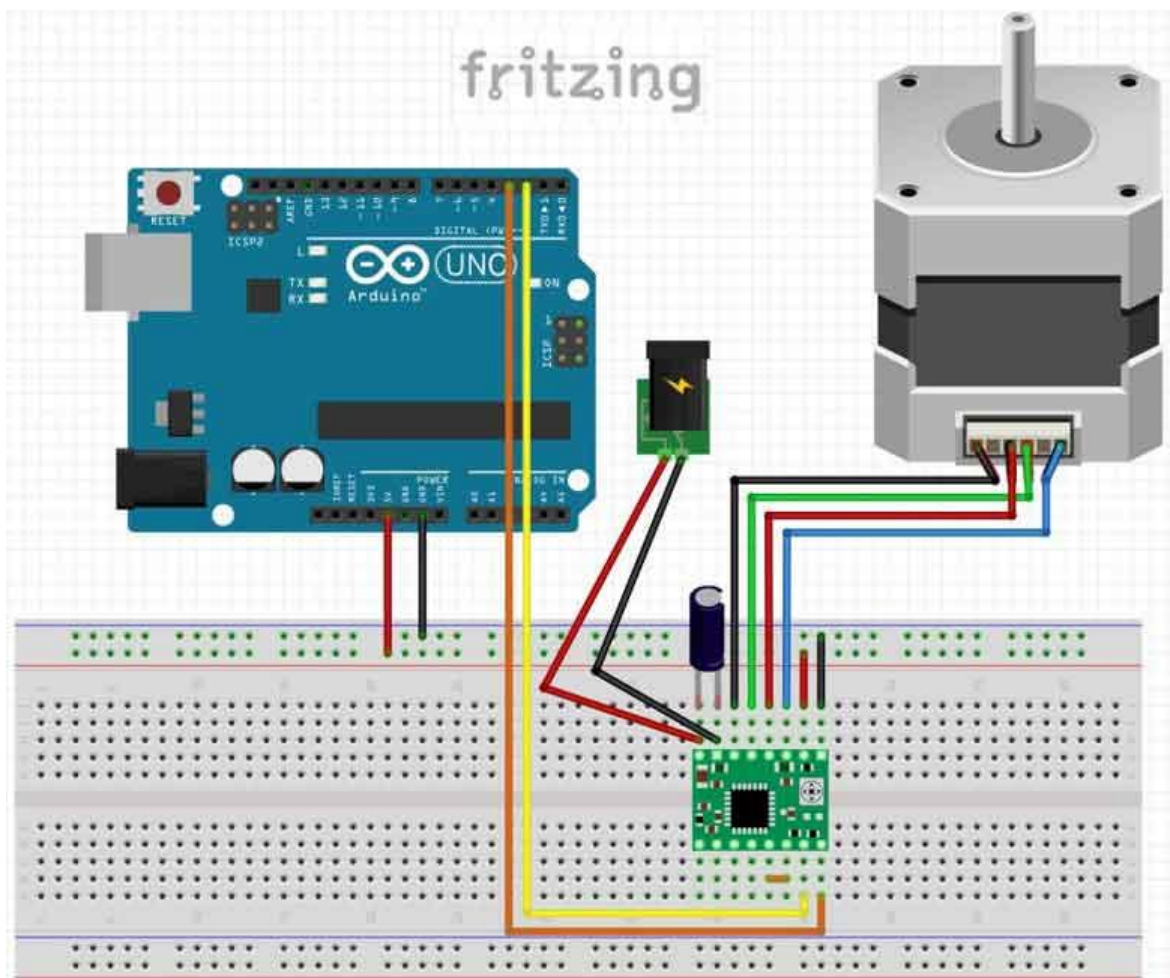


Recuerda: por norma general, no debemos conectar y desconectar las cargas de un circuito electrónico estando encendido, conectar o desconectar el motor paso a paso con el driver **A4988** energizado, no es la excepción, podemos destruir el driver.

Cómo se usa el driver A4988

Para todos nuestros ejemplos usaremos el paso completo: MS1, MS2 y MS3 al aire si usamos los *drivers* en *protoboard*, si usamos tarjeta de expansión, todos los interruptores en 0 (OFF) y si usamos una RAMPS, no colocamos puentes (*jumper*s) en los pines ubicados debajo de cada *driver*. En modo de paso completo tomaremos solo el 70.7% de la intensidad máxima calculada.

También puedes variar la resolución del motor paso a paso variando los pasos según la tabla y observar el cambio en su velocidad, por ejemplo se puede colocar a MS1 en alto o puentear sus pines y trabajar en modo de medio paso y aprovechar el 100% de la intensidad máxima, el recorrido del eje, será la mitad. También se puede ajustar un poco el V_{REF} según las fórmulas, si se nota saltos de pasos en el motor durante su funcionamiento final, siempre respetando la intensidad máxima soportada por el motor.



Consideraciones:

- El controlador **A4988** internamente comunica GND de la fuente de baja tensión de la lógica y GND de la fuente de mayor tensión para los transistores de potencia, por tanto no hay necesidad de hacer comunes los GND externamente, aunque podemos hacerlo ya que es costumbre y es necesario en los circuitos que utilizan dos o más fuentes de alimentación independientes.
- Si usas el módulo **A4988** en una *protoboard* sin su tarjeta de expansión o sin tarjeta RAMPS, debes colocar un condensador de 100 $\mu\text{F}/35\text{ V}$ en los pines de alimentación de la parte de potencia del chip, respetando la polaridad de sus terminales. Es necesario este condensador ya que el controlador **A4988** utiliza condensadores cerámicos de baja Resistencia Equivalente en Serie, ESR (*Equivalent Series Resistance*) que los hace susceptibles de daño con los picos de tensión.

- Los pines de ajuste de resolución de pasos MS1, MS2 y MS3, pueden quedar sin conexión porque tienen resistores de *pull-down* y trabajaremos a paso completo.
- El pin de ENABLE puede quedar sin conexión ya que también tiene su resistor de *pull-down* que habilita el módulo.
- Conectamos con un puente (*jumper*) los pines RESET y SLEEP para aprovechar el resistor de *pull-up* de SLEEP y mantener en un nivel alto a RESET. Podemos hacer esto ya que no usaremos estos pines en nuestro circuito básico.
- Los colores de una bobina del motor paso a paso NEMA 17 son negro y verde, la otra bobina son rojo y azul.

Referencias

<https://programarfacil.com/blog/arduino-blog/a4988-drv8825-tmc2209/>

<https://docs.arduino.cc/hardware/uno-r4-wifi/>

<https://www.pololu.com/product/1182>

<https://lastminuteengineers.com/a4988-stepper-motor-driver-arduino-tutorial/>

Información de material utilizado.

Arduino

Es una plataforma de electrónica abierta basada en hardware y software fáciles de usar, que permite a cualquier persona —incluso sin experiencia previa— crear proyectos electrónicos interactivos.

Protoboard

Una protoboard (también conocida como breadboard) es una placa de plástico con orificios que se usa para construir y probar circuitos electrónicos sin necesidad de soldar.

Tira led WS2812B

Una WS2812B es una tira de LEDs direccionables que permite controlar individualmente el color y brillo de cada LED usando una señal digital. Cada LED contiene un pequeño chip controlador integrado, lo que facilita su programación desde microcontroladores como Arduino, ESP32, entre otros.

Resistencia

Una resistencia (o resistor) es un componente electrónico que se usa para limitar o controlar el flujo de corriente eléctrica en un circuito. Es uno de los componentes más básicos y comunes en electrónica.

Diodo electrolítico

Un condensador electrolítico es un tipo de condensador polarizado que almacena energía eléctrica en forma de campo eléctrico. Se utiliza principalmente para filtrar, suavizar o estabilizar voltajes en circuitos electrónicos.

Fuente de poder

Una fuente de poder es un dispositivo que convierte la energía eléctrica de una forma a otra, generalmente de corriente alterna (AC) a corriente continua (DC), para alimentar circuitos y dispositivos electrónicos.

Reductor de voltaje step-down

Un reductor de voltaje step-down es un dispositivo electrónico que convierte un voltaje alto en uno más bajo de forma eficiente, sin desperdiciar mucha energía.

Driver A4988

El A4988 es un driver o controlador de motor paso a paso (stepper motor driver) muy popular y económico. Se usa para controlar motores bipolares como los NEMA 17, especialmente en impresoras 3D, CNCs, robots y proyectos con Arduino o similares.

Códigos y configuraciones

Arduino Código

```
#include <WiFiS3.h> // Librería para manejar la conexión WiFi en Arduino UNO R4 WiFi
```

```
#include <Adafruit_NeoPixel.h> // Librería para controlar tiras de LEDs WS2812B
```

Configuración WIFI

```
char ssid[] = "CLARO"; // Nombre de la red WiFi a la que nos conectaremos
```

```
char pass[] = "123456"; // Contraseña de la red WiFi
```

```
WiFiServer server(80); // Inicializamos un servidor HTTP en el puerto 80
```

Configuración de LEDS

```
#define LED_PIN 6 // Pin de datos de la tira LED WS2812B
```

```
#define LED_COUNT 30 // Número de LEDs en la tira (1 metro ≈ 30 LEDs)
```

```
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800); // Configuración de la tira
```

```
bool ledsOn = false; // Estado general de los LEDs: encendido (true) o apagado (false)
```

```
int ledMode = 0; // Modo LED activo (0 a 5)
```

```
uint32_t fixedColor = strip.Color(0,128,0); // Color fijo por defecto (verde)
```

```
uint16_t rainbowIndex = 0; // Índice usado en efectos de colores que se desplazan
```

```
uint8_t ledBrightness = 80; // Brillo actual de los LEDs
```

```
uint8_t targetBrightness = 80; // Brillo objetivo (para cambios suaves)
```

Nombres y descripciones cortas de los modos LED para la página web

```
const char* ledNames[] = {  
    "Aleatorio", "Arcoíris", "Parpadeo", "Color Fijo", "Planeta", "Aurora"  
};  
  
const char* ledHints[] = {  
    "Colores cambian aleatoriamente",  
    "Colores se desplazan como arcoíris",  
    "Todos los LEDs parpadean",  
    "Color sólido fijo",  
    "Simula planetas brillando",  
    "Suave efecto aurora boreal"  
};
```

Configuración del motor

```
#define STEP_PIN 3      // Pin que enviará pulsos al motor paso a paso  
#define DIR_PIN 4      // Pin para definir dirección de giro del motor  
#define EN_PIN 5       // Pin para habilitar/deshabilitar el motor  
  
bool motorRunning = false; // Estado del motor: encendido/apagado  
int motorDir = 0;         // Dirección del motor: 0 = derecha, 1 = izquierda  
int motorSpeed = 500;     // Velocidad del motor: mayor valor = más lento  
unsigned long lastStepTime = 0; // Tiempo en micros del último pulso enviado al motor
```

Funciones de inicio

```
void setup() {  
    Serial.begin(115200);    // Inicializamos puerto serial para depuración  
    while(!Serial);         // Esperar a que el puerto serial esté listo  
  
    connectWiFi();           // Conectamos a la red WiFi  
    server.begin();          // Iniciamos el servidor web  
}
```

Configuración inicial de la tira de LEDs

```
strip.begin();              // Inicializamos la tira  
strip.setBrightness(ledBrightness); // Configuramos brillo inicial  
strip.show();               // Apagamos los LEDs al iniciar
```

Configuración del motor

```
pinMode(STEP_PIN, OUTPUT);  // Pin de pulso como salida  
pinMode(DIR_PIN, OUTPUT);   // Pin de dirección como salida  
pinMode(EN_PIN, OUTPUT);    // Pin de habilitación como salida  
digitalWrite(EN_PIN, LOW);  // Habilitar motor al inicio  
}
```

Lool principal

```
void loop() {  
    handleClient(); // Revisamos si hay un cliente web conectado y atendemos peticiones  
    handleMotor();  // Movemos el motor paso a paso sin bloquear el código  
    handleLEDs();   // Actualizamos los LEDs según el modo activo  
    smoothBrightness(); // Ajustamos el brillo suavemente si se cambió  
}
```

Funciones de soporte

```
// --- Conectar a la red WiFi de manera robusta ---  
  
void connectWiFi() {  
    Serial.println("Intentando conectar a WiFi...");  
    int status = WL_IDLE_STATUS; // Estado inicial de conexión  
  
    // Verificar si hay módulo WiFi disponible  
    if (WiFi.status() == WL_NO_MODULE) {  
        Serial.println("No se detecta el módulo WiFi.");  
        while (true); // Detener el código si no hay módulo  
    }  
  
    // Intentar conectarse hasta lograrlo  
    while (status != WL_CONNECTED) {  
        Serial.print("Conectando a SSID: ");  
        Serial.println(ssid);  
        status = WiFi.begin(ssid, pass); // Intentar conexión  
        delay(10000); // Esperar 10 segundos antes de reintentar  
    }  
  
    Serial.print("Conectado a WiFi: ");  
    Serial.println(WiFi.SSID()); // Mostrar SSID conectado  
    Serial.print("IP asignada: ");  
    Serial.println(WiFi.localIP()); // Mostrar IP obtenida  
}
```



```
// --- Manejo del cliente web ---  
  
void handleClient(){  
  
    WiFiClient client = server.available(); // Revisamos si hay un cliente conectado  
  
    if(!client) return;                // Si no hay cliente, salimos  
  
  
    String req = "";                    // Guardaremos la petición HTTP  
  
    unsigned long startTime = millis(); // Para implementar timeout  
  
    // Leer petición HTTP  
    while(client.connected() && millis() - startTime < 50){  
        while(client.available()){  
            char c = client.read(); // Leer byte  
            req += c;               // Agregar a la cadena  
        }  
    }  
  
  
    // --- COMANDOS WEB ---  
  
    if(req.indexOf("/MOTOR=LEFT") != -1) motorRunning = true, motorDir = 1;  
    if(req.indexOf("/MOTOR=RIGHT") != -1) motorRunning = true, motorDir = 0;  
    if(req.indexOf("/MOTOR=OFF") != -1) motorRunning = false;  
    if(req.indexOf("/MOTOR=ON") != -1) motorRunning = true;  
  
  
    if(req.indexOf("/LED=ON") != -1) ledsOn = true;  
    if(req.indexOf("/LED=OFF") != -1) ledsOn = false;  
  
  
    if(req.indexOf("/SPEED=UP") != -1 && motorSpeed > 100) motorSpeed -= 50;
```

```
if(req.indexOf("/SPEED=DOWN") != -1 && motorSpeed < 1500) motorSpeed += 50;

// Cambiar modo LED
for(int i=0;i<=5;i++){
    if(req.indexOf("/LEDMODE="+String(i)) != -1) ledMode = i;
}

// Ajustar brillo desde la web
int idx = req.indexOf("/BRIGHTNESS=");
if(idx != -1){
    targetBrightness = req.substring(idx + 12).toInt();
    if(targetBrightness > 255) targetBrightness = 255;
    if(targetBrightness < 0) targetBrightness = 0;
}

// --- RESPUESTA WEB ---
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html\n");
client.println("<!DOCTYPE HTML>");
client.println("<html><head><meta    name='viewport'    content='width=device-width,
initial-scale=1'>");
client.println("<title>Control Sistema Solar</title>");
client.println("<style>body{font-family:Arial;text-
align:center;background:#111;color:#eee;}\"

\"h2{color:#FFD700;}button{background:#444;border:none;color:#fff;padding:10px
20px;margin:5px;\"
```

```
"font-size:16px;border-
radius:8px;cursor:pointer;}button:hover{background:#666;}"

".active{background:#FFD700;color:#000;}input[type=range]{width:80%;margin:10px
0;}</style>");

client.println("</head><body>");

client.println("<h2>Control Sistema Solar</h2>");

// --- MOTOR ---

client.println("<div class='section'><h3>Motor</h3>");

client.println("<button class='"+String((motorRunning && motorDir==1)?"active":"")+"'
onclick=\"location.href='/MOTOR=LEFT'\">Izquierda</button>");

client.println("<button class='"+String((motorRunning && motorDir==0)?"active":"")+"'
onclick=\"location.href='/MOTOR=RIGHT'\">Derecha</button>");

client.println("<button class='"+String((motorRunning)?"active":"")+"'
onclick=\"location.href='/MOTOR=ON'\">Encender</button>");

client.println("<button class='"+String((!motorRunning)?"active":"")+"'
onclick=\"location.href='/MOTOR=OFF'\">Apagar</button><br>");

client.println("<button onclick=\"location.href='/SPEED=UP'\">+ Velocidad</button>");

client.println("<button onclick=\"location.href='/SPEED=DOWN'\">-
Velocidad</button></div>");

// --- LEDs ---

client.println("<div class='section'><h3>LEDs</h3>");

client.println("<button class='"+String((ledsOn)?"active":"")+"'
onclick=\"location.href='/LED=ON'\">Encender</button>");

client.println("<button class='"+String((!ledsOn)?"active":"")+"'
onclick=\"location.href='/LED=OFF'\">Apagar</button><br>");

for(int i=0;i<=5;i++){
```

```
        client.println("<button                                class='"+String((ledMode==i)?"active":"")+"'
onclick='"+location.href="/LEDMODE="+String(i)+"'>");

        client.print(ledNames[i]); // Nombre del efecto

        client.print(" - ");

        client.print(ledHints[i]); // Descripción corta

        client.println("</button>");

    }
```

```
    client.println("<div class='section'><h3>Brillo LEDs</h3>");

    client.println("<input type='range' min='0' max='255' value='"+String(targetBrightness)+"'
onchange='"+location.href="/BRIGHTNESS="+this.value"></div>");

    client.println("<div                                class='section'><p>Velocidad                                motor:
"+String(map(motorSpeed,100,1500,10,1))+</p></div>");

    client.println("</body></html>");

    client.stop(); // cerrar conexión

}
```

```
// --- MOTOR ---
```

```
// Mueve el motor paso a paso de manera no bloqueante
```

```
void handleMotor(){
```

```
    unsigned long currentTime = micros(); // Tiempo actual en microsegundos
```

```
    if(motorRunning){
```

```
        if(currentTime - lastStepTime >= motorSpeed){ // Control de velocidad
```

```
            digitalWrite(DIR_PIN,motorDir); // Establecer dirección
```

```
            digitalWrite(STEP_PIN,HIGH); // Pulso HIGH
```

```
            delayMicroseconds(50); // Pulso corto
```

```
            digitalWrite(STEP_PIN,LOW); // Pulso LOW
```

```
    lastStepTime = currentTime;    // Actualizar tiempo
  }
}
}

// --- LEDs ---

// Llama al efecto LED correspondiente según ledMode cada 20ms
void handleLEDs(){
  static unsigned long lastUpdate = 0;
  if(!ledsOn || millis() - lastUpdate < 20) return; // actualizar cada 20ms
  lastUpdate = millis();
  switch(ledMode){
    case 0: randomColors(); break;
    case 1: rainbowCycle(); break;
    case 2: blinkAll(); break;
    case 3: solidColor(); break;
    case 4: planetEffect(); break;
    case 5: auroraEffect(); break;
  }
}

// --- EFECTOS LED ---

void randomColors(){
  for(int i=0;i<LED_COUNT;i++)
    strip.setPixelColor(i,strip.Color(random(0,128),random(0,128),random(0,128)));
  strip.show();}
```

```
void rainbowCycle(){  
    for(int i=0;i<LED_COUNT;i++)  
        strip.setPixelColor(i,Wheel((i+rainbowIndex)&255,ledBrightness));  
    strip.show();  
    rainbowIndex++;  
}
```

```
void blinkAll(){  
    static bool on = false;  
    for(int i=0;i<LED_COUNT;i++)  
        strip.setPixelColor(i,on?strip.Color(ledBrightness,ledBrightness,ledBrightness):strip.Color(0,0,0));  
    strip.show();  
    on = !on;  
}
```

```
void solidColor(){  
    for(int i=0;i<LED_COUNT;i++) strip.setPixelColor(i,fixedColor);  
    strip.show();  
}
```

```
void planetEffect(){  
    for(int i=0;i<LED_COUNT;i++){  
        uint32_t baseColor = Wheel((i*8+rainbowIndex)&255,ledBrightness);  
        if(random(0,100)<10) baseColor =  
            strip.Color(ledBrightness,ledBrightness,ledBrightness);  
        strip.setPixelColor(i,baseColor);  
    }
```

```
}

strip.show();

rainbowIndex++;

}

void auroraEffect(){
  for(int i=0;i<LED_COUNT;i++){
    uint8_t r = (sin((i*0.3+rainbowIndex*0.05))*(ledBrightness/2)+(ledBrightness/2));
    uint8_t g = (sin((i*0.2+rainbowIndex*0.03)+2)*(ledBrightness/2)+(ledBrightness/2));
    uint8_t b = (sin((i*0.25+rainbowIndex*0.04)+4)*(ledBrightness/2)+(ledBrightness/2));
    if(random(0,100)<5) {r=ledBrightness; g=ledBrightness; b=ledBrightness;}
    strip.setPixelColor(i,strip.Color(r,g,b));
  }
  strip.show();
  rainbowIndex++;
}

// --- TRANSICIÓN SUAVE DE BRILLO ---
void smoothBrightness(){
  if(ledBrightness == targetBrightness) return; // Si ya llegó al valor, no hacer nada
  if(ledBrightness < targetBrightness) ledBrightness++; // Incrementar brillo
  else ledBrightness--; // Decrementar brillo
  strip.setBrightness(ledBrightness); // Aplicar al strip
}

// --- UTILIDAD PARA EFECTO ARCOÍRIS ---
```

// Convierte un valor de 0-255 a un color RGB para efecto arcoíris

```
uint32_t Wheel(byte WheelPos, uint8_t bright){  
    WheelPos = 255 - WheelPos;  
  
    if(WheelPos < 85) return strip.Color((255 - WheelPos * 3)*bright/255, 0, (WheelPos *  
3)*bright/255);  
  
    if(WheelPos < 170) {WheelPos -= 85; return strip.Color(0,(WheelPos*3)*bright/255,(255 -  
WheelPos*3)*bright/255);}  
  
    WheelPos -= 170;  
  
    return strip.Color((WheelPos*3)*bright/255,(255 - WheelPos*3)*bright/255,0);  
}
```


Herramientas y software

Arduino R4 UNO Wifi: Existen muchos métodos que se pueden utilizar para poder desarrollar el proyecto, sin embargo, la decisión de utilizar el Arduino se dio por varios factores que nos podían simplificar el desarrollo del proyecto:

- El código de Arduino esta basado en C/C++ de manera simplificada y es ideal para principiantes que empiezan en este mundo.
- Existen miles de tutoriales, documentaciones, reportes y soporte para Arduino lo que facilita buscar apoyo a la hora de realizar el proyecto.
- Las placas son accesibles económicamente lo cual facilita adquirir una.
- Se puede conectar fácilmente a sistemas bluetooth, wifi, GSM, GPS, entre otras.

HTML y CSS: Se utilizo el código de etiquetado HTML5 para poder maquetar la página web, agregar la información, imágenes, hipervínculos e iconos para proceder con el diseño; Para el diseño se utilizó CSS3 para darle color a la página web, ubicar las imágenes y texto en cierta posición especifica y agregar ciertas animación sencillas.

JavaScript: Se utilizó JavaScript para darle animaciones avanzadas a la página web y poder desarrollar el menú de hamburguesa que se esta utilizando en la web.

PHP: Se utilizaron archivo con extensión .php para poder realizar una conexión de la página web a la base de datos tanto de inicio de sesión como a la base de datos que conecta con el Arduino para poder enviar la información del panel de control.

MySQL: Se utilizo la base de datos MySQL para poder crear el inicio de sesión de la página web, también se utilizo para crear una base de datos que se encarga de almacenar el dato que se envía al Arduino desde el panel de control.

Hosting y Dominio: Utilizamos como sistema de hosting y dominio InfinityFree.com, que nos permite almacenar una página web con un dominio gratuito, ideal para poder empezar a mostrar la página web desde internet donde cualquiera con accesos pueda conectarse.

IDEs: Como IDEs se utilizaron Visual Studio Code y ArduinoIDE

Control de Versiones (Git y GitHub): Se utilizo el control de versiones Git para tener un entorno compartido de trabajo, Git se conecta a un repositorio de GitHub para que sea un proyecto OpenSource y cualquiera pueda realizar mejoras del código o aportaciones que sumen.

LucidChart: Se utilizo esta aplicación web para poder programar la lógica de los algoritmos y diagramas de flujo que posteriormente iban a servir para posteriormente desarrollar el proyecto.

Tinkercad: Se utilizo este programa de dos manera, la primera fue para desarrollar un boceto de la maqueta para tener una idea de cómo lo íbamos a desarrollar, y en segundo lugar se utilizo para realizar la simulación del Arduino que nos iba a permitir validar su funcionamiento.

Firebase: Se utilizo este sistema para poder realizar la conexión de la página web con el Arduino, para que la página web actualizará la base de datos de Firebase y a la hora del Arduino consultar la base de datos notara el cambio del estado del panel de control.

Twilio: Se utilizo este sistema para el registro de nuevos usuarios, ya que el sistema al registrarse un nuevo usuario envía un mensaje de texto con los datos del usuario para poder realizar la aprobación desde el panel de control.

Documentación (DoxyGen y BookStackApp): Se utilizo un sistema llamado DoxyGen para poder generar la documentación automática en base a los comentarios de los códigos, ya que es un sistema que interprete los comentarios y genera la documentación a partir de ellos, para el manual de usuario se utilizo el sistema BookStackApp para poder hacer un manual entendible del funcionamiento del proyecto.

Propuestas de Mejora

Después de la realización del proyecto podemos decir que obtuvimos un avance realmente bueno, sin embargo, como todo en esta vida siempre se pueden lograr mejoras y siempre se puede seguir escalando en cuanto a funcionalidades y proyecto, a continuación, se brindan algunas propuestas de mejoras que nos puede ayudar a seguir expandiendo la idea del proyecto e incluso sentar las bases para próximos proyectos:

Google Analytics: Contamos con una propuesta de analítica mediante Google Analytics, lo que nos puede ayudar a medir metricas, identificar patrones clave post producción e identificar las áreas de mejora o aquellos apartados que a la gente le interesan más, no solo del proyecto en sí, si no de la empresa como tal.

Inteligencia Artificial: Nuestras propuestas hacia el futuro avanzan cada vez más y hoy en día la inteligencia artificial es una herramienta que no puede hacer falta en ningún lado, por eso una de nuestras propuestas es agregar un asistente personal de inteligencia artificial que permita al usuario interactuar dinámicamente con la página y con los proyectos.

Migrar Página Web a React: Una de las propuestas que más revolucionaría nuestra marca es la de realizar una migración de la página de web de un formato de página estática con HTML5 a la construcción de una aplicación web para que su funcionamiento sea más fluido y sea más dinámica.

Movimiento de los Planetas: Como otra propuesta de mejora se planea mejor la estructura de la maqueta por una donde se puedan tener más funcionalidades, como movimiento de los planetas individualmente y que cada uno pueda incluso tener luz propia la cual se encendería si el usuario elegí mover ese planeta en específico.

Holograma Solar: Y como funcionalidad estrella se pensaba agregar hologramas donde se pudiese visualizar la información de cada planeta.

Referencias Bibliográficas.

- <https://newbiely.com/tutorials/arduino-uno-r4/arduino-uno-r4-web-server>
- <https://docs.arduino.cc/libraries/r4httpclient/>
- <https://docs.arduino.cc/tutorials/uno-r4-wifi/wifi-examples/>
- <https://science.nasa.gov/solar-system/>
- <https://asana.com/es/resources/what-is-scrum>
- <https://asana.com/es/resources/what-is-kanban>
- <https://asana.com/es/resources/scrumban>
- <https://www.ane.gov.co/Sliders/prensa/presentaciones/2021/6.%20JUNIO/CURSO%20CONCEPTOS%20B%3%81SICOS%20DE%20ELECTR%C3%93NICA.pdf>
- <https://forum.arduino.cc/t/conexion-con-firebase/1287173>
- <https://innovadomotics.com/arduino-y-firebase/>
- <https://support.microsoft.com/es-es/topic/conceptos-b%C3%A1sicos-del-dise%C3%B1o-de-una-base-de-datos-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5>

Agradecimientos

Agradezco profundamente a Dios por brindarnos la fortaleza y sabiduría necesarias para realizar y culminar este proyecto

A nuestras familias, por su apoyo incondicional, comprensión y motivación constante durante todo este proceso.

A nuestro ingeniero por dedicación, que fueron fundamentales para la realización de este trabajo.

A mis compañeros de equipo, ya que sin ellos este proyecto no se hubiera podido realizar ya que cada uno tuvo un rol importante dentro del mismo.

Finalmente, agradezco a la Universidad Mariano Gálvez por proporcionarme las herramientas y conocimientos que hicieron posible este proyecto.