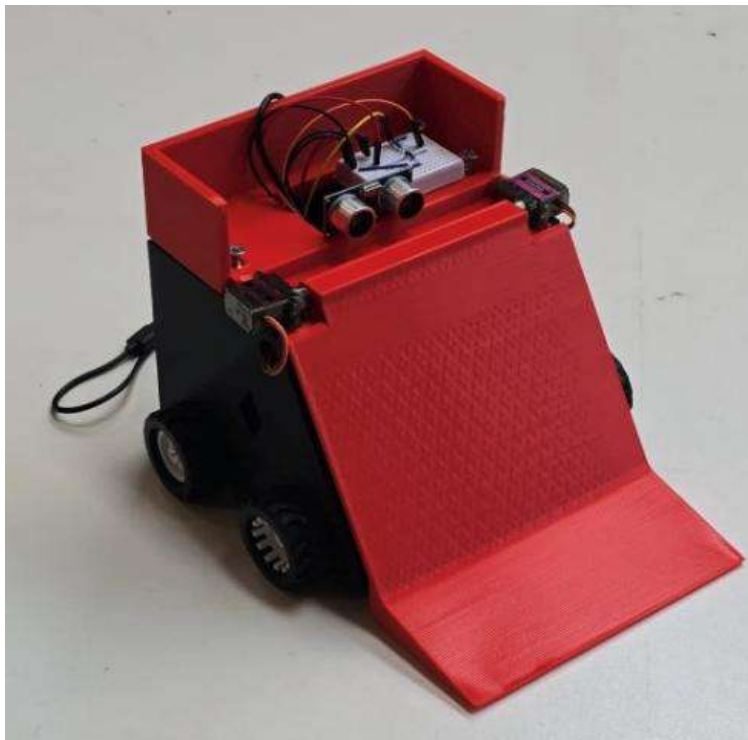


GYOKERES TEAM

Asti Challenge 2025

"La belleza de la ingeniería está en convertir lo que parece imposible en algo funcional y tangible."



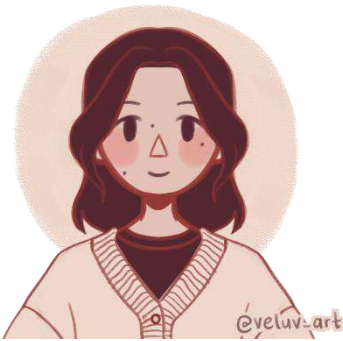


Parte 1: DISEÑO Y DESARROLLO DEL ROBOT

GYOKERES TEAM

Gyokeres team es un equipo que nació del deseo de un grupo de estudiantes de tener un primer acercamiento a la construcción de un robot, pudiendo hacer funcionar algo con nuestras propias manos, defendiendo la idea de: "La belleza de la ingeniería está en convertir lo que parece imposible en algo funcional y tangible."

Formado por 3 estudiantes de ingeniería electrónica industrial y automática de la universidad Carlos III de Madrid, la construcción de Gyokeres es gracias a:



VALENTINA SOTO

ENCARGADA DEL DISEÑO ELECTRÓNICO Y LA PROGRAMACIÓN



SERGIO RODRIGUEZ

ENCARGADO DEL DISEÑO ESTRUCTURAL DEL ROBOT



CELIA MOYA

ENCARGADA DE LA INVESTIGACIÓN E IMPRESIÓN 3D

También, nos gustaría dar especial reconocimiento a la asociación de robótica de la uc3m ASROB por prestarnos sus espacios para el ensamblaje del robot.

PLANIFICACIÓN Y CRONOGRAMA

La planificación llevada a cabo se basó esencialmente en el establecimiento de fechas límites para avanzar de forma progresiva en la realización del robot, aprovechando al máximo el tiempo dado.

Como conocimos del concurso en diciembre, decidimos empezar el proyecto en enero a la salida de los exámenes finales, siguiendo los objetivos pautados por semana:

SEMANA 1 (21 Y 24 DE ENERO): planificamos cómo nos íbamos a organizar, el tiempo que le íbamos a dedicar a cada cosa, los materiales esenciales que necesitábamos y pusimos en común ideas generales sobre el robot como su diseño y funcionalidades.

SEMANA 2 (1 AL 8 DE FEBRERO): Planificación inicial y lista de materiales necesarios. Se determinarán los componentes esenciales para la construcción del robot, investigando proveedores y comparando precios.

SEMANA 3 (9 AL 15 DE FEBRERO): Compra y recepción de materiales. A la vez que se dará iniciación al diseño del robot. Se realizarán bocetos y esquemas para definir la estructura y los sistemas de funcionamiento.

SEMANA 4 (16 AL 22 DE FEBRERO): Realización de los códigos del robot e informe preliminar. Primer Ensamblaje del robot el fin de semana.

SEMANA 5 (23 Y 28 DE FEBRERO): Montaje y construcción del robot. Se procederá al ensamblaje de las piezas y componentes adquiridos.

A la planificación, se le unen reuniones cada viernes para evaluar el progreso del proyecto. Y una reunión el 28 de febrero para la corrección final del informe.

PROCESO DE *Diseño*

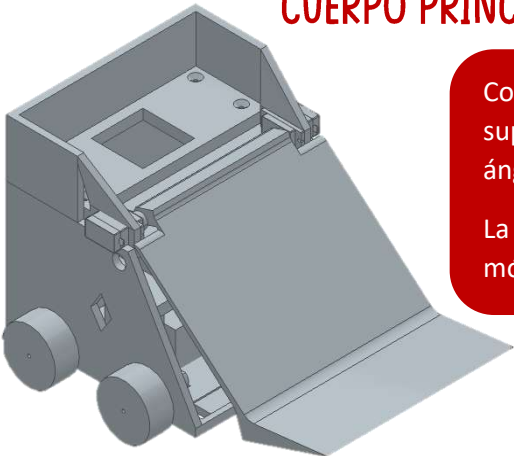
El proceso de diseño se dividió en dos vertientes:

- **Diseño estructural:** Carcasa del robot y partes modulares
- **Diseño electrónico:** Componentes que hacen posible su funcionamiento

Diseño estructural

La estructura del robot se diseñó en el programa CAD solid Edge para su posterior impresión 3D en PLA. Se busco que fuera un robot modular, pudiendo cambiar la pala móvil en función del reto a afrontar

CUERPO PRINCIPAL (SUMO)

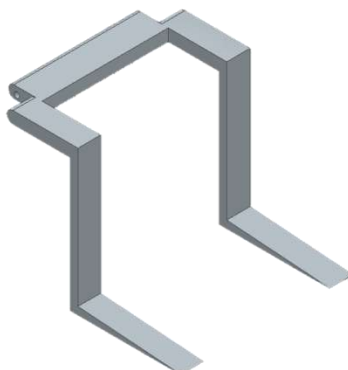


Constando de dos partes desmontables, donde a los servos del módulo superior se les puede montar y desmontar palas capaces de moverse en un ángulo de máximo 90 grados.

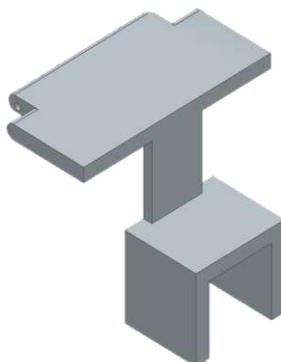
La imagen hace referencia a la configuración empleada en sumo, una pala móvil al nivel del piso para arrastrar y levantar.

VARIANTES

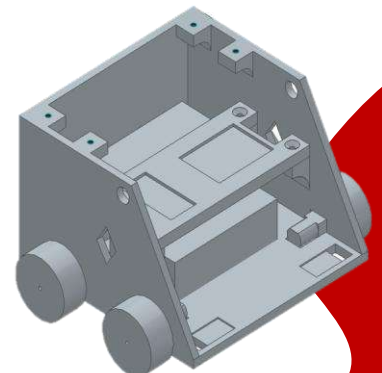
MINI FABRICA



GOLF



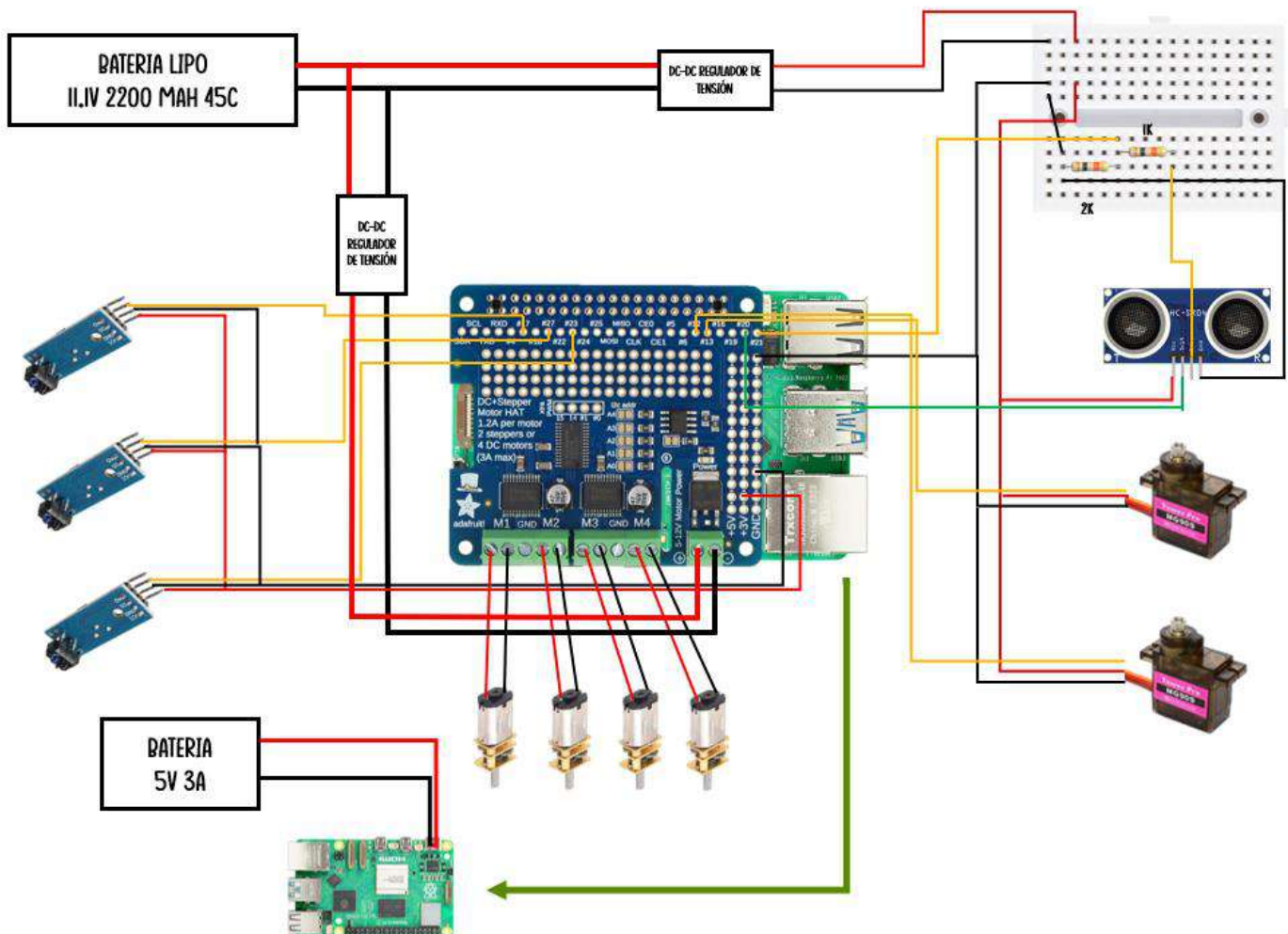
SIGUE LINEAS



Diseño Electrónico

La elección de los componentes se basó en 2 criterios principales: Implementación y reutilización. Al ser un proyecto personal, buscamos materiales que podamos reutilizar para otros años u otros proyectos, a la vez que sean confiables y existan guías de uso claras por ser nuestro primer acercamiento a concursos de robótica estudiantil. Buscamos sobre todo la posibilidad de mejora para años posteriores pudiendo utilizar los mismos componentes como base para múltiples funciones.

DIAGRAMA DE CONEXIONES



LISTA DE COMPONENTES

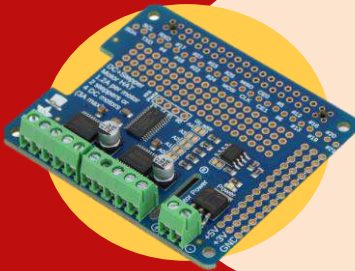
Electrónicos



RASPBERRY PI 5 2GB

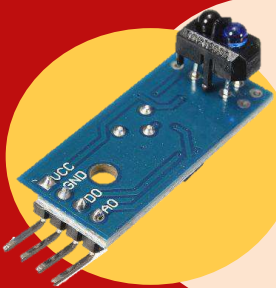
La cabeza del robot, desde ella se ejecutan todos los códigos que hacen posible su funcionamiento.

La elección de este modelo frente a otros es por una relación calidad-precio y la posibilidad de emplearla en futuras evoluciones del robot de mayores prestaciones.



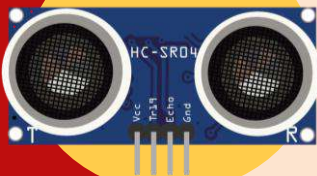
ADAFRUIT 2348

Conocido hat de motores para raspberry pi. Permite manejar 4 motores DC en un rango de 6-12V. A su vez, es posible acceder mediante terminales específicos a los pines libres de la raspberry y extiende la cantidad de pines GND, 5v y 3v, resultando ideal para el proyecto.



TCRT500

Sensor infrarrojo con la opción de mandar señales tanto digitales como analógicas. Tomamos esta opción debido a que la Raspberry no puede manejar señales digitales, así que un sensor que pudiera usarse en ambos sentidos era lo más adecuado.



HC-SR04

Sensor ultrasónico. Altamente usado en robótica, haciéndolo confiable y accesible para un primer acercamiento en la categoría de sumo autónomo.



MOTOR POLOLU 100:1

Motores encargados del movimiento. De pequeña dimensión, pero con un torque de $2.2 \text{ kg} \cdot \text{cm}$ y 360rpm resultan ideales para mover un peso considerable en poco espacio.



RUEDAS TODO TERRENO

Con el mismo eje que los motores y un buen caucho para asegurar buena fricción y con ello empuje.



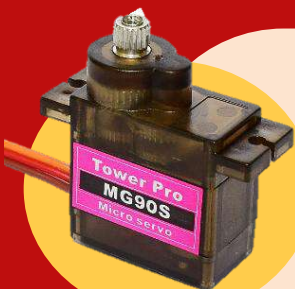
REGULADOR DE TENSIÓN AJUSTABLE

Convertidor DC-DC 4.0-40V a 1.25-37V. Para poder regular el voltaje que va de la batería a los componentes. Cuentan con una pantalla LED que indica el voltaje de salida y de entrada, ayudando a un fácil monitoreo del estado de la batería LIPO. Aun así, son de gran tamaño, siendo poco ideales para esta aplicación.



BATERÍAS

El proyecto consta de dos baterías, una LIPO de 11.1V 2200 mAh para la alimentación de motores y sensores, y un power bank de 5V 3A para la alimentación directa de la raspberry pi 5. La elección de estas fue porque era las que teníamos en casa de otro proyecto y uso cotidiano, resultando adecuadas para la aplicación.



SERVO MG90S

Servos con un torque de $2.2 \text{ kg} \cdot \text{cm}$, una velocidad de 0.1 segundos para los 60° y un ángulo máximo de 90° . Son una opción fiable y fácil de implementar, elegida por la misma razón que las baterías: Teníamos en casa y se adaptaban a las necesidades del proyecto.

Programación

El código cambia dependiendo del reto a enfrentar, teniendo un código distinto para cada reto, todos ellos programados en Python por ser el lenguaje usado por el motor hat. La programación dinámica fue una de las principales razones para elegir una raspberry pi, y es que el control remoto resulta ideal para poder cambiar de código sin tener que desmontar toda la estructura.

PRUEBAS DE CONTROL AUTONOMO

Cada prueba de control autónomo tiene su propio código

SIGUELINEAS

```
'''
SIGUELINEAS
    Sensor Utilizado: IR TRCT5000
    1-> Linea negra
    0-> Fondo claro

    DISTRIBUCION DE MOTORES
    Hat utilizado:Adafruit DC & Stepper Motor HAT for Raspberry Pi - Mini Kit
    Tomandose las medidas con vista de frente al robot
    Motor 1 - Frente Izquierdo
    Motor 2 - Frente Derecho
    Motor 3 - Detras Derecho
    Motor 4 - Detras izquierdo

    Las velocidades se manejan por porcentaje
    kit.motor1.throttle = 1.0  # Máxima velocidad adelante
    kit.motor1.throttle = -0.5 # 50% de la velocidad atrás
    kit.motor1.throttle = 0   # Detener motor

'''
from gpiozero import DigitalInputDevice
from adafruit_motorkit import MotorKit
import time

"""Inicializacion del hat de motores"""
kit = MotorKit()
```

```

"""Inicializacion de infrarrojos"""
sensor_der = DigitalInputDevice(17) #Entiendase derecho con vista de frente al
robot
sensor_izq = DigitalInputDevice(27)

"""Funciones para controlar los motores"""
def detener():
    kit.motor1.throttle = 0
    kit.motor2.throttle = 0
    kit.motor3.throttle = 0
    kit.motor4.throttle = 0

def mov_delante(v):
    kit.motor1.throttle = v
    kit.motor2.throttle = v
    kit.motor3.throttle = v
    kit.motor4.throttle = v

def giro_der(v): #Giro a la derecha con respecto a la vista detras del robot

    kit.motor1.throttle = v/2
    kit.motor2.throttle = v
    kit.motor3.throttle = v/2
    kit.motor4.throttle = v

def giro_izq(v):
    kit.motor1.throttle = v
    kit.motor2.throttle = v/2
    kit.motor3.throttle = v
    kit.motor4.throttle = v/2

# Velocidad
v = 0.5

try:
    while True:

        print(f"Sensor Izq: {sensor_izq.value}, Sensor Der: {sensor_der.value}")
#Este print esta para testear que los valores sean los esperados en todo momento

        if sensor_izq.value == 1 and sensor_der.value == 1: # Ambos sensores ven
negro
            print("Linea negra detectada en ambos sensores. Supuesto de meta")
            detener()

```

```

        elif sensor_izq.value == 0 and sensor_der.value == 0: # Ambos ven blanco
            print("Camino libre")
            mov_delante(v)
        elif sensor_der.value == 1: # Solo el derecho ve negro
            print("Linea detectada a la derecha, girando a la izquierda")
            giro_der(v)

        elif sensor_izq.value == 1: # Solo el izquierdo ve negro
            print("Linea detectada a la izquierda, girando a la derecha")
            giro_izq(v)

        time.sleep(0.005)

except KeyboardInterrupt:
    detener()

```

SUMO

```

from gpiozero import DistanceSensor
from gpiozero import AngularServo
from gpiozero import DigitalInputDevice
import time
from adafruit_motorkit import MotorKit # Control de los motores

"""INICIALIZAR EL HAT DE MOTORES"""
kit = MotorKit()

"""CONFIGURAR LOS SERVOS"""
SERVO_PIN1 = 12
servo1 = AngularServo(SERVO_PIN1, min_angle=0, max_angle=90,
min_pulse_width=0.0006, max_pulse_width=0.0024)
servo_PIN2 = 13
servo2 = AngularServo(servo_PIN2, min_angle=0, max_angle=90,
min_pulse_width=0.0006, max_pulse_width=0.0024)

servo1.angle = None
servo2.angle = None

"""CONFIGURAR LOS INFRARROJOS"""
sensor_der = DigitalInputDevice(17)
sensor_izq = DigitalInputDevice(27)
sensor_atras = DigitalInputDevice(23)

```

```

"""CONFIGURAR EL ULTRASONIDO"""
sensor = DistanceSensor(echo=6, trigger=5)

"""FUNCIONES DE MOVIMIENTO DE MOTORES"""
def detener():
    kit.motor1.throttle = 0
    kit.motor2.throttle = 0
    kit.motor3.throttle = 0
    kit.motor4.throttle = 0

def mov_delante(v):
    kit.motor1.throttle = v
    kit.motor2.throttle = v
    kit.motor3.throttle = v
    kit.motor4.throttle = v

def mov_detras(v):
    kit.motor1.throttle = -v
    kit.motor2.throttle = -v
    kit.motor3.throttle = -v
    kit.motor4.throttle = -v

def giro_der(v):
    kit.motor1.throttle = v/2
    kit.motor2.throttle = v
    kit.motor3.throttle = v/2
    kit.motor4.throttle = v

def giro_izq(v):
    kit.motor1.throttle = v
    kit.motor2.throttle = v/2
    kit.motor3.throttle = v
    kit.motor4.throttle = v/2

# Velocidad
v = 0.5

try:
    while True:

        distancia = sensor.distance * 100 # Convertir a cm
        print(f"Distancia: {distancia:.2f} cm")

```

```

        if sensor_izq.value == 0 and sensor_der.value == 0 and sensor_atras.value
        ==0: #Tomando que blanco es dentro del rink

            if distancia<30: #Si el robot rival esta sobre la pala activa el
servo

                servo2.angle = 50
                servo1.angle = 50
                time.sleep(0.4)
                servo2.angle = 0
                servo1.angle = 0
                servo1.angle = None
                servo2.angle = None

            if distancia<40:
                giro_der(v)
                time.sleep(0.5)
            else:
                detener()

        elif sensor_atras.value == 1: #tomando negro como fuera del rink
            mov_delante(v)
            time.sleep(0.5)
        elif sensor_izq.value == 1 and sensor_der.value == 1:
            mov_detras(v)
            time.sleep(0.5)

except KeyboardInterrupt:
    detener()

```

Con respecto al código de sumo, estamos trabajando en un programa con OPEN CV de detección de objetos, esto con el objetivo de que en futuras ediciones el robot pueda seguir por visión artificial al oponente, favoreciendo una toma inteligente de decisiones.

DIBUJA LA FIGURA

El código de dibuja la figura es muy simple: configuraciones de movimientos según cual figura se tiene que crear, surgiendo del testing de velocidades y giros, cuyos resultados arrojaron:

- 20 cm recorridos a los 0.5s de funcionamiento al 0.5% de su velocidad
- 1 segundo para un giro de 90° al 0.5% de su velocidad

Así que en vez de ese código nos gustaría agregar algo más interesante, un código preliminar que hicimos antes de la publicación de las especificaciones de la prueba. Como no sabíamos que figura tendría que dibujar, quedamos a la mitad del desarrollo de un programa que leyera imágenes vectoriales svg y trazara por movimientos la imagen subida. Nos quedamos en la fase teórica del código, siendo capaz de trazar en un mapa digital la imagen:

```

import matplotlib.pyplot as plt
from svgpathtools import svg2paths

# svg2paths se usa para leer y analizar archivos SVG y extraer las rutas (paths)
y sus atributos.

# Leer el archivo SVG y extraer las rutas y atributos
svg_file = 'C:/Users/Asus/Desktop/Universidad/Gyökeres-ASTI/Dibuja la figura
pre/completo.svg' # Ruta del archivo a dibujar
paths, attributes = svg2paths(svg_file)

"""
La funcion sv2paths lee el archivo y devuelve dos elementos:
paths: Una lista de rutas (Path), cada una compuesta de segmentos (líneas,
curvas, arcos, etc.).
attributes: Una lista de diccionarios con los atributos de cada ruta (como color,
grosor, etc.).
"""

# Crea una nueva figura donde se dibujan los graficos. BIBLIOTCA MATPLOTT
plt.figure(figsize=(8.3, 11.7)) # crea una figura cuadrada de 8x8 pulgadas.
                                # figsize especifica el tamaño de la figura en
pulgadas
                                # Es una tupla (ancho, alto) que define el tamaño de
la figura

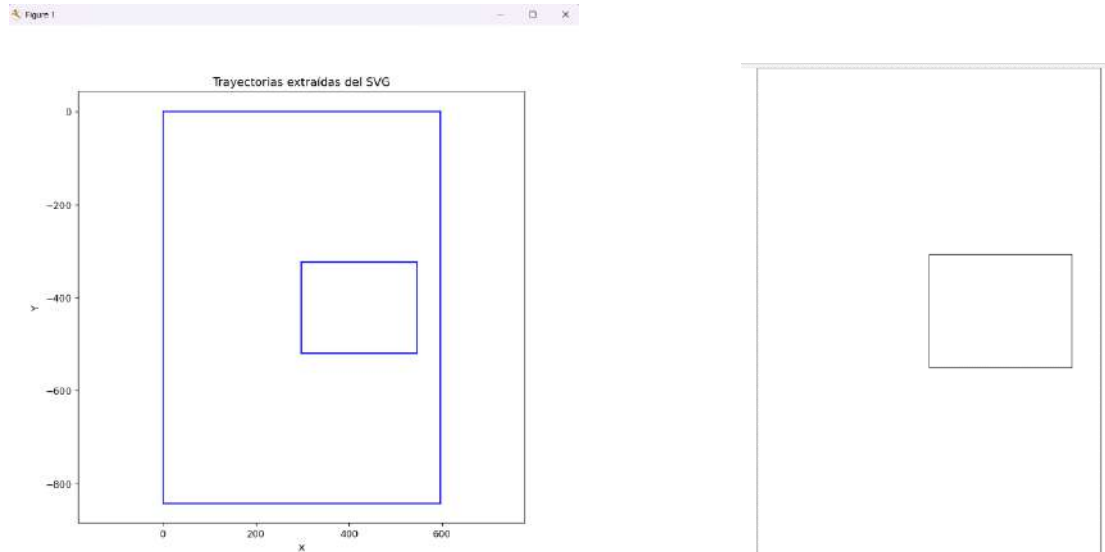
# Procesar cada ruta y graficarla
for path in paths: # Itera sobre todas las rutas (Path) extraídas del archivo
SVG.
    for segment in path: # Itera sobre los segmentos de cada ruta
        if hasattr(segment, "point"):
            # Evalua los puntos en el segmento para curvas suaves
            num_points = 100 # Número de puntos para aproximar el segmento
            points = [segment.point(t) for t in [i / num_points for i in
range(num_points + 1)]]
            x = [point.real for point in points]
            y = [-point.imag for point in points] # Invertir el eje Y. Esto
porque inicialmente dibujaba la figura al reves
            plt.plot(x, y, 'b') # Trazar en azul porque es mi color favorito
        else:
            # Manejar segmentos sin "point()" (por si acaso)
            x = [segment.start.real, segment.end.real]
            y = [-segment.start.imag, -segment.end.imag] # Invertir el eje Y
            plt.plot(x, y, 'r--') # Línea discontinua en rojo

```



```
# Configurar la gráfica
plt.axis('equal') # Mantener proporciones
plt.title('Trayectorias extraídas del SVG')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```

RESULTADOS OBTENIDOS FRENTE A IMAGEN SUBIDA:



Por temas de tiempo, no pudimos llegar a implementar el movimiento de motores y servo para el trazado de la imagen, pero lo guardamos como proyecto para futuras ediciones.

PRUEBAS DE CONTROL MANUAL

Para las pruebas de Mini Fabrica y golf, optamos por un control manual mediante un mando de Xbox.

```
import pygame
import time
from gpiozero import AngularServo
from adafruit_motorkit import MotorKit

kit = MotorKit()

"""Inicializacion de servos"""
SERVO_PIN1 = 12
servo1 = AngularServo(SERVO_PIN1, min_angle=0,
max_angle=90, min_pulse_width=0.0006, max_pulse_width=0.0024)
servo_PIN2 = 13
```

```

servo2 = AngularServo(servo_PIN2, min_angle=0,
max_angle=90, min_pulse_width=0.0006, max_pulse_width=0.0024)

"""Inicializacion del joystick"""
pygame.init()
pygame.joystick.init()

"""Funciones para controlar los motores"""
def detener():
    kit.motor1.throttle = 0
    kit.motor2.throttle = 0
    kit.motor3.throttle = 0
    kit.motor4.throttle = 0

def mov_detras(v):
    kit.motor1.throttle = -v
    kit.motor2.throttle = -v
    kit.motor3.throttle = -v
    kit.motor4.throttle = -v

def mov_delante(v):
    kit.motor1.throttle = v
    kit.motor2.throttle = v
    kit.motor3.throttle = v
    kit.motor4.throttle = v

def giro_der(v):
    kit.motor1.throttle = v/2
    kit.motor2.throttle = v
    kit.motor3.throttle = v/2
    kit.motor4.throttle = v

def giro_izq(v):
    kit.motor1.throttle = v
    kit.motor2.throttle = v/2
    kit.motor3.throttle = v
    kit.motor4.throttle = v/2

servo1.angle=None
servo2.angle=None

# Verificar si hay joysticks conectados
if pygame.joystick.get_count() < 1:
    print("No se encontr♦ el control de Xbox.")

```

```

    exit()

# Inicializar el joystick
joystick = pygame.joystick.Joystick(0)
joystick.init()

# Funcion para leer el control de Xbox
def leer_control():
    pygame.event.pump() # Actualiza los eventos de Pygame
    if pygame.joystick.get_count() < 1:
        print("No se encontr❖ el control de Xbox.")
        return False
    return True

# Bucle principal
try:
    while True:
        control_valido = leer_control() # Verifica si el control esta conectado

        if not control_valido:
            break

        # Leer los botones A y B
        boton_a = joystick.get_button(0)
        boton_b = joystick.get_button(1)

        if boton_a:
            servo2.angle = 0
            servo1.angle=0
            time.sleep(0.5)

        if boton_b:
            servo2.angle = 50
            servo1.angle=50
            time.sleep(0.5)

        """Leer ejes del joystick"""
        eje_vertical = joystick.get_axis(1)
        eje_horizontal = joystick.get_axis(0)

```

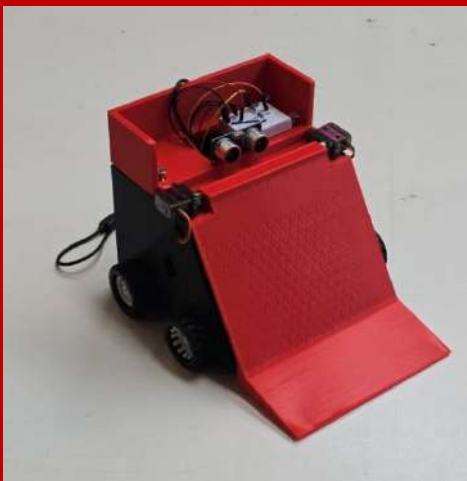
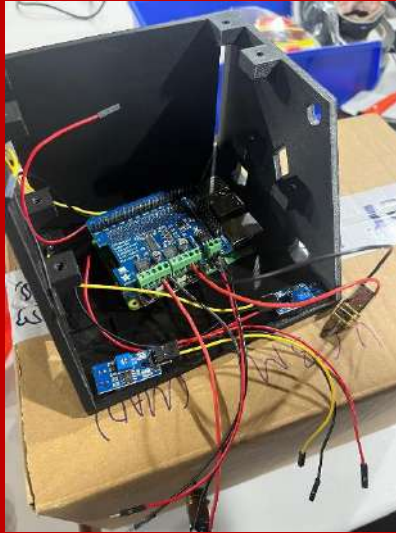
```
        velocidad = abs(eje_vertical) # Cuanto más se incline el joystick, mayor
la velocidad
        giro = abs(eje_horizontal) # Cuanto más se incline lateralmente, mayor
el giro

        if eje_vertical < -0.1: # Mover hacia adelante
            mov_delante(velocidad)
        elif eje_vertical > 0.1: # Mover hacia atrás
            mov_detrás(velocidad)
        elif eje_horizontal < -0.1: # Giro a la izquierda
            giro_izq(giro)
        elif eje_horizontal > 0.1: # Giro a la derecha
            giro_der(giro)
        else:
            detener()

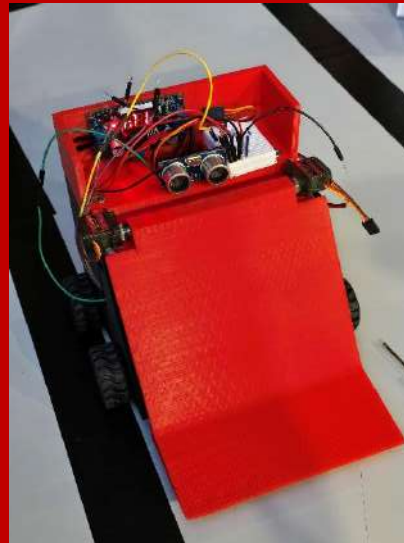
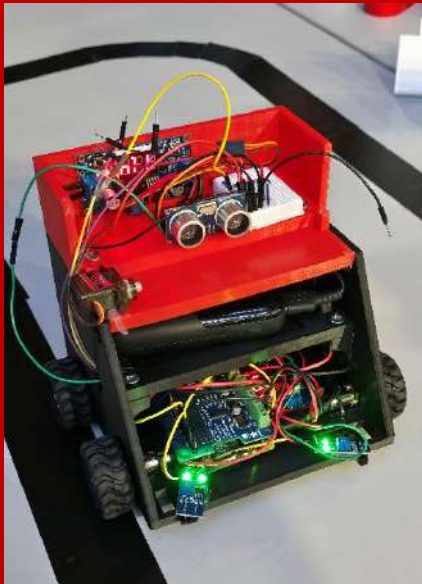
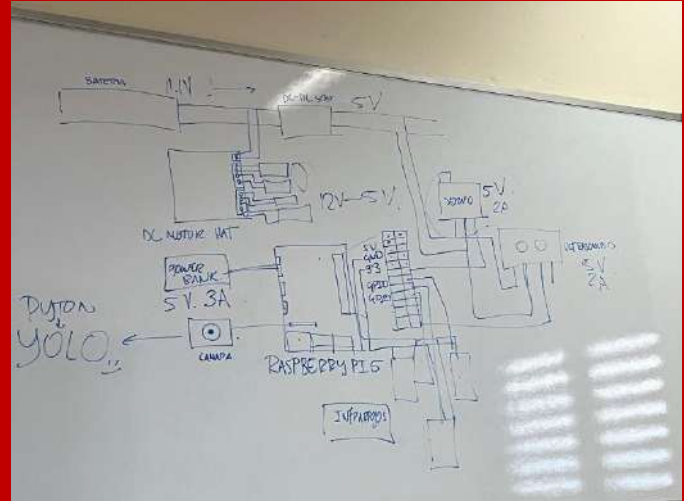
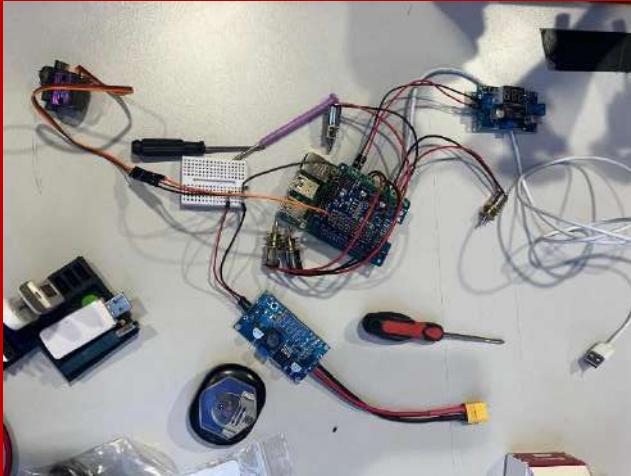
        time.sleep(0.5)

except KeyboardInterrupt:
    print("\nSaliendo del programa...")
    pygame.quit()
```

— Montaje y Construcción —



Testing y pruebas



Los testeos fueron muchos. Desde el primer día que se planifico en una pizarra el posible diagrama electrónico hasta el último probando códigos.

Se probaron los elementos sin montar para comprobar que la electrónica funcionara sin problemas antes de continuar. Esto fue fundamental para poder corregir ángulos en los servos, movimiento de motores y en que pines era mas adecuado conectar los componentes.

Pero, una vez empezado el montaje los testeos solo aumentaron. Se realizaron diversos cambios en la distribución de componentes ya que el diseño inicial tenía problemas con el centro de masas (Pudiendo notarse en las distintas imágenes) y reposicionamiento de sensores por detecciones erróneas.

Presupuesto

El presupuesto se monitorea mediante una tabla en Excel, ayudando a mantenerlo ordenado y dinámico.

El modelo de financiación fue propio. Como comentamos anteriormente, somos solo un grupo de estudiantes en nuestro primer acercamiento a la robótica estudiantil, por lo que financiamos el proyecto mediante lo que teníamos por casa y dividimos los costes totales.

ELEMENTO	PRECIO	CANTIDAD		COSTE FINAL
Raspberry Pi 5	57.47 €	1		57.47
Disipador de calor	6.99 €	0		0
Tarjeta SD 64 gb	8.00 €	0		0
Motor Pololu 6 V - 100:1	16.75 €	4		67
Adafruit 2348	25.85 €	1		25.85
Servo MG90S	3.80 €	1		3.8
Par de ruedas	7.87 €	2		15.74
IR TRCT5000	6.99 €	1	pack de 3	6.99 €
Módulo HC-SR04. Sensor				
Ultrasonico	2.39 €	1		2.39 €
camara v2	16.97 €	1		16.97 €
DC STEP-DOWN BUCK 3A - LM2596	4.49 €	2		8.98 €
Cable xt60 paralelo	6.99 €	1		6.99
Cables 0.5mm 10m	4.66 €	1		4.66
			TOTAL:	216.84 €



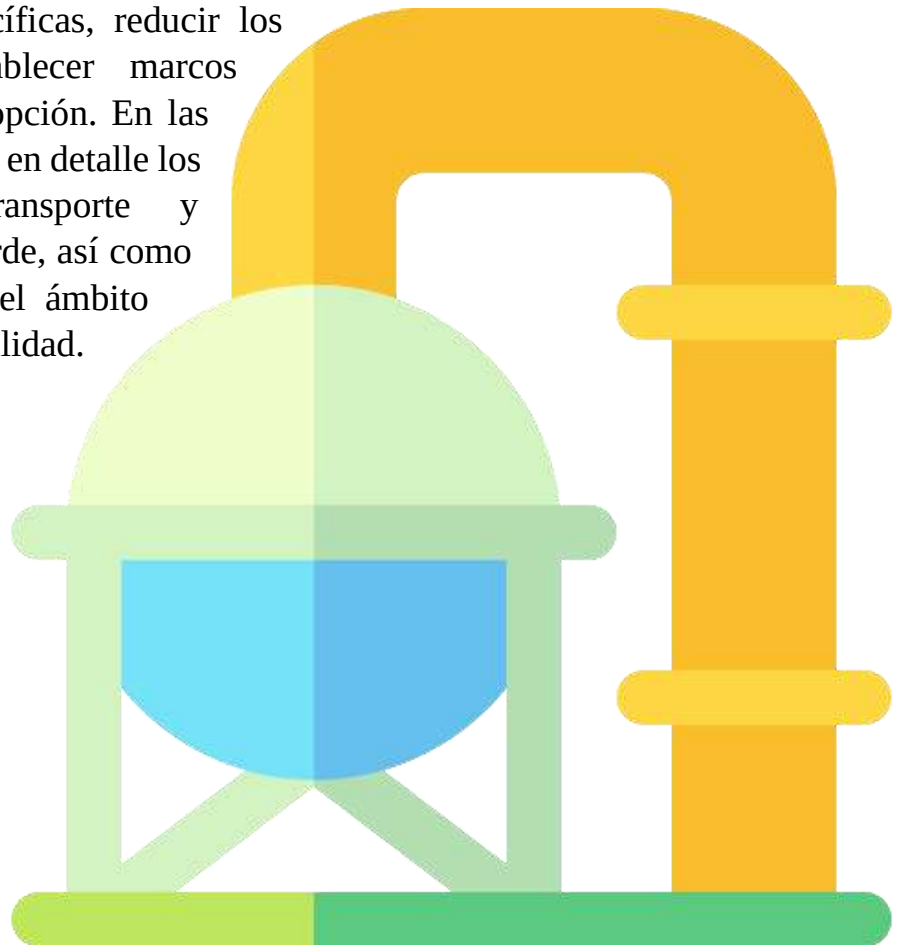
Parte 2: HIDROGENO VERDE



El Futuro De La Energía

El hidrógeno verde se ha convertido en una de las mejores alternativas para avanzar hacia un futuro energético sostenible gracias a su capacidad para generar energía sin emitir gases de efecto invernadero. Esto convierte al hidrógeno verde en una pieza clave en la transición energética global.

El hidrógeno, al ser el elemento más abundante en el universo, ofrece un potencial energético inmenso, pero su obtención en su forma libre requiere de procesos que históricamente han estado asociados a un elevado coste energético y económico. En este contexto, el hidrógeno verde se posiciona como una herramienta clave para alcanzar una economía libre de carbono, permitiendo no solo el almacenamiento eficiente de energía renovable, sino también su aplicación en sectores industriales de difícil electrificación, como la siderurgia, la producción de cemento, el transporte pesado y el marítimo. Asimismo, su capacidad para servir como vector energético le otorga una flexibilidad que puede transformar profundamente la infraestructura energética global, facilitando la integración de las energías renovables en las redes eléctricas y contribuyendo al desarrollo de un sistema energético más resiliente y eficiente. Sin embargo, la expansión del hidrógeno verde enfrenta aún importantes desafíos técnicos, económicos y logísticos, entre los que se incluyen la necesidad de desarrollar infraestructuras específicas, reducir los costes de producción y establecer marcos regulatorios que impulsen su adopción. En las siguientes secciones, se abordarán en detalle los procesos de producción, transporte y almacenamiento de hidrógeno verde, así como sus principales aplicaciones en el ámbito industrial, energético y de la movilidad.

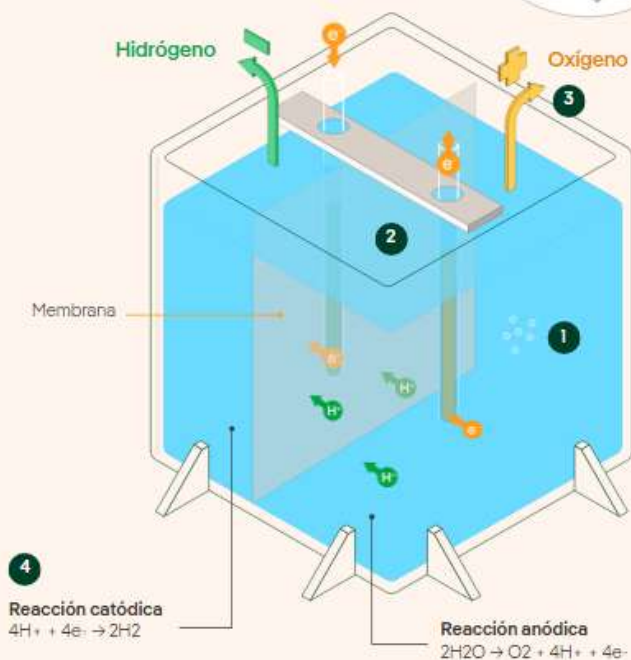


Obtención

La producción de hidrógeno verde se basa en un proceso conocido como electrólisis del agua, que consiste en la separación de las moléculas de agua (H_2O) en hidrógeno (H_2) y oxígeno (O_2) mediante el uso de electricidad. Lo que distingue al hidrógeno verde de otras variedades, como el gris o el azul, es que la electricidad utilizada en este proceso proviene exclusivamente de fuentes de energía renovable, lo que asegura que la producción no genere emisiones contaminantes. Existen distintos tipos de electrólisis, entre los que destacan la electrólisis alcalina, la de membrana de intercambio de protones (PEM) y la de óxido sólido, cada una con sus características, ventajas y desafíos específicos. La electrólisis alcalina es una tecnología madura, ampliamente utilizada debido a su eficiencia y costes relativamente bajos, aunque requiere el uso de electrodos resistentes a la corrosión y presenta limitaciones en su capacidad para adaptarse a las fluctuaciones inherentes a las fuentes renovables intermitentes. Por su parte, la electrólisis PEM ofrece una mayor flexibilidad operativa, adaptándose con rapidez a las variaciones de la generación eléctrica renovable, pero implica mayores costes debido a la necesidad de catalizadores de metales preciosos como el platino e iridio.

¿Cómo se obtiene el hidrógeno verde?

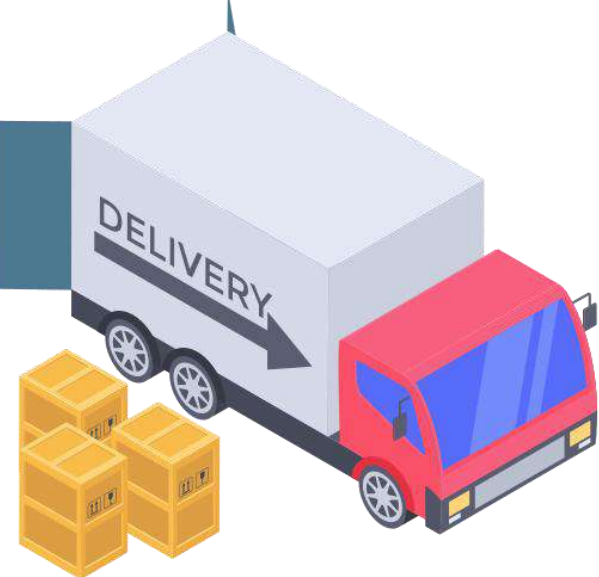
La obtención de hidrógeno verde por electrólisis a partir de fuentes renovables consiste en la descomposición de las moléculas de agua (H_2O) en oxígeno (O_2) e hidrógeno (H_2).



Fuente: Departamento de Energía de EE.UU. y Wood Mackenzie.

Por último, la electrólisis de óxido sólido, aún en fase de desarrollo, opera a altas temperaturas, lo que permite alcanzar una elevada eficiencia energética, aunque presenta desafíos técnicos relacionados con la durabilidad de los materiales empleados. Además de la electrólisis, existen otras tecnologías en fase experimental, como la producción de hidrógeno mediante fotólisis o biogénesis, que buscan aprovechar directamente la energía solar o ciertos microorganismos para descomponer el agua o aprovechar residuos orgánicos.

La elección de la tecnología adecuada dependerá de factores en plantas de electrólisis, así como en la interconexión con parques solares, eólicos o hidroeléctricos, garantizando un suministro estable y competitivo. A medida que se desarrollan nuevas tecnologías y se optimizan los procesos, se espera una reducción progresiva de los costes, lo que facilitará su integración en el sistema energético global.



Transporte y almacenamiento

El transporte y almacenamiento del hidrógeno verde constituyen dos de los principales retos para su implementación a gran escala, dado que este gas presenta características que dificultan su manejo, como su baja densidad energética en condiciones normales y su capacidad para difundirse a través de ciertos materiales.

Para su transporte, existen diversas opciones tecnológicas que permiten adaptarse a las necesidades y condiciones específicas de cada aplicación. Una de las alternativas más utilizadas es el transporte por tuberías, que consiste en canalizar el hidrógeno comprimido a través de infraestructuras similares a las empleadas para el gas natural. Esta opción es eficiente para distancias cortas y medianas, aunque requiere el acondicionamiento o construcción de tuberías resistentes a la permeabilidad del hidrógeno.

Para distancias más largas, se recurre al transporte por carretera mediante camiones cisterna que transportan hidrógeno comprimido o licuado. El hidrógeno comprimido se almacena a altas presiones, generalmente entre 350 y 700 bar, mientras que el hidrógeno licuado se mantiene a temperaturas extremadamente bajas, cercanas a $-253\text{ }^{\circ}\text{C}$, lo que permite incrementar su densidad energética, pero implica mayores costes y desafíos técnicos debido a la necesidad de infraestructuras criogénicas. Otra opción emergente es el transporte marítimo, especialmente relevante para el comercio internacional, donde se desarrollan buques específicos capaces de transportar hidrógeno en forma de amoníaco o metanol, compuestos que permiten una manipulación más segura y eficiente.

En cuanto al almacenamiento, este puede realizarse en estado gaseoso, líquido o en compuestos químicos, como hidruros metálicos o amoníaco. El almacenamiento subterráneo en cavernas salinas ha ganado interés debido a su capacidad para albergar grandes cantidades de hidrógeno a un coste relativamente bajo, convirtiéndose en una solución viable para gestionar la intermitencia de las energías renovables. La mejora y expansión de estas tecnologías es esencial para garantizar la continuidad del suministro y facilitar la integración del hidrógeno verde en el sistema energético global.

— Aplicaciones —

El hidrógeno verde tiene un amplio abanico de aplicaciones que abarcan sectores clave de la economía, lo que lo convierte en un recurso estratégico para la descarbonización y el desarrollo de una infraestructura energética más sostenible.

EN EL SECTOR INDUSTRIAL, se utiliza como materia prima en procesos que requieren altas temperaturas, como la producción de acero, cemento, vidrio y productos químicos, donde tradicionalmente se han empleado combustibles fósiles.

EN LA INDUSTRIA DEL ACERO, por ejemplo, el hidrógeno verde puede reemplazar al carbón en el proceso de reducción directa del mineral de hierro, evitando así la emisión de grandes cantidades de dióxido de carbono.

EN EL ÁMBITO ENERGÉTICO, el hidrógeno verde actúa como un eficaz vector para almacenar el excedente de energía renovable, permitiendo su uso posterior cuando las condiciones climáticas no son óptimas para la generación eléctrica. Este almacenamiento a largo plazo facilita la estabilidad y flexibilidad de las redes eléctricas, especialmente en sistemas con una alta penetración de fuentes intermitentes como la solar y la eólica.

EN EL TRANSPORTE, el hidrógeno verde representa una alternativa limpia para vehículos pesados, trenes, barcos e incluso aviones, sectores en los que la electrificación directa mediante baterías presenta dificultades debido a la necesidad de una elevada autonomía y capacidad energética. Los vehículos impulsados por pilas de combustible convierten el hidrógeno en electricidad para alimentar motores eléctricos, emitiendo únicamente vapor de agua como subproducto.

Por último, en **EL SECTOR RESIDENCIAL Y COMERCIAL**, el hidrógeno verde puede ser utilizado para generar calefacción y electricidad a través de calderas y pilas de combustible, lo que permite sustituir al gas natural en algunos casos. El éxito de estas aplicaciones dependerá del desarrollo de una infraestructura adecuada, así como de la implementación de políticas que fomenten su producción, distribución y consumo, impulsando así la transición hacia un modelo energético más limpio y sostenible.

HIDROGENO EN *España*

España se ha posicionado como uno de los países europeos con mayor potencial para el desarrollo del hidrógeno verde aprovechando su abundante disponibilidad de recursos renovables. El gobierno español ha puesto en marcha diversas iniciativas y estrategias para impulsar la producción, almacenamiento, transporte y uso del hidrógeno verde, alineándose con los objetivos climáticos de la Unión Europea y con su estrategia de transición energética.

En 2020, España presentó su "Hoja de Ruta del Hidrógeno: una apuesta por el hidrógeno renovable", que establece una serie de metas ambiciosas para el desarrollo de esta tecnología hasta 2030, incluyendo la instalación de al menos 4 GW de capacidad de electrólisis, el fomento de proyectos de producción de hidrógeno verde en sectores industriales y el desarrollo de infraestructuras específicas para su transporte y almacenamiento. Regiones como Andalucía, Aragón, Castilla-La Mancha y Galicia se destacan por su potencial para albergar plantas de producción de hidrógeno verde, gracias a sus recursos naturales y su infraestructura industrial. Además, el hidrógeno verde se perfila como un elemento clave en la descarbonización de sectores estratégicos como el transporte de mercancías, la industria siderúrgica y la producción de fertilizantes, impulsando la competitividad y sostenibilidad del tejido económico español.

El éxito de esta apuesta dependerá de la colaboración entre administraciones públicas, empresas y centros de investigación, así como de la atracción de inversiones que permitan escalar las tecnologías y reducir los costes de producción, consolidando a España como un referente en el panorama internacional del hidrógeno renovable.

El futuro



A pesar de su gran potencial, el hidrógeno verde enfrenta importantes desafíos que deben superarse para lograr su implementación a gran escala. Uno de los principales obstáculos es el alto coste de producción, ya que la electrólisis sigue siendo más cara en comparación con los métodos convencionales de obtención de hidrógeno a partir de combustibles fósiles. La eficiencia energética del proceso de electrólisis también es un reto, pues se requieren mejoras tecnológicas para reducir las pérdidas de energía. Además, la infraestructura para su transporte y almacenamiento aún no está completamente desarrollada, lo que dificulta su distribución y comercialización. A nivel regulatorio, es necesario establecer normativas claras y fomentar políticas de incentivos que permitan acelerar la inversión en hidrógeno verde.

que

El futuro del hidrógeno verde dependerá en gran medida de la reducción de sus costes de producción, el desarrollo de nuevas tecnologías de almacenamiento y el establecimiento de una red global de distribución. Se espera que, en los próximos años, con el avance en la investigación y el apoyo gubernamental, el hidrógeno verde juegue un papel clave en la transición energética, ayudando a descarbonizar sectores estratégicos como la industria pesada y el transporte de larga distancia. En este sentido, el compromiso de los gobiernos y la colaboración entre empresas energéticas, industrias y centros de investigación serán determinantes para consolidar al hidrógeno verde como una fuente de energía viable y competitiva a nivel mundial.