

Informe Descriptivo



LA INNOVACIÓN DE LA AGROINDUSTRIA

Abril Contreras - Martin Alejandro Cabrera - Lara Sofia Diaz Steinbrecher
Santino Ferrante - Mateo Kearney - Juan Ignacio Torres

HORUS

ÍNDICE

INTRODUCCIÓN	OBJETIVO ALCANCE BENEFICIOS	04
SOFTWARE	DESCRIPCIÓN GENERAL MICROCONTROLADOR INTELIGENCIA ARTIFICIAL SIST. COMUNICACIÓN SIST. GEOLOCALIZACIÓN LOGS APLICACIÓN FRONTEND INICIO ROBOTS DATA GALERÍA AJUSTE DEPENDENCIA BACKEND ELECTRON	05
HARDWARE	COMPONENTES ESQUEMA DE PLACA DE CONTROL DE MOTORES ESQUEMA DE PLACA RECEPTORA ALIMENTACIÓN DIAGRAMA EN BLOQUES	13

ESTRUCTURA	MATERIALES UTILIZADOS DESCRIPCIÓN DE LA ESTRUCTURA DIAGRAMA 3D	29
PÁGINA WEB	DESCRIPCIÓN GENERAL	32

INTEGRANTES

CONTRERAS, Abril Victoria
DNI 46.646.822 - Tel.: 11 6939 4245

CABRERA, Martin Alejandro Cabrera
DNI 47.260.03 - Tel.: 11 6141 8080

DIAZ, Lara Sofia Steinbrecher
DNI 46.909.041 - Tel.: 11 3700 7432

FERRANTE, Santino Marco
DNI 47.057.431 - Tel.: 11 6148 8088

KEARNEY, Mateo
DNI 47.307.247 - Tel.: 11 3786 4168

TORRES, Juan Ignacio
DNI 47.144.499 - Tel.: 11 6596 9351



ESFUERZO DEL PROYECTO

12 horas semanales
(312 horas de trabajo total)

FECHA DE INICIO

17 de Marzo de 2024

DURACIÓN

26 Semanas

DOCENTES INVOLUCRADOS

BIANCO, Carlos
CARLASSARA, Fabrizio
MEDINA, Sergio
PALMIERI, Diego

PÁGINA WEB

<https://horus-pagina-web.vercel.app>

INSTAGRAM

https://www.instagram.com/agro_horus/

LINKEDIN

<https://www.linkedin.com/company/agro-horus/about/>

INTRODUCCIÓN

OBJETIVO

El objetivo del proyecto es brindar una **herramienta** que permita a los productores de campo en la Argentina **combatir** de forma **más efectiva** las **plagas** que afectan las cosechas, centrandonos inicialmente en la soja. Para esto, buscamos como proyecto diseñar un **sistema de reconocimiento artificial e identificación** a través del análisis de la cosecha. Esta herramienta permitirá **reconocer las 4 plagas artrópodos principales** que pueden estar afectando a la planta: las orugas defoliadoras, las chinches, las arañuelas y los trips; y dar un informe detallado al usuario sobre el tipo y donde se encuentra utilizando una aplicación conectada por radiofrecuencia. La implementación de una **plataforma móvil robótica** permite una eficacia y velocidad mayor en una menor cantidad de tiempo en la que un ser humano podría realizar la misma tarea. La plataforma se traslada con un dispositivo rotatorio que transporta una cámara, la cual se encargará de revisar los cultivos.

ALCANCE

Nuestro proyecto está **destinado a los productores del campo en la Argentina**, principalmente los **productores de soja**. Pero en un futuro, nuestro proyecto podrá utilizarse en cualquier país, ya que la IA se adapta a cualquier plaga artrópoda que afecte al cultivo del terreno. La IA cuenta con la posibilidad de aprender, entonces solo requeriría una etapa de aprendizaje corta para trabajar en cualquier terreno, ya sea un campo en el que se cultive maíz, soja, trigo, tomates, etc.

Por más que nuestro proyecto esté **orientado a la agroindustria**, Horus también se puede utilizar para monitoreo rutinario de cualquier tipo.

BENEFICIOS

Con HORUS buscamos que el sector agrícola pueda reducir la necesidad de labor humano, así como aumentar la tasa de monitoreo tanto en cantidad de días como en horas. Un pequeño grupo de personas podrá trabajar con las imágenes recopiladas que la IA del robot les envíe, evitándose así el trabajo físico a intemperie, como también reduciendo la necesidad de un mayor número de personal capacitado. Además,

1. El proyecto reduce la carga laboral necesaria para el reconocimiento de plagas en el campo.
2. La reducción de pérdidas en la cosecha incrementará la cantidad producida y por lo tanto mayor oferta de la cosecha en el mercado.
3. El uso más eficiente y más bajo de insecticidas tendrá un impacto beneficioso en la salud alimentaria de los consumidores.
4. El personal puede pasar menos tiempo bajo el sol, y monitorear a través del robot, reduciendo la exposición y minimizando la probabilidad de enfermedades.

SOFTWARE

DESCRIPCIÓN GENERAL

El robot utiliza una **IA (Inteligencia Artificial)** de reconocimiento con un modelo de aprendizaje tipo **CNN (Convolutional Neural Network)** que permita la segmentación, clasificación y discriminación de objetos específicos en las imágenes del custom dataset.

Para iniciar, el usuario, desde la **aplicación** de una computadora, indicará en un mapa la zona que tendrá que recorrer HORUS. Al recibir la información y ser ubicado en el área de inicio, el **robot** podrá comenzar a **monitorear el campo** de forma **autónoma**. Mediante la **IA de reconocimiento** y una **cámara** (de la cual el microcontrolador podrá ajustar altitud y dirección mediante servos) monitoreará las plantas de soja en busca de la plaga (ya sea Orugas, Trips, Chinches o Arañuelas) o indicios (como hojas que presenten agujeros o coloración). En **caso positivo**, la IA tomará una **captura de pantalla** que será almacenada en una **memoria adicional**, y transmitirá por medio de un **módulo de radiofrecuencia** la siguiente información a la computadora conectada al segundo módulo de radiofrecuencia:

- Ubicación de la alerta
- Que tipo o tipos de plaga encontró, indicios o especímenes.
- Si la alerta proviene de una trampa de feromonas para plaga o no
- Porcentaje de probabilidad de plaga
- Descripción en formato texto de los indicios
- Código que pertenece a la imagen vinculada

La información es mostrada en la aplicación y guardada en la base de datos del dispositivo para referencia posterior.

MICROCONTROLADOR

El microcontrolador utilizado es una **raspberry pi 4 modelo B**, con el sistema operativo **raspberry pi OS**, siendo el cerebro del robot, está encargada de controlar, por un lado la parte móvil del robot (el brazo y las orugas) y por otro lado el sistema de comunicación, transmisión y de reconocimiento.

El programa que se utilizó fue **Thonny** (lenguaje de programación python) mediante entorno gráfico.

Al iniciar el robot automáticamente se inicia el programa Main.py, y se verifica que el **módulo sx1278** (módulo de comunicación por radiofrecuencia) esté listo para transmitir y recibir información. El programa envía a la app que el robot se encuentra listo para operar, junto con su id, coordenadas y batería restante. Entonces se pone en modo espera hasta recibir las indicaciones de la app con la ruta, las coordenadas de las trampas de feromonas y el momento de inicio:

El microcontrolador pondrá en marcha los motores mediante 2 **drivers a4988** y el robot se moverá de forma autónoma, utilizando 2 **ultrasonidos hc sr04** ubicados al frente para evitar el contacto con las plantas: al leer la distancia entre el robot y el objeto más cercano, se asegurará de mantener una distancia de seguridad y regular su movimiento de derecha a izquierda.

Cada 30 segundos el robot enviará sus coordenadas actuales a la aplicación, y detendrá su movimiento cada 1 metro para comenzar el análisis: el brazo se irá estirando de a 20 cm hasta llegar a su longitud máxima de 80 cm, en esas 5 etapas el servo (arriba del brazo) moverá la cámara para tener un recorrido de 180 grados; la cámara se activará y se iniciará la detección de la IA. En caso de no encontrar nada en las 5 etapas, el brazo se retrae y el robot continúa moviéndose. En caso positivo, la IA hará una foto de la plaga o indicio, la almacenará en los archivos, y transmitirá una alerta a la app con el **tipo de plaga o indicio, la cantidad, la probabilidad** de que la información sea correcta, **id** de la **imagen** vinculada a la información y una **descripción** en formato texto.

En caso de haber trampas de feromonas el microcontrolador se detendrá en las coordenadas de cada trampa al pasar en su recorrido, y hará el proceso antes mencionado, con la diferencia que si la detección es positiva transmitirá que viene de una trampa de feromonas, y si es negativa transmitirá que no se encontró ningún rastro o plaga en la trampa.

Al terminar la primera parte del recorrido el robot dará una vuelta de 180 grados y volverá a hacer un análisis, pero del otro lado, hasta llegar a las coordenadas iniciales, una vez ahí confirmará que el brazo se encuentra retraído, transmitirá un mensaje de

finalización de tarea, y un momento después que se encuentra listo para operar, poniéndose en modo espera para para la siguiente tarea (todavía mandando sus coordenadas cada 30 segundos).

Lectura de corriente de la batería

Para medir la corriente, las baterías se conectan a un **módulo Acs712** que trabaja con un sensor de **efecto Hall** que detecta el campo magnético que se produce por inducción de la corriente que circula. El sensor entrega una salida de voltaje proporcional a la corriente con un máximo de 5v que se envía a un divisor de voltaje que reduce el voltaje a 3,3 para poder ser recibido por el ads 1115 conectado a la raspberry.

INTELIGENCIA ARTIFICIAL

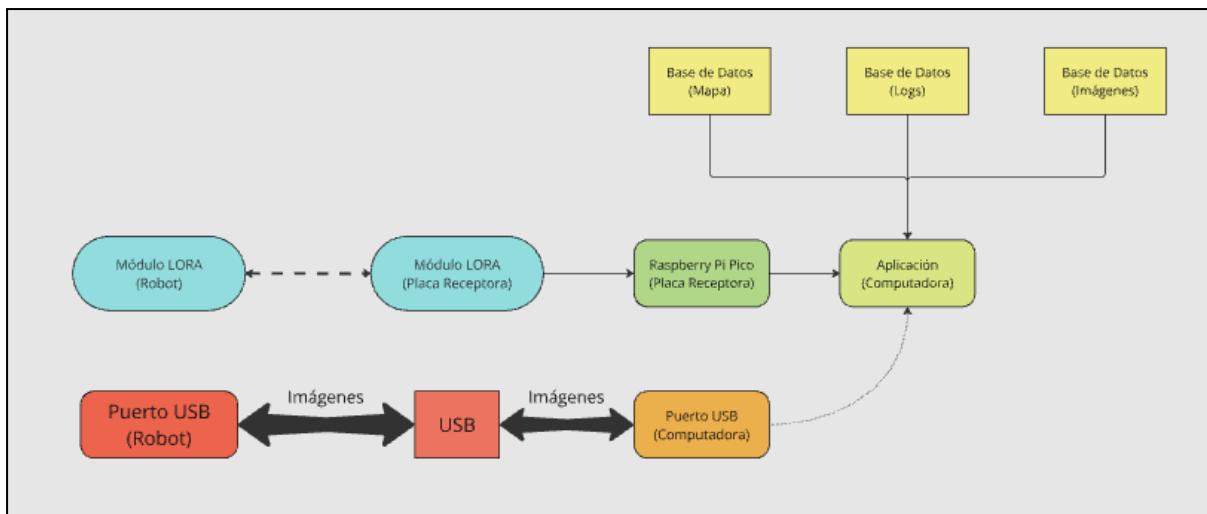
Para la inteligencia Artificial de Horus se utilizó el algoritmo **Yolo version 5s** para detección de objetos, se customizo el código para permitir utilizar datasets custom de entrenamiento, los cuales se armaron en la página **RoboFlow** utilizando imágenes de internet e imágenes sacadas por el equipo utilizando orugas de plástico, la ia se entrenó usando **Google Colab** para aprovechar una mayor velocidad de gpu proporcionada por la web mencionada. Los modelos entrenados actuales ya suman un dataset de al menos 3000 imágenes, estos modelos están guardados en los archivos del programa y pueden ser fácilmente reemplazables por modelos mejorados o diferentes según la necesidad.

SIST. COMUNICACIÓN

La información captada por el robot es enviada a través del **módulo LoRa transmisor** al **módulo LoRa receptor** a 433Mhz, en la placa receptora junto a la **Raspberry Pi Pico**. El microcontrolador se encarga de hacer de intermediario, entre el módulo receptor y la **computadora**. La información recibida, se mostrará en la **aplicación** de **Horus**.

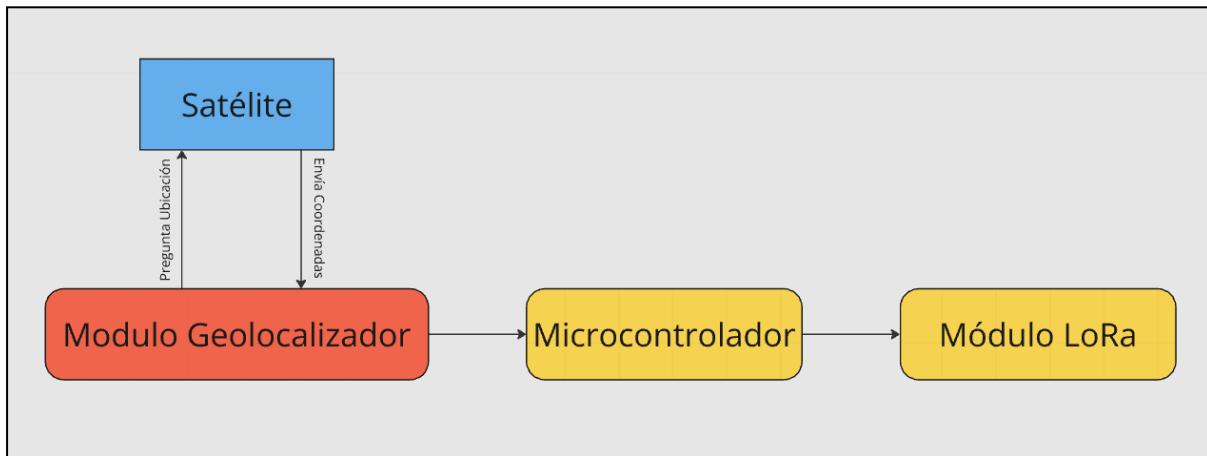
La información se envía letra por letra codificada, y al receptar se decodifica y se forma la oración pertinente.

Las imágenes, al ser muy pesadas para transmitirlas por radiofrecuencia, se guardan en el **microcontrolador** del robot. Son extraíbles a través del **puerto USB** del mismo micro. Un **USB** se puede conectar directamente a la computadora y la **aplicación** mostrará las imágenes de la unidad externa.



SIST. GEOLOCALIZACIÓN

El **módulo localizador GY-NEO6MV2** pide información al satélite cada 3 segundos sobre su posición y este responde enviando sus coordenadas. Las **coordenadas** recibidas se procesan en el **microcontrolador** y se pasan al **módulo LoRa Transmisor**, que se encarga de enviar la información hasta la **aplicación**.



APLICACIÓN

La aplicación tiene como objetivo la interacción del usuario con el sistema y el almacenamiento de datos para su futura referencia. La aplicación está formada por dos partes principales:

- El servidor o backend, que permite realizar gran parte de la lógica y control entre base de datos, interfaz y robot.

- La interfaz o frontend, que el usuario utiliza para interactuar con la base de datos y el robot.

Estas partes se encuentran montadas para ser ejecutadas como aplicación de escritorio a través de la framework de Electron.js.

BACKEND

El backend de la aplicación se encarga de la comunicación entre la base de datos, la interfaz y el robot. Se encuentra programada en javascript a partir de la framework de node.js para el servidor.

La raspberry pi pico conectada a los módulos de radiofrecuencia se conecta a la computadora por medio de un cable USB, el servidor recibe los paquetes o logs cuando hay una detección y los almacena en la base de datos de MongoDB. La base de datos se almacena de manera local por medio de la aplicación de MongoDBCompass. La información propia del robot correspondiente al log no es guardada en la base de datos y es comunicada a la interfaz.

FRONTEND

La interfaz de la aplicación es la forma que el usuario tiene para comunicarse con el sistema. Se encuentra programada en html, css y javascript mediante la framework de Vue.js.

El archivo principal contiene un header, que se mantiene en todas las rutas de la aplicación y le permite navegar entre las diferentes rutas de la misma:

- Inicio
- Robots
- Data
- Galería
- Ajustes

Y también contiene un componente llamado RouterView que proviene de Vue.js. Este componente carga las diferentes vistas o views dependiendo de la ruta designada. Las diferentes rutas se encuentran designadas por un archivo javascript (index.js en router) y en el nav del header cada item de la lista tiene vinculada una dirección de la ruta asignada por el archivo de router.

Algunas rutas, como Inicio, Robots y Data, realizan requests al servidor para obtener información de la base de datos. Esto lo hacen a partir de la api de fetch, tanto para las requests de GET como de POST.

INICIO

El propósito de la ruta de inicio es proveer al usuario de la información general y más importante del sistema. Está formada por dos partes, la primera es el mapa y la segunda es la sección de control.

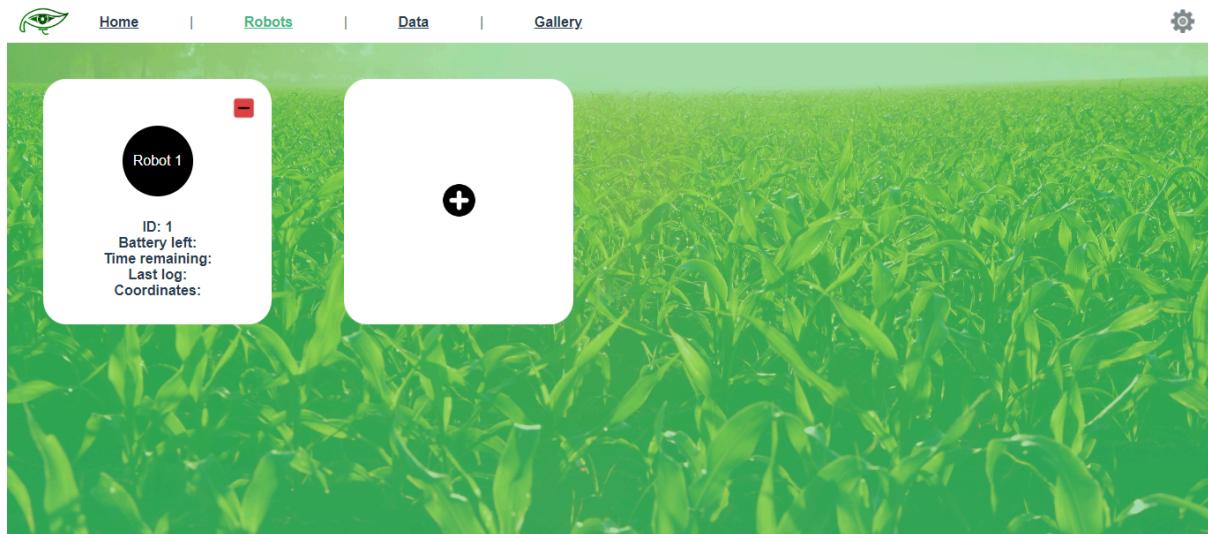
Para el mapa utilizamos la librería de leaflet.js que permite crear mapas a partir de los datos proveídos por un proveedor de datos geográficos (tiles), en nuestro caso OpenStreetMap. En un principio, el mapa trata de buscar los datos al proveedor de manera online, pero si esto falla, carga los datos que hayan sido guardados mediante el importe de archivos offline de la sección de control, si es que fueron cargados. El mapa se carga dentro de un div, y la librería permite que se carguen las imágenes de las tiles de manera dinámica, permitiendo zoom y movimiento en el mapa. Una vez montado el mapa, realiza una request a la base de datos para cargar los datos preexistentes de configuración del mapa. Esto incluye el eje central del mapa, marcadores de detecciones, marcadores de trampas de feromonas, marcadores de puntos de partida que tome cada robot y polígonos relacionados al área a recorrer por el robot. Cada uno de los datos mencionados se cargan en una capa diferente (el mapa se encuentra en una capa base) de modo que puedan ser desactivados en el controlador de capas en la parte superior derecha del mapa.

Con respecto a la sección de control, tiene diferentes controles cuya función se detalla a continuación:

- Permite centrar el eje del mapa - Esto tiene como objetivo centrar el mapa a las coordenadas del usuario.
- Permite importar un archivo offline - Una vez montado el mapa, comprueba si puede conectar con los servicios del proveedor de servicios de mapas y en su defecto, utiliza el archivo importado por este input
- Agregar trampas de feromonas al mapa
- Agregar un área de detección para un robot
- Agregar un punto de partida para un robot
- Filtro: permite filtrar las detecciones que se cargan en el mapa por cualquiera de sus keys, incluido por día, por hora, por ID de robot, por ID de imagen, por coordenadas, por tipo de plaga, por probabilidad o por trampa de feromona.

ROBOTS

La ruta de Robots tiene como objetivo dar a conocer el estado de los robots conectados al servidor. De manera predeterminada no hay robots, y una vez que se agrega uno, se comprueba la conexión con el mismo y se carga un componente si es positiva.



DATA

La ruta de Data tiene como objetivo la visualización de los reconocimientos del robot de manera completa y organizada. Está formado por un filtro, un selector de orden y por el display de información.

- Filtro (DataFilter): la sección de filtro se comporta de la misma manera que el filtro en la ruta de inicio. Se puede elegir entre las diferentes keys (ya sea algunas o todas):
 - Dia
 - Hora
 - ID de robot
 - ID de imagen
 - Coordenadas
 - Tipo de plaga
 - Probabilidad
 - Trampa de feromonas.

Una vez se elige un tipo de key en el selector, se carga la sección correspondiente y permite elegir el dato por el que filtrar. Se pueden cargar todas las keys del selector o ninguna, y cada una puede eliminarse con el botón adjunto.

Cada vez que se elige asigna un valor a alguna de estas keys, se realiza una request POST via api fetch al servidor con los datos en el filtro.

The screenshot shows a user interface for monitoring or data analysis. At the top, there is a navigation bar with links for Home, Robots, Data, and Gallery, along with a gear icon for settings. Below the navigation bar is a sidebar containing two sections: "Filter by" and "Order by". Under "Filter by", there is a dropdown menu set to "Date" with a date selector showing "10/10/2024". Under "Order by", there is a dropdown menu set to "Date and Time" with a sorting order selector. The main content area displays a single log entry with the following details:

```

Date: 2024-10-02
Time: 12:00
Image ID: 123
Robot ID: 1
Plague Type: oruga
Phermone Trap: true
Probability: 89
Location: [ 51.512986, -0.071357 ]

```

Todos los filtros seleccionados:

This screenshot shows the same application interface as the previous one, but with multiple filters selected in the "Filter by" section. The dropdown menu is now set to "Location" and contains several items with "X" icons for removal, including "10/10/2024", "10:53 AM", "Image ID", "Robot ID", "Plague Type", "Yes", "Probability", and "Location". The main content area displays a single log entry with the following details:

```

Date: 10-10-2024
Time: 10:53
Image ID: 1
Robot ID: 1
Plague Type: oruga
Phermone Trap: true
Probability: 100
Location: [ 0, 1 ]

```

- Selector de orden (OrderFilter): el selector de orden permite elegir si la información se presenta de manera ascendente o descendente. La key que se elige en el selector es por el que se ordena. El orden es descendente de manera predeterminada. Estos valores se incluyen junto a los datos del filtro cuando se efectúa la POST request al servidor.
- Display de información (DataLogs): el display carga un componente por cada log que se encuentre en la variable de listLogs, en el que se le asigna el valor a cada key correspondiente de acuerdo a sus valores en el objeto guardado. Cuando se monta los componentes de la ruta, se realiza una request vacía al servidor, lo que devuelve todos los documentos de las instancias de detección de la base de datos (robotdata).

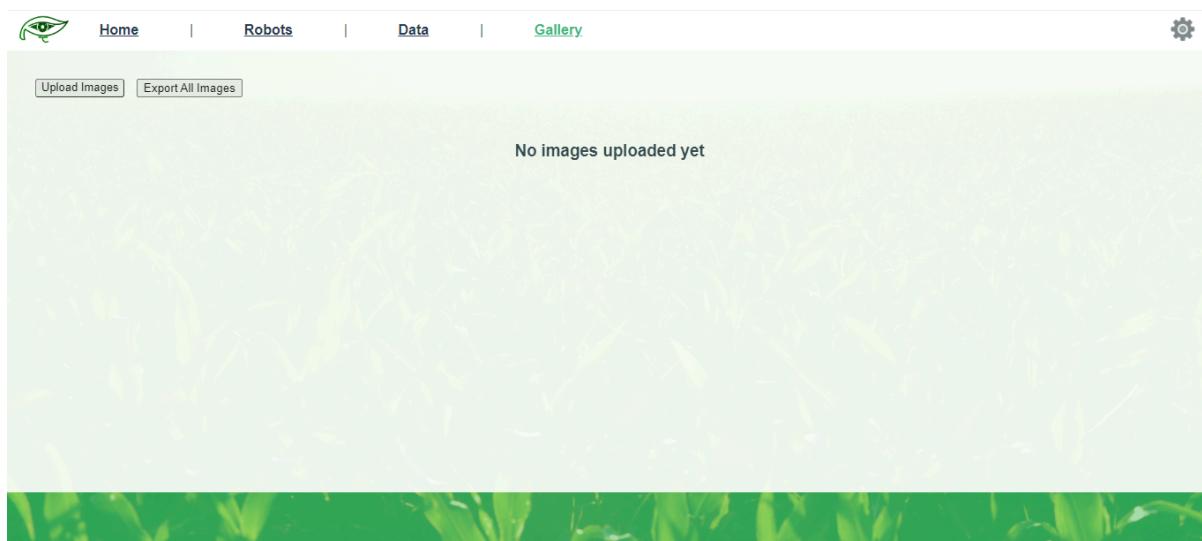
GALERÍA

El propósito de la galería es la visualización y almacenamiento de las imágenes captadas por el robot, y que el mismo almacena en su pendrive. Dicho pendrive puede ser insertado en el puerto de la computadora e importado a la galería.

Debido a que la base de datos de MongoDB tiene limitaciones en cuanto a la cantidad de almacenamiento que puede ocupar cada documento y que en general no está optimizada para almacenar archivos grandes como imágenes, se optó por la base de datos local IndexedDB, que permite almacenar incluso archivos más grandes del lado del cliente (frontend en nuestro caso). Cuando se suben las imágenes desde el pendrive, se convierten las mismas a base64 y se guardan en la colección de ‘image-store’.

Un botón para exportar las imágenes se encuentra junto al de subir imágenes. Este utiliza la librería de JSZip para comprimir todas las imágenes en un archivo zip y que el usuario lo pueda guardar.

Al hacer click en las imágenes en la galería, se sobrepone un overlay con la imagen de tamaño más grande, de modo que al usuario le sea más sencillo enfocar en diferentes aspectos de la imagen.



AJUSTES

El botón de ajustes carga un overlay con diferentes opciones.

- Ayuda: descarga el manual de usuario de la aplicación
- Contactos: contiene un link a la página web de Horus con los contactos del grupo
- Exportar logs: exporta como archivo zip las instancias de detección de toda la base de datos mediante la librería JSZip.

DEPENDENCIAS

- vue-router: permite crear el archivo para las rutas y usar el componente de RouterView para mostrar las views.
- idb: es la librería de IndexedDB que permite guardar información, en este caso imágenes, en el buscador.
- JSZip: es la librería que permite crear archivos comprimidos a partir de la información de logs o de las imágenes.
- leaflet: es la librería que permite crear mapas, capas y marcadores para la ruta de inicio de la aplicación

ELECTRON

Electron.js es una framework que permite la ejecución de la aplicación como aplicación de escritorio. Un archivo de javascript (main.js en electron) indica la creación de una ventana de buscador y controles de ventana cuando se ejecuta. Los paquetes de ejecución se encuentran en el directorio raíz del proyecto.

HARDWARE

COMPONENTES

Raspberry pi 4

Contiene un procesador de cuatro núcleos de 64 bits de alto rendimiento, un soporte de pantalla dual en resoluciones de hasta 4K a través de un par de puertos micro HDMI, hardware de decodificación de vídeo de hasta 4Kp60, hasta 8 GB de RAM, LAN inalámbrica de doble banda de 2,4/5,0 GHz, módulo Bluetooth 5.0, Gigabit Ethernet, USB 3.0 y capacidad PoE (a través de un HAT PoE independiente). Raspberry Pi 4 Modelo B proporciona rendimiento de escritorio comparable a los sistemas de PC x86 de nivel básico.



Especificaciones

Procesador:

Broadcom BCM2711, cuatro núcleos Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz

Memoria:

1GB, 2GB, 4GB or 8GB LPDDR4 (dependiendo del modelo) con on-die ECC

Conectividad:

- 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 × USB 3.0 ports
- 2 × USB 2.0 ports.

GPIO:

Conecotor GPIO estándar de 40 pines

Tarjeta de soporte SD:

Puerto de Micro SD para cargar sistema operativo y almacenamiento de data

Input power:

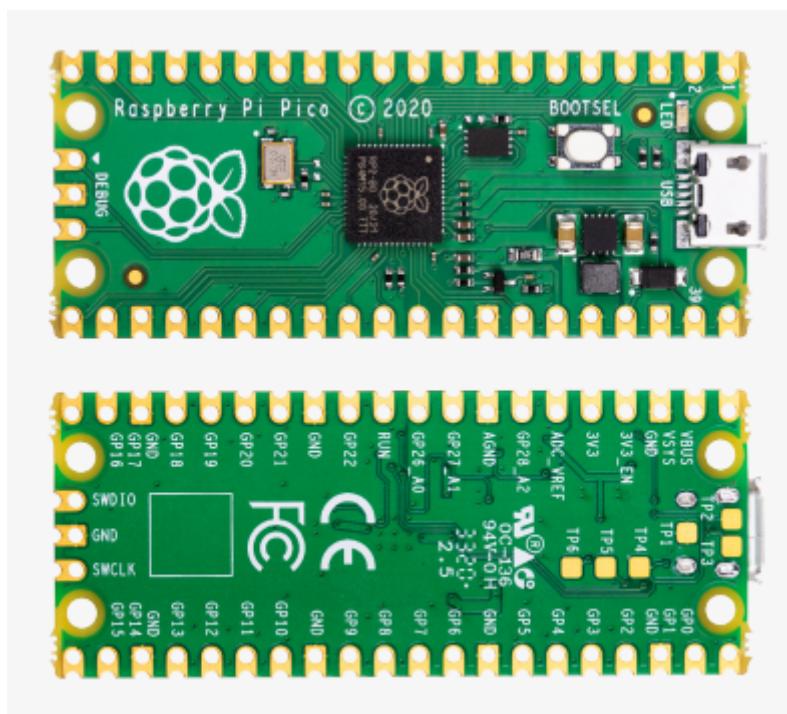
- 5V DC vía conector USB-C (mínimo 3A1)
- 5V DC vía GPIO header (mínimo 3A1)
- Alimentación por Ethernet (PoE)-enabled (requiere PoE HAT separado)

Ambiente:

Temperatura de operación 0–50°C

Raspberry Pi Pico

Pico proporciona un circuito externo mínimo (pero flexible) para soportar el chip RP2040: memoria flash (Winbond W25Q16JV), cristal (Abracon ABM8-272-T3), fuentes de alimentación y desacoplamiento, y conector USB. La mayoría de los pines del microcontrolador RP2040 están disponibles para el usuario en los pines de E/S en el borde izquierdo y derecho de la placa. Cuatro pines de E/S del RP2040 se utilizan para funciones internas: controlar un LED, el control de energía de la fuente de alimentación conmutada (SMPS) a bordo y medir las tensiones del sistema



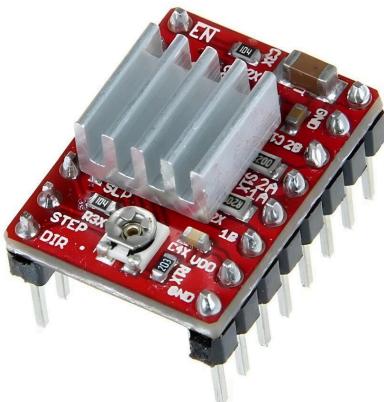
Especificaciones

- RP2040 microcontrolador con 2 MB de Flash
- Puerto Micro-USB B para alimentación y datos (y para reprogramar la memoria Flash)
- PCB de 40 pines en estilo 'DIP' de 21×51 mm, con un grosor de 1 mm y pines pasantes de 0.1", también con festones en el borde.
 - 23 GPIO son sólo digitales y 3 son capaces de funciones de conversión analógica a digital (ADC)
- El puerto de depuración en serie de ARM de 3 pines (SWD)
- Arquitectura de fuente de alimentación simple pero altamente flexible

- Varias opciones para alimentar fácilmente la unidad desde micro-USB, fuentes externas o baterías
- Alta calidad, bajo costo, alta disponibilidad
- SDK completo, ejemplos de software y documentación
- Doble núcleo Cortex M0+ de hasta 133 MHz
 - El PLL integrado permite una frecuencia variable del núcleo
- 264 kB de SRAM de alto rendimiento con múltiples bancos
- Flash Quad-SPI externa con ejecución en el lugar (XIP) y caché integrado de 16 kB
- Bus de alto rendimiento con arquitectura full-crossbar
- USB 1.1 integrado (modo dispositivo o host)
- 30 pines de E/S de propósito general multifunción (4 pueden usarse para ADC)
 - Voltaje de E/S de 1.8 a 3.3V (NOTA: el voltaje de E/S del Pico está fijo en 3.3V)
- 12-bit 500ksps Analogue to Digital Converter (ADC)
- Various digital peripherals ◦ 2 × UART, 2 × I2C, 2 × SPI, 16 × PWM channels
 - 1 × Temporizador con 4 alarmas, 1 × Contador en Tiempo Real
- 2 × bloques de E/S programables (PIO), con un total de 8 máquinas de estado
 - E/S de alta velocidad flexible y programable por el usuario
 - Puede emular interfaces como tarjeta SD y VGA

Drivers A4988

El A4988 es un controlador de motor de micro-paso completo con un traductor integrado para una operación sencilla. Está diseñado para operar motores paso a paso bipolares en modos de paso completo, medio, cuarto, octavo y dieciseisavo, con una capacidad de salida de hasta 35 V y ± 2 A. El A4988 incluye un regulador de corriente de tiempo de apagado fijo que tiene la capacidad de operar en modos de decaimiento lento o mixto.



Especificaciones

- Salidas de bajo RDS(ON)
- Detección/selección automática del modo de decaimiento de corriente
- Modos de decaimiento de corriente mixta y lento
- Rectificación síncrona para baja disipación de potencia
- Protección contra corriente de cruce
- Fuente de lógica compatible con 3.3 y 5 V
- Circuito de apagado térmico
- Protección contra cortocircuitos a tierra
- Protección contra carga en cortocircuito
- Cinco modos de paso seleccionables: completo, 1/2, 1/4, 1/8 y 1/16

Modulo Localizador GY-NEO6MV2

Hay un LED en el módulo GPS NEO-6M que indica el estado de la fijación de la posición.
Parpadeará en diferentes estados dependiendo de su condición:

- **Sin parpadeo** – Está buscando satélites.
- **Parpadeo cada 1 segundo** – Se ha encontrado la fijación de la posición (el módulo puede ver suficientes satélites).



Especificaciones

- Tipo de Receptor: 50 canales, GPS L1 (1575.42 MHz)
- Precisión de Posición Horizontal: 2.5 m
- Velocidad de Actualización de Navegación: 1 Hz (máximo 5 Hz)
- Tiempo de Captura:
 - Arranque en frío: 27 s
 - Arranque en caliente: 1 s
- Sensibilidad de Navegación: -161dBm
- Protocolo de Comunicación: NMEA, UBX Binary, RTCM
- Velocidad de Baudios en Serie: 4800-230400 (default 9600)
- Temperatura de Operación: -40°C ~ 85°C
- Voltaje de Operación: 2.7V ~ 3.6V
- Corriente de Operación: 45mA
- Impedancia TXD/RXD : 510Ω

Módulos LORA SX1278

Los transceptores SX1278 cuentan con el módem de largo alcance LoRa™ que proporciona comunicación de espectro expandido de ultra largo alcance y alta inmunidad a la interferencia, al tiempo que minimiza el consumo de corriente.



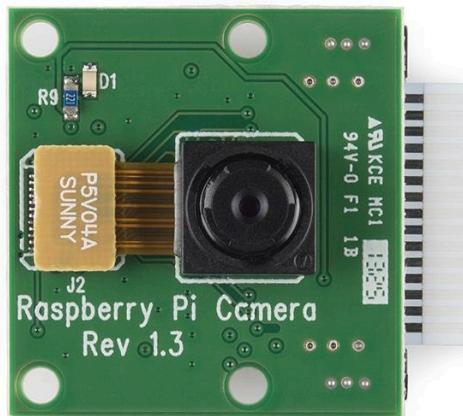
Especificaciones

- Presupuesto de enlace máximo de 168 dB
- Tasa de bits programable de hasta 300 kbps
- Alta sensibilidad: hasta -148 dBm
- Front-end a prueba de balas: IIP3 = -11 dBm
- Excelente inmunidad al bloqueo
- Bajo consumo de corriente RX de 9.9 mA, retención de registro de 200 nA
- Modulación FSK, GFSK, MSK, GMSK, LoRa™ y OOK
- Sincronizador de bits incorporado para recuperación de reloj
- Detección de preámbulo
- Rango dinámico RSSI de 127 dB
- Detección automática de RF y CAD con AFC ultrarrápido
- Motor de paquetes de hasta 256 bytes con CRC
- Sensor de temperatura incorporado e indicador de batería baja
- Paquete QFN de 28 - Rango de operación [-40; +85 °C]

Camara Raspberry Pi 5mp Rev 1.3

Está diseñado específicamente para funcionar con las placas Raspberry Pi 3 y 4 Modelo B.

Se conecta a la Raspberry Pi a través del conector CSI (Interfaz Serial de Cámara) dedicado. El módulo generalmente viene con una lente de enfoque fijo. Algunas versiones pueden permitir el ajuste de enfoque manual. Tiene una resolución de 5 megapíxeles, lo que permite obtener imágenes y videos claros y detallados.



Especificaciones

- Resolución : 5 MP
- Compatibilidad : Raspberry Pi 3B+/4B
- Foco de lente : Fixed Focus
- Tamaño de Imagen(Pixels) : 2592×1944
- Tipo de Interface : CSI(Camera Serial Interface)
- Sensores : Omnivision 5647 fixed-focus
- Apertura : 2.9
- Longitud Focal : 3.29
- FOV : 72.4°
- L x A x H (mm): 25 x 23 x 8
- Peso : 3 g

Motores Stepper NEMA PM-17

Motor paso a paso bipolar con un consumo bajo de corriente por lo que se puede utilizar con casi cualquier controlador de motores como Pi 4 o Arduino. Este motor de excelente calidad cuenta con un cuerpo de metal altamente resistente, fuerte y duradero.



Especificaciones

- Voltaje: 4.8 V
- Corriente: 0.6 A
- Torque: 1600grm/cm
- Temperatura de trabajo -20 a 50 grados
- Par de retención: 12N.cm
- Par de frenado: 1.6N.cm
- Inercia del rotor: 34g.cm²
- Resistencia de aislamiento 100MO 500V DC
- Fuerza eléctrica 820V AC / Min

Servo motor SG90

Pequeño y ligero con alta potencia de salida. El servo puede rotar aproximadamente 180 grados (90 en cada dirección) y funciona igual que los tipos estándar, pero más pequeño. Puedes usar cualquier código de servo, hardware o biblioteca para controlar estos servos.

Es ideal para principiantes que quieren hacer que las cosas se muevan sin tener que construir un controlador de motor con retroalimentación y caja de engranajes, especialmente porque se ajusta en espacios pequeños. Viene con 3 brazos (horns) y hardware.



Especificaciones

- Peso: 9 g
- Dimensión: 22.2 x 11.8 x 31 mm aprox.
- Torque: 1.8 kgf.cm
- Velocidad de Operación: 0.1 s/60 degree
- Voltaje de Operación: 4.8 V (~5V)
- Ancho de banda muerto: 10 μ s
- Rango de Temperatura: 0 °C-55 °C

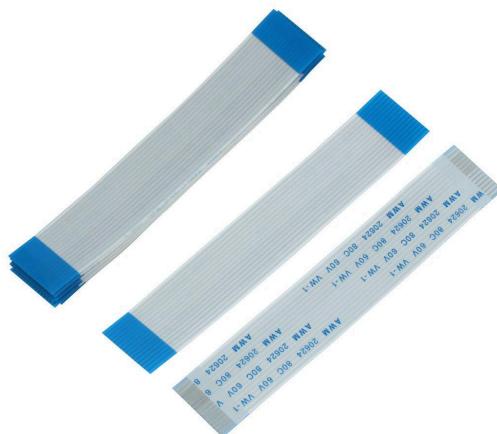
Cable USB

Ideal para aplicaciones de transmisión de datos de alto rendimiento. A diferencia de los cables USB típicos, el cable Cicoil ha sido diseñado para alta flexibilidad, resistencia mecánica, extremos de temperatura y entornos difíciles. Además, la construcción del cable plano garantiza una transmisión de datos clara y de alta velocidad, incluso en los entornos más difíciles.



Cable de Cámara a Rasp

Hecho de metal y PVC con costillas de refuerzo para la reducción del estrés. Paso de 1.0 mm, cable plano flexible de 15 pines y 90 mm de largo



Especificaciones

- Voltaje Máximo: 60V;
- Temperatura: 80C.
- Peso neto: 6g.

Sensor de ultrasonido HC-SR04

Es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición.



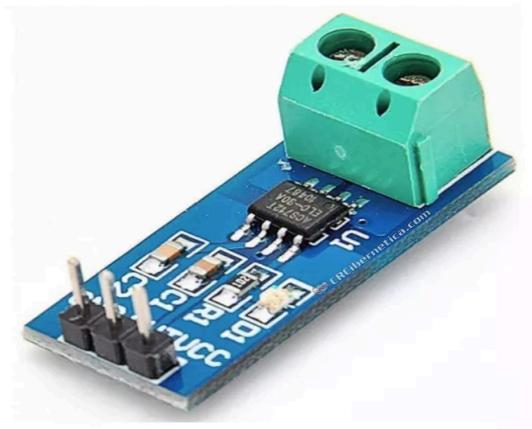
Especificaciones

- Fuente de Alimentación: +5V DC
- Corriente de Reposo: <2mA

- Corriente de Trabajo: 15mA
- Ángulo Efectivo: <15°
- Distancia de Alcance: 2400 cm
- Resolución: 0.3 cm
- Ángulo de Medición: 30°
- Ancho de Pulso de Entrada del Disparador: 10uS
- Dimensión: 45mm x 20mm x 15mm
- Peso: 10 g aprox.

Medidor de corriente Acs712

El ACS712 proporciona soluciones precisas para la detección de corriente CA o CC en sistemas industriales, comerciales y de comunicaciones. El dispositivo consta de un **circuito Hall** lineal, preciso y de baja compensación con una ruta de conducción de cobre ubicada cerca de la superficie de la matriz. La corriente aplicada que fluye a través de esta vía de conducción de cobre genera un campo magnético que el **Hall IC** convierte en un voltaje proporcional. La precisión del dispositivo se optimiza mediante la proximidad de la señal magnética al transductor Hall. El circuito integrado BiCMOS Hall proporciona un voltaje proporcional preciso.



Especificaciones

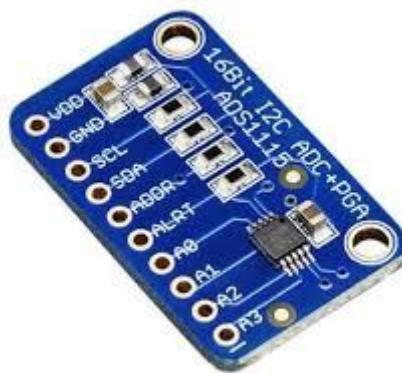
- Ruta de señal analógica de bajo ruido
- Tiempo de aumento de salida de 5 μ s en respuesta a la corriente de entrada escalonada
- Ancho de banda de 80 kHz
- Resistencia del conductor interno de 1,2 m Ω

- 5,0 V, funcionamiento con suministro único
- Sensibilidad de salida de 66 a 185 mV/A
- Tensión de salida proporcional a las corrientes CA o CC
- Tensión de compensación de salida extremadamente estable

Ads 1115

El ADS1115 es preciso, de bajo consumo, 16 bits, compatible con I2C, analógico. Gracias a un amplio rango de suministro operativo del dispositivo, es útil para la medición por sensor con limitaciones de energía y espacio.

Utiliza el comparador digital en el ADS1115 para detección de subtensión y sobretensión.



Especificaciones

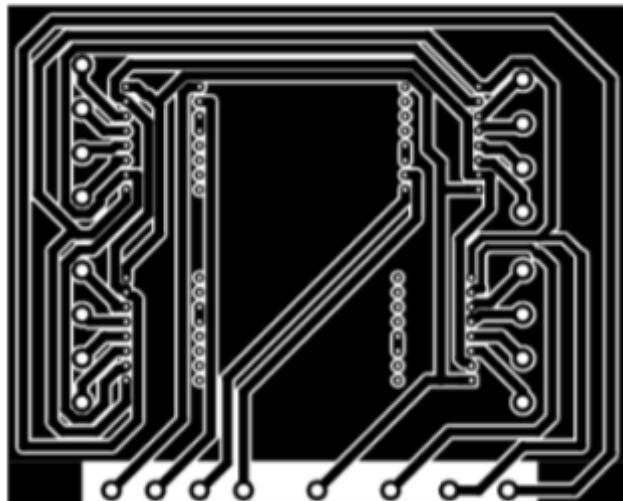
- Paquete QFN ultra pequeño: 2 mm x 1,5 mm x 0,4 mm
- Amplia gama de suministro: 2,0 V a 5,5 V
- Bajo consumo de corriente:
 - Modo continuo: sólo 150 µA
 - Modo de disparo único: apagado automático
- Tasa de datos programable: 8SPS a 860SPS
- Referencia de tensión interna de baja deriva
- Oscilador interno
- Pga Interno

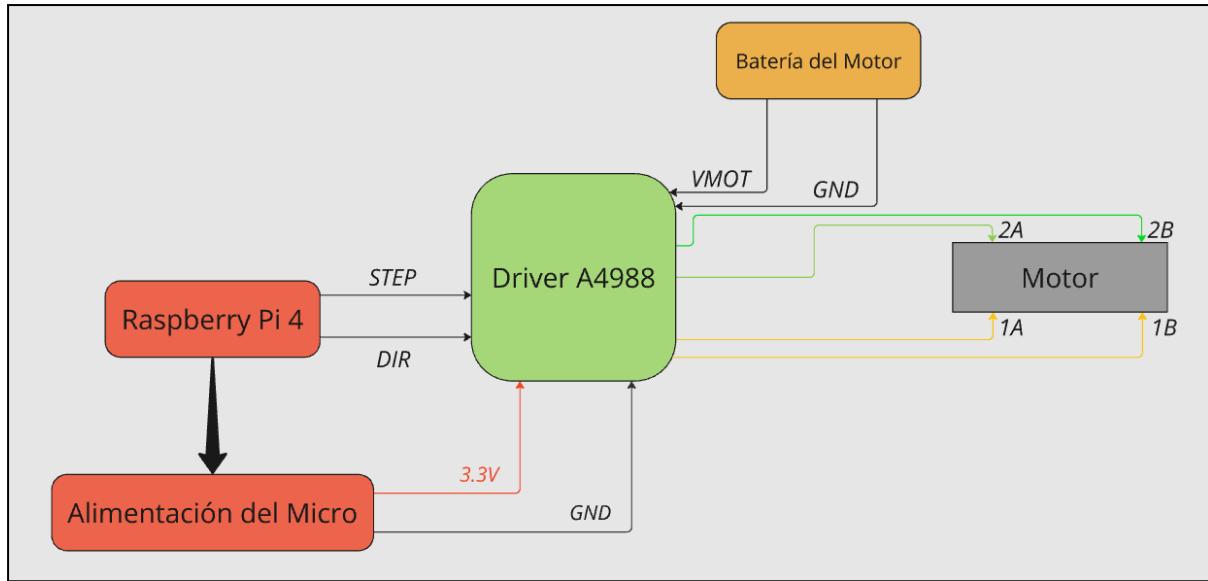
- Interfaz I2C TM: Direcciones seleccionables por PIN
- Cuatro entradas simples o dos diferenciales
- Comparador programable
- Temperatura de funcionamiento: -40°C a +140°C

PLACA DE CONTROL

Los **Drivers** controlan el 1A, 1B, 2A y 2B (estas son las conexiones de las bobinas del motor) a través del **pin DIR**, controlando la polaridad de las bobinas, y envían los pasos que tiene que hacer el motor a través del **pin STEP**. Usamos una señal de **PWM** generada digitalmente con la Raspberry Pi 4 para controlar el paso de cada motor.

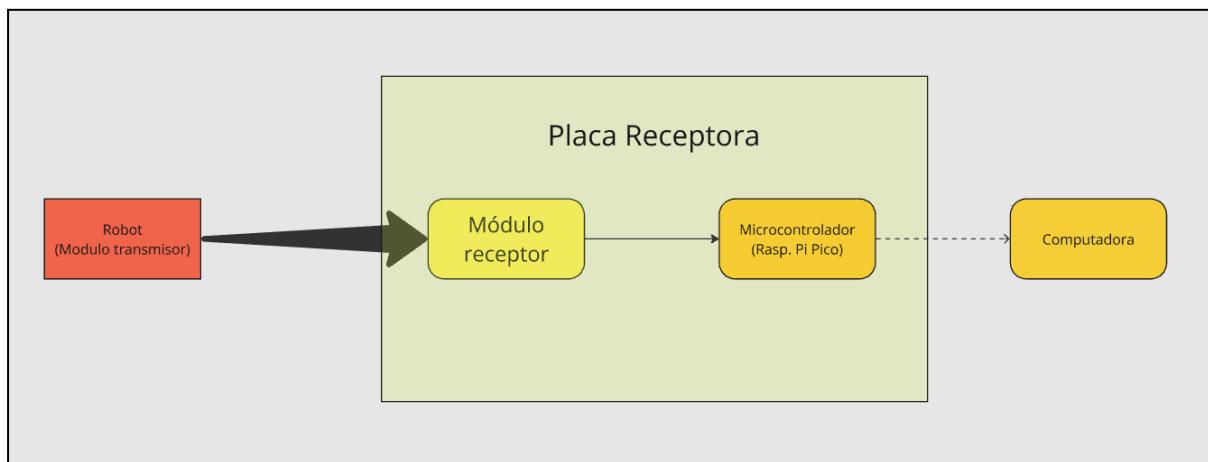
Alimentamos los drivers con la salida del microcontrolador y los motores con la batería. Las conexiones **1A** y **1B**, **2A** y **2B** se pueden alternar para cambiar el sentido de los pasos del motor.





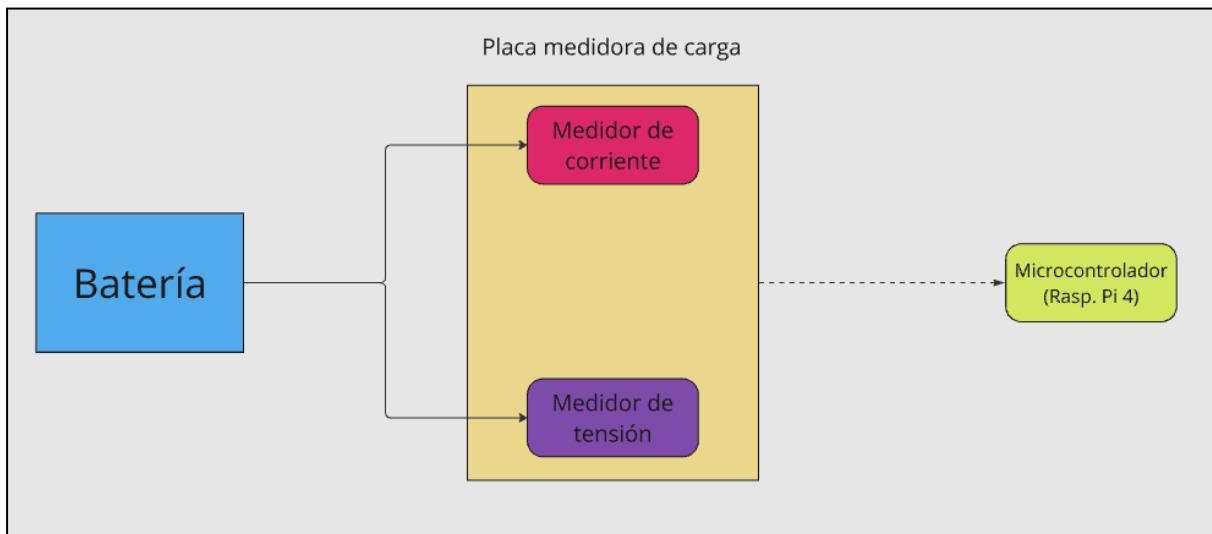
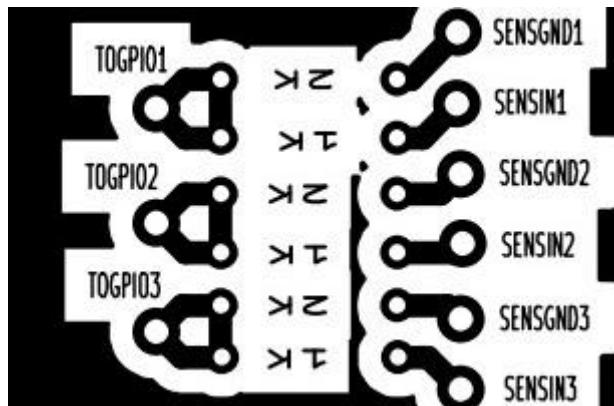
PLACA RECEPTORA

La **placa receptora** es algo simple, cuenta con el **módulo LoRa receptor** y un **microcontrolador Raspberry Pi Pico** que se conecta a la **computadora**. Esta placa le permite a la computadora recibir y procesar toda la información que envía el robot en el campo.



PLACA MEDIDORA DE CORRIENTE

La **placa medidora de corriente** está compuesta por los **ADS** y los **ACS**, y la utilizaremos para saber la cantidad de **carga** y **tensión** que le queda a la batería. La información es enviada a la **Raspberry Pi 4** y esta se encarga de enviarla a través del **módulo LoRa**.



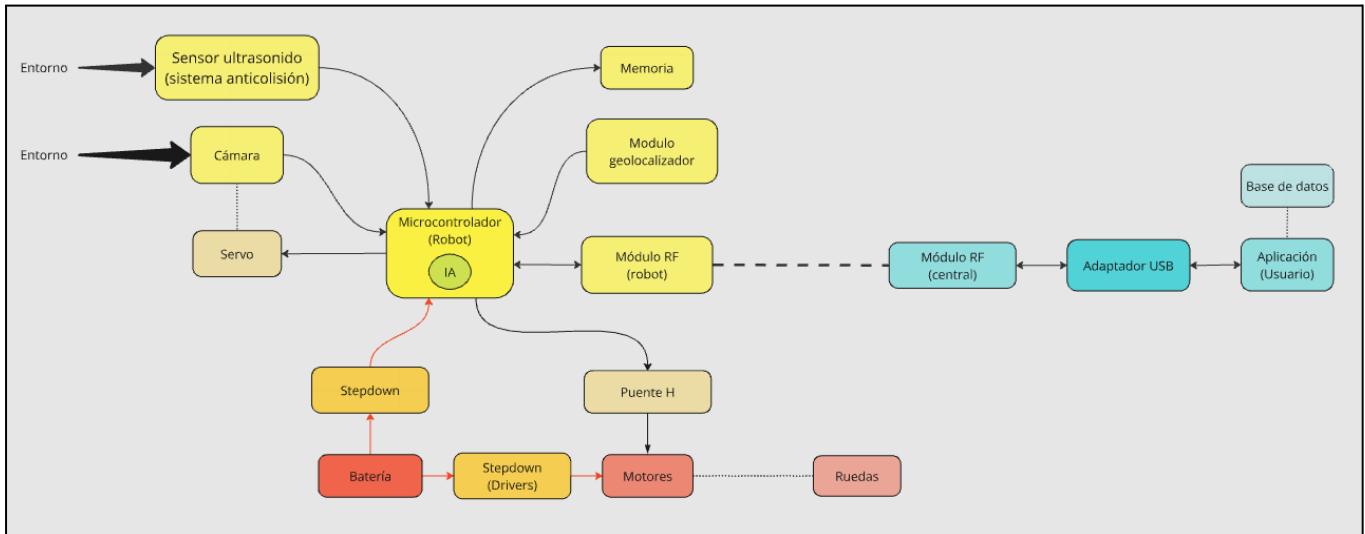
ALIMENTACIÓN

La fuente de energía del robot proviene de una batería de gel Vapex VT 1240, de 12 Volts y 4 Amper-Hora. El tamaño de la batería es de 10.1 cm x 7 cm x 7 cm y pesa alrededor de unos 2 kilos. Utiliza un conector de tipo: FO-01



DIAGRAMA EN BLOQUES

A continuación se muestra el funcionamiento general del prototipo Horus.



ESTRUCTURA

MATERIALES UTILIZADOS

Para la estructura principal utilizamos **Filamento PLA** para impresora 3d 3max Pla+ de 175mm de diámetro, **una Plancha de aluminio**, **Varillas metálicas** y **Pegamento (Gotita)**



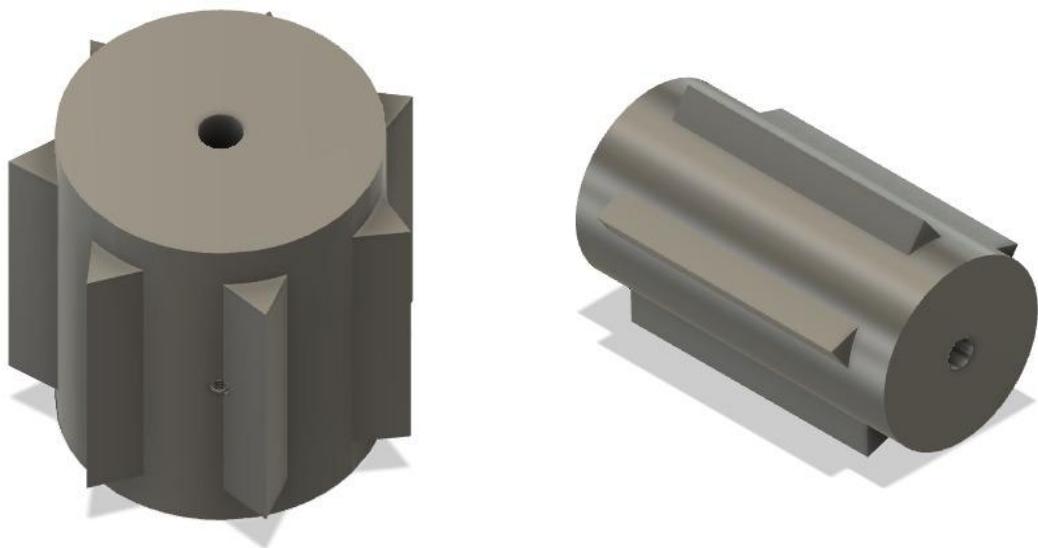
DESCRIPCIÓN DE LA ESTRUC.

El sistema que utilizamos para el movimiento del robot es un tipo de rueda oruga, un sistema de rodamiento implementado en tanques, permitiéndonos que el robot se desplace fácilmente por terrenos difíciles e irregulares. Al tener espacios reducidos para que el robot se desplace, el sistema de rodado de oruga nos permite girar muchos más grados en

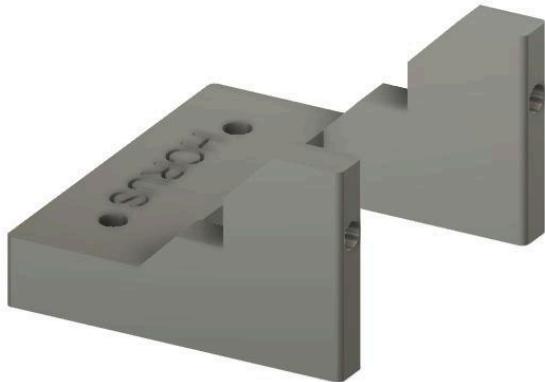
un recorridos cortos. El robot cuenta con una cadena de eslabones (oruga) en ambos costados, contando con 33 eslabones por oruga.



La oruga gira debido a un engranaje acoplado al eje del motor, los dientes del engranaje se encastren entre el espacio de cada eslabón y arrastra la cadena consigo. La imagen debajo es una imagen de referencia, nuestro robot solo cuenta con los cilindros engranados de los costados, en el centro hay un espacio libre.



Los soportes de motores cumplen la función de sostener los motores a la placa de apoyo. Fueron hechos con 100% de relleno para que no se rompan fácilmente y puedan soportar el torque del motor



La estructura principal está formada por la carcasa que cubre la placa de apoyo y las bases donde se apoyan los sistemas eléctricos. La carcasa está separada en 4 paredes con $\frac{1}{4}$ del techo cada parte. Toda la carcasa fue hecha con PLA de color negro y cuenta con un grabado del nombre del proyecto. Las estructura tiene unas dimensiones de 300x300x200 mm y cuenta con una perforación en la parte superior central de unos 50 mm de diámetro.

Las paredes tienen un espesor de 10 mm y perforaciones de 5 mm de diámetro para el encastre de todas las partes. Los encastres se completan al introducir varillas metálicas entre perforaciones y pegando las juntas.

El brazo es un sistema de gato mecánico de tijera, diseñado de esta forma para poder elevarse. Un eje rodado pasante de un a extremo a otro es girado por un motor paso a paso, lo que junta los extremos del brazo y eleva su altura. El brazo tiene una altura máxima de 30 mm y tiene un ancho de 150 mm.

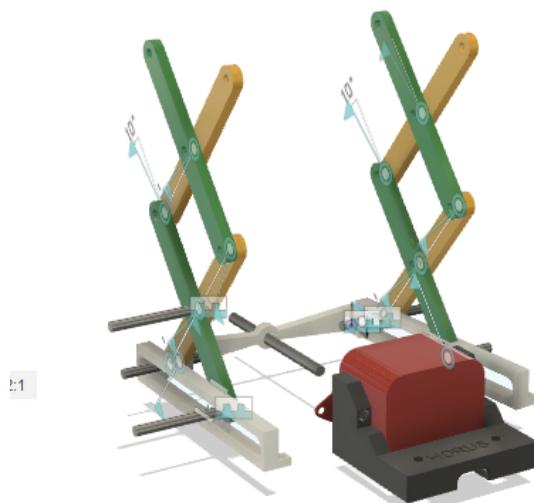


DIAGRAMA 3D

La estructura final mide unos 300x300x200 mm y pesa alrededor de unos 11 kilogramos. Todos los diseños se realizaron en **Fusión** y se utilizó una impresora **ENDER 3 V2** para la impresión de las partes.

PÁGINA WEB

DESCRIPCIÓN GENERAL

Los diseños de la página web se realizaron en Canva. Para la página web utilizamos **CSS**, **HTML** y **JS** como lenguajes bases, y el framework de **Vue.js** para el acceso a diferentes rutas dentro de la página. Todo el programa se realizó en Visual Studio Code, cumpliendo con los estándares de página responsive. La página está hosteada a través del servicio gratuito de **Vercel**.

La página cuenta con dos rutas, la ruta de Home y la de Contactos. La primera está seccionada en 6 partes principales, el **Navbar**, **Portada**, **Objetivo**, **Sponsor**, **Equipo** y **Footer**. La ruta de contactos cuenta con toda la información necesaria para comunicarse con nosotros e incluso los links para nuestras diferentes redes sociales.



Nuestro Objetivo

Nuestro objetivo es brindar una herramienta que permita a los productores de campo en la Argentina combatir de forma más efectiva las plagas que afectan las cosechas de soja.

Esta herramienta permitirá reconocer las 4 plagas principales que pueden estar afectando a la planta: las orugas defoliadoras, las chinches, las arañuelas y los trips; y dar un informe detallado al usuario sobre el tipo y donde se encuentra.



"Agradecidos con todos
aquellos que nos brindaron
su apoyo"

Equipo HORUS



Nuestro Equipo

Conozcan un poco más sobre nuestro equipo y sus opiniones en base al proyecto que desarrollamos



El desarrollo de la IA fue muy complicado pero eso no me detuvo. Superé los desafíos y alcancé las metas establecidas.

Díaz, Lara



Si le preguntas a Chat GPT que te dé un proyecto invaluable, HORUS sería su respuesta. Propuse una idea innovadora y mi equipo no dudó en trabajar en ella para desarrollarla.

Kearney, Mateo



Sin comentarios.

Torres, Juan Ignacio



Para la gente común diseñar es complicado, para mí es muy fácil.

Ferrante, Santino



Muy lindo.

Contreras, Abril



Sin mí no serían nada.

Cabrera, Martín



Tel +54 9 11 6939-4245 agroohoruss@gmail.com

HORUS © All Rights Reserved. Designed by Kearney Mateo