

Carpeta de Campo



LA INNOVACIÓN DE LA AGROINDUSTRIA

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

HORUS

BITÁCORA

-----Semana del 11/03

13/03

- Anteproyecto

15/03

- Anteproyecto

-----Semana del 18/03

20/03

- Anteproyecto
- Investigación de encuentros agrarios y rurales para asistir y publicitar el proyecto.

22/03

- Anteproyecto
- Comienzo del diseño del logo en app de código abierto InkSpace

Ideas:

Ojo de Horus

Cámara -> ojo

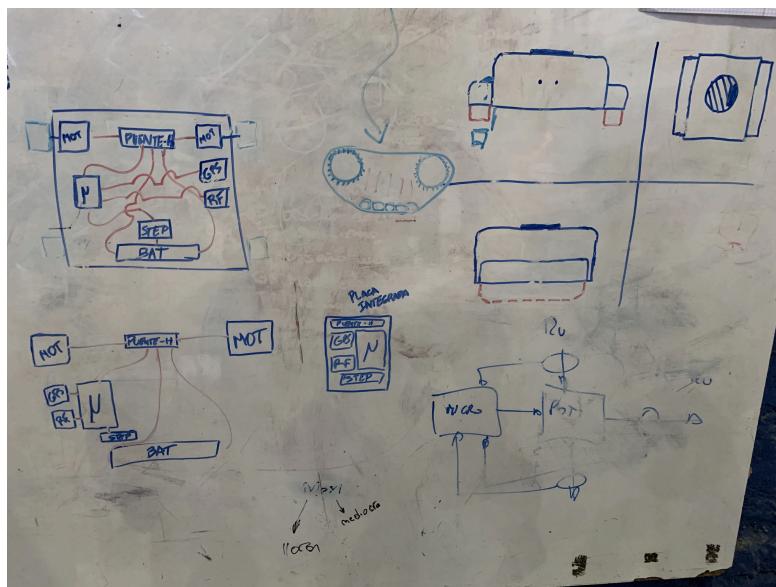
Hoja de planta



-----Semana del 25/04

27/03

- Introducción al sistema de scrum.
- Configuración de cuentas para github y trello.
- Ordenamiento del espacio de trabajo.
- Diseño del robot, boceto de ideas iniciales:
Círculo interno, croquis de 3 vistas.



29/03

- Feriado

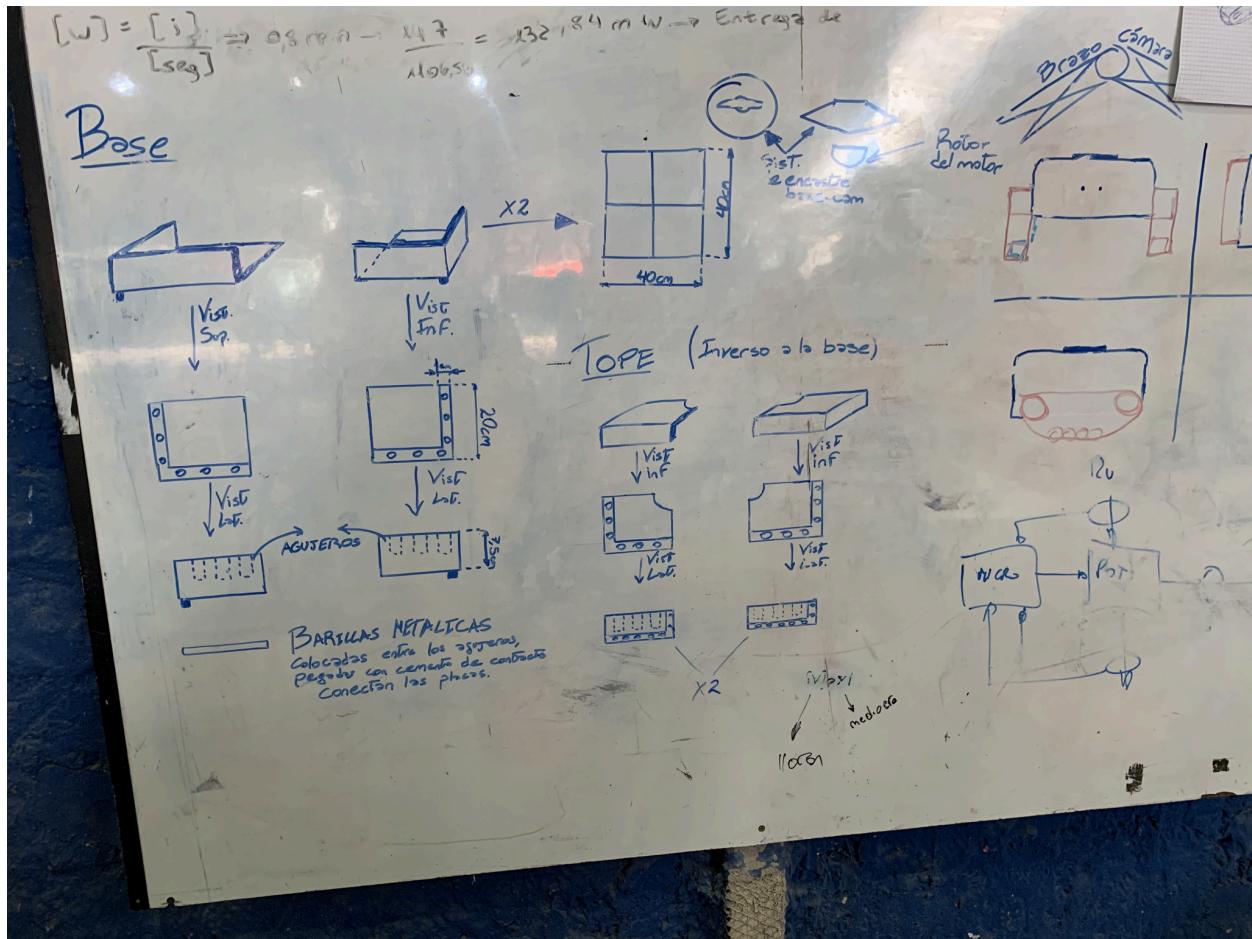
-----Semana del 01/04

01/04

- Feriado

03/04

- Seguimiento del diseño del robot:



- Seguimiento del diseño del logo.



Fondo -> colorea

- Investigación - Frameworks para el desarrollo del backend de la aplicación.
 - Corrección de problemas de sobreescritura en el código base para el armado de datasets para la IA.

- Redacción de mail para sponsors y confección de lista de los mismos.

----- Semana del 08/04 -----

08/04

- Se realizó linktree.



- Continuar con la investigación de herramientas y métodos para el desarrollo de la aplicación y la base de datos.
- Crear un dataset de imágenes de prueba, anotar cada imagen creada en el dataset con CVAT, y desarrollar el código del modelo CNN YOLOv 5.
- Se finalizó el logo del proyecto Horus.



- Comienzo de diseño de folleto e infografía del proyecto:
<https://www.canva.com/design/DAGB2SzHacY/ZXoww8wi6I7IgXtqEkR33w/view>

09/04

- Se continuó con el diseño del folleto.
<https://www.canva.com/design/DAGB2SzHacY/ZXoww8wi6I7IgXtqEkR33w/view>
- Se comenzó el diseño del primer post de instagram.

10/04

- Se realizó documentación de tamaño y peso de componentes
 - Pesos y tamaño**
 - Estructura 400x400 peso 2kg
 - motores 150x60 peso 560g
 - servo 40x20 peso 55g
 - camara 25x24 peso 15g
 - raspberry pi 4 85x56 peso 100g
 - lectora de sd 50x30 peso 10g
 - geolocalizador 35x25 peso 17g
 - adaptador 20x30 peso 15g
 - modulo rf 30x50 peso 20g
- Se terminó el código de inicio del modelo CNN YOLOv5, probar su entrenamiento con imágenes de prueba.
<https://colab.research.google.com/drive/1NvxYCrydiz7HrLsaFG-PK0eY3rEnZiT?usp=sharing>
- Se terminó el folleto para los eventos rurales
<https://www.canva.com/design/DAGB2SzHacY/ZXoww8wi6I7IgXtqEkR33w/view>

- Se realizó el boceto de aplicación

- Charla de sponsors con Medina:

Anotaciones

Comunicación

Escrito

-Mail (cc) -Wpp -Ig -Telegram -LinkedIn -Github

Voz

-Llamada -Discord -Wpp -Meet (Videocall)

Personal

-Charla -Eventos -Entrevista

A las empresas se les puede aportar conocimientos nuevos, fuerza de trabajo (/capacitada), innovación, visibilidad [medios de comunicación (radios/ TV/ Stream)], soporte social [conciencia (cosas que no se hacen que se empiecen a hacer) e interés]. Los medios más sencillos son los escritos y gráficos.

Informar que estamos trabajando en algo y decir que actualizariamos a futuro para una posible entrevista (cómo reservar una entrevista)

Como se presenta uno es muy importante. Respetuosos y formales

Objetivo

Condensar en el menor tiempo posible lo que quiero decir. Menos de 2 minutos. Generar interés

Speech elevator

- Inicio de la interfaz de la aplicación (Frontend)

<https://colab.research.google.com/drive/1NvxYCrydiz7HrLsaFG-PK0eY3rEnZiXt?usp=sharing>

Entrenamiento de 60 épocas realizado de google Colab para la IA de prueba, modelo RECOM de entrenamiento dataset personalizado.
Código de entrenamiento finalizado.

-----Semana 15/04

15/04

- Se continuó enviando mails a empresas para posible sponsoreo
- Se empezó a escribir el Pitch Elevator
- Comenzó el diseño de Mockup de la aplicación en Marvel
- Realización de código de detección y probarlo usando el modelo de prueba creado con el custom DATASET
- Recopilación de imágenes de orugas y armado de DATASET para crear un nuevo modelo

17/04

- Se enviaron mails a posibles sponsors
- Diseño de suspensión y del brazo de cámara
- Desarrollo del programa IA para detección basada en YOLOv 5
- Se completó el Mockup en Marvel de la aplicación
<https://marvelapp.com/prototype/c6i1ga7/screen/94383038>
- Comenzó la redacción de las historias de usuario
- Investigación de fechas de eventos rurales/ agrarios
- Comienzo de diseño del Segundo post de Instagram
- Empezó el desarrollo del Front-End

18/04

- Se continuó con el desarrollo del Front-End
- Se terminó el diseño del Segundo post de Instagram y se subió

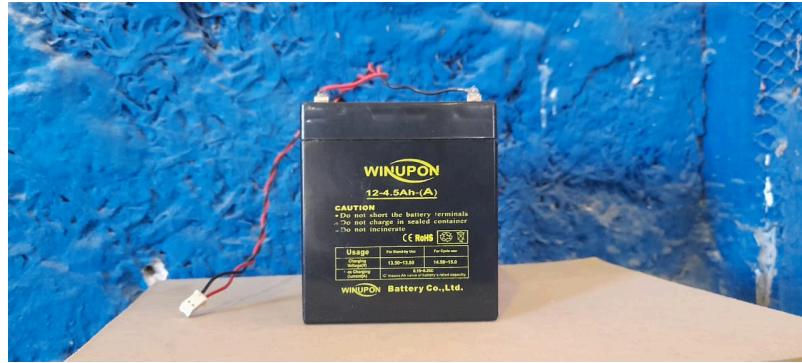


- Desarmado de Impresora para reciclar sus partes y componentes



19/04

- Se cargó y midió una batería para comprobar si estaba apta para su uso

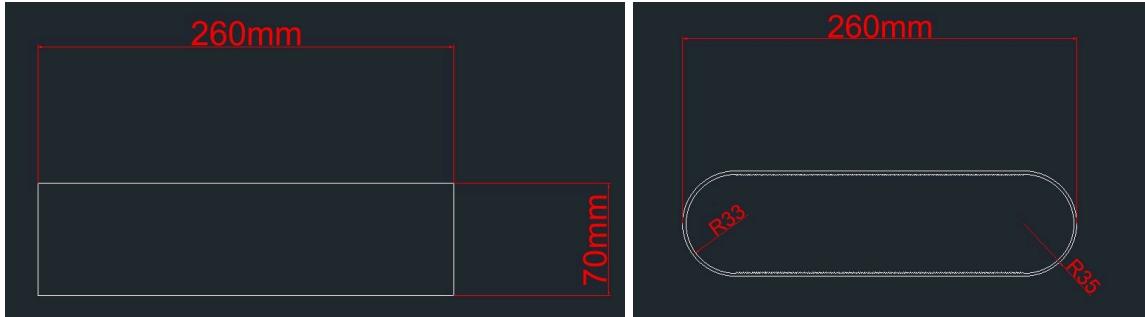


- Se solucionaba el error de archivo modelo.pt para detectar en computadora local
- Se continuó con el desarrollo del Front-End

-----Semana 22/04

22/04

- Se tomaron las medidas de las correas sacadas de la impresora



- Se terminó el documento de las historias de usuario
<https://docs.google.com/document/d/1Auik0VvRMzf7xbtGVN5kx15p3YYn1d4K1Fu4YquHcns/edit?usp=sharing>
- Se solucionó el error de archivo modelo.pt para detectar en computadora local

```

1  # YOLOv5 by Ultralytics, AGPL-3.0 license
2
3  Run YOLOv5 detection inference on images, videos, directories, globs, YouTube, webcam, streams, etc.
4
5  Usage - sources:
6      $ python detect.py --weights yolov5s.pt --source 0                         # webcam
7      $ python detect.py --weights yolov5s.pt --source img.jpg                      # image
8      $ python detect.py --weights yolov5s.pt --source vid.mp4                      # video
9      $ python detect.py --weights yolov5s.pt --source screen                           # screenshot
10     $ python detect.py --weights yolov5s.pt --source path/                         # directory
11     $ python detect.py --weights yolov5s.pt --source list.txt                      # list of images
12     $ python detect.py --weights yolov5s.pt --source list.streams                  # list of streams
13     $ python detect.py --weights yolov5s.pt --source path/img                      # glob or pattern
14     $ python detect.py --weights yolov5s.pt --source https://youtu.be/IuMxDXcvI4"    # YouTube
15     $ python detect.py --weights yolov5s.pt --source rtsp://example.com/media.mp4"   # RTSP, RTMP, HTTP stream
16
17  Usage - formats:
18      $ python detect.py --weights yolov5s.pt                                         # PyTorch
19      $ python detect.py --weights yolov5s.torchscript                            # TorchScript
20      $ python detect.py --weights yolov5s.onnx                                    # ONNX Runtime or OpenCV DNN with --dnn
21      $ python detect.py --weights yolov5s.openvino_model                         # OpenVINO
22      $ python detect.py --weights yolov5s.engine                                # TensorRT
23      $ python detect.py --weights yolov5s.mlmodel                             # CoreML (macOS-only)
24      $ python detect.py --weights yolov5s.saved_model                          # TensorFlow SavedModel
25      $ python detect.py --weights yolov5s.pb                                    # TensorFlow Model
26      $ python detect.py --weights yolov5s.tflite                               # TensorFlow Lite
27      $ python detect.py --weights yolov5s.edgetpu.tflite                        # TensorFlow Edge TPU
28      $ python detect.py --weights yolov5s.paddle_model                         # PaddlePaddle
29
30

```

- Se realizó la lista de eventos festivos/ importantes para los posts/ historias de Instagram
<https://docs.google.com/document/d/1IGMzDqZMR-gsLaUyZ9C1PUZXLC5Lfvy/edit?usp=sharing&ouid=105637000038282676180&rtpof=true&sd=true>
- Se corrigió error de muestreo de etiquetas de inferencia (Modelo IA)
- Continúa el desarrollo del Front-End de la aplicación

24/04

- Comenzó el diseño inicial de la carcasa en fusión 360
- Terminó el DATASET de orugas para modelo oficial y se hizo el modelo 2.0
- Investigación sobre el protocolo de comunicación
- Inició el diseño de la historia para el día mundial de la seguridad y salud en el trabajo
- Se filmó a los encargados de desarrollo de la IA para un futuro video
- Se continuó el desarrollo del Front-End de la aplicación

-----Semana 29/04

29/04

- Se diseñó la imagen para una de las historias destacadas de Instagram.
- Sigue el diseño estructural en 3D del robot
- Se subió una historia del progreso de la IA en Instagram
- Se empezó a desarrollar el backend de la aplicación (Conectar una database de mongodb a node.js)
- Se revisó la potencia de los motores nema 17 PM-K018
- Comienzo de diseño del post por el día del trabajador para Instagram

30/04

- Enviamos mails a Sponsors

- Terminó el diseño del post por el día del trabajador
- Continuó el diseño en 3D de la estructura del robot

1/05

- Feriado

----- **Semana 06/05**

06/05

- Continuó el envío de mails para juntar sponsors
- Se continuó con el diseño en 3D de la estructura del robot
- Llenado de planilla de compras

07/05

- Inicio del diseño de la historia y publicación sobre el día de la Cruz Roja para Instagram

08/05

- Terminó el diseño de la historia y publicación sobre el día de la Cruz Roja para Instagram
- Se comenzó con el diseño de un modelo de prueba de la estructura
- Comienzo del Mockup de la página web
- Conectar el código de Capturador.py con detector.py
- Empezó la búsqueda de ideas para futuros reels de Instagram
- El diseño de stickers se empezó
- Continuó la búsqueda de posibles sponsors
- Resolver problemas de librería “concurrently” de la aplicación

10/05

- Terminamos el diseño del sticker

----- **Semana 13/05**

13/05

- Continuó el Mockup de la página

15/05

- Terminamos el mail para enviarlo a las empresas de medios
- Se empezó a mandar mails a las estaciones de radio para una promoción
- Se siguió con el código de conexión del código de Capturador.py con el Detector.py
- Continuó el diseño de prueba de la estructura
- Siguió la conexión del archivo main.py con el server de node en la computadora
- Continuó el diseño del MockUp de la página Web
- Cambios de foto de perfil en Trello
- Se hizo una historia por el día mundial del reciclaje

16/05

- Se siguió con el Mockup de la página Web

17/05

- Continuó el diseño del MockUp de la página
- Enviamos mails a estaciones de radio
- Se realizó el primer reel de Instagram y se publicó

Semana 20/05

21/05

- Retoque del MockUp de la página Web

22/05

- Continuamos con el desarrollo del modelo de pruebas de la estructura
- Se pensaron nuevas ideas para futuros reels
- Continuamos mandando mails a sponsors

- Investigamos sobre la geolocalización
- Se comenzó a comentar y organizar el código de la IA
- Se sacaron fotos a orugas falsas en plantas reales
- Se resolvió el error en la conexión con la base de datos

24/05

- Terminó el diseño del MockUp de la página Web
- Continuó el desarrollo de la App

----- **Semana 27/05**

28/05

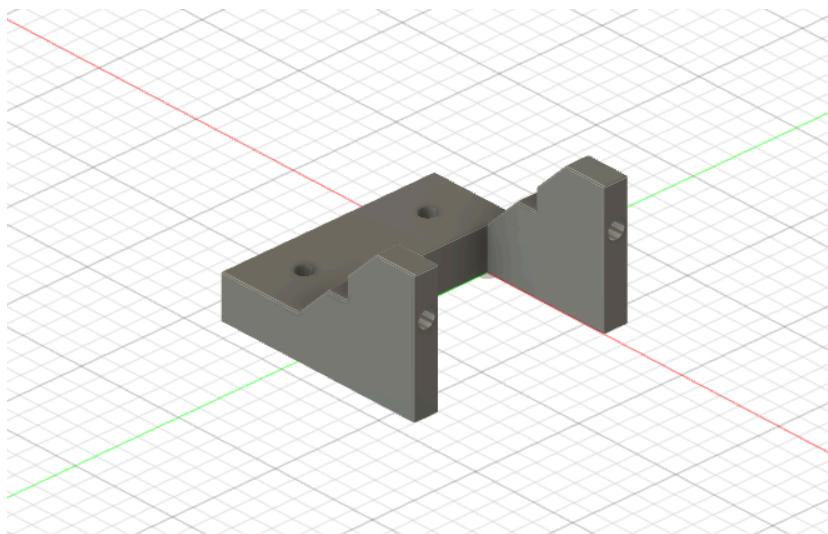
- Terminó el desarrollo del modelo de prueba del robot y se probó

29/05

- Fuimos a la base de la Fuerza Aérea
- Comenzó el diseño de la página Web

31/05

- Se investigó y comenzó el desarrollo de los drivers para los motores stepper
- Empezó el diseño del modelo de soporte de motores



- Se continúo entrenando a la IA con nuevas fotos
- Empezó el diseño de una de las rutas de la App

- Continuó el desarrollo de la página Web
- Empezó la anotación de 3500 imágenes para el dataset nuevo

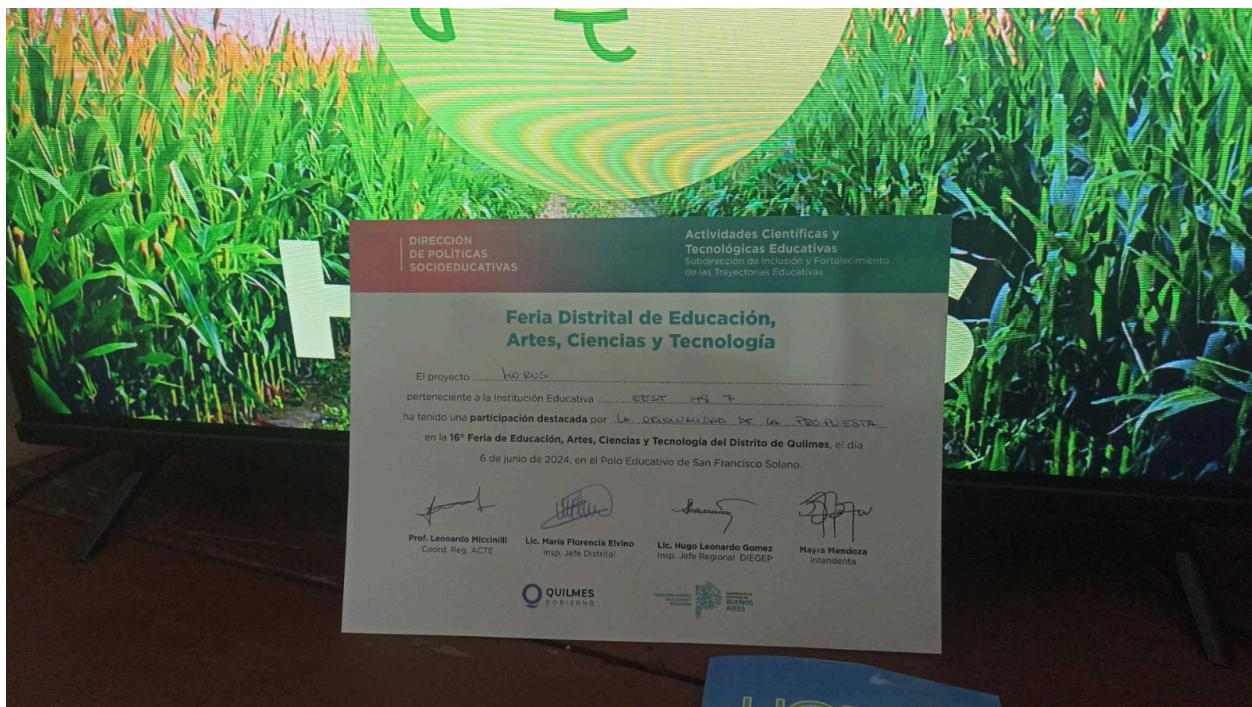
-----Semana 03/06-----

05/06

- Preparación para exposición en Solano

06/06

- Participación de la exposición en Solano



07/06

Investigación de modelo de geolocalización python

Nota

- El voltaje de operación del módulo es de 3.3V - 5V, pero la lógica del puerto serial (Tx, Rx) es de 3.3V. Si se va a utilizar el pin Rx con lógica de 5V, se debe emplear un divisor de tensión para reducirla a 3.3V. Por lo general solo se usa el pin Tx ya que para conocer el posicionamiento solo se requiere leer las sentencias NMEA que genera el chip.

-----Semana 10/06-----

10/06

- Se avanzó con la página web (html y vue completo, css de "contactos")

- Se avanzó con la aplicación (ruta de data)

12/06

- Entrevista con la radio Cocktail en Solano.

-----**Semana 17/06**

19/06

- Continuar con el desarrollo del sistema de control de los steppers con el driver.
- Continuar diseño de brazo para elevar la cámara

-----**Semana 24/06**

26/06

- Realizar entrenamiento cruzado IA de orugas con 5 datasets:
Debido al uso limitado de google colabs decidimos separar los entrenamientos en fases.
- Continuar con el diseño de la página web (contactos)
- Reel para ig (Dia con el equipo Horus)
- Seguir con la ruta de data de la app
- Realizar modificaciones soporte de motor
- Probar el funcionamiento del driver a4988 y continuar con el circuito para el stepper
- Recibimos raspberry pi 4 y tarjeta micro sd de cooperadora

28/06

- Recibimos los stickers Horus

-----**Semana 01/07**

01/07

- Set up de raspberry pi 4: instalación de sistema operativo y visualización por VNC desde computadora externa.
- Continuar diseño de página web (home)
- Comunicación frontend-backend (app)

02/07

- Comienzo de comunicacion backend-frontend (fetch requests)

03/07

-----**Semana 08/07**

09/07

- Armamos un post para Ig
- Continuó el desarrollo de la página web
- Conseguimos materiales para la PCB de los steppers
- Se hizo código para raspberry y servo en python y se probó.
- Se programó el filtro para los logs de Data

10/07

- Se diseñó la sección de contactos de la página web

-----**Semana 29/07**

31/07

- Se empezó a modificar el google colab para trabajo AFK
- Solución de error con el ingreso sucesivo de archivos json al servidor
- Averiguamos el tipo y dureza de resorte para diseño de suspensión
- Se realizó y postear un nuevo reel

02/08

- Se desarrolló código para driver a4988 y steppers con micropython
- Probamos código de motores stepper variando RPM
- Empezó el diseño de la PCB para alimentación y Control de motores Stepper

-----**Semana 05/08**

07/08

- Se hizo el código de prueba del módulo lora lora para raspberry pi Sx1278 433mhz

09/08

- Se consiguieron los materiales para la placa PCB de los motores stepper.
- Empezamos a realizar código de servo motor en python y probarlo
- Finalización de comunicación backend-frontend (fetch requests)
- Comienzo de selección de filtro para logs (Data)
- Corrección de error de enrutado de PCB para motores stepper
- Conseguimos los materiales para la placa PCB de los motores stepper.

Semana 12/08

13/08

- Subimos un nuevo post en ig sobre arañuelas

16/08

- Rehacer PCB de drivers y steppers para 2 drivers a4988, terminando el diseño y solucionando el error
- Imprimimos y planchamos el circuito en la PCB de motores stepper
- Probamos el código modulo LORA con las raspberrys
- Realización de post de ig sobre Trips

Semana 19/08

19/08

- Agujereamos la PCB para los motores steppers
- Soldamos los pines al módulo NEO6M-GPS
- Se inició el diseño del adaptador para módulo Lora sx1276 a Protoboard
- Empezamos a escribir el código para módulo gps y probarlo en la raspberry

21/08

- Se diseñó la suspensión del robot

22/08

- Terminó el diseño del Footer - Página Web
- Terminó el diseño de la sección de nuestro equipo - Página Web
- Se logró modificar el google colab para trabajo AFK
- Entrenamos el Modelo 1.0v y Prueba
- Comenzamos a Imprimir, Planchar y Hacer los Agujeros para la placa adaptador para módulos lora
- Preparamos y soldamos cables a la PCB para motores stepper

23/08

- Se diseñó la sección de sponsors - Página Web
- Se diseñó la sección de Nuestro Objetivo - Página Web

24/08

- Terminó el diseño de la sección de Nuestro Objetivo - Página Web

25/08

- Terminó el diseño de la sección de sponsors - Página Web
- Terminar el diseño de la sección de inicio - Página Web

Semana 26/08

26/08

- Se terminaron de soldar los cables de la PCB para motores stepper
- Se terminó el adaptador para módulo Lora sx1276 a Protoboard
- Creamos las huellas con Logo y QR Mockup para PCB's
- Armaron el prototipo de suspensión v1 y cortaron los resortes
- Se diseñó el Navbar - Página Web

28/08

- Terminamos la selección de filtro para logs (Data) y empezó la selección de Orden para logs (ruta de Data)
- Soldamos módulos Lora RF a cables y de cables a placa
- Comenzamos a ordenar y comentar el código hasta ahora

-----**Semana 02/09**

02/09

- Terminamos el diseño de la Página Web
- Empezamos a investigar cómo subir la Página a Internet
- Corregir medidas para Modulo Adaptador Lora sx1278 a Protoboard
- Post de Chinches para IG
- Realización de reel y publicación a IG
- Hicimos los Agujeros para la placa adaptador para módulos lora lora
- Programamos la ruta de Data en la app

06/09

- Logramos subir la Página Web a la red
<https://horus-pagina-web.vercel.app/>
- Soldamos los cables de los módulos LORA a las placas

-----**Semana 09/09**

09/09

- Probamos los motores stepper con la PCB
- Continuó la investigación para comunicación entre Módulos LORA

10/09

- Empezó la investigación sobre los tópicos a mencionar en la Carpeta Técnica y Carpeta de Campo
- Empezamos a diseñar la portada de la Carpeta de Campo y Carpeta Técnica

11/09

- Terminó el diseño de la portada de Carpeta Técnica y de la Carpeta de Campo



- Diseñamos el índice y conversamos sobre algunos de los anexos que debe contener

ÍNDICE

INTRODUCCIÓN	OBJETIVO ALCANCE BENEFICIOS	00
Software	EXPLICACIÓN GENERAL ANEXO MICROCONTROLADOR ANEXO COMUNICACIÓN ANEXO GEOLOCALIZACIÓN ANEXO LOGS APLICACIÓN DESCRIPCIÓN GENERAL BASE DE DATOS	01
Hardware	DIAGRAMA EN BLOQUES COMPONENTES ESQUEMA DE PLACA DE CONTROL DE MOTORES COMPONENTES ELECTRÓNICOS UTILIZADOS ALIMENTACIÓN	02
Estructura	DIAGRAMA 3D DESCRIPCIÓN DE LA ESTRUCTURA MATERIALES UTILIZADOS	03
Página Web	DESCRIPCIÓN GENERAL	04

12/09

- Empezamos a diseñar el banner del proyecto

13/09

- Terminó el modelo 1.5 de la IA
- Continuamos con la prueba de los módulos LORA

----- **Semana 16/09**

16/09

- Prueba de código de módulos LORA
- Avanzamos con la aplicación
- Empezamos a diseñar tensores de orugas y marcar agujeros para soportes de motores y suspensión
- Transcribimos el código de control de motores stepper de Micro Python a Python

17/09

- Revisamos asuntos financieros del proyecto

18/09

- Empezamos a desarrollar la información de la documentación técnica
- Continuaron las pruebas de los módulos LORA

-----**Semana 23/09**

23/09

- Probamos nuevamente los motores stepper con la placa de control, ya con los ejes cambiados
- Se crearon las huellas con Logo y QR para las últimas PCB's
- Se escondió la scrollbar de la página web y de la aplicación de escritorio, y se solucionaron problemas de adaptación a la resolución pantalla
- El modelo IA 1.5v tuvo un entrenamiento intensivo
- Se realizó el código de servo motor en python y lo probamos

24/09

- La selección de orden para logs (ruta de Data) para la aplicación se terminó
- Empezamos a programar ruta de Robots (App)
- Hicimos un post nuevo para nuestro instagram

25/09

- Se montaron piezas a la estructura principal



- Empezó a la ruta de Galería (App) y continuamos programando la ruta de Robots
- El diseño de tensor de orugas se empezó
- Escribimos el código para módulo gps y probarlo en la raspberry
- Se terminó de probar el código de los módulo LORA con las raspberrypi pico
- Empezamos a buscar la solución al error SPI de los módulos LORA y al error de bloqueo del módulo gps
- Continuó el desarrollo del Informe Descriptivo y de la Carpeta Técnica, agregandole información sobre el capítulo: Página Web, Componentes, Introducción y Estructura
- Se terminó el diseño de tensor de orugas

-----Semana 30/09

30/09

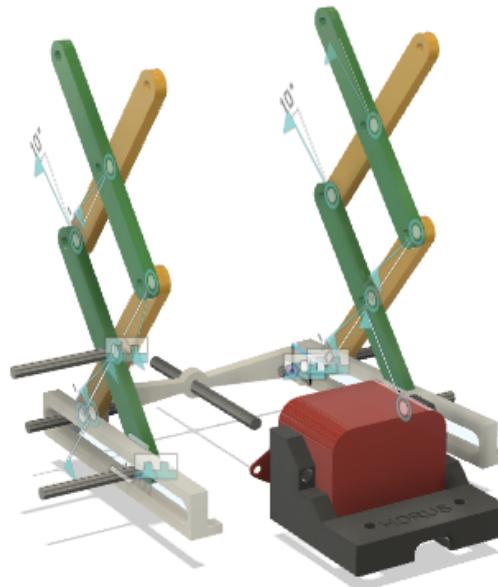
- Se hizo el banner del proyecto
- Agregamos diversos diagramas en bloques e información para explicar mejor el funcionamiento del proyecto y sus partes
- Terminó de programarse la ruta de Galería y empezamos a programar la ruta de Home
- Realizamos el código de servo motor en python y probarlo
- Continuó y terminó la búsqueda de una solución para error SPI de los módulos LORA

02/10

- Continuamos agregando información a la Carpeta Técnica e Informe Descriptivo
- Terminamos el diseño del tensor de orugas



- Adaptamos el programa de Lora sx1278 para la raspberry pi 4
- Se hizo el diseño de brazo elevador de camara



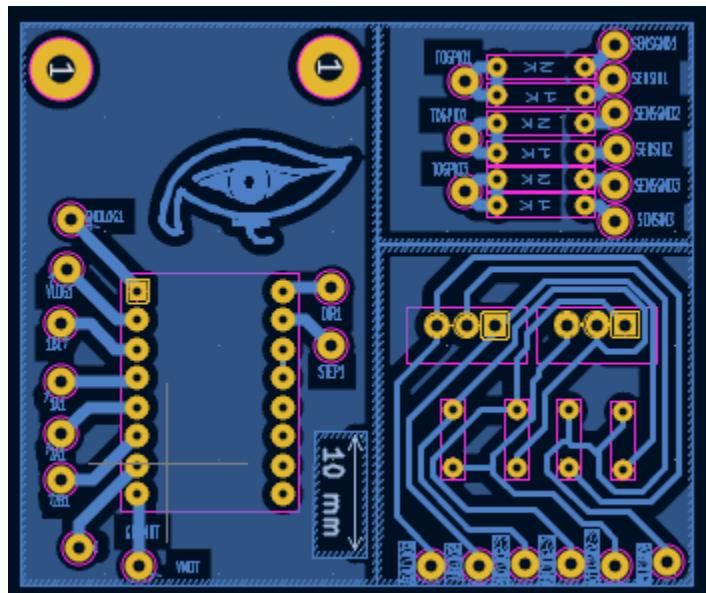
03/10

- Empezamos a hacer el PCB con Driver A4988 para Control del Brazo y divisores de tensión para Compatibilidad Sensores-GPIO
- Armamos el sistema de suspensión y tracción en el robot



04/10

- Terminamos de hacer el PCB para Control del Brazo y divisores de tensión



- Se siguió agregando información a la Carpeta Técnica y el Informe Descriptivo
 - Empezamos a agujerear y soldar componentes a PCB con Driver A4988 para control de brazo y divisores de tensión



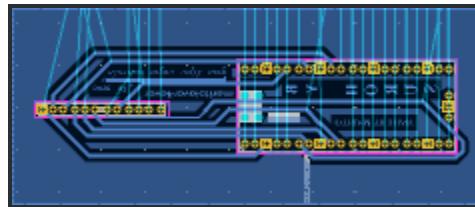
-----Semana 07/10

07/10

- Empezamos a cortar gomitas para grip de orugas y pegarlas a las orugas

08/10

- Montamos las orugas nuevamente, los tensores y el brazo robótico
- Un croquis preliminar de la carcasa para organización de los controladores steppers se hizo
- Hicimos el PCB para compatibilidad LORA ADAPTER to Raspberry Pi PICO



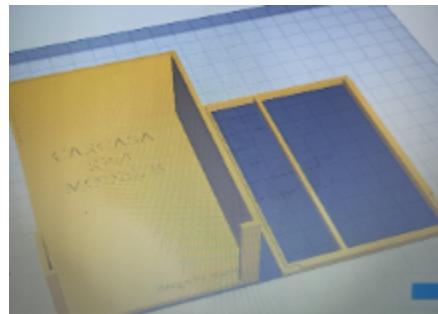
09/10

- Continuamos programando ruta de Home
- Empezamos a probar el código modulo camara con raspberry e IA

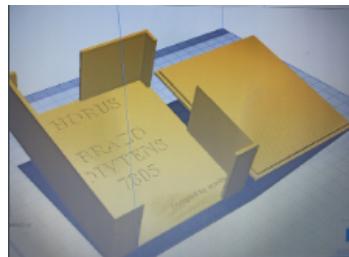
11/10

- Probamos nuevamente el código modulo camara con raspberry e IA
- Pintamos la base del robot

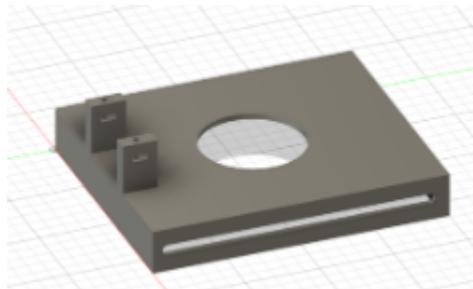
- Diseñamos las paredes frontal y posterior del robot
- Terminó de programar la ruta Home (App)
- Encontramos el fallo de cámara no reconocida
- Se hizo un diseño 3D de la carcasa para Raspberry PI 4



- Se hizo el diseño 3D de carcasa para pcb multipurpose (divisores de tensió)



- Hicimos el diseño de base para brazo elevador y soporte de servo



-----Semana 14/10
14/10

- Pasamos los programas de la IA a la raspberry pi 4
- Terminamos de cortar y pintar el techo de nuestro robot
- Terminamos de armar el brazo de la cámara
- Solucionamos fallas con los módulos de radiofrecuencia

- Terminamos el programa para el funcionamiento del brazo y el servo de la cámara

15/10

- Terminamos de armar y pintar las paredes del robot
- Armamos todas las placas, motores y estructura del robot para posteriormente probar los motores de movimiento
- Solucionamos fallas con la raspberry pi 4 y terminamos de instalar la IA
- Probamos la cámara del robot con la IA
- Solucionamos una falla de la placa receptora
- Solucionamos falla con el programa y los drivers de los motores
- Terminamos código de comunicación server - Raspberry pico

16/10

- Solucionamos problemas estructurales del robot
- Terminamos código de comunicación Raspberry pico - robot

//////////////////////////////ONIET/////////////////////////////

17/10

- Armamos el robot
- Probamos los códigos y la cámara
- Programamos recepcion y transmision de datos clave para iniciar el robot (ID y Pin de seguridad)

18/10

- Hicimos un código de secuencia para que el robot se vea en funcionamiento (Sube el brazo de la cámara, Se mueve la cámara, baja el brazo y se mueven los motores)
- La cámara se rompió e hicimos un código provvisorio para utilizar la cámara del celular mediante wifi
- Se colocaron funciones para el mapa (incluyendo borrar marcadores) y solucionamos varios problemas (con el zoom) de la aplicación

19/10

- Expusimos
- Desarmamos y embalamos el robot

||||||||||||||||||||||||||||||||ONIET||||||||||||||||||||||||||||

-----**Semana 28/10**

28/10

- Desoldamos y soldamos los módulos LoRa a las placas adaptadoras
- Reiniciamos el sistema operativo de la Rasp Pi 4 para probar la cámara
- Probamos nuevamente la cámara Raspberry y no anduvo
- Continuamos retocando la aplicación
- Empezamos la redacción del Manual de Usuario

29/10

- Continuamos con el Manual de usuario

30/10

- Seguimos con el Manual de Usuario

31/10

- Empezamos con la redacción de la Carpeta Técnica
- Hicimos la Portada, el índice y colocamos algo de información



- Continuamos con el Manual de Usuario
- Buscamos una forma de solucionar el error de transmisión raspberry pi 4 - Lora

01/11

- Terminamos el Manual de usuario

-----Semana 04/11

04/11

- Conseguimos un motor servo que pueda tener suficiente torque para mover la oruga del robot, a diferencia de los motores actuales. Se le removieron su tope mecánico y su tope electrónico para que pueda girar 360° de manera continua

08/11

- Varias features de la app que habían quedado terminadas de forma local pero no commiteadas fueron subidas al Github del proyecto, incluido pero no limitados a: modo offline del mapa, funcionalidad para agregar zonas de detección y puntos de partida aparte de las trampas, y ahora los tres marcadores son guardados en la base de datos para que se carguen incluso luego de cerrarse la app. Se agregó para que los marcadores que se agregan se puedan eliminar con doble click (junto con su entrada en la base de datos). Los

controles de marcadores en el mapa también incluyen a los puntos de partida y las áreas de detección.

El servidor ahora tiene funcionalidades para recibir apropiadamente de la conexión USB las detecciones para agregarlas en la base de datos (ahora la key de la trampa de feromonas lo agrega el propio servidor dependiendo de sus datos de las trampas)

RobotsView ahora tiene requests para pedir datos del servidor para crear perfiles y obtener datos de ellos (solo al iniciar por ahora)

10/11

- Agrego modo offline a la aplicación para el importe manual de logs por medio de pendrive
- Eliminación de alerts en el código de la app porque provocan que los inputs y otras funciones de la aplicación dejen de funcionar hasta abrir los dev tools
- Actualización de símbolos en el mapa de Home (app)

Semana 11/11

11/11

- Revisión de proyectos, videos de presentación y manual de usuario
- Se terminó de agregar un modo offline a la aplicación para el importe manual de logs por medio de pendrive
- Finalización de MVP documento
- Se agregó a la sección de Hardware y mantenimiento al manual de usuario

Semana 18/11

18/11

- Se adaptó el segundo servo para que funcione como motor de continua, removiéndole el tope mecánico y electrónico
- Actualización de manual de usuario con nuevas imágenes de la aplicación
- Actualización de carpeta técnica (frontend)

- Actualización de css y marcadores de la app
- Actualización del main.py del módulo comunicador con funciones de Lora
- Creación de ejecutables para la aplicación (se descartó el uso de electron-builder para packaging o deployment)

19/11

- Release de la la versión 1.0.0 de la app en github
- Actualización del Readme.md de la parte de app
- Corrección de errores en perfiles y zoom de mapa

20/11

- Actualización de carpeta técnica (backend)
- Se agregó a la app que el servidor sea el que designa el date y time de cada log, debido a que el robot no logra poder acceder a datos de hora y día sin conexión a internet.

21/11

- Actualización de carpeta técnica antes de imprimirla, e impresión de la misma
- Impresión de carpeta de campo
- Corrección de error de zoom de las imágenes de la galería de la app

//////////EXPOSICIÓN//////////

22/11

- Armamos todo el robot y probamos una última vez que todo funcione para la exposición
- Expusimos

23/11

- Expusimos

//////////EXPOSICIÓN//////////

-----Semana 25/11

25/11

- Modificamos el código de los motores para que ambos giren para el mismo lado y por el tiempo que querramos
- Implementamos el programa del gps en el programa main
- Colocamos el reemplazo de la pieza del brazo que se nos había roto
- Terminamos la carpeta técnica