



# PROTORP

Carpeta de Campo

Año de desarrollo: 2025

E.E.T. N°7 de Quilmes

**Sitio web:** <https://protorp2025.netlify.app/>

**Trello:** <https://trello.com/b/dudjmzhY/kanban>

**GitHub:** <https://github.com/impatrq/Protorp>

**Redes sociales:** @protorp25

**Escuela de Educación Técnica N°7 “Taller Regional Quilmes”**

**Especialidad: Electrónica**

**Año lectivo: 2025**

# Índice

1. Equipo	2
2. Objetivo del Proyecto PROTORP	2
Objetivo del Proyecto PROTORP	2
Documentación Día a Día	3

## 1. Equipo

### Integrantes del proyecto PROTORP

Joaquín Gonzalez Nogueira

Santiago Nicolás Vila

## 2. Objetivo del Proyecto PROTORP

El proyecto **PROTORP** (Protección y Regulación de Trenes mediante Organización y Respuesta Programada) tiene como objetivo principal el desarrollo de un **sistema automático de protección ferroviaria (ATP)** y un **sistema de frenado automático (ATS)** a escala, combinando hardware y software en un entorno educativo.

El propósito es crear una maqueta funcional que simule un circuito ferroviario real, con señales, cambios de vía, sensores de ocupación y control mediante microcontroladores como **ESP32** y **Raspberry Pi**.

Además, el sistema busca representar cómo la tecnología puede mejorar la seguridad, eficiencia y automatización en el transporte ferroviario, estableciendo un puente entre la educación técnica y las soluciones tecnológicas reales aplicadas al ámbito ferroviario argentino.

# PROYECTO PROTORP: DOCUMENTACIÓN DÍA A DÍA

## ABRIL

14/4: Creamos la cuenta de instagram y Santiago empezo a comunicarse con sponsors.  
15/4: Joaquin empezo con la programación de la página web y Santiago con la búsqueda de sponsors.  
16/4: Joaquin continuo con la programación de la pagina web, y Santiago con búsqueda de sponsors.  
17-20/4: Feriado  
21/4: Joaquin continuo con el desarrollo de la pagina web  
23/4: Joaquin continuo con la pagina web  
30/4: Joaquin continuo con la pagina web

## MAYO

7/5: Joaquin continuo con la pagina web  
9/5: Joaquin continuo con la programación de la pagina web  
Semana del 12/5 a 18/5: Ambos tuvimos una reunion con SIEMENS, leimos la documentación y Joaquin continuo la programación de la pagina web  
Semana 19/5 a 25/5: Tuvimos una segunda reunion con SIEMENS para hablar sobre nuestras dudas y Joaquin termino la programación de la pagina web, consiguio un link en Netlify.app y Santiago la subio al Instagram  
Semana 26/5 a 1/6: Santiago realizo un diagrama del funcionamiento de las vías y Joaquin añadio un poco de información a la pagina ya que recibimos comentarios de que faltaba información

## JUNIO

Semana 2/6 a 8/6: Esta semana nos dedicamos a preguntarle a distintos profes sobre que componentes usar y a empezar con unas reformas en la pagina web  
Semana 9/6 a 15/6: Santiago realizo la lista de materiales y la mando a la empresa, mientras Joaquin siguio con las reformas en la pagina web  
Semana 16/6 a 22/6: Joaquin instalo Blender y aprendio a usarlo para poder realizar el modelo 3D de la señalizacion de los cambios de via  
Semana 23/6 a 29/6: Joaquin realizo el modelo 3D de la primera señal y la subio al Github como STL

## JULIO

16/7: Tuvimos una reunion sobre los avances del proyecto. Nos sugirieron trabajar durante el receso invernal para ganar tiempo y pedirle los componentes a cooperadora.

18/7: Fuimos a pedir 2 Raspberrys Pi Pico 2W y 2 modulos RF de 433mhz a Cooperadora. Tambien nos prestaron resistencias y LEDs del pañol.

Receso Invernal: Joaquin estuvo haciendo simulaciones en Wokwi, que luego serian probadas en agosto.

## AGOSTO

4/8: Joaquin probo en Wokwi el modulo de comunicacion de tren - via y semaforos con 2 ESP32 y 3 leds.

5/8: Joaquin probo en Wokwi el modulo de sensor de ocupacion de via con un ESP32 y un led

6/8: Joaquin probo en Wokwi el funcionamiento de la señalización ferroviaria con un ESP32, un servo, un boton y 3 leds

8/8: Joaquin actualizo el sensor de ocupacion. Desde cooperadora nos mandaron llas 2 Raspberrys que pedimos. Santiago subio una historia agradeciendole a cooperadora por las Raspberry y Joaquin instalo la configuracion de micropython en vscode para configurar la Raspberry.

11/8: Empezamos a hacer el diagrama eléctrico mientras inicializabamos la Raspberry en VScode.

13/8: A la mañana empezamos el curso de drones. A la tarde visitamos Retiro donde nos dieron una capacitacion y nos permitieron recorrer los andenes que estan en obra.

18/8: Joaquin empezo a preparar el codigo para que funcione el wifi de la Raspberry que ira en la via.

20/8: Este dia, a pesar de tener practicas profesionalizantes, no pudimos hacer nada porque se fue la luz y no pudimos cargar nuestras compus. Este dia tambien llegaron los sensores RF de 433 mhz de parte de cooperadora, entonces fuimos a retirarlos.

21/8: Joaquin termino de hacer el codigo de la raspberry que iria en la locomotora y probamos la comunicacion wifi entre ambas raspberry

27/8: Tuvimos el curso de dron

28/8: Tratamos de probar la raspberrys en fisico pero no tenian las patitas. Joaquin actualizo el codigo entre la Raspberry y la locomotora de la via

## SEPTIEMBRE

1/9: Santiago empezó a hacer un segundo modelo 3D de las señales ferroviarias ya que el primero tenía errores en el modelo

2/9: Santiago continuo con el segundo modelo 3D de las señales

3/9: Tuvimos curso de dron

4/9: Este dia ambos faltamos pero mandamos a imprimir una nota para llevar al Ferroclub Escalada y se la hicimos firmar al vicedirector Arguello

6/9: Este dia, a pesar de ser fin de semana y no tener practicas profesionaliantes, fuimos al Ferroclub Escalada con la nota. Este dia no habiamos llevado los componentes y nos dijeron que debian consultarle a la fundacion del Ferroclub por los materiales, asi que nos pidieron que volvieramos la siguiente semana con los componentes para darles una mejor explicacion de como iban a ir los componentes en el circuito

9/9: Santiago termino el segundo modelo 3D de las señales ferroviarias

10/9: Fuimos al curso de dron

13/9: Este dia volvimos al Ferroclub Escalada junto con los componentes. Les explicamos mejor el funcionamiento del circuito y como iban a ir montados los componentes. Nos dijeron que volvieramos la otra semana con otra nota para que nos den el material, pero nos confirmaron que nos iban a prestar una locomotora y un par de vías

16/9: Hicimos una lista de materiales y fuimos a pedirlos a cooperadora.

17/9: A la mañana estuvimos en el curso de drones. En la tarde Joaquin empezo a realizar el banner y Santiago el primer diseño de la placa en KiCad.

18/9: Joaquin termino de hacer el banner y Santiago continuo con el diseño de la placa en kicad

19/9: Santiago continuo haciendo el diseño de la placa en KiCad y le añadio una base al modelo en 3D

20/9: Santiago volvio al Ferroclub Escalada y le proporcionaron el tren y un par de vias

22/9: Probamos la corriente en las vias que nos proporciono el Ferroclub Escalada

23/9: Santiago soldao las patitas al ESP32 y continuo con el modelo en Kicad

24/9: Santiago finalizo el diseño en Kicad. Tambien, Joaquin empezo a hacer el codigo en una de las compus de la escuela ya que su compu andaba mal. Joaquin estuvo toda la mañana en el aula del profesor Montoni haciendo el codigo. A la tarde, el profesor Garabato nos proporciono mas vias, cambios de via y locomotoras para probar el circuito en fisico. Dedicamos un rato a ver que nos servia y que le podiamos devolver y luego Joaquin eempezo a instalar las librerias necesarias en Thonny y VScode para probar el codigo

25/9: Joaquin trato de probar el codigo del ESP32 en ARDUINO IDE pero no funciono. Tambien empezamos a limpiar las vias que nos dio el profesor Garabato y armamos el circuito para probar la corriente y verificar que el tren andara correctamente sobre las vias.

26/9: Continuamos limpiando las vias con virulana y alcohol. Este dia tambien nos mu-

damos a la jaula del fondo ya que se nos hacia incomodo armar y desarmar el circuito cada vez que queriamos probarlo, asi que decidimos llevar 2 mesas a la jaula y dejar las vias armadas ahi. Tambien Joaquin trato de probar la comunicacion wifi entre esp32 y raspberry en una de las computas de la jaula pero estas no tenian internet, asi que decidimos tratar de probarla en la compu de Santiago pero ahi tampoco funciono.

30/9: Este dia, Joaquin trato de probar por ultima vez el codigo en su compu con Platfor-mio. El codigo se subio correctamente al esp32 pero al momento de abrir el serial monitor, este tardaba mucho en cargar o no mostraba nada. Finalmente, decidio probar el codigo del ESP32 y el de la Raspberry en Thonny

## OCTUBRE

1/10: A la mañana, pasamos el codigo de la Raspberry a Thonny y como aun no tenia-mos programada la esp32 para que mandara el estado de las vias, probamos mandandole mensajes desde una app llamada MyMQTT. A la tarde, Joaquin modifico el codigo del ESP32 para que mandara el estado de las vias.

2/10: Joaquin modifico el codigo de ambos componentes para incluir los leds y los sensores RF de 433mhz. Tratamos de probar el circuito en fisico pero nuestro cable no transmitia datos, entonces decidimos que lo probariamos el 7/10. Este dia, Santiago hizo la impre-sion del primer semaforo en 3D. El modelo 3D tenia una base aparte, asi que luego de imprimirlo en 3D, Santiago lijo y quito todo el sobrante, luego separo la base y la pego con la gotita. Por ultimo, luego de la escuela, fuimos a lo del señor Daniel Raley para conseguir mas vias rectas y otros 2 cambios de via. Nos proporciono un par de vias pero no nos pudo proporcionar los cambios de via que nos faltaban.

6/10: Este dia armamos el circuito para probarlo en fisico, pero nos dimos cuenta de algo importante. Los codigos originales se hicieron para que la Raspberry y la ESP32 se comu-nicaran por wifi, pero la gracia es que los datos se manden por los sensores de 433mhz. Luego de darnos cuenta de eso, Joaquin empezo a modificar todo el codigo de nuevo. Este dia, Santiago imprimio el segundo semaforo

7/10: Este dia, Santiago imprimio el 3er y 4to semaforo. Joaquin busco las librerias para poder programar los sensores de 433mhz junto con la Raspberry en Thonny. Las librerias que encontro para los sensores no eran compatibles con micropython, entonces le pidio a una IA que le cree una libreria para usar. Luego, empezo a programar nuevamente la ESP32 y la Raspberry con esta libreria. El codigo fue hecho para probar una sola señal

8/10: Este dia, Joaquin termino de programar la Raspberry y la ESP32 y tratamos de probar el circuito en fisico. Al principio no funciono ya que uno de los transmisores estaba dañado, luego lo intercambiamos con otro transmisor y funciono. El transmisor mostraba un estado entre “LIBRE”, “PRECAUCION” y “OCUPADO” en el shell de Thonny con la Raspberry y cambiaba el led dependiendo del estado. La ESP32 captaba este estado con su receptor y lo mostraba en el shell de Thonny que este fue recibido junto con la

velocidad simulada.

9/10: Este día, Joaquín cambió el código para que se probaran dos señales en lugar de solo una. Luego empezamos el circuito con un ESP32, una raspberry, 3 TX y 3 Rx en 3 protoboards para probar su funcionamiento pero se nos hizo tarde y no pudimos probarlo, así que lo probamos el día 13/10

10/10: Este día Santiago fue a la casa de un vecino a conseguir los cambios de vía y vías rectas faltantes.

13/10: Este día volvimos a armar el circuito en físico para probarlo. No anduvo correctamente debido a que había algunos errores en el código, mayormente relacionados a la lógica de funcionamiento entre los componentes, así que Joaquín empezó a modificar el código nuevamente para respetar esa lógica.

14/10: Joaquín continuó modificando el código de ambos componentes y Santiago modificó el modelo en Blender para que esta vez el modelo tuviera 2 semáforos y se pudieran imprimir más rápido. Este día también fuimos a cooperadora a solicitar un doble puente H driver L298N y unos pines hembra. También teníamos pensado empezar a imprimir la placa pero este día no estaba el profesor Fabricio Carlassara, que es el dueño de la fresadora y del software, así que no pudimos imprimirlas.

15/10: Joaquín terminó de armar ambos códigos. A la tarde, quisimos armar el circuito nuevamente, pero la profesora Lara necesitaba los protoboards, así que probamos el circuito al día siguiente.

16/10: Este día armamos el circuito por última vez. El código parecía funcionar pero se nos quemó la raspberry porque accidentalmente pusimos 10 volts en la fuente. Decidimos dejar de armar el circuito en protoboard cada vez que íbamos a probarlo para evitar accidentes como estos y empezar a armar la placa. También, Joaquín modificó el código para poder manejar la esp32 por bluetooth.

17/10: Joaquín continuó modificando el código para controlar la esp32 (locomotora) por bluetooth, aparte de adquirir la librería `ble_simple_peripheral.py` que es necesaria para que la esp32 se anuncie como un dispositivo bluetooth. Este día también llegó el driver L298N así que fuimos a buscarlo a cooperadora.

20/10: A la mañana no pudimos hacer mucho debido a que no había wifi. A la tarde, Joaquín continuó con el código, añadiéndole la lógica de control del motor e incluyéndole el driver L298N

21/10: Este día, el profesor Fabricio Carlassara nos aconsejó cambiar el diseño en KiCad para hacer módulos de 1 raspberry en vez de tenerlas todas en una sola placa. Santiago modificó el diseño en KiCad y también cortó la placa de cobre en nueve partes. Mientras tanto, Joaquín se instaló la app nRF connect para mandarle una velocidad aleatoria a la esp32 desde la app y verificar que apareciera en el Thonny. Esta app también tiene una opción de lectura que muestra las velocidades aleatorias que genera la esp32 en la app. También instaló la librería `uwebsockets` en la raspberry para coordinarla con la página web. Santiago rediseñó el banner ya que nos pareció que el que hicimos antes quedó ob-



soleto, además que el anterior estaba en 90x120 px, lo que hacia que fuera muy pequeño, así que en la segunda version tratamos de que saliera con la mayor calidad posible.

22/10: Santiago imprimio los semaforos 11 y 12 en 3D. Tambien imprimio la primera placa y la dejamos lista para soldar. Este dia tambien enviamos el banner al profesor Palmieri y luego de la escuela fuimos a retirar los pines hembra que compramos.

23/10: Santiago abrio en Blender el modelo 3D de los semaforos normales y tomo sus bases. Luego, las exporto al modelo 3D de los semaforos de cambio de via y se las añadio a ese modelo para poder imprimirlas con el fin de que tengan el mismo soporte que los semaforos normales. Luego fuimos a imprimirlos. Mientras tanto, Joaquin se enfoco en ver ejemplos de años pasados para realizar el manual de usuario, carpeta tecnica y video de presentación de la mejor manera posible. Tambien nos comunicamos con nuestro patrocinador SIEMENS para ver si nos podian proporcionar manuales de usuario para usar de inspiracion.

24/10: Joaquin realizo la carpeta del manual de usuario y empezo a programar el nuevo codigo de la raspberry

27/10: Joaquin continuo realizando el codigo de la raspberry y Santiago modifico el modelo del semaforo del cambio de via. Luego, Santiago imprimio el 13vo y 14vo semaforo en 3D

28/10: Este dia, Santiago imprimio los semaforos 15 y 16. Joaquin continuo el codigo de la raspberry y empezo a armar el texto del manual de usuario

29/10: Este dia fuimos a hacer pasantias a la empresa NEWTON. A la tarde, Joaquin continuo con el codigo de la Raspberry y con el texto del manual de usuario. Santiago armo el tren con el puente H, soldo algunas Raspberrys y lijo los sobrantes de los semaforos que imprimimos

30/10: Este dia, Joaquin termino el manual de usuario del proyecto implementado en un entorno real, pero nos recomendaron cambiarlo para que sea la explicación de la maqueta. Joaquin empezo con el nuevo texto y siguio programando las raspberrys. Mientras tanto, Santiago empezo a grabar tomas para el video presentación

31/10: Joaquin termino el manual de usuario de la maqueta y ambos grabamos los audios para el video presentación. Santiago hizo todas las placas y se encargo de editar el video.

## NOVIEMBRE

3/11: Santiago agujereo todas las placas e hizo el modelo 3D de la cajita del motor.

4/11: Santiago armo todos los semaforos, con sus leds y sus cables. Joaquin empezo a hacer la carpeta tecnica.

5/11: A la mañana fuimos a NEWTON. A la tarde, Joaquin continuo con la carpeta tecnica y probamos el funcionamiento de la ESP32 y el driver por Bluetooth, observamos que al mandarle un valor se movia un poco y despues se detenia, decidimos cambiar los cables

al día siguiente. Mientras, nos pidieron rehacer el video presentación, así que Santiago grabo nuevas tomas y se encargo de editar el video.

6/11: Probamos nuevamente el funcionamiento de la ESP32 por Bluetooth. Seguía pasando lo mismo que el día anterior. Entonces, Joaquín trató de hacer una app por RemoteXY de Arduino, pese a que no le haya funcionado para programar antes. Instaló todas las librerías necesarias y se puso a hacer la app. Mientras, Santiago agujereó más placas y probó el funcionamiento de uno de los semáforos junto con la Raspberry, que ya tenía el código cargado

7/11: A la mañana, Joaquín trató de seguir programando la app en arduino. Mientras, vino el profesor Medina y trató de ayudarnos a ver cuál podía ser el problema. Santiago trató de armar nuevamente el circuito junto con el profe, mientras Joaquín le cargaba el código antiguo de Thonny al ESP32. Cuando lo probamos nuevamente, Santiago se dio cuenta que había puesto la corriente de las vías al revés, entonces al invertirlo y mandarle valores por la app nRF connect, anduvo normalmente. Hicimos un par de pruebas más pero la ESP32 se calentó excesivamente. Le preguntamos al profesor Medina cuál podría haber sido el problema, y junto con él verificamos que la ESP32 se había quemado. Decidimos probar el código con la otra ESP32 que teníamos pero esta también estaba quemada, entonces Santiago pidió 2 ESP32 que llegarían el día siguiente. Mientras, Santiago siguió armando las placas y los semáforos y Joaquín continuó con la carpeta técnica y se puso a actualizar la carpeta de campo.