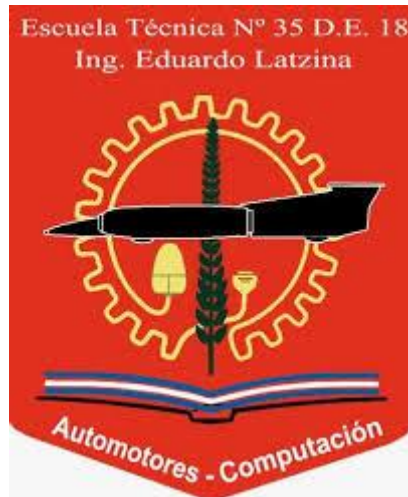


## **Escuela Técnica N°35 - “ing - Ed. Latzina”**



**Año: 6to 3ra**

**Ciclo lectivo: 2025**

**Profesor: Lescano Daniel Carlos**

## **Documentación – Proyecto Carrito de Seguridad**

### **Grupo Raspberry Pi**

- Gerónimo Calvo

- Rodrigo Tapia
- Emmanuel Pilco
- Franco De Acha

## **Grupo Arduino Py**

- Nicolás Gómez
- Emanuel Oliveira
- Adrián Prado

**26/05/2025**

### **Raspberry Pi**

El primer día empezamos con la instalación de la Raspbian OS en microSD con una computadora con puerto microSD. La programamos con el sistema operativo Raspbian 64 bits. Para programar ajustamos el sistema operativo con una red del teléfono de un alumno; después de instalar el sistema operativo en la microSD, lo conectamos al Raspi.

Mientras tanto, con una computadora del colegio conectada al ethernet utilizamos un programa llamado Acceso Remoto al Escritorio que nos proporcionó el profesor para utilizarlo en la Raspberry. Intentando probar la IP que sabíamos que tenía la Raspberry, probamos en la computadora, pero falló.

Para verificar que todo andaba bien empezamos a conectar el Raspberry al teclado, al ratón, el conversor \*\*\*\* a HDMI. Visualizamos mediante la pantalla el escritorio del Raspberry, y al visualizarlo notamos que se trataba de un sistema operativo parecido a Linux. Entonces buscamos un código que nos muestre la IP de la Raspberry, y volvimos a utilizar el programa con la IP que nos marcó y volvió a fallar.

Lo que hicimos hoy no estaba conectado a ethernet sino a la red del alumno. Intentamos varias veces y no logramos nada.

### **Arduino Py**

El primer día empezamos a buscar las librerías que podríamos usar para conectar Python con Arduino. Empezamos planteando el uso de funciones y procedimientos almacenados en Arduino para controlar su hardware.

**02/06/2025**

### **Raspberry Pi**

El segundo día estuvimos concentrados en realizar el documento con la propuesta inicial del proyecto con el diagrama de hitos.

### **Arduino Py**

En el segundo día el profesor nos recomendó el uso de una librería llamada Firmata, que facilita el uso mediático entre Python y Arduino. Se nos facilitó mucho el uso y control de las corrientes de Arduino Mega, pero teníamos errores con el PWM.

**04/06/2025**

### **Raspberry Pi**

El tercer día hicimos las conexiones de HDMI, el teclado, la pantalla y el ratón al Raspberry, y le conectamos el ethernet directamente. Volvimos a probar una computadora conectada al ethernet. En el programa nos marcaba un error, el cual buscamos en internet, en foros. Aplicamos ajustes a la Raspberry mediante comandos y no se solucionaba el problema.

Volvimos a la Raspberry y decidimos usar otro programa. Lo utilizamos en la computadora conectada al ethernet y tampoco logramos nada. Además, en otra Raspberry probamos una versión anterior de 32 bits y aplicamos los procedimientos que habíamos hecho hasta ese momento, pero también falló.

### **Arduino Py**

Al tercer día logramos resolver los problemas de PWM. Actualizamos el uso de Firmata a Firmata 2, que contiene más comandos y controles de un nivel más alto y cómodo.

**09/06/2025**

### **Raspberry Pi**

En el cuarto día usamos la Raspberry de 32 bits y volvimos a hacer todos los procedimientos de instalar la aplicación. Después probamos otro método para controlarlo remotamente con VNC entre la computadora y la Raspberry. Nos pidió iniciar sesión en la computadora. Creamos una cuenta para la aplicación, iniciamos sesión y pudimos entrar.

Sacamos la IP, completamos un formulario de usuario y contraseña, y logramos conectarnos. Luego instalamos la base de datos MariaDB. Al final, el profesor nos dio la tarea de crear una base de datos.

### **Arduino Py**

En el cuarto día seguimos con los errores con Firmata. Intentamos controlar distintas luces pero daba error de comando. El profesor nos recomendó volver a Firmata 1 varias veces. Luego de intentos fallidos en controlar más de un LED, volvimos a Firmata 1, pero seguíamos con fallas.

**11/06/2025**

### **Raspberry Pi**

En el quinto día continuamos con la Raspberry a la que ya habíamos instalado MariaDB. Al principio nos resultó difícil conectarnos desde otra computadora, ya que VNC no mostraba nada. Reiniciamos la Raspberry para ver si respondía. También estuvimos cambiando opciones sin lograr cambios.

Después de varios intentos de conectarnos remotamente, lo hicimos manualmente y funcionó. Entramos a la base de datos para revisar todas las tablas y decidir cuál eliminar. Finalmente decidimos que no eliminaríamos ninguna por el momento y usaríamos las tablas de usuarios y movimientos.

### **Arduino Py**

En el quinto día continuamos con el control de LED. Firmata 1 no fue solución y Firmata 2 daba error. Consideramos volver a usar funciones almacenadas en la memoria del Arduino Mega, un método más complicado. Pudimos resolver el problema usando Firmata 2 y cambiando la declaración de los pines al almacenarlos en variables. PWM y salidas digitales funcionaban. Faltaba la lectura analógica. Intentamos con read() y no funcionó; volvimos a errores de versión.

**23/06/2025**

## **Raspberry Pi**

El profesor nos mostró dos computadoras que la regencia nos había dado para desarmar y usar las cámaras para los carritos. Un grupo estuvo desarmando una computadora para extraer la cámara y otros componentes necesarios. Fue difícil porque había que tener cuidado de no dañar otras partes útiles. Los cables eran finitos y había que tratarlos con cuidado.

Mientras tanto, descargamos la versión de Python que estaba programada en el Arduino hacia la Raspberry. Después de mucho trabajo, lograron desarmar la primera computadora y extraer la cámara, junto con otros componentes como disco rígido, parlantes, chip de wifi y micrófono. También desarmaron la otra computadora para extraer otra cámara.

Soldamos los cables de una de las cámaras a un cable USB. Al conectarla a la PC, la cámara funcionó.

## **Arduino Py**

Seguimos con la lectura de entradas en Arduino para que el programa de Python pueda procesarlas. Con el profesor revisamos la documentación oficial de Firmata y encontramos un programa precargado que ayudó a leer un botón en una entrada digital, pero los valores estaban invertidos: el GND daba 1 y los 5.5V daban 0.

**25/06/2025**

## **Raspberry Pi**

Hablamos con el profesor sobre cómo conectar una Raspberry y una cámara con USB soldada.

## **Arduino Py**

Resolvimos los errores de lectura digital invertida y pudimos avanzar más con el read(). Teníamos un último error: los pines digitales extra del Arduino Mega aparecían como inexistentes en el programa Python. Luego de revisar variables, descubrimos que la tabla se estaba declarando como un "Arduino", no "Arduino Mega".

**30/06/2025**

## **Raspberry Pi**

Intentamos conectar la cámara a la Raspberry, pero apareció un error. Verificamos que no era error, sino que faltaba una aplicación para visualizarla. Instalamos imágenes y un código para sacar fotos que guardaba la imagen en una carpeta.

Como la Raspberry podía sacar fotos, descargamos una aplicación para ver en tiempo real lo que mostraba la cámara: VLC Media Player. Tras configurarlo, la cámara funcionó y mostraba video en vivo. Por acceso remoto, la imagen tenía lag.

## **Arduino Py**

Logramos usar un callback para una entrada analógica con valores flotantes.

**02/07/2025**

## **Raspberry Pi**

Revisamos la cámara conectada a la Raspberry. Como funcionaba, comenzamos el siguiente paso: crear un script Python que grabara video mientras la cámara esté encendida y genere un archivo de video al apagarse.

**07/07/2025**

## **Raspberry Pi**

Como el Arduino aún no estaba listo para unirse al Raspberry, el profesor nos dio nuevas indicaciones:

1. Crear un sistema para que el nombre del archivo de video sea incremental e incluya día y hora.
2. Crear una función que borre archivos si pasa una semana.
3. Chequear si se pueden bajar los frameworks para reducir el lag.
4. Aprender a usar hilos de ejecución en Python.

**11/08/2025**

## **Raspberry Pi**

Revisamos la base de datos de la Raspberry para asegurarnos de que todas las tablas estuvieran preparadas para almacenar información.

**18/08/2025**

## **Raspberry Pi**

Intentamos crear una interfaz de usuario para la Raspberry. Conectamos la Raspberry a una computadora y accedimos desde otra. Revisamos la base de datos. Intentamos conectar la Raspberry con Python pero sin éxito. Creamos la interfaz en Visual Studio Code y evaluamos su diseño.

**20/08/2025**

## **Raspberry Pi**

Datos: Usuario de la base de datos: root@localhost.

Mejoramos la interfaz de usuario. Luego pensamos mover la base de datos hacia una carpeta de Drive mediante un archivo. Buscamos código para hacerlo. Intentamos enviar directamente la base de datos al Drive, pero finalmente decidimos crear un archivo SQL aparte. Creímos haberlo logrado, pero nos equivocamos.

**03/09/2025**

Subimos la base de datos a XAMPP para conectarla. Revisamos y vimos muchas tablas repetidas o innecesarias. Decidimos que la base quedaría con las tablas: Usuario, Movimientos y Casillero.

**08/09/2025**

Calvo creó una landing page para registrar usuarios que quieran retirar computadoras del carrito. Se ingresaba con correo y contraseña. Permitía retirar computadoras disponibles (más de una) o devolverlas.

También discutimos el diseño de la página y el rol del administrador, que tendría una lista completa de usuarios y opciones para crear, editar o eliminar.

**24/09/2025**

El profesor nos dió nuevas consignas:

- Hacer que la interfaz de usuario sea en una sola pantalla.
- Unir la Raspberry con las funciones del Arduino.
- Hacer que todo funcione con el código de barras.

**29/09/2025**

Presentamos los avances del proyecto. El profesor pidió:

- Simplificar la interfaz en una sola pantalla.
- Unir la Raspberry con el Arduino.
- Unir la base de datos con la interfaz y hacer que funcione.
- Conectar Arduino a la interfaz.

**01/10/2025**

Trabajamos todos juntos en seguir ajustando la interfaz de usuario para que funcione en una sola pantalla.

**15/10/2025 – Raspberry Pi**

Continuamos trabajando en la interfaz de usuario. La probamos directamente en la Raspberry y ajustamos algunos elementos que no se veían bien, como botones y textos. También probamos el lector de código de barras para confirmar que la Raspberry lo reconozca como entrada de teclado. Logramos que lea varios códigos, aunque por ahora solo los muestra y todavía no ejecuta ninguna acción dentro de la interfaz.



## **22/10/2025 – Raspberry Pi / Arduino Py**

Ese día nos enfocamos en conectar la Raspberry con el Arduino usando Firmata. Instalamos las librerías necesarias y probamos un script sencillo para verificar la comunicación. El Arduino respondió, pero la conexión era inestable.

## **29/10/2025 – Raspberry Pi**

Trabajamos en que la interfaz funcione correctamente dentro de la Raspberry. Ajustamos detalles visuales y revisamos el código para asegurarnos de que la interfaz cargue sin errores. Todavía no tiene integrada la lógica completa, pero ya se abre bien y queda lista para conectarla con el resto del sistema. La próxima etapa es hacer que la interfaz responda cuando se escanee un código de barras.

## **05/11/2025 – Raspberry Pi / Arduino Py**

Ese día revisamos cómo estábamos integrando el código entre la Raspberry y el Arduino. Organizamos mejor los archivos de Python para que sea más fácil unir la interfaz con las funciones del Arduino.