Escuela de Doctorado
y Estudios de Posgrado
Universidad de La Laguna

Máster Universitario en
Ingeniería Industrial

Trabajo Fin de Máster

MuROB, automatización del baño de órganos por métodos ópticos

Autor/a: Eduardo Miquel Hernández

Tutor/a: Cándido Caballero Gil

Cotutor/a: Ricardo Borges Jurado

1 de julio de 2022

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

Resumen

En este trabajo se presentan, en primer lugar, la introducción al MuWOB y la interfaz de usuario que se ha encargado de desarrollar el estudiante, la cual permite enviar los mensajes necesarios al autómata (Arduino Mega). Para ello, se ha trasladado la interfaz básica del LabVIEW al entorno de desarrollo de software App Inventor, obteniendo un modelo de aplicación para dispositivos Android (.apk), facilitando el uso del MuWOB por parte del usuario, a través de su Smartphone o Tablet.

A su vez, este entorno ha permitido enviar mensajes mediante la conexión Bluetooth, pudiendo cambiar la comunicación serial, la cual necesitaba de conexión directa con el Arduino. Para ello, se ha adquirido un módulo de Bluetooth compatible con el autómata, siendo configurado y conexionado con el objetivo previamente descrito.

En relación con la interfaz, se añadieron los nuevos requerimientos solicitados por el Dr. Ricardo Borges, el interesado del proyecto, configurando la programación y definición de las funcionalidades de la botonera y diferentes aspectos de diseño de la interfaz, consiguiendo una transición entre pantallas fluida y atractiva, dotando de sencillez, robustez y uso intuitivo a la app. Además, se han implementado mejoras al código Arduino, modificaciones y adaptaciones mecánicas y de redes, funcionalidades básicas y avanzadas u otras.

Por otro lado, se añadieron una serie de restricciones de acceso y modificación de parámetros dentro de la aplicación; de esta forma, la persona que esté haciendo uso de ella, debe tener conocimiento de su usuario, y hará uso de las pantallas y procesos a los que se le haya facilitado permiso.

Y, por último, las conclusiones del proyecto, en las que se describe cómo se han logrado los objetivos de este TFM y los resultados obtenidos, así como las posibilidades de mejoras si se continuara trabajando en este proyecto, y las limitaciones encontradas durante su desarrollo.

Abstract

First of all, this paper presents the introduction to MuWOB and the user interface that the student has been in charge of developing, which allows sending the necessary messages to the automaton (Arduino Mega). For it, the basic interface of LabVIEW was switched to the App Inventor software development environment, obtaining an application model for Android devices (.apk), facilitating the use of MuWOB by the user, through their Smartphone or Tablet.

In turn, this environment has allowed sending messages through the Bluetooth connection, being able to change the serial communication, which needed a direct connection with the Arduino. For this, a Bluetooth module compatible with the automaton has been acquired, being configured and connected with the objective previously described.

In relation to the interface, the new requirements requested by Dr. Ricardo Borges, the person interested in the project, were added, configuring the programming and definition of the functionalities of the button panel and different aspects of the interface design, achieving a smooth transition between screens and attractive, providing simplicity, robustness and intuitive use to the app. In addition, improvements to the Arduino code, mechanical and network modifications, adaptations, basic and advanced functionalities, and others have been implemented.

On the other hand, a series of access restrictions and modification of parameters within the application were added; in this way, the person who is making use of it must be aware of its user, and will make use of the screens and processes to which permission has been provided.

Finally, the conclusions of the project, in which it is described how the objectives of this TFM have been achieved and the results obtained, as well as the possibilities of improvements if work continues on this project, and the limitations encountered during its development.

1. Introducción

A lo largo de los últimos años, se ha ido desarrollando la técnica de la monitorización de la respuesta contráctil de tejidos utilizando sistemas basados en el análisis de imagen (MuWOB, por Multi-Well-Organ-Bath), como alternativa a los sistemas basados en transductores. Este sistema ya ha sido patentado y publicado.

Multi-Well Organ Bath (MuWOB): consiste en un innovador instrumento para medir la respuesta contráctil que provocan las sustancias químicas en porciones pequeñas de tejido vivo. Su técnica de medida es totalmente diferente a la empleada por los equipos que se utilizan hoy en día, y aporta ventajas que no ofrecen los instrumentos convencionales. El MuWOB, se distingue por su capacidad para analizar múltiples sustancias con hasta noventa y seis porciones de tejido en paralelo. Con ello, se consigue, entre otras cosas, un ahorro importante de sustancias, y un mejor aprovechamiento de los animales en cada experimento. Además, el MuWOB ocupa muy poco espacio de poyata y, económicamente, está al alcance de pequeños y medianos laboratorios.

Ahora, el objetivo es poner en marcha su automatización, básicamente en lo que se refiere al manejo de líquidos. Se va a concluir la implementación de un brazo robótico (MuROB, por Multiwell Robot). El sistema irá alojado en una estufa para mantener constantes la temperatura y la humificación del ambiente. La misión encargada será el diseño de los programas para el manejo del MuROB y de comunicación con el mismo.

Con este fin, el estudiante se ha encargado de desarrollar una interfaz de usuario que permita enviar los mensajes necesarios al autómata (en este caso, Arduino Mega). Para ello, se ha trasladado la interfaz básica del LabVIEW al entorno de desarrollo de software App Inventor, obteniendo un modelo de aplicación para dispositivos Android (.apk), facilitando el uso del MuWOB por parte del usuario, a través de su Smartphone o Tablet.

A su vez, este entorno permite enviar mensajes mediante la conexión Bluetooth, pudiendo modificar la comunicación serial, la cual necesitaba de conexión directa con el Arduino. Para ello, se ha adquirido un módulo de Bluetooth HC-05 compatible con el autómata, siendo configurado y conexionado con el objetivo previamente descrito.

En relación a la interfaz, se añadieron los nuevos requerimientos a implementar, configurando la programación y definición de las funcionalidades de la botonera y diferentes aspectos de

diseño de la interfaz, consiguiendo una transición entre pantallas fluida y atractiva, dotando de sencillez, robustez y uso intuitivo a la app.

Por otro lado, se añadieron una serie de restricciones de acceso y modificación de parámetros dentro de la aplicación; de esta forma, la persona que esté haciendo uso de la misma, debe tener conocimiento de su usuario, y hará uso de las pantallas y procesos a los que se le haya facilitado permiso.

En resumen, se han implementado mejoras al código Arduino (mejoras implementadas para eliminar la burbuja de aire, cambio de los parámetros del motor paso a paso encargado del desplazamiento vertical de las jeringas, etc), cambios mecánicos para ajustar la inserción de las agujas, funcionalidades básicas y avanzadas u otras.

1.1. Introduction

Over the last few years, the technique of monitoring the contractile response of tissues has been developed using systems based on the analysis (MuWOB, for Multi-Well-Organ-Bath), as an alternative to transducer-based systems. This system has already been patented and published.

Multi-Well Organ Bath (MuWOB): consists of an innovative instrument to measure the contractile response caused by chemical substances in small portions of living tissue. Its measurement technique is very different from the equipment used today, and provides advantages that conventional instruments do not offer. The MuWOB is distinguished by its ability to analyze multiple substances with up to ninety-six tissue portions in parallel. This achieves, among other things, a significant saving of substances, and a better use of the animals in each experiment. In addition, the MuWOB occupies very little bench space and is economically within the reach of small and medium-sized laboratories.

Now, the objective is to start its automation, in what refers to the handling of liquids. The implementation of a robotic arm (MuROB, by Multiwell Robot) will be completed. The system will be housed in an oven to keep the temperature and humidification of the environment constant. The mission in charge will be the design of the programs for the management of the MuROB and communication with it.

To this end, the student has been in charge of developing a user interface that allows sending the necessary messages to the automaton (in this case, Arduino Mega). For it, the basic LabVIEW interface has been switched to the App Inventor software development environment, obtaining an application model for Android devices (.apk), facilitating the use of MuWOB by the user, through their Smartphone or Tablet.

In turn, this environment allows you to send messages through the Bluetooth connection, being able to modify the serial communication, which required a direct connection with the Arduino. For this, a Bluetooth module HC-05 compatible with the automaton has been acquired, being configured and connected with the objective previously described.

In relation to the interface, the new requirements to be implemented were added, configuring the programming and definition of the functionalities of the button panel and different design aspects of the interface, achieving a fluid and attractive transition between screens, providing simplicity, robustness and use intuitive to the app.

On the other hand, a series of access restrictions and modification of parameters within the application were added; in this way, the person who is making use of it must be aware of its user, and will make use of the screens and processes to which permission has been provided.

In summary, improvements to the Arduino code have been implemented (improvements implemented to eliminate the air bubble, change of the parameters of the stepper motor in charge of the vertical displacement of the syringes, etc.), mechanical changes to adjust the insertion of the needles, basic and advanced functionalities or others.

2. Marco Teórico

La inmensa mayoría de los procesos fisiológicos del organismo se llevan a cabo bien contrayendo un músculo, bien liberando una sustancia y, la mayor parte de las veces haciendo ambas cosas. La medición de la actividad contráctil ha sido históricamente más sencilla que la evaluación de la liberación (secreción) de una sustancia. Es por ello que desde hace un siglo y medio se ha recurrido a los ensayos de contracción muscular en tejidos aislados (baño de órganos).

Así, para estudiar los efectos farmacológicos de una sustancia en tejidos u órganos aislados de animales [1], los investigadores utilizan sistemas de incubación que mantienen funcionales los tejidos durante el experimento. Estos sistemas intentan medir la actividad de un tejido, la contracción suele desencadenarse con fármacos o con estimulación eléctrica.

Todos estos aparatos apenas han cambiado en décadas, pero siguen utilizándose porque devuelven resultados muy rigurosos y fiables. Sin embargo, su desarrollo ha quedado obsoleto e inconexo con las investigaciones modernas por ser demasiado lentos, tediosos y poco rentables para producir sustancias terapéuticas a gran escala, especialmente para los cribados farmacológicos en la industria farmacéutica. Con estos aparatos resulta imposible atender el crecimiento exponencial que ha experimentado la demanda de fármacos, así que actualmente, los estudios con órganos y tejidos vivos solo se llevan a cabo para estudios muy concretos y en las últimas etapas del cribado farmacológico.

En respuesta a esta circunstancia, los grandes laboratorios farmacéuticos han optado por la utilización de sistemas robóticos diseñados, específicamente, para el análisis de sustancias de forma masiva. Estos sistemas han posibilitado la automatización de multitud de procesos que, anteriormente, el investigador realizaba manualmente; como preparar las muestras, inyectar líquidos, registrar los datos y lavar el material, en otras labores.

Como resultado, las grandes compañías farmacéuticas evalúan librerías de cientos de miles de sustancias diariamente. Sin embargo, los resultados obtenidos por estos mega-robots han sido decepcionantes. Hace unos años se tenía un paradigma: “de cada diez mil sustancias que salen del laboratorio de química farmacéutica, solo una alcanza la clínica”. Hoy, un robot de una gran compañía analiza unas cien mil sustancias diarias y ni de lejos la clínica recibe diez sustancias

nuevas cada mañana. Es más, el número de fármacos notablemente nuevos ha disminuido de forma considerable en las dos últimas décadas.

¿Dónde está el problema? Una de las respuestas a dicha cuestión, la encontramos en que los sistemas autómatas no trabajan con tejidos vivos y órganos aislados, la auténtica diana de los fármacos. En su lugar, las sustancias se ensayan en células cultivadas, enzimas inmovilizadas, receptores, transportadores o ADN sobre placas multipocillos. Como consecuencia, proporcionan una inmensa cantidad de información que no siempre es de fácil de procesar y, con frecuencia, carece de utilidad.

Se cuenta, como ejemplo, con el caso de los Laboratorios Schering, donde una serie de citostáticos proporcionó unos resultados muy prometedores tras el análisis masivo de su actividad anticancerosa, sin embargo, en realidad se demostraría posteriormente que era fruto de una contaminación cruzada con hipoclorito sódico. En otras palabras, a nivel celular, la lejía es un eficiente antitumoral pero aplicada en tejidos, solo es lejía.

La conclusión obtenida es que los aparatos para la experimentación con órganos enteros y tejidos vivos de animales proporcionan resultados más rigurosos, no obstante, tienen en su contra la lentitud, el tedio y la poca rentabilidad. En contraposición, las plataformas robóticas de producción masiva cuentan con la rapidez y el automatismo, aunque con resultados menos provechosos. Por tanto, parece evidente que la solución al problema pasa por el diseño de nuevos instrumentos de laboratorio que permitan experimentar con muestras de tejido vivo y que, al mismo tiempo, estén adaptados a los sistemas robóticos que ya existen para trabajar de forma rápida y automática.

3. Objetivos

El grupo del Dr. Borges, en intima colaboración con el Dr. Rodríguez-Valido, desarrolló un sistema de medición de la actividad contráctil, diseñado particularmente para el ensayo farmacológico de vasodilatadores (en anillos de arteria aorta) y de broncodilatadores (en anillos traqueales). Este sistema combina el uso de placas de 96 multipocillos (cada pocillo alberga un anillo) con métodos ópticos de medición. Este dispositivo bautizado como Multi-Well-Organ-Bath (MuWOB) ha sido descrito [2] [3], patentado [4] y ha dado lugar a dos Tesis Doctorales, correspondientes a *Nuevos instrumentos para la investigación en el laboratorio de Farmacología y Fisiología. Diseño e implementación* [5] y *Nuevas herramientas para la investigación en Farmacología* [6].

Sin embargo, los experimentos con este sistema se llevaban a cabo de forma manual (administrando los fármacos con una pipeta multicanal) y con muchos problemas derivados de la termostatización a 37ºC y de la evaporación de los medios líquidos durante la realización de los experimentos. Además, el disparo de la cámara que llevaba a cabo las mediciones no estaba sincronizada con el manejo de fármacos y debía hacerse de forma manual. Si bien el sistema permite realizar experimentos sobre 96 preparaciones de forma simultánea, seguía teniendo deficiencias en cuanto a su optimización.

Hace unos años el equipo de investigación, con una importante ayuda por parte del Servicio de Electrónica de la ULL, acometió la mejora del sistema MuWOB con dos importantes aportaciones:

1- El desarrollo de un brazo robótico para el manejo de líquidos (MuROB).

2- La colocación de todo el sistema en una incubadora para el control de la temperatura y la humedad.

3- La coordinación de los sistemas MuWOB y MuROB de tal forma que la administración de fármacos estuviese sincronizada con la adquisición de imágenes.

Una de las dificultades para llevar a cabo esta integración es que los programas de manejo fueron desarrollados en sistemas distintos y, en muchos casos, obsoletos. Así, el MuWOB se controla con la plataforma MatLab (The MathWorks, Inc., Portola Valley, California, USA) y el MuROB con LabVIEW (National Instruments, Austin, TX, USA).

Este TFM tiene como objetivo el diseño de una aplicación (app) intuitiva y fácil de usar para controlar el MuROB y sincronizarlo con el MuWOB. También incluye el perfeccionamiento del código Arduino u otros aspectos relevantes y que influyan directamente en el comportamiento del MuROB.

Por un lado, se ha establecido la meta de desarrollar una aplicación con una interfaz atractiva para el usuario, ajustada a las guías de diseño para Android actuales [7] y modernas, con la prioridad de facilitar una transición ágil y un uso versátil de la misma, teniendo como referencia la comunicación constante con el MuROB.

Por otro lado, debido a los requerimientos establecidos por el Dr. Borges, el usuario interesado en este sistema de automatización, se requiere la modificación de los parámetros de movilidad del brazo, jeringas y electroválvulas. A su vez, este proyecto introduce mejoras en el código de Arduino y en la comunicación con el mismo. Esta comunicación se realizaba mediante serial, directamente desde la pantalla del software Arduino IDE [8] en el ordenador y, por tanto, se ha propuesto la utilización del bluetooth y un Smartphone o Tablet que permita manejar cómodamente este brazo automatizado.

Por último, se ha definido el requisito de establecimiento de diferentes niveles de usuario, a fin de restricciones de accesos y modificación de parámetros relativos al funcionamiento del MuWOB. Estos niveles de usuarios previamente considerados son:

- Nivel ingeniero/programador. El cual se encarga solamente del ajuste de las variables relativas al movimiento general del sistema.

- Nivel responsable senior. Este usuario tendrá la mayor parte de los permisos asociados con aspectos específicos de los diferentes procesos llevados a cabo.

- Nivel usuario. En este caso, se le habilita el acceso a parámetros básicos que no comprometan la integridad de la máquina y su entorno, siendo estos las posiciones y los volúmenes a tomar por las jeringas.

3.1. Objectives

Dr. Borges' group, in close collaboration with Dr. Rodríguez-Valido, developed a system for measuring contractile activity, particularly designed for pharmacological testing of vasodilators (in aortic artery rings) and bronchodilators (in tracheal rings). This system combines the use of 96 multiwell plates (each well houses one ring) with optical measurement methods. This device named Multi Well Organ Bath (MuWOB) has been described, patented and has given rise to two Doctoral Theses, corresponding to *Nuevos instrumentos para la investigación en el laboratorio de Farmacología y Fisiología. Diseño e implementación* and *Nuevas herramientas para la investigación en Farmacología*.

However, the experiments with this system were carried out manually (administrating the drugs with a multichannel pipette) and with many problems derived from the thermostatization at 37ºC and the evaporation of the liquid media during the performance of the experiments. In addition, the triggering of the camera that carried out the measurements was not synchronized with the handling of drugs and had to be done manually. Although the system allows experiments to be carried out on 96 preparations simultaneously, it still had shortcomings in terms of its optimization.

A few years ago, the research team, with significant help from the ULL Electronics Service, undertook the improvement of the MuWOB system with two important contributions:

1- The development of a robotic arm for handling liquids (MuROB).

2- Placing the entire system in an incubator for temperature and humidity control.

3- The coordination of the MuWOB and MuROB systems in such a way that the administration of drugs was synchronized with the acquisition of images.

One of the difficulties in carrying out this integration is that the management programs were developed in different systems and, in many cases, obsolete. Thus, the MuWOB is controlled with the MatLab platform (The MathWorks, Inc., Portola Valley, California, USA) and the MuROB with LabVIEW (National Instruments, Austin, TX, USA).

This TFM aims to design an intuitive and easy-to-use application (app) to control the MuROB and synchronize it with the MuWOB. It also includes the improvement of the Arduino code or other relevant aspects that directly influence the behavior of the MuROB.

On the one hand, the goal of developing an application with an attractive user interface, adjusted to current and modern Android design guidelines, with the priority of facilitating an agile transition and versatile use of the application, has been established itself, having as reference the constant communication with the MuROB.

On the other hand, due to the requirements established by Dr. Borges, the user interested in this automation system, requires the modification of the mobility parameters of the arm, syringes and solenoid valves. At the same time, this project introduces improvements in the Arduino code and in the communication with it. This communication was carried out via serial, directly from the screen of the Arduino IDE software on the computer and, therefore, the use of bluetooth and a Smartphone or Tablet has been proposed to comfortably handle this automated arm.

Finally, the requirement to establish different user levels has been defined, in order to restrict access and modify parameters related to the operation of the MuWOB. These user levels previously considered are:

- Engineer/programmer level. Which is responsible only for the adjustment of the variables related to the general movement of the system.

- Senior responsible level. This user will have most of the permissions associated with specific aspects of the different processes carried out.

- User level. In this case, access to basic parameters that do not compromise the integrity of the machine and its environment is enabled, these being the positions and volumes to be taken by the syringes.

4. Esquema general

En este apartado, se muestran los diferentes componentes empleados, así como las imágenes y una descripción general de las funciones de cada uno de ellos.

En primer lugar, se visualiza la figura 1, correspondiente a la placa prototipo (ver mejoras propuestas) que contiene el conexionado eléctrico de cada uno de los motores (jeringas, movimiento vertical y horizontal) y los pines a los que se le atribuye cada señal al Arduino Mega que se ve en la imagen. Además, en la parte izquierda de la imagen se aprecia el módulo bluetooth HC-05, el cual está soldado a los pines de GND (tierra), VCC (tensión de alimentación de +5V), RXD (receptor de datos por serial) y TXD (transmisor de datos por serial).

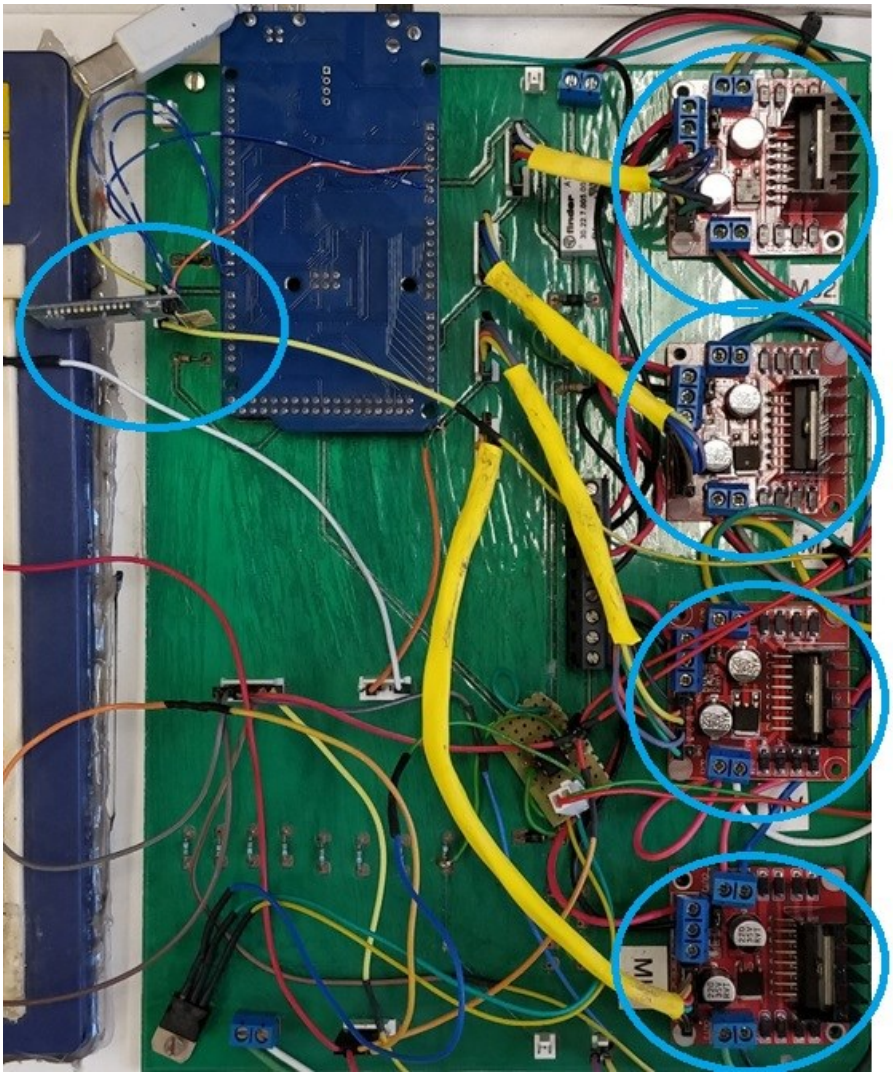

Figura 1. Elementos principales de la PCB.

Por otro lado, además de lo mencionado anteriormente, la figura 2 muestra, en la parte izquierda de la imagen, los botones de control manual que se utilizaban con anterioridad, y la entrada del cableado en la manguera que lo traslada hasta las zonas de interés del MuROB.

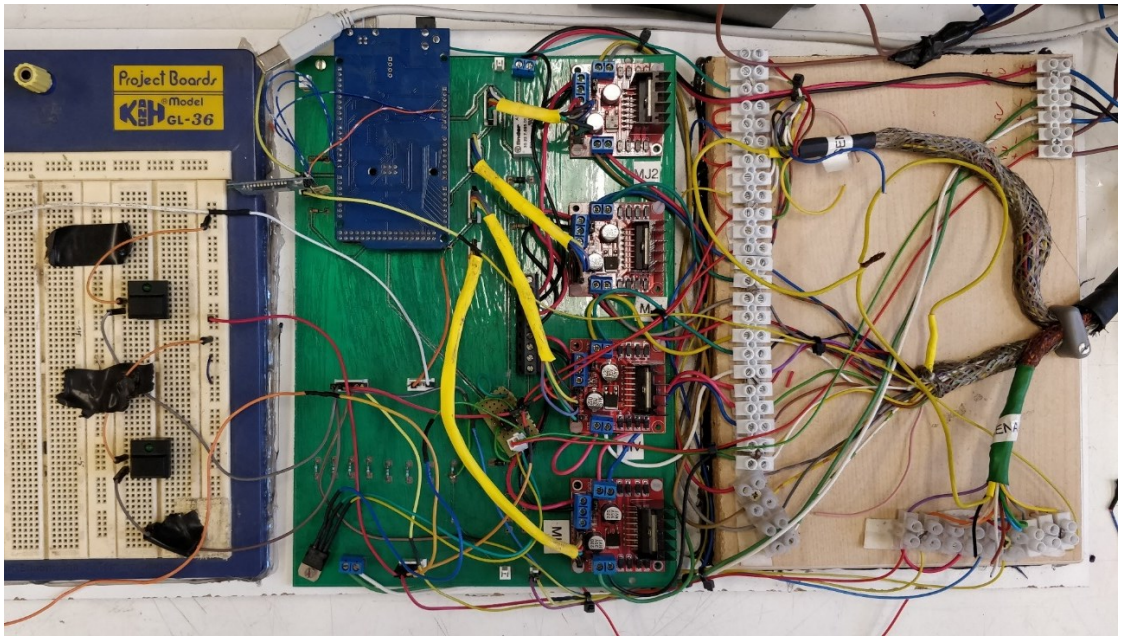

Figura 2. Vista general del prototipo la de PCB y la botonera.

Una vez se ha descrito el esquema de conexionado eléctrico, se procede a comentar la figura 3, la cual contiene una imagen superior de lo que sería el MuROB que se situará en el interior de la estufa. Así, en el centro de la imagen se pueden observar los pocillos en los que se depositará el fármaco, y los que contendrán las arterias; también se distinguen los conductos que unen las jeringas y las agujas, y la carrera del motor horizontal (limitado por la cinta negra de la parte superior y dos finales de carrera).

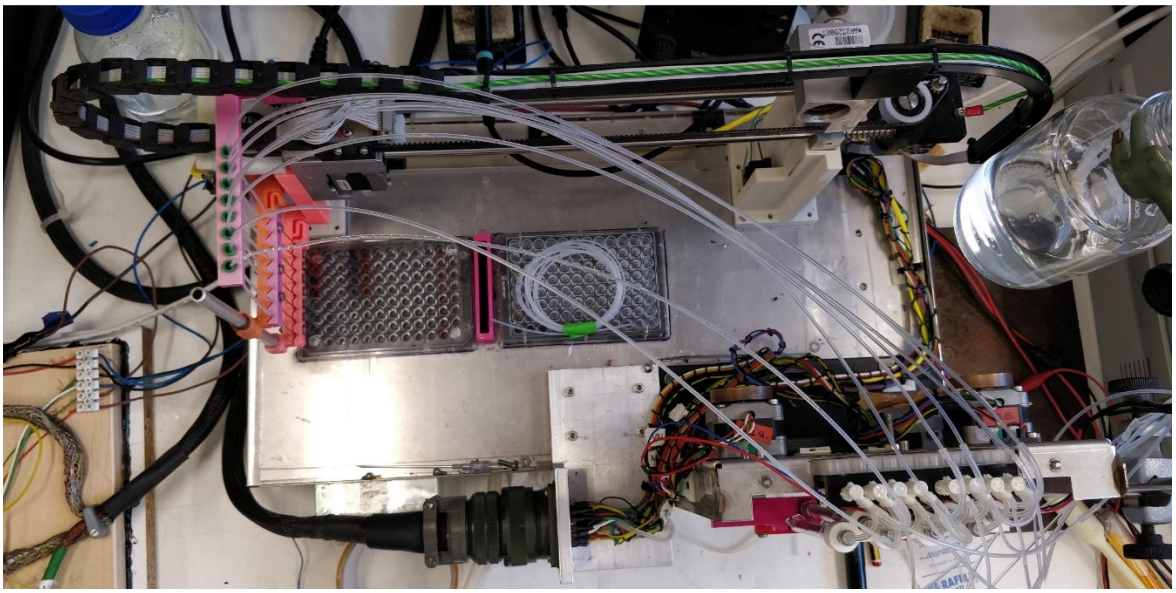

Figura 3. Vista superior del brazo robótico.

Por último, se expone una imagen frontal del sistema (Figura 4), donde se pueden ver con claridad las jeringas, el sistema que regula su altura, las válvulas de aspiración y eyección y la manguera, entre otros elementos.

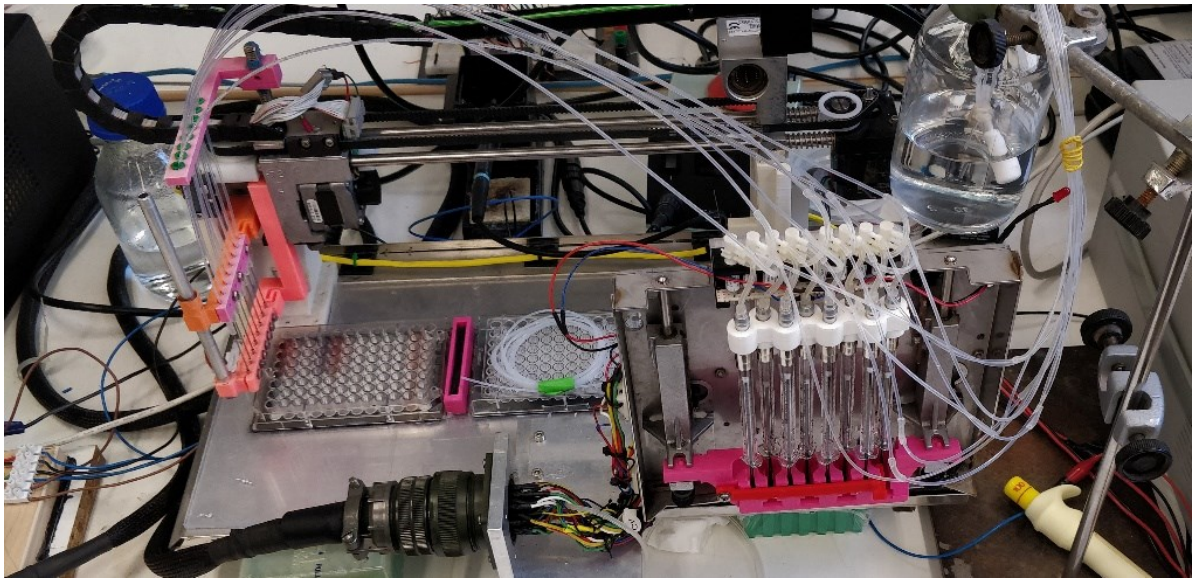

Figura 4. Vista frontal/alzado del MuROB y sus componentes.

5. Diseño de la interfaz de usuario

En relación a este apartado, se expondrán las diferentes acciones que se han desempeñado para elaborar la interfaz de usuario final, la cual será expuesta en este Trabajo de Fin de Máster.

5.1. Diseño

Para llevar a cabo el desarrollo de la interfaz de la app, se ha hecho uso del entorno de desarrollo de software MIT App Inventor [9], como se ha comentado anteriormente. A partir del mismo, se han programado y diseñado un total de 7 pantallas, conteniendo las funcionalidades y capacidades necesarias para comunicarse con el autómata y permitir al usuario una experiencia amena y de calidad. A continuación, se muestra la pantalla principal y su contenido (ver figura 5), las pantallas restantes se insertan más adelante, en el apartado de resultados.

Figura 5. Interfaz de la pantalla principal.

5.2. Programación y funcionalidades

En este subapartado, se insertarán algunas de las imágenes correspondientes al código de App Inventor, seleccionando los bloques más destacados y con funcionalidades diferentes.

En primer lugar, dentro de la pantalla número 1, se puede resaltar los bloques de comunicación Bluetooth del botón flotante (Figura 6), mediante los cuales se permite al usuario acceder a la lista de dispositivos bluetooth previamente conectados con el Smartphone/Tablet, así como realizar la conexión o desconexión de esta misma (ver Figura 4 para visualizar la botonera correspondiente a esta pantalla).

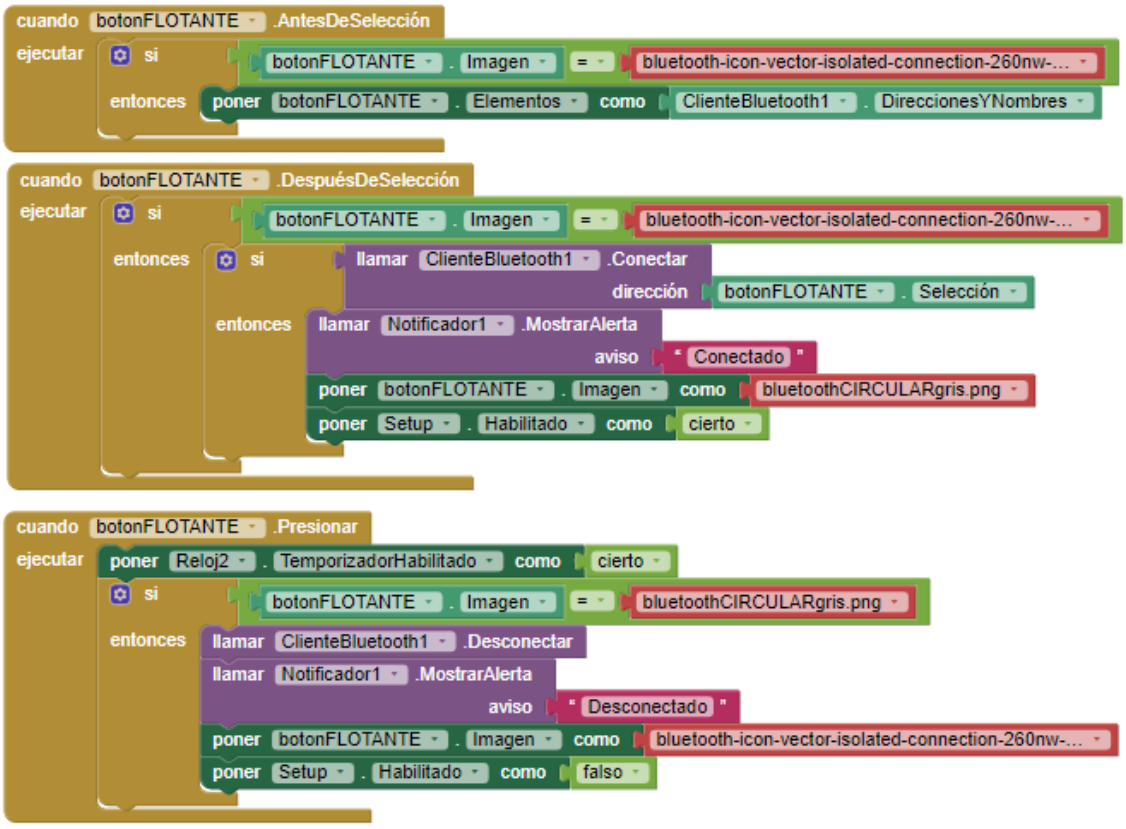

Figura 6. Configuración del botón flotante.

Por otro lado, se encuentra los bloques de la Figura 7, usados exclusivamente en esta pantalla. En esta ocasión, se corresponden con los selectores de fecha y hora, con el fin de que el usuario sea capaz de visualizar el momento exacto en el que ha realizado una actividad, simplemente a modo informativo.

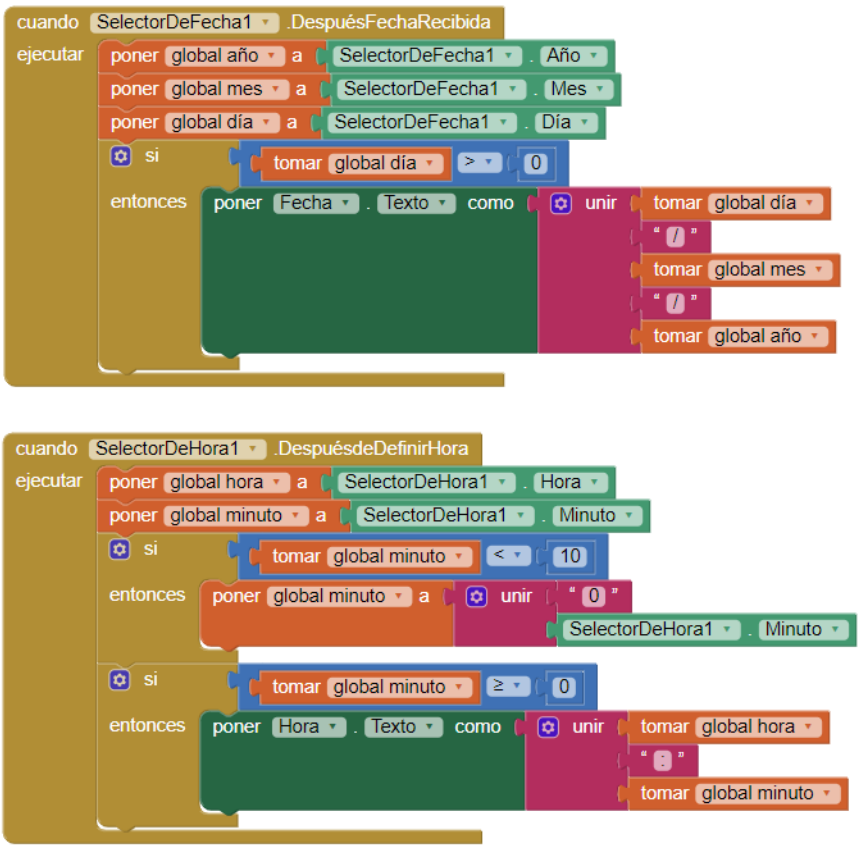

Figura 7. Selector de fecha y hora.

Con respecto a la figura 8, se observa el bloque correspondiente al código de los botones Setup y Stop. Por un lado, el botón de Stop se encarga de abortar la ejecución del programa, ante cualquier señal de emergencia detectada por el usuario (influye directamente en el comportamiento del MuROB, ya que éste sólo se puede parar mediante la fuente de alimentación, con interrupciones de Arduino o enviándole un string vacío); por otro lado, el botón de Setup envía la cadena contenida en el cuadro de texto del String decimal, a la vez que muestra un mensaje por pantalla: “Se ha enviado al string”, con el fin de que el usuario tenga la certeza de que el mensaje ha sido reportado al Arduino y, además, comprueba que el intento de envío ha sido realizado por cualquiera de los usuarios con permiso.

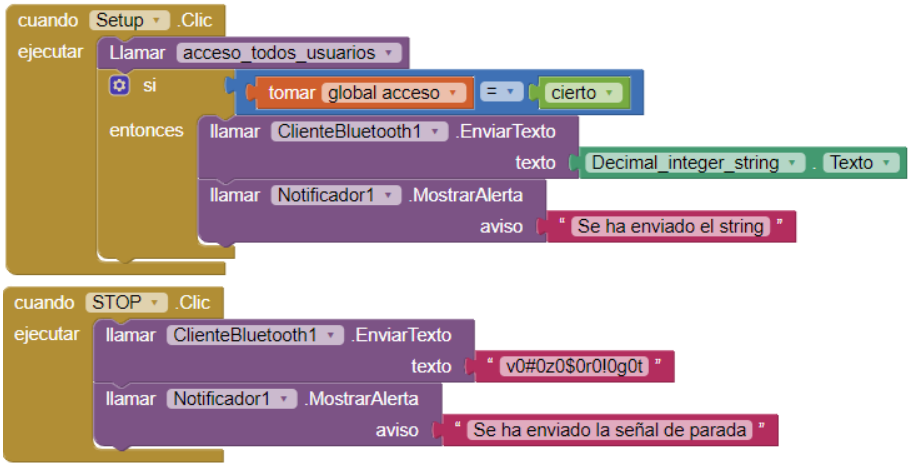

Figura 8. Envío del string y parada de emergencia.

En la figura 9, se aprecia la función de verificación de acceso todos los usuarios (para los casos en los que los 3 niveles de usuarios pueden acceder a la pantalla en concreto), esta función permite ahorrar líneas de código al no repetir varias veces los mismos bloques. También se muestra que, en el caso de la pantalla 6, se le guarda el texto presente en el cuadro de texto de "User", mientras que al seleccionar la pantalla 7, se guarda el valor del string ya presente (todo esto según interés en el diseño de cada pantalla).

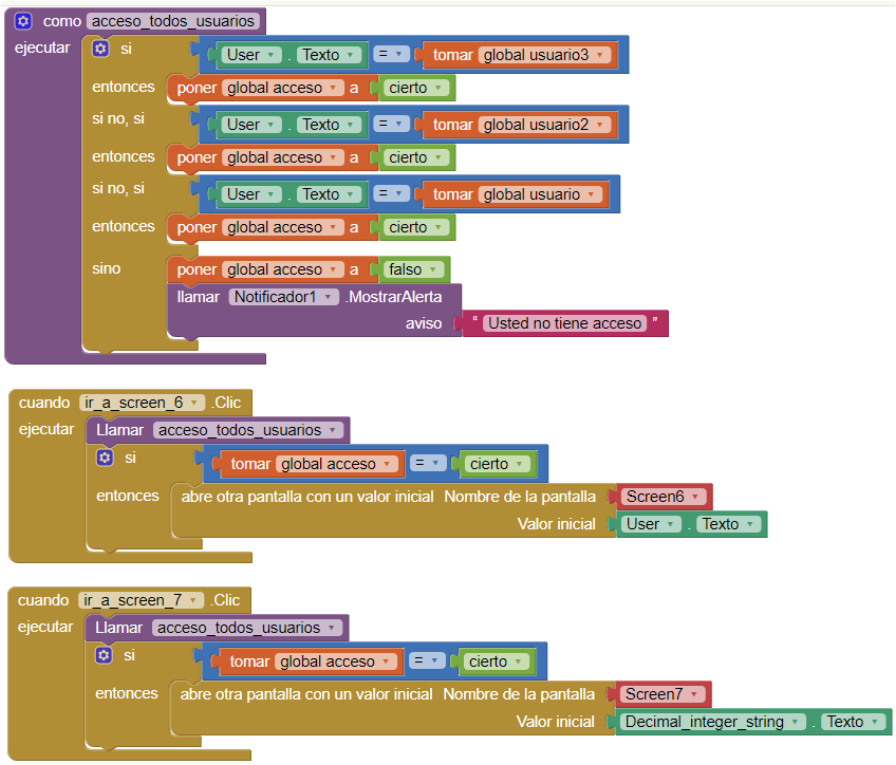

Figura 9. Configuración niveles de usuario y transición entre pantallas.

En la siguiente imagen (figura 10), se describe el funcionamiento de la pantalla principal una vez es iniciada, tanto cuando se inicia por primera vez la aplicación como cuando transita desde una pantalla secundaria. En este caso, capta el valor del string (de una pantalla secundaria), fuerza la conexión al módulo bluetooth HC-05 y, por último, indica que es necesario introducir el usuario para realizar cualquier tipo de interacción usuario-app.

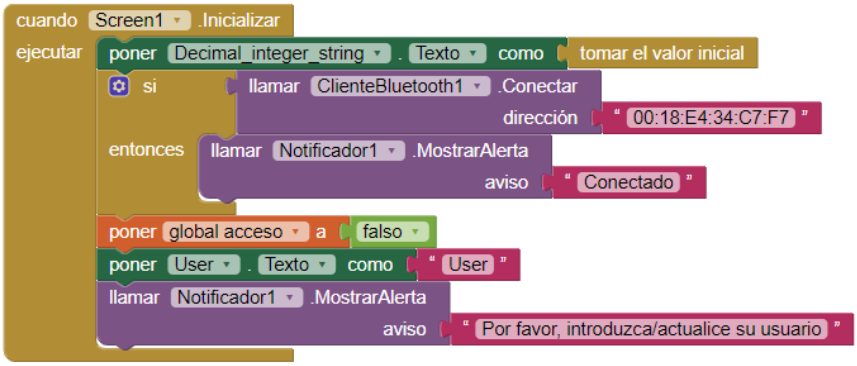

Figura 10. Inicialización de la pantalla principal.

Como otra parte destacable de la pantalla 1, temática que se comenta más en profundidad en los subapartados siguientes (5.4 de este documento), se definen los nombres de usuarios que serán admitidos por la aplicación, por defecto no se concede acceso hasta que se introduzca una credencial válida (ver figura 11).

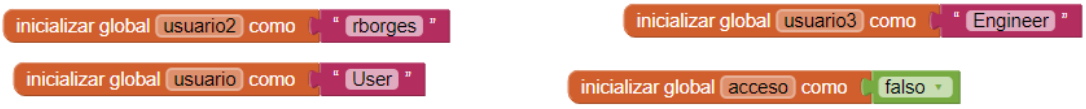

Figura 11. Definición de niveles de usuarios.

Por último, la configuración del menú lateral se ha llevado a cabo en todas y cada una de las pantallas, con exactamente la misma configuración (figura 12). En este caso, se hace uso de un temporizador que, en el momento en el que se presione el botón "≡", permite actualizar el ancho del menú lateral, haciéndolo visible y desplazando el contenido de la pantalla hacia la derecha, con el fin de interactuar con los diferentes botones presentes en este menú, los cuales permiten transitar entre todas las pantallas de la aplicación.

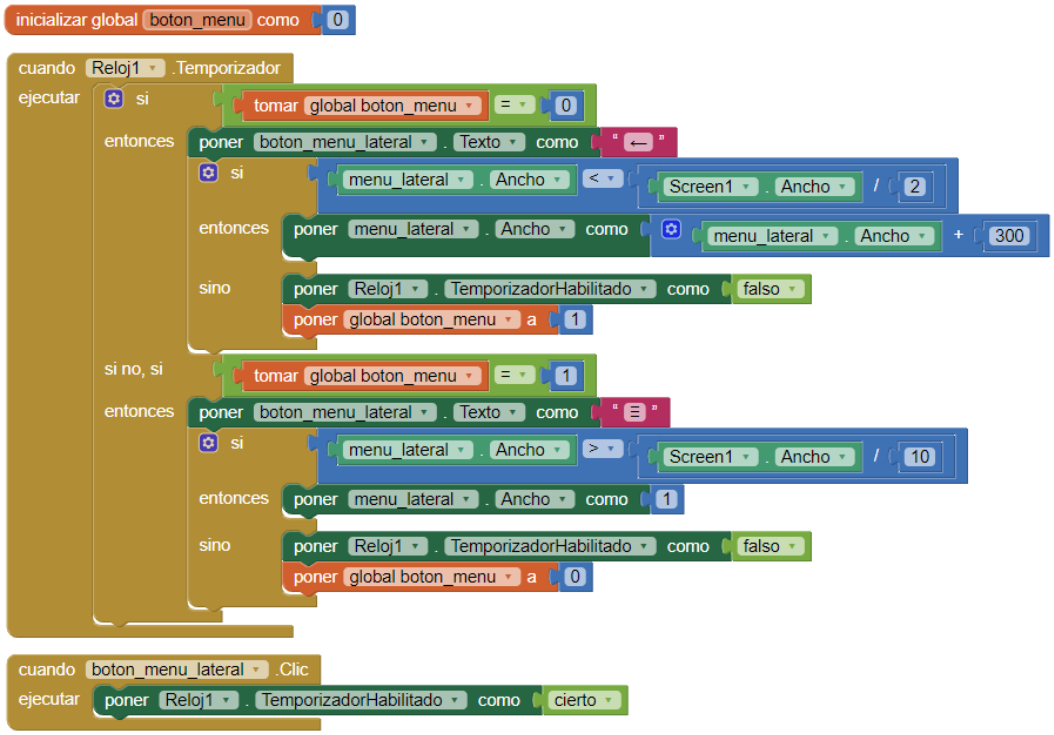

Figura 12. Configuración del menú lateral.

A continuación, en la figura 13, se visualiza el envío del string correspondiente al proceso de purga desde esta misma pantalla, sin necesidad de retornar a la pantalla principal; también se contempla la opción de volver a la pantalla 1, por ello, si se hace clic en la flecha de "atrás" en el Smartphone, guarda el string y lo plasma en la main pantalla.

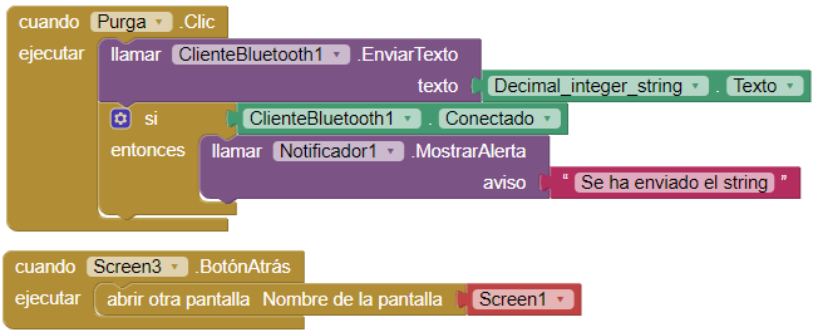

Figura 13. Envío del string de purga.

Además, la pantalla de purga también fuerza la conexión con el módulo bluetooth (figura 14), ya que, debido al núcleo de diseño del App Inventor, la conexión se pierde con la transición entre pantallas, por tanto, se requiere para poder enviar otro mensaje. A su vez, también se observa la actualización del string al pulsar el botón correspondiente, y las condiciones de modificación de parámetros para los usuarios, en este caso solamente el Administrador.

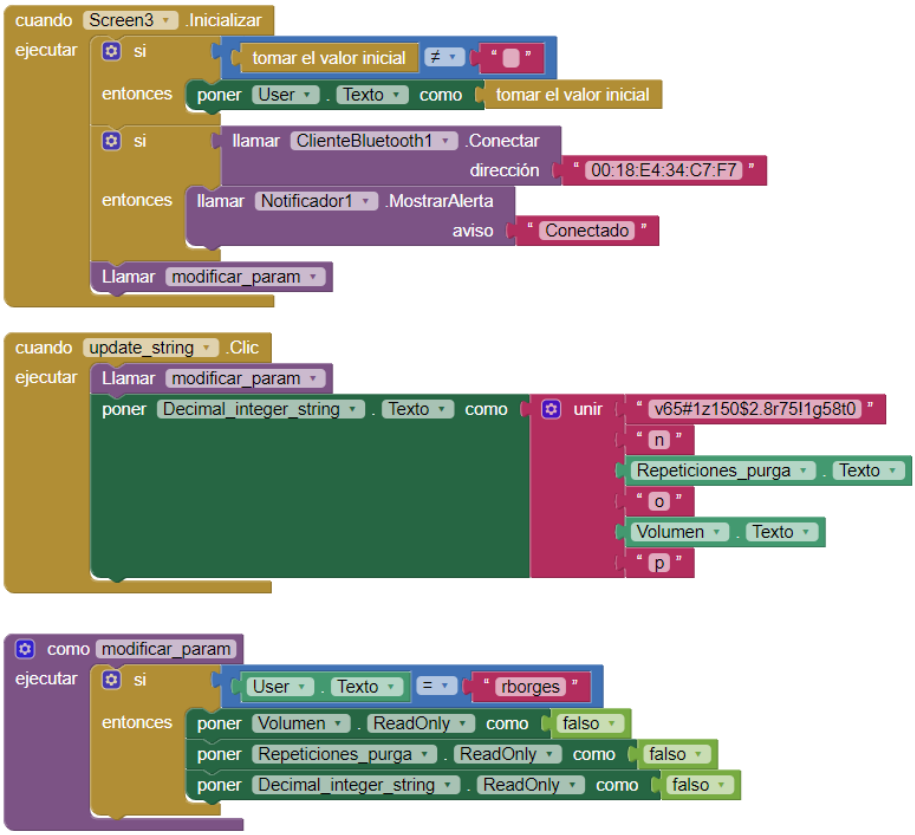

Figura 14. Otras funcionalidades de la pantalla del proceso de purga.

Al mismo tiempo, en la pantalla 4 se podía habilitar/deshabilitar el funcionamiento de ciertos parámetros de movilidad en modo manual, aunque esta funcionalidad estaba en desuso y, debido a su no utilidad actual, se eliminó. Asimismo, esta misma pantalla permite modificar los parámetros claves de movimiento del sistema (figura 15), los cuales puede cambiar el ingeniero, con el fin de un comportamiento estable y no variable.

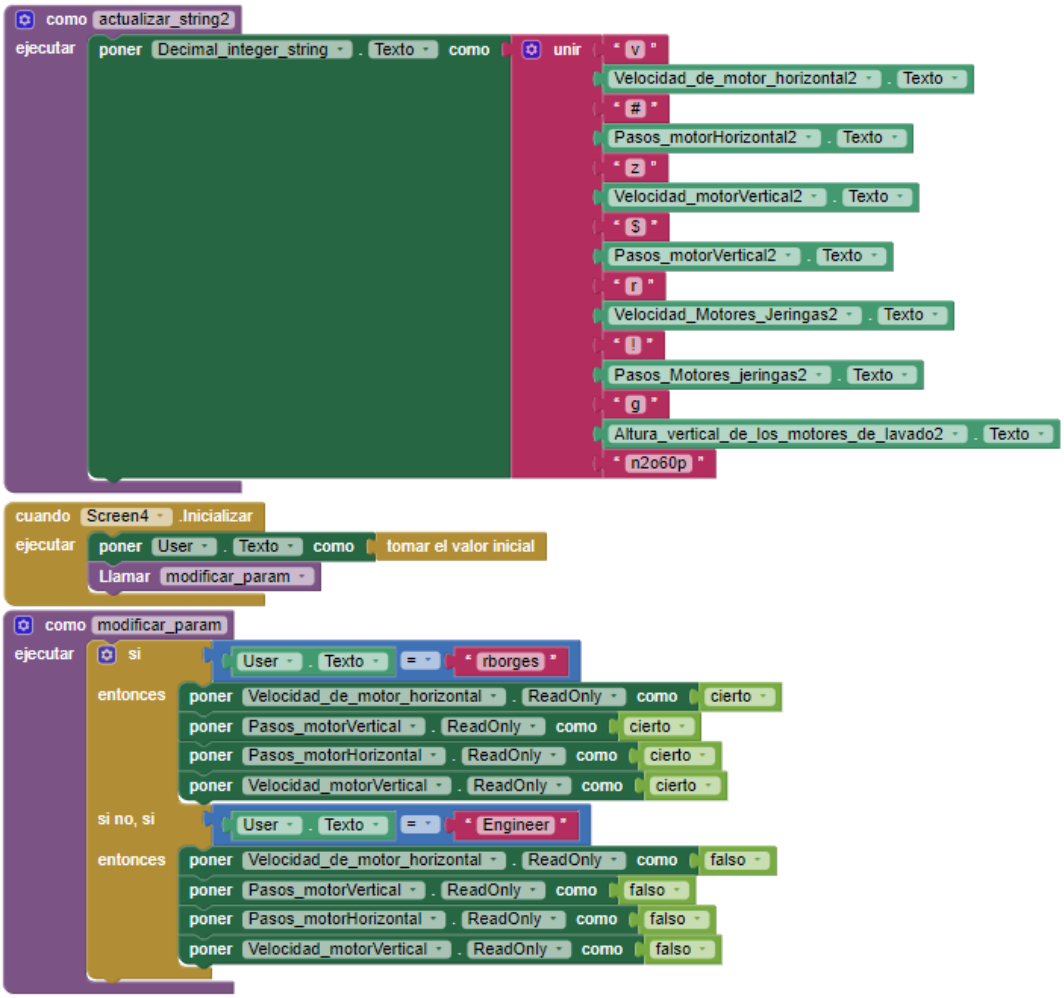

Figura 15. Actualización de los parámetros profesionales y permiso para modificarlos.

Acerca de la pantalla 6, comprende el proceso de lavado. Para ello, se emplea también la función de modificación de parámetros y, a su vez, la actualización del string tras las modificaciones que hayan sido realizadas (figura 16).

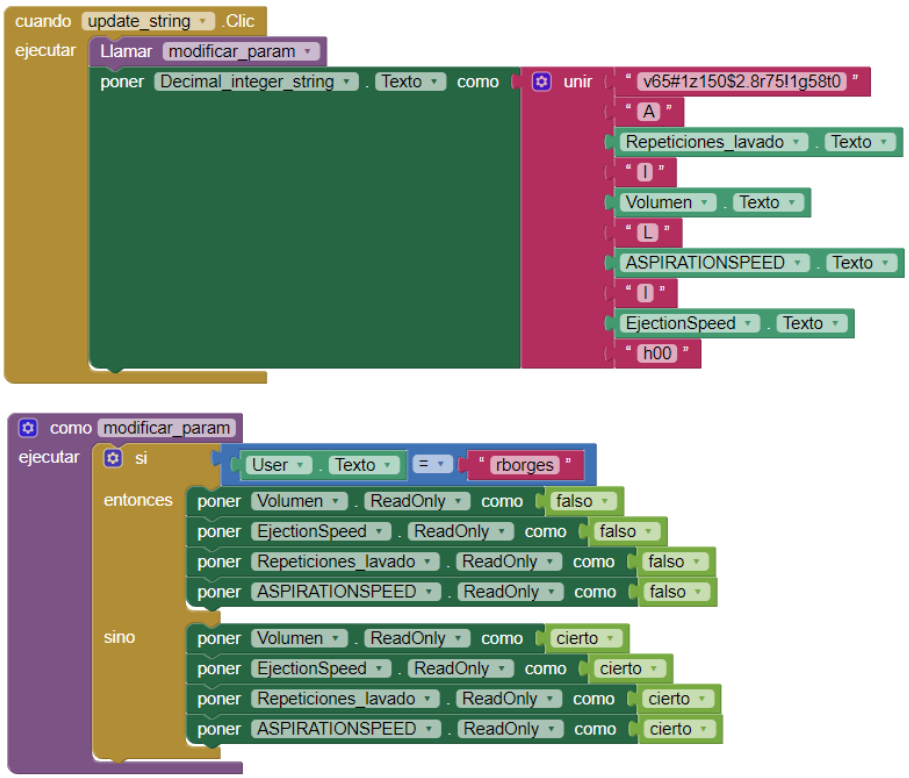

Figura 16. Actualización string del lavado y permiso para modificar parámetros.

Por último, en cuanto al proceso de SDA, lo más llamativo es la cadena de más de 55 caracteres, siendo la más completa de todos los procesos (figura 17), ya que contiene también un lavado inicial y los diferentes parámetros de este proceso,

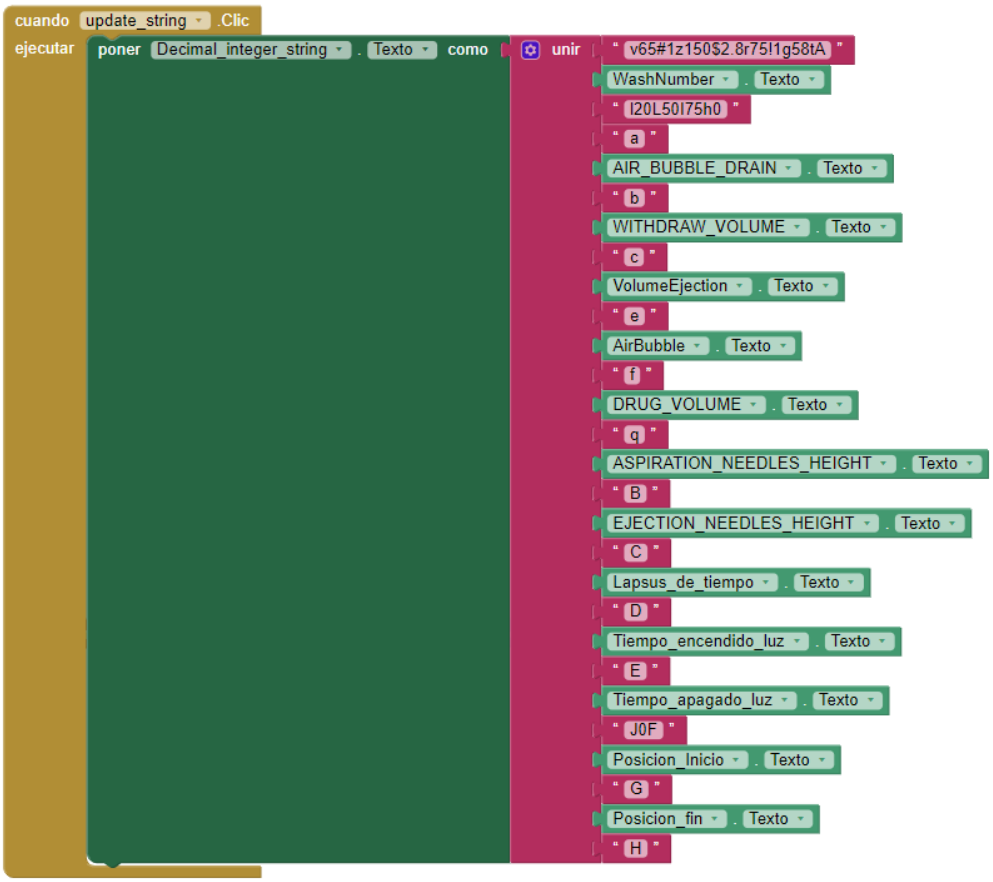

Figura 17. Actualización del string del SDA.

5.3. Comunicaciones

En primer lugar, se adquirió un módulo bluetooth inicial, concretamente el HM-10 BLE (Bluetooth Low Energy) que, tras intentar vincularlo con el Smartphone, se observaron una serie de incompatibilidades. En este sentido, la única manera de vincularlo y enviarle mensajes desde un Smartphone, era descargando una app auxiliar denominada *BLE Scanner*; el problema de esto es que, una vez se accede a la conexión de la app del App Inventor, se perdía el vínculo directo con el módulo, perjudicando notoriamente en la comunicación inalámbrica. Además, incluso sabiendo la dirección del módulo, no se permitía forzar la conexión a este dispositivo, debido a sus incompatibilidades con Android/Apple.

Es por ello que, tras realizar una serie de investigaciones, se decidió emplear el módulo bluetooth HC-05. Este último, tiene la característica de que [10] es un módulo Maestro-Esclavo, es decir, además de recibir conexiones desde un PC, Smartphone o Tablet, también permite generar conexiones hacia otros dispositivos bluetooth.

Como se mencionó con anterioridad (ver apartado 4), el módulo se soldó a los pines de GND (tierra), VCC (tensión de alimentación de +5V), RXD (receptor de datos por serial) y TXD (transmisor de datos por serial); aunque el Modulo puede operar a 5V, lo recomendable son 3.3V, para obtener un mayor ahorro de energía. Los pines en concreto se pueden ver en la siguiente imagen (ver figura 18).

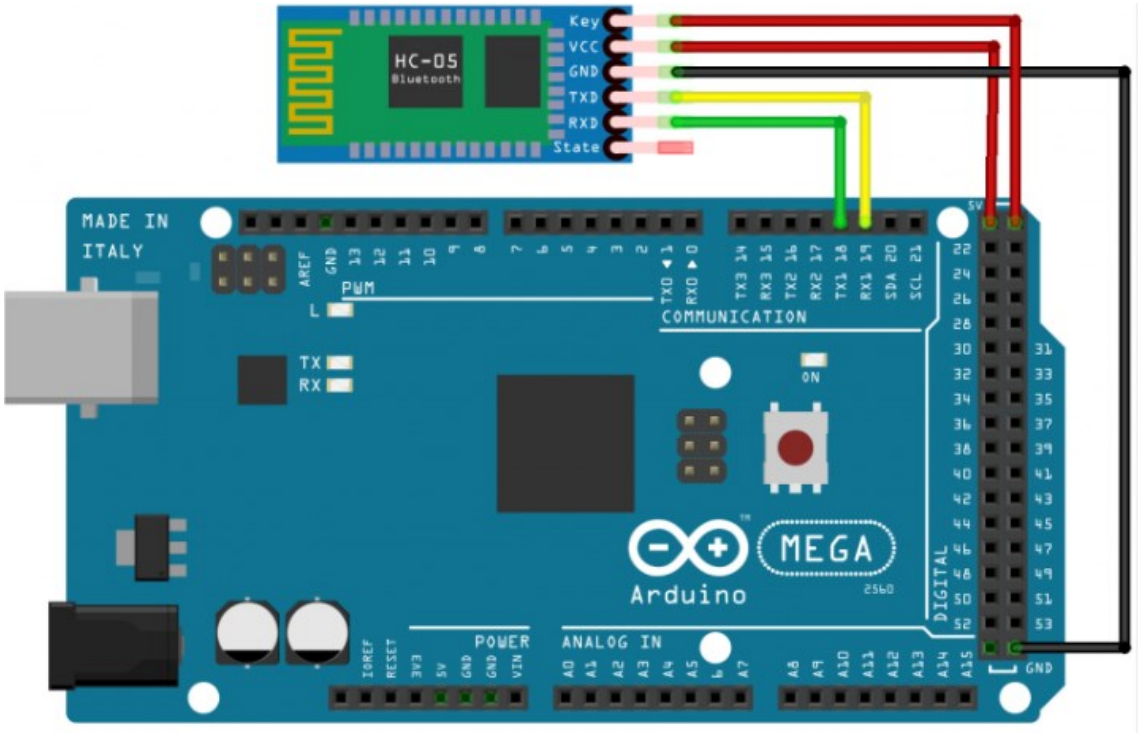

Figura 18. Conexionado del módulo bluetooth HC-05 al Arduino [11].

Tras realizar el pineado con el Arduino Mega, se programó su conexión en el App Inventor; para ello, debido a que, tras realizar una transición entre pantallas se perdía la conexión con el módulo, se tuvo que forzar la conexión al bluetooth en cada una de las pantallas que la requerían, incluyendo la dirección del módulo HC-05 en los bloques del entorno de desarrollo y, consiguiendo, el objetivo de la vinculación constante para envío de mensajes.

5.4. Restricciones de usuario

Para la aplicación de estas restricciones, el administrador estableció los requisitos de acceso a las pantallas y/o modificación de parámetros, en función del usuario introducido. A partir de estas indicaciones, las cuales se enunciarán a continuación, se procedió a imponer las restricciones en cada una de las pantallas.

5.4.1. Usuarios

Como se ha mencionado anteriormente, en función del usuario introducido, se tendrá acceso a unas pantallas u otras, así como permisos de lectura o escritura. Para ello, se definieron tres niveles de usuarios:

1. Ingeniero → se le permite configuración de parámetros de motores, velocidades, u otros básicos relacionados con el movimiento general del sistema.

2. Administrador → tiene acceso a la modificación de las velocidades de las jeringas, la altura de expulsión y aspiración del líquido, y la configuración de los parámetros de la purga y lavado, entre otros.

3. Nivel usuario → sus autorizaciones son más limitadas, está restringido a realizar procesos como la purga y el lavado, pero sólo se le permite la modificación de la posición de inicio y fin del SDA y, además, el volumen que desea retirar e inyectar.

Así pues, en el siguiente apartado se especificarán los diferentes permisos establecidos por pantalla.

5.4.2. Permisos

En este contexto, se enumeran las correspondientes opciones de edición y visualización para cada una de las pantallas, en función del usuario introducido:

- Pantalla 1: Se permite el acceso a todos y cada uno de los usuarios, ya que es la pantalla principal, desde la cual se iniciará el proceso de navegación a través de la aplicación, así como se solicita el usuario con el fin de evaluar la transición a las otras pantallas. No dispone de datos a modificar.

- Pantalla 2: Sólo se le permite acceder al ingeniero. Esto es debido a que contiene los parámetros fundamentales de movimiento del MuWOB, los cuales son predefinidos por el mismo, con el fin de un correcto funcionamiento y, si se requieren modificaciones de los mismos, el ingeniero los podría reajustar.

- Pantalla 3: Todos los niveles de usuarios pueden acceder a esta pantalla. En cambio, en este caso sólo el Administrador tiene la opción de modificar ciertos parámetros (volumen y repeticiones de la purga).

- Pantalla 4: En esta pantalla, tanto el ingeniero como el Administrador podrán acceder y, a su vez, modificar los parámetros pertinentes. Estos valores corresponden con las alturas y velocidades de los motores y jeringas.

- Pantalla 5: Esta pantalla no requiere restricciones de lectura o escritura, debido a que su contenido es información y, además, no tiene funcionalidades volátiles ni procesos vinculados.

- Pantalla 6: Para este caso, podrán acceder todos los usuarios. Sin embargo, el Administrador es el único con permisos de escritura, principalmente las velocidades de inyección y aspiración, así como el volumen y repeticiones de lavado a realizar.

- Pantalla 7: Finalmente, esta pantalla será visible por todos los usuarios. Además, todos tendrán permiso de escritura; por un lado, el ingeniero y el administrador podrán variar todos los valores a su antojo, mientras que el nivel de usuario estará limitado a la posición de inicio y fin del SDA y los volúmenes del mismo.

5.5. Tabla de variables en uso

En este apartado, se muestra un desglose de los diferentes caracteres que están configurados y enviados mediante la manguera que conecta con el autómata; de esta forma, son parámetros que se utilizan a fecha de mayo de 2022 y, además, otros parámetros que pueden llegar a aplicarse en un futuro.

Número de variable	Variable	Parámetro
1	a	Air Bubble Drain
2	b	Withdraw Volume
3	c	Volume Ejection
4	e	Air Bubble
5	f	Drug Volume

6	g	Lim. Vertical descent
7	h	Wash activation
8	l	Wash Volume
9	n	Purge Repetitions
10	o	Purge Volume
11	p	Purge activation
12	q	Needle Suction Height
13	r	Syringes Speed
14	s	Stop
15	t	Parameters OK
16	v	Horizontal Motor Speed
17	z	Vertical Motor Speed
18	$	Steps Vertical Motor
19	!	Steps Syringes Motors
20	#	Steps Horizontal Motor
21	A	Wash Repetitions
22	B	Ejection Needles Height
23	C	Time Lapse
24	D	Light ON Time
25	E	Light OFF Time
26	F	Starting Position SDA
27	G	End Position SDA
28	H	SDA activation
29	I	Ejection Speed
30	J	Drug Analysis activation
31	L	Aspiration Speed

Tabla 1. Caracteres reconocibles por el Arduino.

A continuación, se muestra la tabla de variables que se utilizan para indicar al MuWOB que se encuentra en control manual, facilitando al usuario la prueba de acciones concretas o movimientos de interés para un cierto proceso, los cuales no hayan sido previamente automatizados.

Número de variable	Variable	Parámetro
32	d	Motor horizontal a la derecha
33	i	Motor horizontal a la izquierda
34	j	Jeringas arriba
35	k	Jeringas abajo
36	m	Control manual
37	u	Subir motor vertical
38	w	Bajar motor vertical

Tabla 2. Caracteres de control manual.

5.6. Cadenas (strings) de caracteres

Tras visualizar los caracteres que identifica el Arduino, se procede a enumerar los diferentes mensajes (cadenas/strings de caracteres) que será capaz de leer, mediante la comunicación bluetooth, el autómata.

Estos mensajes tienen, como objetivo principal, el indicar los valores de las diferentes variables de configuración del brazo [ejemplo: v65#1…, 65 equivale al valor dado a la velocidad horizontal de los motores (v), y el 1 corresponde a los pasos de ese mismo motor (#)], siendo imprescindible que la comunicación con el Arduino Mega esté activa, para poder realizar los movimientos y procesos oportunos.

Para corroborar el significado de cada carácter, dirigirse a la tabla anterior (Tabla 1).

5.6.1. Parámetros motores avanzados

En primer lugar, los parámetros de configuración del movimiento horizontal y vertical, y distancias a recorrer por parte del manipulador. Estos son parámetros no modificables, los cuales se han establecido y restringido por parte del ingeniero, con el fin del correcto funcionamiento del sistema.

Además, cabe destacar que esta cadena es imprescindible, y se envía continuamente, junto la información anexa del proceso específico a desarrollar (purga, lavado u otro).

Así pues, la cadena por defecto es la siguiente (String1):

String1: *v65#1z150$3r75!1g58t.*

Como se puede observar, este mensaje se corresponde con el establecimiento de los pasos y velocidades de los motores horizontales, verticales y de las jeringas.

5.6.2. Purga

En cuanto al proceso de purgado, consiste en eliminar las burbujas de aire que puedan estar presentes en las jeringas y conductos. A su vez, esta cadena se compone de los parámetros avanzados de los motores y, además, del número de repeticiones y el volumen de la purga a llevar a cabo. El string por defecto se muestra a continuación (String2).

String2: *v65#1z150$3r75!1g58t0n2o60p.*

5.6.3. Lavado

En relación al mensaje correspondiente al lavado de las jeringas y agujas, contiene los parámetros de número de lavados y volumen del mismo y, además, la velocidad de inyección y aspiración del fármaco. Por lo tanto, el string predeterminado se puede visualizar en el siguiente párrafo (String3).

String3: *v65#1z150$3r75!1g58t0 A1l20L50I75h00.*

5.6.4. Análisis del fármaco (Drug Analysis)

Con respecto al mensaje del *Drug Analysis* (no utilizado), se compone del tamaño de la burbuja de aire (establecida en 10 µm para un mejor funcionamiento y, evitar así, posibles mezclas con el líquido ya presente en los conductos), volumen de retiro e inyección, burbuja de aire de drenaje, volumen de fármaco, y la altura de aspiración e inyección de las agujas.

Por lo tanto, el string predeterminado es el siguiente (String4):

String4: *v65#1z150$3r75!1g58t0a20b10c50e10f10q1B5J.*

5.6.5. Adición simple del fármaco (SDA)

La cadena del *Simple Drug Analysis* (no utilizado), se compone del tamaño del lapsus de tiempo, el tiempo de encendido y apagado del led, y la posición de inicio y fin del SDA.

Así pues, el string por defecto es el siguiente (String5):

String5: *v65#1z150$3r75!1g58t0C15D5E5F1G1H.*

5.6.6. Análisis optimizado del fármaco

El mensaje del *Drug Analysis* optimizado en cuestión, viene a ser una combinación del *Drug Analysis* y del *SDA;* así, se compone del tamaño de la burbuja de aire (establecida en 10 µm para un mejor funcionamiento y, evitar así, posibles mezclas con líquido ya presente en los conductos), volumen de retiro e inyección, burbuja de aire de drenaje, volumen de fármaco, la altura de aspiración e inyección de las agujas, el lapsus de tiempo, el tiempo de encendido y apagado del led, y la posición de inicio y fin del SDA.

El string predeterminado se muestra a continuación (String6, cadena total del SDA):

String6: *v65#1z150$3r75!1g58t0A1l20L50I75h0a20b10c50e10f10q1B5C15D5E5J0F1G1H.*

5.6.7. SDA optimizado

Por último, el string del *Simple Drug Analysis* optimizado (no utilizado), se compone de, solamente, la posición de inicio y fin del SDA. Este proceso tampoco se utiliza, debido a su sencillez y carencia de algunos parámetros necesarios.

Así pues, el string por defecto es el siguiente (String7):

String7: *v65#1z150$3r75!1g58t0F1G1H*.

6. Mejoras, limitaciones y errores

6.1. Mejoras propuestas

Por un lado, en cuanto a las opciones de mejora del código Arduino, una de las posibilidades más inmediatas a programar sería la opción de elegir la columna de fármaco a aspirar, y su destino final. Esto es debido a que, tal y como está el diseño actual, se ha preferido actuar sobre un proceso básico que funcione, sin actuar mucho más allá, por las limitaciones de tiempo que conlleva un TFM, así como las limitaciones de horario de acceso al laboratorio del Servicio de Electrónica, donde su ubicaba el MuROB.

El proceso actual, como se ha mencionado anteriormente, permite elegir la posición de inicio de succión y eyección, siendo la misma columna para el fármaco que la del pocillo de depósito del mismo. Por ejemplo, introducir el fármaco de la columna 7 en la columna 7 de las arterias, de forma que, no se puede introducir el fármaco de, por ejemplo, la columna 9 en la columna 9 de los órganos.

Por otro lado, otras opciones que se han planteado es la de, como proceso ya más avanzado, escoger el fármaco de una columna, e introducirla en todas las columnas de arterias, con el fin de ver su comportamiento en cada tipo de fragmento.

A su vez, se pretende también controlar la temperatura de la estufa (permitiendo al software indicar cuándo encenderla y la temperatura de la misma), así como lanzar un mensaje para captar imágenes en el momento concreto que el usuario la requiera. Todo esto no era viable de implementar en este corto período de tiempo, sobre todo teniendo en cuenta que los procesos básicos tenían sus limitaciones y la interfaz del LabVIEW estaba al límite de lo obsoleto. Hay que tener en cuenta que, como se ha mencionado, el proyecto se ha centrado más en fijar y dejar funcionando los procesos fundamentales y, además, el desarrollo de una interfaz intuitiva, sencilla y agradable de manejar.

Por otro lado, debido a las limitaciones del desarrollo de software App Inventor, se realizó la adición de una simulación de un botón flotante (empleando un botón básico y atajos de código para imitar su comportamiento), ya que no se dispone de botones flotantes como elemento para añadir a la interfaz de usuario.

Junto con esto, se propone la realización de una PCB más compacta, debido a que, desde un inicio, se han ido realizando modificaciones en el circuito impreso inicial. Así, en un futuro, cuando se finalicen las nuevas adaptaciones, se rediseñará esta PCB, con el fin de mejorar el conexionado, la seguridad y la estabilidad de las diferentes señales, así como optimizar el espacio que ocupa.

Además de lo ya mencionado, con respecto al diseño de la interfaz de usuario, la apariencia se podría mejorar dentro del App Inventor, añadiendo extensiones de subscripción no gratuita, pero que le podrían dotar al diseño un atractivo adicional; actualmente la app ha sido diseñada con el único enfoque de uso por parte del administrador, no con fines comerciales, por lo que no se ha considerado esta implementación.

Por último, como se ha comentado anteriormente, el MuROB irá situado en el interior de una estufa, con control de temperatura incluido; esta estufa en cuestión se puede ver en la figura 19.

Figura 19. Vista frontal y trasera de la estufa.

6.2. Limitaciones y errores

A parte de la limitación de presupuesto, se ha dispuesto de una limitación horaria para acceder al aparato, además de un tiempo y unos objetivos limitados acordes al mismo, quedando pendiente la automatización de la cámara y el control de la temperatura de la estufa.

El módulo bluetooth HM-10 BLE no sirve para conectar con Android, ya que no son HM-10 originales, son copias y, por lo tanto, tienen un firmware distinto. Más concretamente estos módulos son los MLT-BT05. Por la información obtenida en internet, la única forma de comunicarse con estos módulos era mediante una app llamada "BLE Scanner" [12], a partir de la cual se le puede enviar un mensaje al módulo. El problema de la misma es que, dado que la conexión se realiza mediante la aplicación y no con el servicio bluetooth del terminal, no quedaba registrado en los dispositivos bluetooth anteriormente conectados con el Smartphone. Además, el App Inventor no permite acceder a otra aplicación dentro de la misma, por lo que eran incompatibles.

Debido a esto, se comenzó a usar el módulo bluetooth HC-05. Así, este módulo permitió realizar el vínculo con el Smartphone sin ningún problema, funcionando correctamente la comunicación, tras su posterior configuración y conexionado [10].

6.3. Modificaciones mecánicas/eléctricas realizadas

Con respecto a las adaptaciones que se han aplicado, se enumeran a continuación:

1. Desplazamiento de la sujeción de las agujas; esto es debido a que, por una mala fijación al llevar a cabo el descenso vertical de las agujas, acabó cediendo y desalineándolas, provocando que las agujas no entraran correctamente en el centro de cada uno de los pocillos.

2. Realineación de las agujas con respecto a la horizontal. Debido al problema descrito en el punto número 1 de este apartado, las agujas golpeaban los pocillos, variando así la posición vertical de algunas de ellas; por lo tanto, se han tenido que reajustar a la misma altura.

3. Adición del módulo bluetooth encargado de la comunicación entre el terminal móvil y el Arduino. Este módulo implicó soldar los cables que lo conectaban a los pines de la placa, ya que se encuentra boca abajo, y no se podía conectar directamente.

4. Desconexión del cable correspondiente a la comunicación serial con el ordenador, debido a la nueva conexión por bluetooth (con la posibilidad de volver a realizar la comunicación serial, no se ha deshabilitado).

5. Cambio del módulo bluetooth inicial, como se explicó anteriormente, debido a que el HM-10 BLE no es compatible con Android, ya que no son HM-10 originales, son copias y, por lo tanto, tienen un firmware distinto. Más concretamente estos módulos son los MLT-BT05 [13]. El HM-10 es un módulo BLE Bluetooth 4.0, lo que significa que no se puede conectar a los módulos Bluetooth 2 / 2.1 como los viejos HC-05 y HC-06. Esto se debe a que BLE no es una actualización de Bluetooth Classic; funciona de una manera muy diferente.

6. Modificación de las válvulas, ya que no inyectaban lo aspirado. En este caso, se ha tenido que modificar la lógica implementada en el código de Arduino, de forma que las válvulas estén en reposo, excepto cuando las jeringas procedan a aspirar y/o eyectar líquido en los procesos de purga y lavado. Antes de realizar esta modificación, el sistema aspiraba correctamente el líquido, pero no lo eyectaba en los lugares de interés.

7. Resultados

En este apartado, se muestran las imágenes de funcionamiento de la app, así como el menú lateral implementado (figura 20); dado que se ha insertado la pantalla principal en el apartado de diseño, se muestran a partir de la pantalla número dos en adelante. Durante la ejecución de la aplicación, aparecen notificaciones por pantalla al enviar un mensaje al Arduino y algunos errores de usuario al intentar acceder a una pantalla no permitida, así como otros aspectos que no son tan representativos, pero que conviene mencionar y tener en cuenta.

Figura 20. Menú lateral.

Por otro lado, con respecto a la figura 21, se trata de la pantalla nº2, en la que se pueden ver los valores correspondientes a todos los procesos que lleva a cabo el sistema robótico, de una manera genérica y predeterminada, como pueden ser los procesos de purga, lavado, análisis de fármaco y los parámetros de movimiento básico del MuROB.

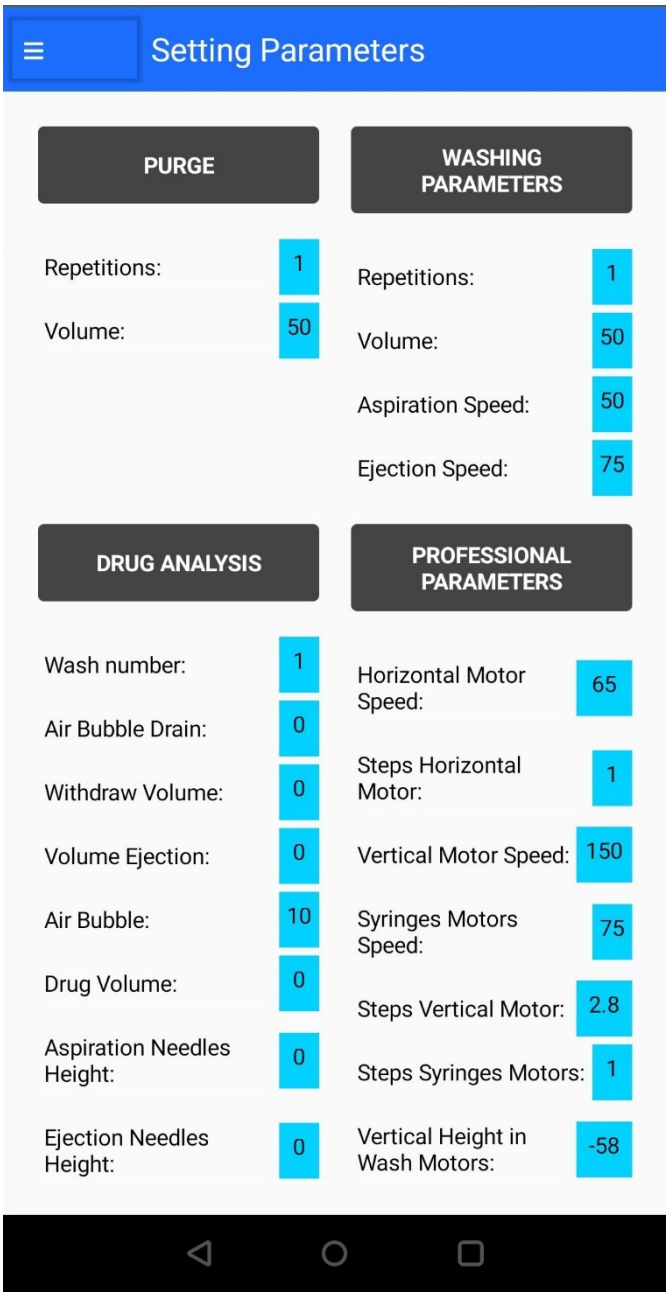

Figura 21. Pantalla número 2, vista general de los parámetros de configuración.

En relación a la figura 22, se muestra el contenido de la pantalla relativa al proceso de purgado, donde se maneja el volumen y el número de repeticiones del mismo, con el fin de eliminar las posibles burbujas de aire encontradas en las jeringas, consiguiendo que no afecten al análisis de fármaco. También se puede enviar la cadena de caracteres directamente desde esta pantalla.

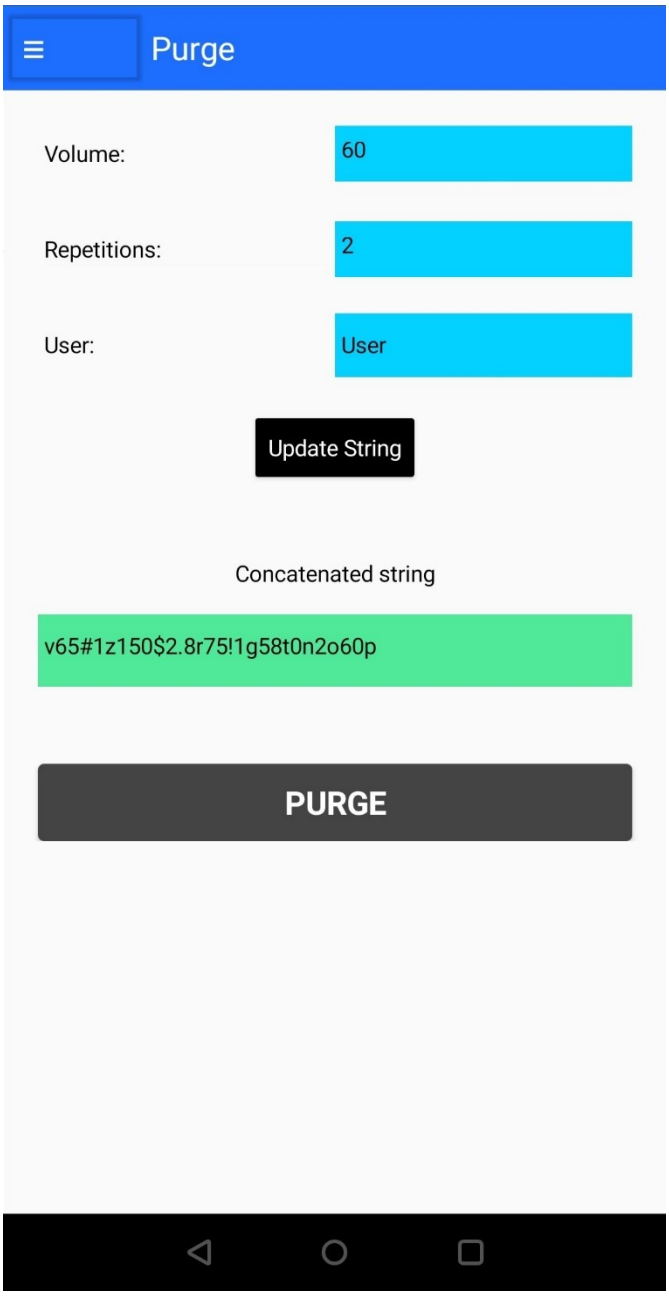

Figura 22. Pantalla número 3, proceso de purga.

A continuación, se visualizan los parámetros profesionales (figura 23), mediante los que se puede modificar directamente el comportamiento de los motores del MuROB, variando su velocidad, distancias horizontales y verticales u otros.

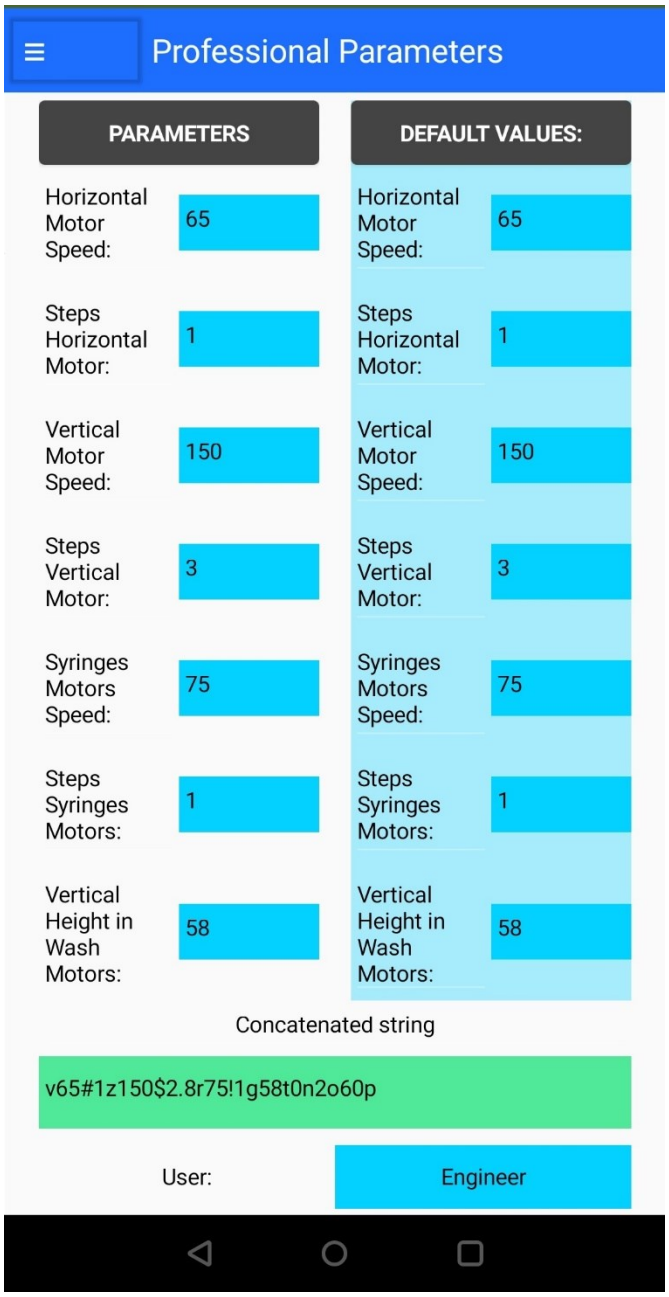

Figura 23. Pantalla número 4, parámetros descriptivos del movimiento del MuROB.

Por otro lado, la pantalla número 5 (figura 24) tiene contenido meramente informativo, y actualmente está en desuso, ya que no se están realizando movimientos manuales, por lo que el deslizador asociado al interruptor (switch) no tiene función alguna a fecha de junio 2022.

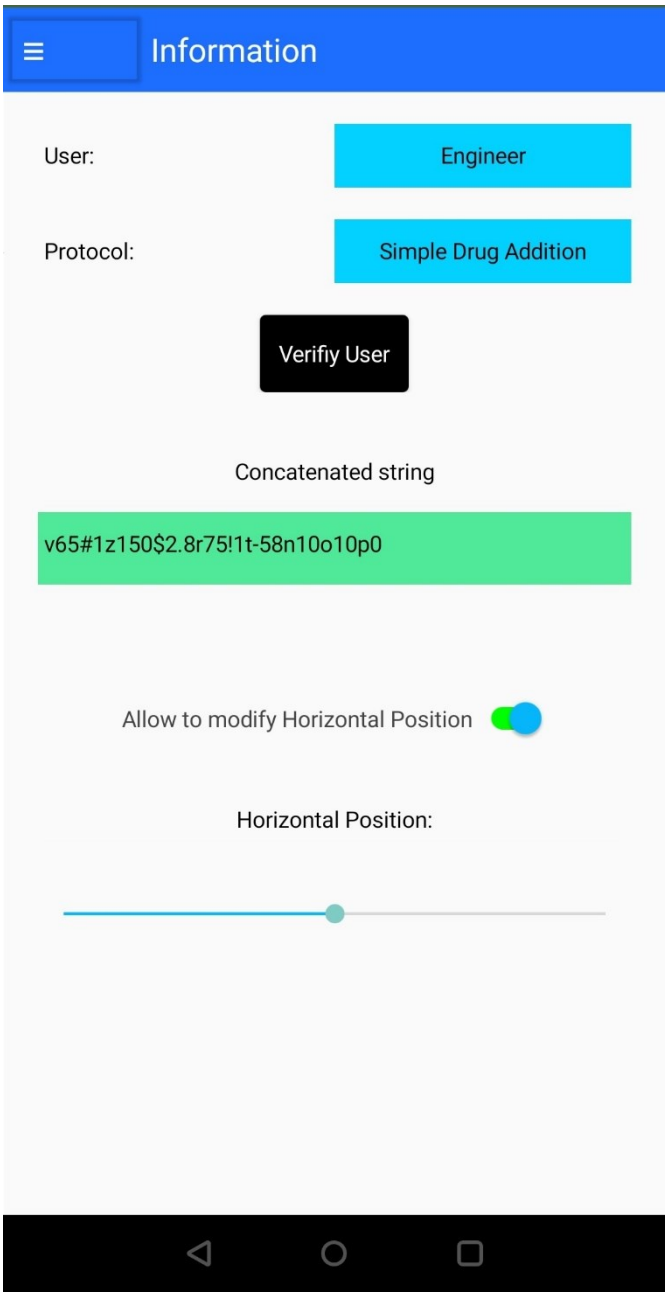

Figura 24. Pantalla número 5, información.

Acerca de la siguiente imagen, la figura 25 muestra el proceso de lavado y sus parámetros principales. Como se observa en la ilustración, se permite modificar los valores del volumen, repeticiones, velocidad de aspiración y de eyección de este proceso, habilitando al usuario o administrador el poder limpiar con total libertad el sistema, según se estime. Además, también permite el envío de la cadena de caracteres directamente desde esta pantalla, o volviendo a la pantalla principal.

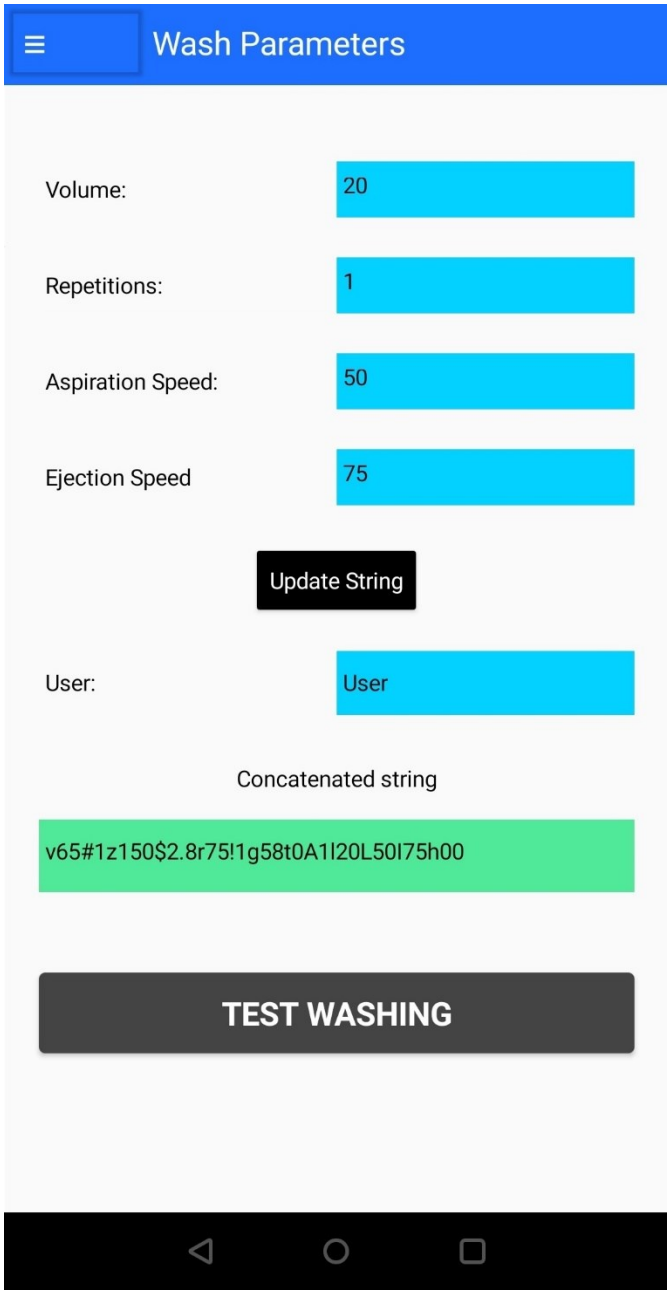

Figura 25. Pantalla número 6, lavado.

Por último, se visualiza la pantalla número 7 (figura 26), que corresponde con el proceso de análisis del fármaco, en la misma se puede seleccionar el tipo de protocolo a realizar (aspecto sin implementar por no ser un requerimiento actual, sino de futuro), variar los valores de las diferentes características del proceso (como pueden ser la posición de inicio y fin), y también se encuentra la opción de actualizar el mensaje y enviarlo al Arduino desde esta misma pantalla.

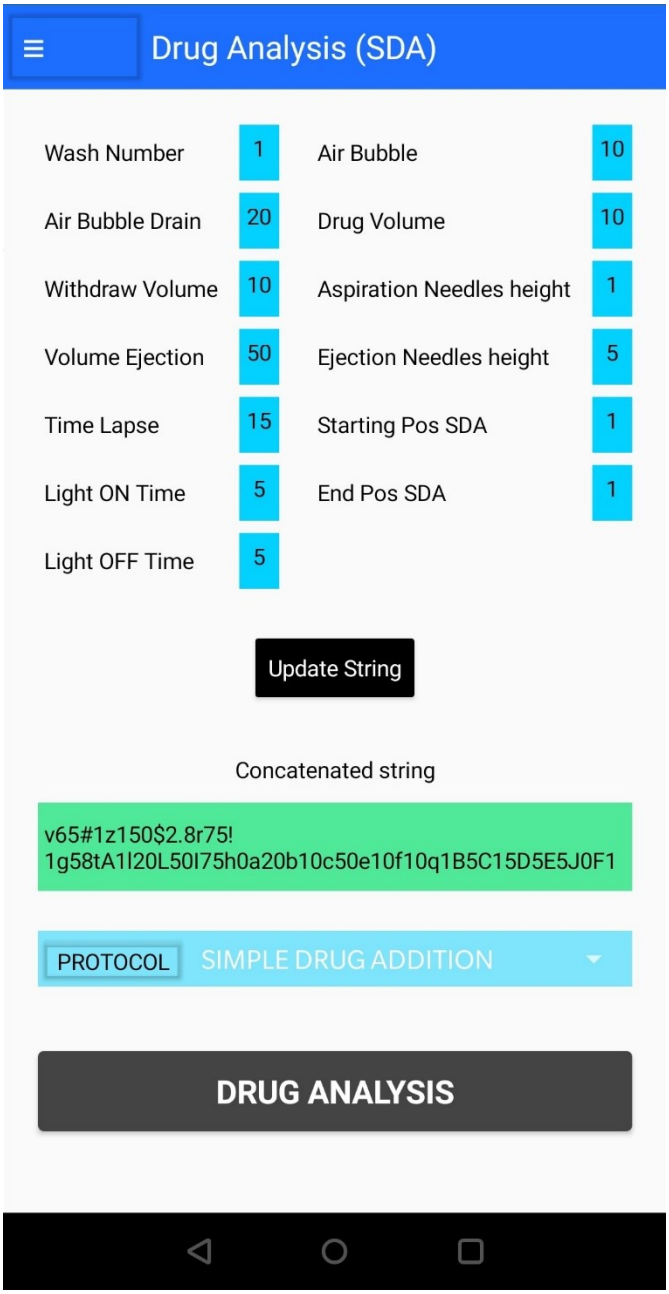

Figura 26. Pantalla número 7, análisis del fármaco.

8. Conclusiones

A modo de cierre, los resultados obtenidos han sido bastante satisfactorios, adquiriendo nociones bastante útiles para un futuro profesional y, a su vez, obteniendo conocimientos de otras ramas y tecnologías que no se conocían previamente. En relación a este aspecto, se ha dedicado bastante tiempo al desarrollo y perfeccionamiento de la app, consiguiendo una interfaz versátil e intuitiva para el usuario, así como la especialización del alumno en el diseño de aplicaciones móviles interactivas en tiempo real.

Además, el alumno ha profundizado en aspectos relacionados con la automatización industrial y la robótica, el cual era uno de sus objetivos personales, con el manejo de líquidos. De esta manera, se han empleado tecnologías que no había manejado con anterioridad y herramientas como la comunicación bluetooth y sus características. Por otro lado, el manejo de Arduino y App Inventor ha sido una grata experiencia.

En relación a la programación mediante App Inventor, se partió de una base de diseño sencilla, estudiando este entorno a partir de una serie de documentos aportados por el Dr. Caballero-Gil. Mediante estos conceptos, se creó una app básica que cumpliera con los objetivos iniciales de envío de mensajes al Arduino y que fuera intuitiva para, posteriormente, obtener un diseño de interfaz más moderno, atractivo y que cumpliera con los requerimientos del Dr. Borges y su proyecto, pudiendo manejar el MuROB desde un Smartphone o Tablet.

Finalmente, con el desarrollo de esta herramienta, se alcanzarían los objetivos futuros de vinculación con el resto de componentes del MuWOB, como son la estufa y el control de temperatura y humedad de la misma, de tal forma que la administración de fármacos estuviese sincronizada con la adquisición de imágenes.

8.1. Conclusions

To sum up, the results obtained have been quite successful, acquiring quite useful notions for a professional future and, in turn, obtaining knowledge of other branches and technologies that were not previously known. In relation to this aspect, considerable time has been devoted to the development and improvement of the app, achieving a versatile and intuitive interface for the user, as well as the specialization of the student in the design of interactive mobile applications in real time.

In addition, the student has delved into aspects related to industrial automation and robotics, which was one of his personal goals, with the handling of liquids. In this way, technologies that had not been used before and tools such as bluetooth communication and its characteristics have been used. On the other hand, handling Arduino and App Inventor has been a pleasant experience.

In relation to App Inventor programming, we started from a simple design basis, studying this environment from a series of documents provided by Dr. Caballero-Gil. Through these concepts, a basic app was created that met the initial objectives of sending messages to the Arduino and that was intuitive to later obtain a more modern, attractive interface design that met the requirements of Dr. Borges and his project, being able to manage the MuROB from a Smartphone or Tablet.

Finally, if this tool were to continue in developing, the future objectives of linking with the rest of the MuWOB components would be achieved, such as the stove and its temperature and humidity control, in such a way that the administration of drugs would be synchronized with image acquisition.

9. Referencias y bibliografía

[1] R. S. M. y. J. D. Aboud, *Investigation of the subtypes of alpha 1-adrenoceptor mediating contractions of rat aorta, vas deferens and spleen,* Br J Pharmacol, 1993.

[2] D. Díaz-Martín, J. Hernández-Jiménez, M. Rodríguez-Valido and R. Borges, *Measuring the contractile response of isolated tissue using an image sensor,* Sensors 15, 9179-9188. doi: 10.3390/s150409179., 2015.

[3] R. Borges, D. Díaz-Martín, J. Hernández-Jiménez, M. Rodríguez-Valido and B. Beltrán, *Analyzing isolated blood vessel contraction in multi-well plates,* Naunyn-Schmiedeberg's Ach. Pharmacol. 389, 521-528. doi: 10.1007/s00210-016-1218-6, 2016.

[4] R. Borges, M. Rodríguez-Valido, M. J.D., B. Beltrán, M. Montesinos, D. Díaz-Martín y Y. González Morales, «Dispositivo y procedimiento para la evaluación múltiple y simultánea de la actividad contráctil de sustancias farmacológicas mediante métodos ópticos». España. ES2390961 A1 (20.11.2012) Patente P201000640, 31 Enero 2014.

[5] D. D. Martín, *Nuevos instrumentos para la investigación en el laboratorio de Farmacología y Fisiología. Diseño e implementación.,* Universidad de La Laguna, 2017.

[6] J. Hernández Jiménez, *Nuevas herramientas para la investigación en Farmacología,* Santa Cruz de Tenerife. Universidad de La Laguna: Tesis Doctoral. Departamento de Medicina Física y Farmacología Facultad de Medicina, 2015.

[7] A. Developers. [En línea]. Available: https://developer.android.com/design. [Último acceso: mayo 2022].

[8] «Software Arduino,» Arduino, [En línea]. Available: https://www.arduino.cc/en/software.

[9] Massachusetts Institute of Technology, [En línea]. Available: http://appinventor.mit.edu/.

[10] HETPRO/TUTORIALES. [En línea]. Available: https://hetpro-store.com/TUTORIALES/bluetooth-hc-06-app-arduino/. [Último acceso: 13 marzo 2022].

[11] Prometec.net, [En línea]. Available: https://www.prometec.net/configurando-bluetooth-hc-05-y-hc-06/.

[12] «Arduino Forum,» Arduino, 9 mayo 2017. [En línea]. Available: https://forum.arduino.cc/t/comandos-at-y-comunicacion-con-modulo-bluetooth-4-0-ble-hm10/456185. [Último acceso: 12 marzo 2022].

[13] Un Electricista En La Casa Y En El Lugar De Trabajo, [En línea]. Available: https://spa.answersexpress.com/hands-review-hm-10-ble-module-10908. [Último acceso: 09 mayo 2022].

ANEXO I. Código de Arduino

```
// 1.6.21 Arduino


#include
<string.h>
#include
<Stepper.h>

const int stepsPerRevolution = 200;  // cambie este valor por el numero de pasos de su motor 360/1.8 -> Para los motores PAP, en este caso son iguales const int FCizquierdo = 26; const int FCderecho = 24;  // Finales de carrera derecho e izquierdo const int FCsube = 28; // FC motor de agujas const int opto = 31;  // Activo a nivel bajo const int fcjeringasabajo = 33; // activo a nivel bajo const int valvulas = 37;  //8 valvulas const int Vreservorio = 45; // Valvula reservorio

const int EnableH = 35; // Habilita o deshabilita el movimiento del motor por el driver const int EnableV = 47;  // Se cambia por el 41 porque se fastidio la pista const int EnableJ = 39;

boolean FCizActual;  // Valores de los Finales de Carrera boolean
FCizAnterior;

boolean FCderActual;  // Valor del FC derecho Actual boolean FCderAnterior; boolean valorFCSubida; boolean valorOpto; boolean valorFCjeringas; int cuentapasos = 0; // Variable que actúa incrementando o decrementando la cuenta de los pasos dependiendo si estos van en un sentido u otro int cuentapasosvert = 0; // Idem, para motor agujas int cuentapasosjer = 0; // Idem, motores jeringas int cuentapasosinyeccion = 0; // Contador de pasos para la inyeccion del lavado int limbajada = 0; // Limite para que no se pueda bajar mas llas agujas int LIMVERT = 0; // Limite por parametros que metemos desde Labview

String cadena; // Lee desde el USB el Labview

String numcadena [50]; // Array que almacena los datos que les llega desde el Labview int velocidadH;  // Velocidad motor horizontal desde Labview int pasos; // pasos para que se mueva el motor horizontal un orificio int pasosvert = 0; // Conteo de los pasos del motor de las agujas int pasosjeringa = 0; int pasosH;  // Parámetros que llegan
```

```
desde Labview int velocidadV; int pasosV; int velocidadJ; int pasosJ; String OK; int
posInicio = 0; // Posiciones para la lectura de la cadena que llega desde serial,
supongamos v43q , v sería posInicio y posFinal q int posFinal; // posicion final

int Manual = 0; // Rutina manual int Izquierda = 0; // Moverse manualmente a la
izquierda (Izquierda es que le llega la orden desde Labview e I es la rutina en Arduino)
int I = 0; int Derecha = 0; int D = 0; int AgujasArriba = 0; // En caso de que llegue una
"u" por serial se activará esta rutina int SubirAgujas = 0; // int AgujasAbajo = 0; int
BajarAgujas = 0; int JeringasArriba = 0; int SubirJeringas = 0; int JeringasAbajo = 0;
int BajarJeringas = 0;

int Stop = 0; // Rutina para el STOP int Npurga = 0;
// Parámetros y activación de la purga int Volpurga =
0; // Volumen purga int VolPurga = 0; int Purga = 0;

int Nrepeticiones = 0; // constante con la que se compara Npurga int
desague = 0; // Motor horizontal en posicion desague int Lavado =
0; // Parámetros y activación del lavado int Nlavado = 0; int
Microlitroslavado = 0; // Lee lo que le llega por serial int
MicrolitrosLavado = 0; // Multiplica el valor anterior por 5 int
velInyeccion = 0; int velAspiracion = 0; int NrepeticionesL = 0; int
Jerpurga = 0; // Habilita la ultima rutina de la purga int Jerlavado =
0; int Farmaco = 0; // Rutina Farmaco

int inicial = 0; // Rutina que se ejecuta el primer ciclo del arduino y las veces que se necesite
volver al estado de reposo int inicialS = 0; // Rutina inicial para agujas int inicialJ = 0; //
Rutina inicial para jeringas int inicialPurga = 0; // Rutina inicial dentro de la purga int
inicialLavado = 0; // Rutina inicial dentro del lavado int limiteInyeccion = 0; // Si llega al
optoacoplador no se puede inyectar mas int DesagueLavado = 0; // int LavadoSDA = 0;


int Air1 = 0;          // Parámetros comunes drug analysis int
withdraw = 0; int VolEject = 0; int Air2 = 0; int
drugvolume = 0;

int AltAgAspir = 0; int AltAgInyec = 0; int Aire1 = 0;
// 1 microlitro aprox 5 pasos int Withdraw = 0; int
LiquidEjection = 0; int Aire2 = 0; int DrugVolume = 0;
```

```
int TimeLapse = 0; //
Parámetros SDA int
LightsONtime = 0; int
LightsOFFtime = 0; int
StartSDA = 0; int EndSDA
= 0; int SDA1 = 0 ;

int
cuentapasosj
erSDA = 0;
int
pasosSDA =
0; int
cuentapasos
SDA = 0; int
DerSDA =
0; int IzSDA
= 0;
```

```
const int Temp = A0; // Sensor analógico de temperatura float Temperatura [100]; // Buffer que almacenará los 100 ultimos valores de temperatura int indexT; // Posición que se moverá en el bucle int ActivarTemp = 0; // Activa la lectura de la temperatura long previousMillis = 0;   // almacenará el último cambio

long interval = 400;   // intervalo de envío de temperatura(en milisegundos)
```

```
Stepper myStepper(stepsPerRevolution, 2, 3, 4, 5); // Motor Horizonal, inicialización

Stepper PAP(stepsPerRevolution, 8, 9, 6, 7); // Motor vertical de agujas Stepper
PAPS(stepsPerRevolution, 10, 11, 12, 13);  // Motor de jeringas
```

```
void setup() {   pinMode(valvulas, OUTPUT); // Declaración de
salidas del arduino   pinMode(EnableH, OUTPUT);
pinMode(EnableV, OUTPUT);   pinMode(EnableJ, OUTPUT);
pinMode(Vreservorio, OUTPUT);

  attachInterrupt(2, Emergencia, RISING); // attachInterrupt(PIN 21 CORRESPONDE A

INTERRUPCION 2 ,FUNCION QUE LLAMAMOS,MODO);

  Serial.begin(115200);


  // CONDICIONES INICIALES. Comprobamos en qué estado se encuentra cada motor
mediante los finales de carrera y ejecutamos las rutinas iniciales

  FCderActual = digitalRead(FCderecho);
valorFCSubida = digitalRead(FCsube);
valorFCjeringas =
digitalRead(fcjeringasabajo);   valorOpto =
digitalRead(opto);


  if
(FCderActual
!= HIGH) {
inicial = 1;   }

//  if (valorFCSubida != HIGH) {

//    inicialS = 1;

//  }

//

//  if (valorFCjeringas == HIGH) {

//    inicialJ = 1;
```

```
//  }
//
//  if (valorOpto == LOW) {
////    digitalWrite(EnableJ , HIGH);
//    PAPS.setSpeed(75);
//    PAPS.step(100);
//
//  }
//
//  if (FCderActual == HIGH) {
//    inicial = 1;
//    inicialS = 1;
//  }



}

// PROGRAMA DE LA EMERGENCIA. Con el botón del pánico, se parará todo y para volver a empezar debemos realizar de nuevo la comunicación entre el arduino y el Labview
void Emergencia() { // Rutina de emergencia   digitalWrite(EnableV , LOW); PAP.setSpeed(0); // velocidad de paso   PAP.step(0);
  digitalWrite(EnableJ , LOW);
  PAPS.setSpeed(0); // velocidad de paso
  PAPS.step(0);   digitalWrite(EnableH , LOW);   myStepper.setSpeed(0); //
```

```
velocidad de paso   myStepper.step(0);
inicialS = 1;   inicial = 1;   inicialJ = 1;

}



void loop() {

//   ActivarTemp = 1;       // Descomentar para calibrar la temperatura

  // Lectura de temperatura: La lectura del sensor nos da 9.05 mV por cada grado centigrado

  if (ActivarTemp == 1) {     int
valorTemperatura =
analogRead(Temp);

    float temperatura = ((valorTemperatura / 1024.0) * 5000) / 9.05;

    Temperatura [indexT] = temperatura;
indexT = ++indexT % 100;

    float Tmedia = 0;

    for (int i = 0 ; i < 100 ; i++)
Tmedia = Tmedia +
Temperatura[i] ;

    Tmedia = Tmedia / 100;  // Esto sirve para promediar la muestra de temperatura en 100
muestras     unsigned long currentMillis = millis(); // tiempo actual     if (currentMillis -
previousMillis > interval) { // si el tiempo actual - el anterior es menor que el intervalo
entra aqui       previousMillis = currentMillis;

     Serial.print ("T");

     Serial.print( Tmedia ) ;

     Serial.println("t");
```

```
  }

 }
```

```
 // ESTE PROGRAMA LEE LO QUE LE LLEGA DESDE LABVIEW COMO STRING
Y LO TRANSFORMA EN LOS PARÁMETROS QUE USAREMOS COMO PUEDEN
SER VELOCIDADES Y NUMERO DE PASOS   if (Serial.available() > 0) {     cadena =
Serial.readString();  // Leemos la cadena por serial

   int j = 0;     for (int i = 0 ; i < cadena.length() ; i++) {      // Recorremos las
posiciones de la cadena


     switch (cadena[i]) {        case 'v': // VELOCIDAD Motor HORIZONTAL
posInicio = i + 1;          break;        case '#': // PASOS MOTOR HORIZONTAL
posFinal = i;                    // La posicion final será lo que recorra el bucle
numcadena[j] = cadena.substring(posInicio, posFinal);  // Numcadena almacena en un string
lo que va despues de v, PE(v43q) pues coge el 43          velocidadH = numcadena[j].toInt();
// velocidad será el valor anterior convertido en entero          j++;          posInicio = i + 1;
break;

      case 'z': // VELOCIDAD MOTOR VERTICAL
posFinal = i;

        numcadena[j] = cadena.substring(posInicio, posFinal);
pasosH = numcadena[j].toInt();          j++;          posInicio = i + 1;
break;        case '$': // PASOS MOTOR VERTICAL
posFinal = i;          numcadena[j] = cadena.substring(posInicio,
posFinal);          velocidadV =  numcadena[j].toInt();          j++;
posInicio = i + 1;          break;

      case 'r': // VELOCIDAD JERINGAS          posFinal = i;
numcadena[j] = cadena.substring(posInicio, posFinal);
pasosV =  numcadena[j].toInt();          j++;          posInicio = i +
```

```
1;          break;        case '!': // PASOS JERINGAS          posFinal
= i;          numcadena[j] = cadena.substring(posInicio, posFinal);
velocidadJ =  numcadena[j].toInt();          j++;          posInicio = i
+ 1;          break;        case 'g': // PASOS JERINGAS
posFinal = i;

        numcadena[j] = cadena.substring(posInicio, posFinal);
pasosJ =  numcadena[j].toInt();          j++;          posInicio = i + 1;
break;        case 't':   // LIMITE VERTICAL AGUJAS ABAJO
posFinal = i;

        numcadena[j] = cadena.substring(posInicio, posFinal);

        LIMVERT  = numcadena[j].toInt();

        OK = cadena.substring(posFinal); // Este corresponde al pulsador que indica que
los parametros introducidos son correctos          break;

      case 'm':  // PARTE MANUAL QUE LLEGA DESDE LABVIEW


Manual = 1;
ActivarTem
p = 0;
break;
case 'i':
Izquierda =
1;
Derecha =
0;
break;
case 'd':
Derecha =
1;
Izquierda =
0;
break;
case 'u':
```

```
AgujasArriba
= 1;
AgujasAbajo
= 0;
break;
case 'w':


AgujasArriba
= 0;
AgujasAbajo
= 1;
break;
case 'j':


JeringasArriba =
1;
JeringasAbajo =
0;          break;
case 'k':


JeringasArriba =
0;
JeringasAbajo =
1;          break;
case 's':
Stop = 1;
break;

      case 'n': //PARAMETROS DE LA PURGA Y
ACTIVACIÓN          posInicio = i + 1;          break;
case 'o':

        posFinal = i;                    // La posicion final será lo que recorra el bucle
numcadena[j] = cadena.substring(posInicio, posFinal);  // Numcadena almacena en un string
lo que va despues de ¡, PE(A2l) pues coge el 2
```

```
Npurga = numcadena[j].toInt();  // velocidad será el valor anterior convertido en
entero j++; posInicio = i + 1; break; case 'p':

posFinal = i;

numcadena[j] = cadena.substring(posInicio, posFinal);

Volpurga = numcadena[j].toInt();

Purga = 1;

Manual = 0;

Lavado = 0;

SDA1 = 0;

ActivarTemp = 0;
Nrepeticiones = 0;
break;

case 'A': //PARÁMETROS DE LAVADO Y ACTIVACIÓN posInicio = i +
1; break; case 'l': posFinal = i; // La posicion final será lo
que recorra el bucle numcadena[j] = cadena.substring(posInicio, posFinal);  //
Numcadena almacena en un string lo que va despues de ¡, PE(A2l) pues coge el 2

Nlavado = numcadena[j].toInt();  // velocidad será el valor anterior convertido en
entero j++; posInicio = i + 1; break; case 'L': posFinal = i;
// La posicion final será lo que recorra el bucle numcadena[j] =
cadena.substring(posInicio, posFinal);  // Numcadena almacena en un string lo que va
despues de ¡, PE(l4L) pues coge el 4

Microlitroslavado = numcadena[j].toInt();  // microlitros para el lavado j++;
posInicio = i + 1; break; case 'I': posFinal = i; // La posicion
final será lo que recorra el bucle

numcadena[j] = cadena.substring(posInicio, posFinal);  // Numcadena almacena
en un string lo que va despues de ¡, PE(L3I) pues coge el 3 velAspiracion =
```

```
numcadena[j].toInt(); // velocidad será el valor anterior convertido en entero
j++;

        posInicio
= i + 1;
break;
case 'h':

        posFinal = i;          numcadena[j] =
cadena.substring(posInicio, posFinal);          velInyeccion =
numcadena[j].toInt();          MicrolitrosLavado =
Microlitroslavado * 5;


Lavado = 1;
Purga = 0;

        Manual = 0;

        ActivarTemp = 0;

        NrepeticionesL = 0;
Nrepeticiones = 0;
limiteInyeccion = 0;
break;

      case 'a':       // PARÁMETROS DRUG ANALYSIS          posInicio = i + 1;
break;       case 'b':          posFinal = i;                    // La posicion final será lo que
recorra el bucle          numcadena[j] = cadena.substring(posInicio, posFinal);  //
Numcadena almacena en un string lo que va despues de ¡, PE(a2b) pues coge el 2

        Air1 = numcadena[j].toInt();  // velocidad será el valor anterior convertido en
entero          j++;          posInicio = i + 1;          break;          case 'c':          posFinal = i;
// La posicion final será lo que recorra el bucle          numcadena[j] =
cadena.substring(posInicio, posFinal);  // Numcadena almacena en un string lo que va
despues de ¡, PE(a2b) pues coge el 2          withdraw = numcadena[j].toInt();  // velocidad
será el valor anterior convertido en entero          j++;          posInicio = i + 1;          break;
case 'e':          posFinal = i;                    // La posicion final será lo que recorra el bucle
```

```
numcadena[j] = cadena.substring(posInicio, posFinal);  // Numcadena almacena en un string lo que va despues de ¡, PE(a2b) pues coge el 2

        VolEject = numcadena[j].toInt();  // velocidad será el valor anterior convertido en entero        j++;        posInicio = i + 1;        break;

      case 'f':        posFinal = i;                  // La posicion final será lo que recorra el bucle        numcadena[j] = cadena.substring(posInicio, posFinal);  // Numcadena almacena en un string lo que va despues de ¡, PE(a2b) pues coge el 2

        Air2 = numcadena[j].toInt();  // velocidad será el valor anterior convertido en entero        j++;        posInicio = i + 1;        break;        case 'q':        posFinal = i;  // La posicion final será lo que recorra el bucle        numcadena[j] = cadena.substring(posInicio, posFinal);  // Numcadena almacena en un string lo que va despues de ¡, PE(a2b) pues coge el 2

        drugvolume = numcadena[j].toInt();  // velocidad será el valor anterior convertido en entero        j++;        posInicio = i + 1;        break;        case 'B':        posFinal = i;  // La posicion final será lo que recorra el bucle        numcadena[j] = cadena.substring(posInicio, posFinal);  // Numcadena almacena en un string lo que va despues de ¡, PE(a2b) pues coge el 2

        AltAgAspir = numcadena[j].toInt();  // velocidad será el valor anterior convertido en entero        j++;        posInicio = i + 1;        break;        case 'C':        posFinal = i;  // La posicion final será lo que recorra el bucle

        numcadena[j] = cadena.substring(posInicio, posFinal);  // Numcadena almacena en un string lo que va despues de ¡, PE(a2b) pues coge el 2

        AltAgInyec = numcadena[j].toInt();  // velocidad será el valor anterior convertido en entero        j++;        posInicio = i + 1;        break;        case 'D':        posFinal = i;  // La posicion final será lo que recorra el bucle        numcadena[j] = cadena.substring(posInicio, posFinal);  // Numcadena almacena en un string lo que va despues de ¡, PE(a2b) pues coge el 2

        TimeLapse = numcadena[j].toInt();  // velocidad será el valor anterior convertido en entero        j++;        posInicio = i + 1;
```

```
break; case 'E': posFinal = i; // La posicion final será lo que recorra el bucle numcadena[j] = cadena.substring(posInicio, posFinal); // Numcadena almacena en un string lo que va despues de ¡, PE(a2b) pues coge el 2 LightsONtime = numcadena[j].toInt(); // velocidad será el valor anterior convertido en entero j++; posInicio = i + 1; break; case 'J':

posFinal = i;

numcadena[j] = cadena.substring(posInicio, posFinal);

LightsOFFtime = numcadena[j].toInt();

Aire1 = Air1 * 5;

Withdraw = withdraw * 5;

LiquidEjection = VolEject * 5;

Aire2 = Air2 * 5;

DrugVolume = drugvolume * 5;
break;

case 'F': // PARÁMETROS SDA posInicio = i + 1; break;
case 'G': posFinal = i; // La posicion final será lo que recorra el bucle numcadena[j] = cadena.substring(posInicio, posFinal); // Numcadena almacena en un string lo que va despues de ¡, PE(a2b) pues coge el 2

StartSDA = numcadena[j].toInt(); // velocidad será el valor anterior convertido en entero j++;

posInicio
= i + 1;
break;
case 'H':

posFinal = i;

numcadena[j] = cadena.substring(posInicio, posFinal);
```

```
        EndSDA = numcadena[j].toInt();


Lavado =
1;
SDA1 = 1;

        Manual = 0;

        Purga = 0;

        Manual = 0;

//          ActivarTemp = 1;

        NrepeticionesL = 0;
Nrepeticiones = 0;
limiteInyeccion = 0;          break;


     }

   }

  }

  FCderActual = digitalRead (FCderecho); // Lectura de pulsadores y de Finales de Carrera

  FCizActual = digitalRead (FCizquierdo);
valorFCSubida = digitalRead(FCsube);
valorOpto = digitalRead(opto);
valorFCjeringas =
digitalRead(fcjeringasabajo);

  if (cuentapasos == pasos) { // Establecemos que si la cuenta de pasos es igual a los pasos
asignados, paramos el motor horizontal     digitalWrite (EnableH, LOW);

  }
```

```
 // PROGRAMA INICIAL


//

//  if ((inicialS == 1) && (valorFCSubida != HIGH) && (Purga == 0) && (Lavado ==

0) && (SDA1 == 0)) { // Ponemos en el cero el motor de las agujas

////       Serial.println("Entra en 1");

//    digitalWrite(EnableV , HIGH);

//    PAP.setSpeed(150);

//    PAP.step(4);

//  }

  if ((inicial == 1) && (FCderActual != HIGH) && (Purga == 0) && (Lavado == 0)) { //
Ponemos el motor horizontal en el desague //       Serial.println("Entra en 2");
digitalWrite(EnableH , HIGH);     myStepper.setSpeed(55);     myStepper.step(1);

  }

  if (FCderActual == 1 && FCderAnterior == 0 && inicial == 1 && DerSDA == 0 ) {

// Condicion para que el cero sea el desague //       Serial.println("Entra
en 3");     digitalWrite(EnableH , LOW);     myStepper.setSpeed(0); //
velocidad de paso     myStepper.step(0);// numero de pasos y dirección
(- izquierda + derecha)       // si llega al fc derecho paramos motor*/

    Derecha = 0;

    Izquierda = 0;     D = 0;     delay(3);
digitalWrite(EnableH , HIGH);     myStepper.setSpeed(55);
// Damos un paso a la izquierda     myStepper.step(-671);
inicial = 0;     desague = 1;

  }
```

```
//  if ((inicialJ == 1) && (valorFCjeringas != HIGH) && (inicial == 0) && (Purga
== 0) && (Lavado == 0)) { // Ponemos los motores de las jeringas en el cero
////        Serial.println("Entra en 4");
//    digitalWrite(EnableJ , HIGH);
//    PAPS.setSpeed(70);
//    PAPS.step(1);
//  }


  // CONDICION DE STOP, PARA TODO Y VUELVE A
CONDICIONES INICIALES   if (Stop == 1) {
//      Serial.println("Entra en
5");     inicial = 1;     inicialS
= 1;     inicialJ = 1;
Lavado = 0;
    Manual = 0;
    Purga = 0;
    SDA1 = 0;
  }
  Stop = 0;


  // CONTROL MANUAL   if
((Manual == 1) && (inicial ==
0)) {
```

```
//      Serial.println("Entra en 6");

   Purga = 0;     Lavado = 0;     if (Izquierda == 1 && ((FCderActual != HIGH) ||
(FCizActual != HIGH ))) {

//          Serial.println("Entra en 7");

     I = 1;       D = 0;       pasos = -50;   // - Izquierda + derecha maximo 1316 , 50 pasos entre
huecos       cuentapasos = 0;

   }

   Izquierda = 0;

   if ((I == 1) && (cuentapasos > pasos) && (FCizActual != HIGH)) {

//      Serial.println("Entra en 8");       digitalWrite(EnableH , HIGH);
myStepper.setSpeed(velocidadH); // velocidad de paso  myStepper.

setSpeed(velocidadH);       myStepper.step(-pasosH);   // numero de pasos y dirección (-
izquierda + derecha)   myStepper.step(-pasosH);       cuentapasos = cuentapasos - 1;
desague = 0;

   }

   if (Derecha == 1 && (FCderActual != HIGH)) {

//          Serial.println("Entra en 9");

     D = 1;       I = 0;       pasos = 50;  // Izquierda +
Derecha maximo 1316       cuentapasos = 0 ;

   }

   Derecha = 0;

   if ((D == 1) && (cuentapasos < pasos) && (FCderActual != HIGH)) {

//      Serial.println("Entra en 10");       digitalWrite(EnableH , HIGH);
myStepper.setSpeed(velocidadH); // velocidad de paso  myStepper.
```

```
setSpeed(velocidadH);       myStepper.step(pasosH);   // numero de pasos y dirección (-
izquierda + derecha)  myStepper.step(pasosH);       cuentapasos = cuentapasos + 1;
desague = 0;

   }



   if ((AgujasArriba == 1) && (valorFCSubida != HIGH) ) {

//          Serial.println("Entra en 11");
SubirAgujas = 1;

     pasosvert = 100;    // numero de pasos (parametro) + arriba - abajo // Maximo 61
pasos       cuentapasosvert = 0;

     AgujasArriba = 0;

   }

   if ((SubirAgujas == 1) && (cuentapasosvert < pasosvert) && valorFCSubida != HIGH) {

//          Serial.println("Entra en
12");       digitalWrite(EnableV ,
HIGH);
PAP.setSpeed(velocidadV); //
velocidad de paso
PAP.setSpeed(velocidadV);
PAP.step(pasosV);   // numero de
pasos y dirección (- abajo + arriba)
PAP. step(pasosV);
cuentapasosvert = cuentapasosvert
+ 1;

   }

   if (AgujasAbajo == 1 ) {

//          Serial.println("Entra en 13");       BajarAgujas = 1;       pasosvert = -100;
// numero de pasos (parametro) + arriba - abajo       cuentapasosvert = 0;
```

```
    AgujasAbajo = 0;

  }

  if ((BajarAgujas == 1) && (limbajada == 0) && (cuentapasosvert > pasosvert)&&
(Lavado == 0)) {

//        Serial.println("Entra en 14");       digitalWrite(EnableV ,
HIGH);

    PAP.setSpeed(velocidadV); // velocidad de paso   PAP.setSpeed(velocidadV);
PAP.step(-pasosV);   // numero de pasos y dirección (- abajo + arriba)  PAP.

step(-pasosV);

    cuentapasosvert = cuentapasosvert - 1;

  }

  if ((cuentapasosvert == -LIMVERT)&& Lavado == 0) {

//        Serial.println("Entra en 15");
BajarAgujas = 0;

    digitalWrite(EnableV , LOW);

    PAP.setSpeed(0); // velocidad de paso

    PAP.step(0);// numero de pasos y dirección (- abajo + arriba)       limbajada = 1; //
Si esto está activo no baja mas

  }


  if (JeringasArriba == 1 && valorOpto == HIGH ) {

//        Serial.println("Entra en 16");       SubirJeringas = 1;
pasosjeringa = 20;  // pasos jeringas + arriba - abajo
cuentapasosjer = 0;       JeringasArriba = 0;
digitalWrite(EnableJ , HIGH);
```

```
    }

    if ((SubirJeringas == 1) && (cuentapasosjer < pasosjeringa) && valorOpto == HIGH) {

//          Serial.println("Entra en 17");

      PAPS.setSpeed(velocidadJ); // velocidad de paso   PAPS.setSpeed(velocidadJ);

      PAPS.step(-pasosJ);   // numero de pasos y dirección (- arriba + abajo)

PAPS.step(-pasosJ);

      cuentapasosjer = cuentapasosjer + 1;

    }

    if (JeringasAbajo == 1 && valorFCjeringas == LOW ) {

//          Serial.println("Entra en 18");       BajarJeringas = 1;
pasosjeringa = -20;  // pasos jeringas + arriba - abajo
cuentapasosjer = 0;       JeringasAbajo = 0;
digitalWrite(EnableJ , HIGH);

    }

    if ((BajarJeringas == 1) && (cuentapasosjer > pasosjeringa) && valorFCjeringas ==
LOW) {

//          Serial.println("Entra en 19");

      PAPS.setSpeed(velocidadJ); // velocidad de paso   PAPS.setSpeed(velocidadJ);

      PAPS.step(pasosJ);   // numero de pasos y dirección (- arriba + abajo)
PAPS.step(pasosJ);       cuentapasosjer = cuentapasosjer - 1;     }


  }
```

```
 // CONDICIONES DE LOS FINALES DE
CARRERA   if (FCizActual == 1 && FCizAnterior ==
0 && SDA1==0) {

//    Serial.println("Entra en 20");

   Izquierda = 0;     I = 0;     digitalWrite(EnableH , LOW);
myStepper.setSpeed(0); // velocidad de paso     myStepper.step(0);//
numero de pasos y dirección (- izquierda + derecha)       // si llega al fc
izquierdo paramos motor*/     delay(3);     digitalWrite(EnableH ,
HIGH);     myStepper.setSpeed(25); // Damos un paso a la izquierda
myStepper.step(20);

 }

 FCizAnterior = FCizActual;

 if (FCderActual == 1 && FCderAnterior == 0 && inicial == 0 && SDA1 == 0 &&

DerSDA == 0 && Purga == 0 && Lavado == 0&& Farmaco!=7) {

//    Serial.println("Entra en 21");     digitalWrite(EnableH , LOW);
myStepper.setSpeed(0); // velocidad de paso     myStepper.step(0);//
numero de pasos y dirección (- izquierda + derecha)       // si llega al fc
derecho paramos motor*/

   Derecha = 0;

   Izquierda = 0;     D = 0;     delay(3);
digitalWrite(EnableH , HIGH);     myStepper.setSpeed(25);
// Damos un paso a la izquierda     myStepper.step(-2);   }

 FCderAnterior = FCderActual;



 if (valorFCSubida == HIGH && Farmaco !=4  ) {      // Si toca el FC de subida, paramos el
motor y luego damos un pequeño paso para separarlo
```

```
//        Serial.println("Entra en 22");     digitalWrite(EnableV ,
LOW);

   PAP.setSpeed(0); // velocidad de paso

   PAP.step(0);// numero de pasos y dirección (- abajo + arriba)
delay(3);     inicialS = 0;     limbajada = 0;     SubirAgujas = 0;

   digitalWrite(EnableV , HIGH);

   PAP.setSpeed(150);
PAP.step(-10);
digitalWrite(EnableV ,
LOW);

  }


  if (valorOpto == LOW) {  // Si toca el optoacoplador, se establece el limite de inyección
para que no se pueda inyectar más y se separa del optoacoplador un paso para que no lo
vuelva a detectar     limiteInyeccion = 1;

//        Serial.println("Entra en
23");     SubirJeringas = 0;
digitalWrite(EnableJ , LOW);

   PAPS.setSpeed(0); // velocidad
de paso     PAPS.step(0);
delay(3);

   digitalWrite(EnableJ , HIGH);

   PAPS.setSpeed(75); // velocidad de paso

   PAPS.step(1);   // numero de pasos y dirección (+ abajo - arriba)

  }
```

```
  if ((valorFCjeringas == HIGH) && (limbajada == 0) && (Jerpurga == 0)) { // Si toca el FC de las jeringas abajo, hacemos que el motor suba 330 pasos para que quede inicializado     limiteInyeccion = 0;

//        Serial.println("Entra en 24");

    digitalWrite(EnableJ , LOW);

    PAPS.setSpeed(0); // velocidad de paso     PAPS.step(0);     inicialJ = 0;     BajarJeringas = 0;
delay(3);

    digitalWrite(EnableJ , HIGH);

    PAPS.setSpeed(75); // velocidad de paso

    PAPS.step(-330);   // numero de pasos y dirección (+ abajo - arriba)
digitalWrite(EnableJ , LOW);

  }



 //v65#1z150$3r75!1g58t0n2o60p

  // CONDICIONES DE LA PURGA. En la posición de desague, bajar las agujas y realizar las purgas que el usuario desee por parámetro. Para cada purga, se aspira al tope(final de carrera abajo)

  // por parte de las jeringas, luego sube 660 pasos, la secuencia se repite las veces que se han solicitado
```

```
  if ((Purga == 1) && (desague == 0) && (inicialS == 0) && (inicial == 0)) { // Si se
detecta que el motor horizontal no está en el desague, se coloca en el mismo y empieza la
rutina de la purga //        Serial.println("Entra en 25");     inicialPurga = 1;

  }   if
(inicialPurg
a == 1) {

//          Serial.println("Entra en
26");     inicial = 1;     inicialS = 1;

  }

  if ((Purga == 1) && (inicialS == 1) && (valorFCSubida != HIGH) && (desague ==

0)) { // Ponemos en el cero el motor de las
agujas //          Serial.println("Entra en 27");
inicialPurga = 0;

    digitalWrite(EnableV , HIGH);

    PAP.setSpeed(velocidadV);

    PAP.step(pasosV);

  }

  if ((Purga == 1) && (valorFCSubida ==
HIGH)) { //          Serial.println("Entra en
28");     digitalWrite(EnableV , LOW);
PAP.setSpeed(0); // velocidad de paso

    PAP.step(0);// numero de pasos y dirección (- abajo + arriba)
inicialS = 0;     limbajada = 0;     SubirAgujas = 0;     delay(3);

    digitalWrite(EnableV , HIGH);
```

```
   PAP.setSpeed(velocidadV);
PAP.step(-pasosV);
digitalWrite(EnableV , LOW);

  }

  if ((Purga == 1) && (inicial == 1) && (inicialS == 0) && (FCderActual != HIGH))

{ // Ponemos el motor horizontal en el desague

//    Serial.println("Entra en 29");
digitalWrite(EnableH , HIGH);
myStepper.setSpeed(velocidadH);
myStepper.step(pasosH);

  }

  if ((Purga == 1) && (FCderActual == 1 && FCderAnterior == 0 && inicial == 1 &&

DerSDA == 0 )) { // Condicion para que el cero sea el desague

//    Serial.println("Entra en 30");     digitalWrite(EnableH , LOW);
myStepper.setSpeed(0); // velocidad de paso     myStepper.step(0);//
numero de pasos y dirección (- izquierda + derecha)       // si llega al fc
derecho paramos motor*/

   Derecha = 0;

   Izquierda = 0;     D = 0;     delay(3);     digitalWrite(EnableH , HIGH);
myStepper.setSpeed(velocidadH); // Damos un paso a la izquierda
myStepper.step(-650);     inicial = 0;     desague = 1;     inicialPurga = 0;

  }



  if ((Purga == 1) && (desague ==
1)) { //       Serial.println("Entra en
31");
```

```
   if ((Nrepeticiones == Npurga) && (limbajada == 0)) { // Si las repeticiones se han hecho
se para la purga a espera de otra instrucción

//          Serial.println("Entra en 32");

//      ActivarTemp = 1;
cuentapasos = 0;
cuentapasosvert = 0;
Jerpurga = 0;       inicialS
= 0;       inicialJ = 0;
inicial = 0;
digitalWrite(Vreservorio,
LOW);
digitalWrite(valvulas,
LOW);

   }

   if ((Nrepeticiones == Npurga) && (limbajada == 1)) { // Condición para subir las aguja
una vez que ha terminado la purga       if (valorFCSubida != HIGH) {
digitalWrite(EnableV , HIGH);

      PAP.setSpeed(velocidadV);

      PAP.step(pasosV);

     }

     if (valorFCSubida ==
HIGH) {
digitalWrite(EnableV ,
LOW);        limbajada = 0;
}    }


   if ((Nrepeticiones < Npurga) && (desague == 1) && (BajarAgujas == 0) &&

(limbajada == 0)&&(Lavado==0)) {
```

```
//          Serial.println("Entra en 33");       BajarAgujas = 1;       inicial = 0;
inicialS = 0;       inicialJ = 0;       pasosvert = -100;    // numero de pasos
(parametro) + arriba - abajo       cuentapasosvert = 0;

   }

   if ((BajarAgujas == 1) && (limbajada == 0) && (cuentapasosvert > pasosvert)

&& (Nrepeticiones <
Npurga)&&(Lavado==0)) { //
Serial.println("Entra en 34");
digitalWrite(EnableJ , LOW);

     PAPS.setSpeed(0);
PAPS.step(0);

     digitalWrite(EnableV , HIGH);

     PAP.setSpeed(velocidadV); // velocidad de paso  PAP.setSpeed(velocidadV);
PAP.step(-pasosV);   // numero de pasos y dirección (- abajo + arriba)  PAP. step(-pasosV);
cuentapasosvert = cuentapasosvert - 1;

   }

   if ((cuentapasosvert == -LIMVERT) && (limbajada == 0) && (Nrepeticiones < Npurga))
{

//          Serial.println("Entra en 35");       BajarAgujas
= 0;

     digitalWrite(EnableV , LOW);

     PAP.setSpeed(0); // velocidad de paso

     PAP.step(0);// numero de pasos y dirección (- abajo + arriba)       limbajada = 1; //
Si esto está activo no baja mas

   }
```

```
   if ((limbajada == 1) && (Jerpurga == 0) && (Nrepeticiones < Npurga)) {

//          Serial.println("Entra en 36");
digitalWrite(valvulas, LOW);
digitalWrite(Vreservorio, HIGH);
digitalWrite(EnableV, LOW);
delay(4);

     digitalWrite(EnableJ , HIGH);

     PAPS.setSpeed(velocidadJ); // PAPS.setSpeed(velocidadJ);

     PAPS.step(pasosJ); // Motores jeringas ( - arriba + abajo ) PAPS.

step(pasosJ);

   }

   if ((valorFCjeringas == HIGH) && (limbajada == 1) && (Nrepeticiones <

Npurga)&&(Jerpurga==0)) {

//     Serial.println("Entra en
37");       Jerpurga = 1;
cuentapasosjer =0;
VolPurga = Volpurga * 5;
digitalWrite(valvulas,
HIGH);
digitalWrite(Vreservorio,
LOW);       delay(10);
digitalWrite(EnableJ ,
HIGH);

   }

   if ((Jerpurga == 1) && (Nrepeticiones < Npurga)&&(cuentapasosjer < VolPurga) ) {

//      Serial.println("Entra en 38");
```

```
    PAPS.setSpeed(velocidadJ);
PAPS.step(-pasosJ);
cuentapasosjer++;

  }


  if ((Jerpurga == 1) && (Nrepeticiones < Npurga)&&(cuentapasosjer == VolPurga) ) {

//        Serial.println("Entra en 39");

    Nrepeticiones = Nrepeticiones + 1;

    Jerpurga = 0;
inicialS = 0;      inicialJ
= 0;      inicial = 0;
cuentapasosjer = 0;
digitalWrite(EnableJ ,
LOW);

  }




 }


 // v65#1z150$3r75!1g58t0A2l50L100I75h

 // CONDICIONES DEL LAVADO. Se coloca inicialmente en el desague como en la
purga. El proceso empieza en el desague, bajamos las agujas. Aspiramos con el valor
de un parámetro e inyectamos sobre ese

 // mismo valor en microlitros, pudiendo modificar la velocidad de inyeccion y de aspiracion


 if ((Lavado == 1) && (desague == 0) && (inicialS == 0) && (inicial == 0)) {
```

```
//    Serial.println("Entra en 39");
inicialLavado = 1;

  }

  if (inicialLavado == 1) {

//    Serial.println("Entra en
40");     inicial = 1;
inicialS = 1;

  }

  if ((Lavado == 1) && (inicialS == 1) && (valorFCSubida != HIGH) && (desague ==

0)) { // Ponemos en el cero el motor de las agujas

//    Serial.println("Entra en
41");     inicialLavado = 0;
digitalWrite(EnableV ,
HIGH);

    PAP.setSpeed(velocidadV);

    PAP.step(pasosV);

  }

  if ((Lavado == 1) && (valorFCSubida == HIGH) && (desague == 0)) {

//    Serial.println("Entra en
42");
digitalWrite(EnableV ,
LOW);     limbajada = 0;

//    inicialS = 0;

  }
```

```
  if ((Lavado == 1) && (inicial == 1) && (inicialS == 0) && (FCderActual != HIGH)
&& (desague == 0)) { // Ponemos el motor horizontal en el desague
//    Serial.println(DerSDA);
digitalWrite(EnableH , HIGH);
myStepper.setSpeed(velocidadH);
myStepper.step(pasosH);
  }
  if ((Lavado == 1) && (FCderActual == 1 && FCderAnterior == 0 && inicial == 1 &&
DerSDA == 0 && DesagueLavado==0)) { // Condicion para que el cero sea el desague
//    Serial.println("Entra en 44");     digitalWrite(EnableH , LOW);
myStepper.setSpeed(0); // velocidad de paso     myStepper.step(0);//
numero de pasos y dirección (- izquierda + derecha)       // si llega al fc
derecho paramos motor*/
    Derecha = 0;
    Izquierda = 0;
    D = 0;
    DesagueLavado = 1;
  }
  if((Lavado == 1) && (FCderActual == 1 && FCderAnterior == 0 && inicial == 1 &&
DesagueLavado==1 )){
//    Serial.println("Entra en 45");     digitalWrite(EnableH , HIGH);
myStepper.setSpeed(velocidadH); // Damos un paso a la izquierda
myStepper.step(-671);     inicial = 0;     desague = 1;     inicialLavado =
0;     DesagueLavado = 0;
  }
```

```
  if ((Lavado == 1) && (desague == 1) && (Stop == 0)) {


    if ((NrepeticionesL == Nlavado) && (limbajada == 0)) { // Si las repeticiones se han
hecho se reinicia el lavado //      Serial.println("Entra en 45");       Jerlavado = 0;
cuentapasosjer = 0;       cuentapasosvert = 0;       cuentapasosinyeccion = 0;

      inicialS = 0;       inicialJ = 0;       inicial = 0; //      ActivarTemp = 1;
digitalWrite(EnableV , LOW);       digitalWrite(EnableH , LOW);
digitalWrite(EnableJ , LOW);       digitalWrite (Vreservorio , LOW);       digitalWrite
(valvulas , LOW);       if (SDA1 == 1) {  // Condición para que acabe el lavado y
empiece el SDA

//        Serial.println("Entra en
45");         Lavado = 0;
cuentapasosjer = 0;
Nrepeticiones = 0;
NrepeticionesL = 0;
cuentapasosjerSDA = 0;
DerSDA = 0;
cuentapasosvert = 0;



}
}


    if ((NrepeticionesL == Nlavado) && (limbajada == 1)) { // Si las repeticiones se han
hecho se reinicia el lavado //        Serial.println("Entra en 46");       if (valorFCSubida !=
HIGH) { //        Serial.println("Entra en 47");         digitalWrite(EnableV , HIGH);

       PAP.setSpeed(velocidadV);

       PAP.step(pasosV);

      }


      if (valorFCSubida == HIGH)
{ //        Serial.println("Entra en
48");         digitalWrite(EnableV
, LOW);         limbajada = 0;
```

```
}
}

   if ((NrepeticionesL < Nlavado) && (desague == 0)) { // hacemos que el motor vaya al
desague

//      Serial.println("Entra en
49");       inicial = 1;
inicialS = 1;

   }

   if ((NrepeticionesL < Nlavado) && (desague == 1) && (BajarAgujas == 0) &&

(limbajada == 0) && (Lavado == 1)) {

//      Serial.println("Entra en 50");

     BajarAgujas = 1;       inicial = 0;       inicialS = 0;       inicialJ = 0;       pasosvert
= -100;    // numero de pasos (parametro) + arriba - abajo       cuentapasosvert = 0;
digitalWrite (valvulas , LOW);       digitalWrite (Vreservorio , LOW);
digitalWrite(EnableJ , LOW);

   }

   if ((BajarAgujas == 1) && (limbajada == 0) && (cuentapasosvert > pasosvert) &&
(Lavado == 1)) {

//      Serial.println("Entra en 51");

//      delay(4);

     digitalWrite(EnableV , HIGH);

     PAP.setSpeed(velocidadV); // velocidad de paso PAP.setSpeed(225);
PAP.step(-pasosV);   // numero de pasos y dirección (- abajo + arriba)
cuentapasosvert = cuentapasosvert - 1;
```

```
}
/
/
   if ((cuentapasosvert == -LIMVERT) && (limbajada == 0) && (Lavado == 1)) {
//      Serial.println("Entra en 52");
BajarAgujas = 0;
     digitalWrite(EnableV , LOW);
     PAP.setSpeed(0); // velocidad de paso
     PAP.step(0);// numero de pasos y dirección (- abajo + arriba)       limbajada = 1; //
Si esto está activo no baja mas
   }
   if ((limbajada == 1) && (cuentapasosjer < MicrolitrosLavado) && (Jerlavado ==
0) && (NrepeticionesL < Nlavado) && (Lavado == 1)) {
//      Serial.println("Entra en 53");
digitalWrite(valvulas, LOW);
digitalWrite (Vreservorio , HIGH);
//      delay(6);
     digitalWrite(EnableJ , HIGH);
PAPS.setSpeed(velAspiracion);
     PAPS.step(1); // Motores jeringas ( - arriba + abajo )       cuentapasosjer =
cuentapasosjer + 1;
   }
   if ((cuentapasosjer == MicrolitrosLavado) && (Nrepeticiones < Nlavado) && (Lavado
== 1)) {
```

```
//      Serial.println("Entra en 54");       Jerlavado = 1;       if (limiteInyeccion == 1) {
// Si llegó al limite de inyeccion que no inyecte más

       Nrepeticiones = Nlavado;

       Jerlavado = 0;
digitalWrite(EnableJ , LOW);



}
}

   if ((cuentapasosjer == MicrolitrosLavado) && (cuentapasosinyeccion <

MicrolitrosLavado) && (Jerlavado == 1) && (limiteInyeccion == 0)) {

//      Serial.println("Entra en 55");
digitalWrite(valvulas, HIGH);
digitalWrite (Vreservorio , LOW);

//      delay(4);

     digitalWrite(EnableJ , HIGH);

     PAPS.setSpeed(velInyeccion);

     PAPS.step(-1); // Motores jeringas ( - arriba + abajo )       cuentapasosinyeccion =
cuentapasosinyeccion + 1;

   }

   if (((cuentapasosinyeccion == MicrolitrosLavado) || (limiteInyeccion == 1))

&& (cuentapasosinyeccion != 0)) {

//      Serial.println("Entra en 56");

     Jerlavado = 2;

   }
```

```
   if (Jerlavado == 2) {

//      Serial.println("Entra en 57");       digitalWrite(EnableJ
, LOW);

     PAPS.setSpeed(0);

     PAPS.step(0); // Motores jeringas ( - arriba + abajo )
NrepeticionesL = NrepeticionesL + 1;
cuentapasosinyeccion = 0;       cuentapasosjer = 0;

     Jerlavado = 0;

   }


  }


  //v65#1z150$3r75!

1g58t0A2l20L50I75h0a20b10c50e20f10q1B5C15D15E15J0F4G6H

//v65#1z150$3r75!1g58t0A1l20L50I75h0a20b10c50e20f10q1B5C15D5E5J0F1G2H

  // CONDICIONES SIMPLE DRUG ADDITION (SDA)

  if ((SDA1 == 1) && (Lavado == 0) && (StartSDA < EndSDA + 1 )) {

//    ActivarTemp = 1;              // Descomentar para la lectura de temperatura en el simple
drug addition

   if ((cuentapasosjerSDA < Aire1) && (Lavado == 0) && (IzSDA == 0) &&

(cuentapasosjer != Withdraw) && (Farmaco == 0)) {

//      Serial.println("Entra en
59");       digitalWrite(valvulas ,
HIGH);
digitalWrite(Vreservorio ,
```

```
LOW); //
digitalWrite(EnableJ, HIGH);

    PAPS.setSpeed(velAspiracion);

    PAPS.step(1); // Motores jeringas ( - arriba + abajo )      cuentapasosjerSDA =
cuentapasosjerSDA + 1;

  }

  if ((cuentapasosjerSDA == Aire1) && (IzSDA == 0) && (Farmaco == 0)) {

//     Serial.println("Entra en 60");      cuentapasosjerSDA = 0;
digitalWrite(valvulas , LOW);      digitalWrite(EnableJ, LOW);      pasosSDA =
((-StartSDA * 50) - 80);  // Nos movemos a target plate( izquierda)  pasosSDA =
((-StartSDA*50)-50);      cuentapasosSDA = 0;


DerSD
A = 0;
IzSDA
= 1;
delay(
5);

  }

  if ((IzSDA == 1) && (cuentapasosSDA > pasosSDA) && (FCderActual != HIGH) &&
(Farmaco == 0)) {

//     Serial.println("Entra en 61");      desague = 0;      digitalWrite(EnableH , HIGH);
myStepper.setSpeed(velocidadH); // velocidad de paso      myStepper.step(-pasosH);   //
numero de pasos y dirección (- izquierda + derecha)      cuentapasosSDA =
cuentapasosSDA - 1;

  }

  if ((IzSDA == 1) && (cuentapasosSDA == pasosSDA) && (cuentapasosjer != Withdraw)
&& (limbajada == 0) && (cuentapasosvert > pasosvert) && (Farmaco == 0)) {
```

```
//      Serial.println("Entra en 62");
digitalWrite(EnableH , LOW);
digitalWrite(EnableV , HIGH);

    PAP.setSpeed(velocidadV); // velocidad de paso PAP.setSpeed(225);
PAP.step(-pasosV);   // numero de pasos y dirección (- abajo + arriba)
cuentapasosvert = cuentapasosvert - 1;

   }

   if ((cuentapasosvert == -LIMVERT) && (limbajada == 0) && (SDA1 == 1) &&

(cuentapasosjer != Withdraw) && (Farmaco == 0)) {

//      Serial.println("Entra en 63");       digitalWrite(EnableV
, LOW);

    PAP.setSpeed(0); // velocidad de paso

    PAP.step(0);// numero de pasos y dirección (- abajo + arriba)       limbajada = 1; //
Si esto está activo no baja mas

   }


   if ((limbajada == 1) && (cuentapasosjer < Withdraw) && (Farmaco == 0)) {

//      Serial.println("Entra en 64");
digitalWrite(valvulas, HIGH);
digitalWrite(Vreservorio , LOW);

//      delay(5);

    digitalWrite(EnableJ , HIGH);
PAPS.setSpeed(velAspiracion);

    PAPS.step(1); // Motores jeringas ( - arriba + abajo )       cuentapasosjer =
cuentapasosjer + 1;

   }
```

```
   if ((cuentapasosjer == Withdraw) && (desague == 0) && (limbajada == 1) && (Farmaco
== 0)) {

//      Serial.println("Entra en
65");
digitalWrite(valvulas , LOW);
digitalWrite(EnableJ, LOW);
//      digitalWrite(EnableV,
HIGH);

     PAP.setSpeed(velocidadV); // velocidad de paso PAP.setSpeed(225);

     PAP.step(pasosV);   // numero de pasos y dirección (- abajo + arriba)

   }

   if ((cuentapasosjer == Withdraw) && (valorFCSubida == HIGH) && (desague == 0) &&
(Farmaco == 0)) {

//      Serial.println("Entra en 66");
limbajada = 0;


digitalWrite(EnableV,
LOW);       Farmaco =
1;      delay(5);

   }

   if (Farmaco == 1 && desague == 0 && DerSDA == 0) {

//      Serial.println("Entra en 67");
digitalWrite(EnableH, HIGH);
myStepper.setSpeed(velocidadH);
myStepper.step(pasosH);

   }

   if (FCderActual == 1 && DerSDA == 0 && (Farmaco == 1)) { // Condicion para que el
cero sea el desague
```

```
//      Serial.println("Entra en 68");       digitalWrite(EnableH , LOW);
myStepper.setSpeed(0); // velocidad de paso       myStepper.step(0);//
numero de pasos y dirección (- izquierda + derecha)       // si llega al fc
derecho paramos motor*/       delay(7);       digitalWrite(EnableH , HIGH);
myStepper.setSpeed(70); // Damos un paso a la izquierda
myStepper.step(-671);       desague = 1;       cuentapasosvert = 0;

   }

   if (desague == 1 && limbajada == 0 && (cuentapasosvert > -LIMVERT) && Farmaco

== 1 && cuentapasosjer !=
LiquidEjection) { //
Serial.println("Entra en 69");
digitalWrite(EnableH, LOW);
digitalWrite(EnableV , HIGH);

    PAP.setSpeed(velocidadV);       PAP.step(-
pasosV);

    cuentapasosvert = cuentapasosvert - 1;

   }

   if (desague == 1 && limbajada == 0 && (cuentapasosvert == -LIMVERT) &&

(Farmaco == 1)) {

//      Serial.println("Entra en
70");       limbajada = 1;
cuentapasosjer = 0;
digitalWrite(EnableV ,
LOW);

   }

   if (desague == 1 && limbajada == 1 && (cuentapasosjer < LiquidEjection)) {
```

```
//      Serial.println("Entra en
71");
digitalWrite(valvulas, HIGH);
digitalWrite(Vreservorio,
LOW);

//      delay(5);

     digitalWrite(EnableJ , HIGH);

     PAPS.setSpeed(velInyeccion);

     PAPS.step(-1); // Motores jeringas ( - arriba + abajo )       cuentapasosjer =
cuentapasosjer + 1;

   }

   if (desague == 1 && limbajada == 1 && (cuentapasosjer == LiquidEjection) &&

(Farmaco == 1) && (valorFCSubida
!= HIGH)) { //      Serial.println("Entra
en 72");       digitalWrite(valvulas,
HIGH);       digitalWrite(EnableJ ,
LOW);       cuentapasosvert = 0;

//      digitalWrite(EnableV, HIGH);

     PAP.setSpeed(velocidadV); // velocidad de paso PAP.setSpeed(225);

     PAP.step(pasosV);   // numero de pasos y dirección (- abajo + arriba)

   }

   if (desague == 1 && limbajada == 1 && (cuentapasosjer == LiquidEjection) &&

(Farmaco == 1) && (valorFCSubida
== HIGH)) { //
Serial.println("Entra en 73");
```

```
digitalWrite(EnableV , LOW);
limbajada = 0;

   }

   if (desague == 1 && limbajada == 0 && (cuentapasosjer == LiquidEjection) &&
(Farmaco == 1)) {

//      Serial.println("Entra en 74");       pasosSDA = ((StartSDA * 50) - 723);  // Nos
movemos a drug plate( derecha)       cuentapasosSDA = 0;

     DerSDA = 1;

     IzSDA = 0;

   }


   if ((DerSDA == 1) && (cuentapasosSDA > pasosSDA)) {

//      Serial.println("Entra en 75");       desague = 0;       digitalWrite(EnableH , HIGH);
myStepper.setSpeed(velocidadH); // velocidad de paso       myStepper.step(pasosH);
// numero de pasos y dirección (- izquierda + derecha)       cuentapasosSDA =
cuentapasosSDA - 1;

   }

   if ((DerSDA == 1) && (cuentapasosSDA == pasosSDA) && (cuentapasosjer !=
Withdraw) && (limbajada == 0) && (cuentapasosvert > pasosvert) && (Farmaco == 1)) {

//      Serial.println("Entra en
76");
digitalWrite(EnableH ,
LOW);       cuentapasosvert =
0;

     Farmaco = 2;

   }
```

```
   if (desague == 0 && limbajada == 0 && (cuentapasosvert > -LIMVERT
+AltAgAspir) && Farmaco == 2 && DerSDA == 1) {
//     Serial.println("Entra en 77");
digitalWrite(EnableH, LOW);
digitalWrite(EnableV , HIGH);
    PAP.setSpeed(velocidadV);
PAP.step(-pasosV);
cuentapasosvert = cuentapasosvert - 1;
   }
   if (desague == 0 && limbajada == 0 && (cuentapasosvert == -LIMVERT +AltAgAspir)
&& Farmaco == 2 && DerSDA == 1 && cuentapasosjer != DrugVolume) {
//     Serial.println("Entra en
78");      limbajada = 1;
digitalWrite(EnableV ,
LOW);      cuentapasosjer =
0;
   }
   if (Farmaco == 2 && limbajada == 1 && (cuentapasosjer < DrugVolume)) {
//     Serial.println("Entra en
79");
digitalWrite(valvulas, HIGH);
digitalWrite(Vreservorio,
LOW);
//     delay(5);
    digitalWrite(EnableJ , HIGH);
PAPS.setSpeed(velAspiracion);
```

```
    PAPS.step(1); // Motores jeringas ( - arriba + abajo )       cuentapasosjer =
cuentapasosjer + 1;

  }

  if (Farmaco == 2 && limbajada == 1 && (cuentapasosjer == DrugVolume)) {

//     Serial.println("Entra en
80");
digitalWrite(valvulas, LOW);
digitalWrite(EnableJ , LOW);
//     digitalWrite(EnableV,
HIGH);

    PAP.setSpeed(velocidadV); // velocidad de paso PAP.setSpeed(225);

    PAP.step(pasosV);   // numero de pasos y dirección (- abajo + arriba)

  }

  if ((cuentapasosjer == DrugVolume) && (valorFCSubida == HIGH) && (desague == 0)
&& (Farmaco == 2)) {

//     Serial.println("Entra en 81");
limbajada = 0;


digitalWrite(EnableV,
LOW);      Farmaco =
3;
cuentapasosjerSDA =
0;

  }

  if ((cuentapasosjerSDA < Aire2) && (Farmaco == 3)) {

//     Serial.println("Entra en
82");
```

```
digitalWrite(valvulas ,
HIGH);
digitalWrite(Vreservorio,
LOW);

//     delay(5);

//     digitalWrite(EnableJ, HIGH);

    PAPS.setSpeed(velAspiracion);

    PAPS.step(1); // Motores jeringas ( - arriba + abajo )       cuentapasosjerSDA =
cuentapasosjerSDA + 1;

  }

  if ((cuentapasosjerSDA == Aire2) && (Farmaco == 3) && (FCderActual != HIGH)

&& (desague == 0)) { // Air bubble 2 //      Serial.println("Entra en 83");
digitalWrite(valvulas,LOW);       digitalWrite(EnableJ, LOW);
digitalWrite(EnableH, HIGH);       myStepper.setSpeed(velocidadH); // velocidad de
paso       myStepper.step(pasosH);   // numero de pasos y dirección (- izquierda +
derecha)     }

  if ((cuentapasosjerSDA == Aire2) && (Farmaco == 3) && (FCderActual == HIGH) &&
(desague == 0)) {

//     Serial.println("Entra en 84");       digitalWrite(EnableJ, LOW);
digitalWrite(EnableH , LOW);       myStepper.setSpeed(0); // velocidad de
paso       myStepper.step(0);// numero de pasos y dirección (- izquierda +
derecha)       // si llega al fc derecho paramos motor*/       delay(5);

    digitalWrite(EnableH , HIGH);

    myStepper.setSpeed(70); // Damos un paso a la izquierda
myStepper.step(-671);       desague = 1;       pasosSDA = ((-StartSDA * 50) - 80);
// Nos movemos a target plate( izquierda)

    cuentapasosSDA = 0;

    DerSDA = 0;
```

```
    IzSDA =
1;
Farmaco =
4;
cuentapasos
vert = 0;
delay(5);

  }

  if ((IzSDA == 1) && (cuentapasosSDA > pasosSDA) && (FCderActual != HIGH) &&
(Farmaco == 4)) {

//    Serial.println("Entra en 85");    desague = 0;    digitalWrite(EnableH , HIGH);
myStepper.setSpeed(velocidadH); // velocidad de paso    myStepper.step(-pasosH);  //
numero de pasos y dirección (- izquierda + derecha)    cuentapasosSDA =
cuentapasosSDA - 1;

  }


  if ((IzSDA == 1) && (cuentapasosSDA == pasosSDA) && (Farmaco == 4) &&

(limbajada == 0) && (cuentapasosvert > -LIMVERT +AltAgInyec)) {

//    Serial.println("Entra en 86");

    digitalWrite(EnableH, LOW);
digitalWrite(EnableV , HIGH);

    PAP.setSpeed(velocidadV);
PAP.step(-pasosV);
cuentapasosvert = cuentapasosvert - 1;

  }

  if ((IzSDA == 1) && (cuentapasosSDA == pasosSDA) && (Farmaco == 4) &&

(limbajada == 0) && (cuentapasosvert == -LIMVERT +AltAgInyec)) {
```

```
//      Serial.println("Entra en
87");       limbajada = 1;
digitalWrite(EnableV ,
LOW);       cuentapasosjer =
0;

     IzSDA = 0;

     Farmaco = 5;

    }

    if ((limbajada == 1) && (cuentapasosjer < DrugVolume) && (Farmaco == 5)) {

//      Serial.println("Entra en
88");
digitalWrite(valvulas, HIGH);
digitalWrite(Vreservorio,
LOW);

//      delay(5);

     digitalWrite(EnableJ , HIGH);

     PAPS.setSpeed(velInyeccion);

     PAPS.step(-1); // Motores jeringas ( - arriba + abajo )       cuentapasosjer =
cuentapasosjer + 1;

    }


    if ((cuentapasosjer == DrugVolume)&& (desague == 0) && (limbajada == 1) &&

(Farmaco == 5)&& (StartSDA <
EndSDA + 1)) { //
Serial.println("Entra en 89");
digitalWrite(EnableJ, LOW);
```

```
//      StartSDA =
StartSDA + 1;
Farmaco = 6;
cuentapasosSDA = 0;
cuentapasosvert = 0;
cuentapasosjer = 0;

   }




  if ((Farmaco == 6) && (valorFCSubida != HIGH) ) { // Ponemos en el cero el motor de las
agujas

//       Serial.println("Entra en 1");     digitalWrite(EnableV ,
HIGH);

   PAP.setSpeed(150);

   PAP.step(4);

  }

  if (valorFCSubida == HIGH && Farmaco ==6  ) {      // Si toca el FC de subida, paramos
el motor y luego damos un pequeño paso para separarlo

//       Serial.println("Entra en 22");     digitalWrite(EnableV ,
LOW);

   PAP.setSpeed(0); // velocidad de paso

   PAP.step(0);// numero de pasos y dirección (- abajo + arriba)
delay(3);     inicialS = 0;     limbajada = 0;     SubirAgujas = 0;

   digitalWrite(EnableV , HIGH);
```

```
   PAP.setSpeed(150);
PAP.step(-10);

   digitalWrite(EnableV , LOW);

   Farmaco = 7;

  }

  if ((Farmaco == 7) && (FCizActual != HIGH) ) { // Ponemos el motor horizontal en el
desague

//       Serial.println("Entra en
2");     digitalWrite(EnableH ,
HIGH);
myStepper.setSpeed(55);
myStepper.step(-1);

  }

  if (FCizActual == 1 && Farmaco == 7 &&(StartSDA < EndSDA + 1) ) { // Condicion para
que el cero sea el desague

//       Serial.println("Entra en 3");
digitalWrite(EnableH , LOW);
myStepper.setSpeed(0); // velocidad de
paso

   myStepper.step(0);// numero de pasos y dirección (- izquierda +
derecha)       // si llega al fc derecho paramos motor*/     delay(3);
digitalWrite(EnableH , HIGH);     myStepper.setSpeed(55); // Damos un
paso a la izquierda     myStepper.step(690);     desague = 1;     Lavado =
1;

   Farmaco = 0;

   StartSDA = StartSDA + 1;
```

```
  }



   if ((cuentapasosjer == DrugVolume) && (valorFCSubida == HIGH) && (desague == 0)
&& (Farmaco == 5) && (StartSDA == EndSDA + 1)) {

//      Serial.println("Entra en 90");

     SDA1 = 0;

     Lavado
= 0;
Farmaco =
6;
cuentapasos
SDA = 0;
cuentapasos
vert = 0;
cuentapasosj
er = 0;
limbajada =
0;

    }









  }


//v65#1z150$3r75!1g58t0A2l20L50I75h0a20b10c50e20f10q1B5C15D15E15J0F1G2H


}
```