



**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO**  
**CENTRO UNIVERSITARIO UAEM ATLACOMULCO**



“INTERFAZ GRÁFICA DE CONTROL E INSTRUMENTACIÓN  
DE UN SISTEMA MECÁNICO DE EJES COORDENADOS  
APLICADO EN EL PROCESO DE PULIDO ELECTROQUÍMICO”

**T E S I S**

Que para obtener el Título de:

**Ingeniero en Computación**

Presenta:

**Brayan Uriel Farfán González**

Director de Tesis:

**Dr. en C. I. E. Everardo Efrén Granda Gutiérrez**

**Atlacomulco, México; octubre de 2018**



Universidad Autónoma del Estado de México  
Centro Universitario UAEM Atlacomulco

**Área de Titulación**

Atlacomulco, Méx., a 17 de Abril de 2018.

**P.ICO. BRAYAN URIEL FARFÁN GONZÁLEZ  
P R E S E N T E .**


Por este conducto me permito comunicar a usted (es) que el Comité Técnico de Titulación autorizó el proyecto de TESIS que presentó (aron) y que será asesorado por el (la) DR. EVERARDO EFRÉN GRANDA GUTIÉRREZ titulado:

**“INTERFAZ GRÁFICA DE CONTROL E INSTRUMENTACIÓN DE  
UN SISTEMA MECÁNICO DE EJES COORDENADOS APLICADO  
EN EL PROCESO DE PULIDO ELECTROQUÍMICO”**

Al mismo tiempo solicito a usted (es) de la manera más atenta que al concluir el desarrollo del trabajo sea (n) tan amable (s) de comunicarlo por escrito a este Departamento.

**ATENTAMENTE**  
PATRIA, CIENCIA Y TRABAJO

2018, Año del 190 Aniversario de la Universidad Autónoma del Estado de México

  
**M. EN P.C. NEPHTALÍ PIERRE ROMERO NAVARRETE**  
SUBDIRECTOR ACADÉMICO  
CENTRO UNIVERSITARIO UAEM ATLACOMULCO



Km. 60 Carretera Toluca – Atlacomulco  
C.P. 50450  
Atlacomulco, Estado de México  
Tels. (712) 122 04 36, 122 04 46, 122 0535  
e-mail: cuatla@uaemex.mx





Universidad Autónoma del Estado de México  
Centro Universitario UAEM Atlacomulco

*Área de Titulación*

Atlacomulco, Méx., a 17 de Abril de 2018.

DR. EVERARDO EFRÉN GRANDA GUTIÉRREZ  
P R E S E N T E :

Por este conducto me permito comunicar a usted que la Comisión Revisora de Proyectos de Tesis, acordó nombrarlo ASESOR del Trabajo de TESIS titulado:

“INTERFAZ GRÁFICA DE CONTROL E INSTRUMENTACIÓN DE UN SISTEMA MECÁNICO DE EJES COORDENADOS APLICADO EN EL PROCESO DE PULIDO ELECTROQUÍMICO”

Que presenta (n) el (los) Pasante (s) de ICO. BRAYAN URIEL FARFÁN GONZÁLEZ al mismo tiempo solicito a usted de la manera más atenta, que al concluir su asesoría sea tan amable de comunicarlo por escrito a este Departamento.

A T E N T A M E N T E  
PATRIA, CIENCIA Y TRABAJO

“2018, Año del 190 Aniversario de la Universidad Autónoma del Estado de México”

M. EN P.C. NEPHTALI PIERRE ROMERO NAVARRETE  
SUBDIRECTOR ACADÉMICO  
CENTRO UNIVERSITARIO UAEM ATLA COMULCO



Km. 60 Carretera Toluca – Atlacomulco  
C.P. 50450  
Atlacomulco, Estado de México  
Tels. (712) 122 04 36, 122 04 46, 122 0535  
e-mail: [cuatla@uaemex.mx](mailto:cuatla@uaemex.mx)





Universidad Autónoma del Estado de México  
Centro Universitario UAEM Atzacomulco

Atzacomulco, México a 5 de octubre de 2018

**M. EN P. C. NEPHTALI PIERRE ROMERO NAVARRETE**  
**SUBDIRECTOR ACADEMICO DEL CENTRO**  
**UNIVERSITARIO UAEM ATZACOMULCO**

**P R E S E N T E:**

En contestación a su atento oficio con fecha del 17 de abril de 2018, en el cual se me designó como **ASESOR** del Trabajo de **TESIS** titulado "**Interfaz gráfica de control e instrumentación de un sistema mecánico de ejes coordenados aplicado en el proceso de pulido electroquímico**", presentado por el pasante de la Licenciatura en Ingeniería en Computación **Brayan Uriel Farfán González**, informo que:

He concluido satisfactoriamente la revisión del trabajo, haciendo las observaciones pertinentes, mismas que han sido solventadas por el interesado. Así mismo, **manifiesto que el trabajo cumple con los requisitos y cualidades que corresponden a la opción de evaluación profesional**; por lo cual no tengo ningún inconveniente en dar mi **Voto Aprobatorio** para que el egresado pueda continuar con los trámites de titulación.

Sin otro particular por el momento, agradezco su atención.

**ATENTAMENTE**

DR. EN C. I. E. EVERARDO EFRÉN GRANDA GUTIÉRREZ  
PROFESOR DE TIEMPO COMPLETO

*Recibi 5-10-18  
J. Farfán*

Km. 60 Carretera Toluca - Atzacomulco, C.P. 50450  
Atzacomulco, Estado de México  
Tels. (712) 122 04 36, 122 04 46, 122 0535  
e-mail: cuatia@uaemex.mx  
Centro Universitario UAEM Atzacomulco





Universidad Autónoma del Estado de México  
Centro Universitario UAEM Atlacomulco

**Departamento de Evaluación Profesional**  
c.c.p. expediente del (la) interesado(a)

Atlacomulco, Méx., a 5 de Octubre de 2018.

**DR. EN C. EN I. JOSÉ ARTURO PÉREZ MARTÍNEZ**  
**PRESENTE:**

Por este conducto me permito comunicar a usted que la Comisión Revisora de Proyectos de Tesis acordó nombrarlo **REVISOR(A)** del Trabajo de **TESIS** titulado:

**"INTERFAZ GRÁFICA DE CONTROL E INSTRUMENTACION DE UN SISTEMA MECÁNICO DE EJES COORDENADOS APLICADO EN EL PROCESO DE PULIDO ELECTROQUÍMICO"**

Que presenta(n) el (los) Pasante(s) de **ICO. BRAYAN URIEL FANFAN GONZALEZ**, al mismo tiempo solicito a usted de la manera más atenta, que al concluir su revisión sea tan amable en comunicar por escrito su **VOTO APROBATORIO** a este Departamento, en un lapso **no mayor a 10 días** a partir de la recepción del trabajo.

**ATENTAMENTE**  
**PATRIA, CIENCIA Y TRABAJO**

"2018, Año del 190 Aniversario de la Universidad Autónoma del Estado de México"

**M. EN P.C. NEPHTALI PIERRE ROMERO NAVARRETE**  
**SUBDIRECTOR ACADÉMICO**  
**CENTRO UNIVERSITARIO UAEM ATLACOMULCO**



*Recibo Original*  
*José Arturo Pérez Martínez*

c.c.p. expediente del (la) interesado(a)

Km. 60 Carretera Toluca - Atlacomulco  
C.P. 50450  
Atlacomulco, Estado de México  
Tels. (712) 122 04 36, 122 04 46, 122 0535  
e-mail: cuatla@uaemex.mx







Universidad Autónoma del Estado de México  
Centro Universitario UAEM Atlacomulco

*Departamento de Evaluación Profesional*

Atlacomulco, Méx., a 5 de Octubre de 2018.

**DR. EN C. EN I. CARLOS EDUARDO TORRES REYES  
P R E S E N T E:**

Por este conducto me permito comunicar a usted que la Comisión Revisora de Proyectos de Tesis acordó nombrarlo **REVISOR(A)** del Trabajo de **TESIS** titulado:

**"INTERFAZ GRÁFICA DE CONTROL E INSTRUMENTACIÓN DE UN SISTEMA MECANICO DE EJES COORDENADOS APLICADO EN EL PROCESO DE PÚLIDO ELECTROQUIMICO"**

Que presenta(n) el (los) Pasante(s) de **ICO. BRAYAN URIEL FARFAN GONZALEZ**, al mismo tiempo solicito a usted de la manera más atenta, que al concluir su revisión sea tan amable en comunicar por escrito su **VOTO APROBATORIO** a este Departamento, en un lapso **no mayor a 10 días a partir de la recepción del trabajo.**

**A T E N T A M E N T E  
PATRIA, CIENCIA Y TRABAJO**

*"2018, Año del 190 Aniversario de la Universidad Autónoma del Estado de México"*

Recibi:  
08-10-18  
*[Handwritten signature]*

*[Handwritten signature]*

**M. EN P.C. NEPHTALI PIERRE ROMERO NAVARRETE  
SUBDIRECTOR ACADÉMICO  
CENTRO UNIVERSITARIO UAEM ATLACOMULCO**



c.c.p. expediente del (la) interesado(a)

Km. 60 Carretera Toluca - Atlacomulco  
C.P. 50450  
Atlacomulco, Estado de México  
Tels. (712) 122 04 36, 122 04 46, 122 0535  
e-mail: [cuatla@uaemex.mx](mailto:cuatla@uaemex.mx)





Universidad Autónoma del Estado de México  
Centro Universitario UAEM Atlacomulco

Atlacomulco, México a 17 de octubre de 2018

**M. EN P.C. NEPHTALI PIERRE ROMERO NAVARRETE**  
**SUBDIRECTOR ACADEMICO DEL CENTRO**  
**UNIVERSITARIO UAEM ATLACOMULCO**

**P R E S E N T E:**

En contestación a su atento oficio con fecha del 5 de octubre de 2018, en el cual se me designó como **REVISOR** del Trabajo de **TESIS** titulado "**Interfaz gráfica de control e instrumentación de un sistema mecánico de ejes coordenados aplicado en el proceso de pulido electroquímico**", presentado por el pasante de la Licenciatura en Ingeniería en Computación **Brayan Uriel Farfán González**, informo que:

He concluido satisfactoriamente la revisión del trabajo, haciendo las observaciones pertinentes, mismas que han sido solventadas por el interesado. Así mismo, **manifiesto que el trabajo cumple con los requisitos y cualidades que corresponden a la opción de evaluación profesional**; por lo cual no tengo ningún inconveniente en dar mi **Voto Aprobatorio** para que el egresado pueda continuar con los trámites de titulación.

Sin otro particular por el momento, agradezco su atención.

**ATENTAMENTE**

  
\_\_\_\_\_  
**DR. EN C. P. E. JOSÉ ARTURO PÉREZ MARTÍNEZ**  
**PROFESOR DE TIEMPO COMPLETO**

*Recibo 18-01-18  
J. Pérez*

Km. 60 Carretera Toluca - Atlacomulco, C.P. 50450  
Atlacomulco, Estado de México  
Tels. (712) 122 04 36, 122 04 46, 122 0535  
e-mail: cuatia@uaemex.mx  
Centro Universitario UAEM Atlacomulco





Universidad Autónoma del Estado de México  
Centro Universitario UAEM Atlacomulco

Atlacomulco, México a 17 de octubre de 2018

**M. EN P.C. NEPHTALI PIERRE ROMERO NAVARRETE  
SUBDIRECTOR ACADEMICO DEL CENTRO  
UNIVERSITARIO UAEM ATLACOMULCO**

**P R E S E N T E:**

En contestación a su atento oficio con fecha del 5 de octubre de 2018, en el cual se me designó como **REVISOR** del Trabajo de **TESIS** titulado "**Interfaz gráfica de control e instrumentación de un sistema mecánico de ejes coordenados aplicado en el proceso de pulido electroquímico**", presentado por el pasante de la Licenciatura en Ingeniería en Computación **Brayan Uriel Farfán González**, informo que:

He concluido satisfactoriamente la revisión del trabajo, haciendo las observaciones pertinentes, mismas que han sido solventadas por el interesado. Así mismo, **manifiesto que el trabajo cumple con los requisitos y cualidades que corresponden a la opción de evaluación profesional**, por lo cual no tengo ningún inconveniente en dar mi **Voto Aprobatorio** para que el egresado pueda continuar con los trámites de titulación.

Sin otro particular por el momento, agradezco su atención.

**ATENTAMENTE**

**DR. EN C. I. E. CARLOS EDUARDO TORRES REYES  
PROFESOR DE TIEMPO COMPLETO**

*Recibido 15-10-2018  
Torres Reyes*

Km. 60 Carretera Toluca – Atlacomulco, C.P. 50450  
Atlacomulco, Estado de México  
Tels. (712) 122 04 36, 122 04 46, 122 0535  
e-mail: [cuatla@uaemex.mx](mailto:cuatla@uaemex.mx)  
Centro Universitario UAEM Atlacomulco





## **DEDICATORIAS**

Este trabajo es dedicado a mis padres, por todo su apoyo y cariño incondicional; aun cuando se encontraban cansados o en sus peores días, que dieron lo mejor de sí para que pudiera salir a delante. Dándome sus mejores consejos y educación que hoy en día hacen que sea como soy, y por enseñarme que nunca hay que rendirse, que todo esfuerzo tiene su recompensa.

A mis hermanos, por soportar mi mal humor, pleitos y grandes momentos como toda familia, que aun cuando estemos al otro lado del mundo, siempre contaré con ustedes.

BYK. Por todos esos momentos grandiosos y los que no, que me dejan conocer lo mejor de ti, tu tenacidad y dedicación, eres quien me inspira a ser una mejor persona día con día. Quien me propone nuevos retos para salir de esa zona de confort y conocer juntos nuevos horizontes: ¡esto es por ti!

A mis compañeros y amigos de la “ratonera”, por todo ese tiempo que nos soportamos mutuamente en esos momentos de estrés y confortamientos, como también de aquellas amenas platicas de tontería y anécdotas de vida, espero que al igual que yo, lo hayan disfrutado. Gracias amigos por esos momentos.

## **AGRADECIMIENTOS**

El autor agradece el apoyo otorgado por la Secretaría de Educación Pública, mediante la beca para el desarrollo de esta tesis, como parte del proyecto denominado “Desarrollo de un prototipo para el maquinado electroquímico de metales utilizando voltaje pulsado de alta frecuencia”, con número de convenio DSA/103.5/16/14424, cuyo responsable técnico fue el Dr. Everardo Efrén Granda Gutiérrez.

Se agradece también al equipo de trabajo del Laboratorio de Investigación de Posgrado del Centro Universitario UAEM Atlacomulco, en donde se desarrolló la totalidad de la experimentación de este trabajo, así como al Cuerpo Académico “Desarrollo de software, dispositivos y sistemas aplicados a la innovación tecnológica”, del mismo espacio académico, por la vinculación de este trabajo a sus líneas de investigación.

Estoy enormemente agradecido con mi asesor, el Dr. Everardo Efrén Granda Gutiérrez por su inmensa paciencia, apoyo, consejos y motivación para la realización y culminación de este proyecto. Que durante esta excelsa travesía confió en mí y me alentó para salir adelante: ¡MIL GRACIAS!

Al cuerpo académico de Ingeniería en Computación y en especial a los PTC's por permitirme ocupar las instalaciones y laboratorios que con su apoyo se han podido mejorar a lo largo de su estancia en este Centro Universitario.

## RESUMEN

En este documento se presenta el desarrollo de un sistema de cómputo embebido, específicamente diseñado para la ejecución de pruebas de pulido electroquímico en piezas metálicas conductoras. Este proceso pertenece a las técnicas de manufactura avanzada no convencional, debido a que la herramienta de trabajo no se encuentra en contacto directo con la pieza.

El proceso de pulido electroquímico se lleva a cabo en un medio electrolítico, en el que existe una polarización entre los electrodos: herramienta y pieza de trabajo. Esta polarización ocasiona el desprendimiento de material del ánodo (pieza que se desea pulir), permitiendo la remoción de material a escala micrométrica y mejorando el acabado superficial de la misma. Se caracteriza porque la distancia entre los electrodos se encuentra en una escala entre 100 y 400  $\mu\text{m}$ . El proceso de pulido se realiza recorriendo paulatinamente la superficie de la pieza con el electrodo o herramienta de trabajo, mientras esta se encuentra polarizada y el electrolito se hace fluir en el espacio que existe entre ambos elementos.

El sistema desarrollado permite que el usuario tenga un control manual o semiautomático para manipular un dispositivo de movimiento mecánico de tres ejes (plano cartesiano  $x$ ,  $y$ ,  $z$ ), pudiendo definir la trayectoria que debe seguir el electrodo durante la ejecución del proceso, así como la duración de este. Al mismo tiempo, el sistema permite el monitoreo de las variables más importantes del proceso: voltaje de polarización del electrodo y corriente consumida. Como parámetros adicionales se permite introducir las características físicas de la pieza (tipo de aleación y dimensiones), del electrodo (material del que está hecho) y del electrolito (composición química y concentración).

El sistema de control se desarrolló en el lenguaje de programación Python, el cual permite utilizar bibliotecas de funciones, generando recursos gráficos por medio de la biblioteca Tkinter para crear una Interfaz Gráfica de Usuario; de esta forma el usuario puede interactuar con el proceso y observar los diversos cambios que se puedan generar durante el desarrollo de este.

Para validar este trabajo, se realizaron pruebas experimentales de pulido electroquímico con diversas aleaciones de materiales metálicos y con una solución salina como electrolito. Estas pruebas permitieron demostrar que el sistema embebido es capaz de efectuar satisfactoriamente el movimiento del sistema mecánico mediante la sincronización de los ejes cartesianos por medio de motores a pasos, con una resolución seleccionable de paso completo, medio paso, octavo de paso o dieciseisavo de paso. Asimismo, se presentan gráficas del monitoreo en tiempo real de las variables del proceso y se almacena un reporte para el usuario con todos los datos del proceso.

Adicionalmente, se probó que el sistema puede ser empleado para maquinado electroquímico, ya que se efectuó el maquinado de un canal lineal en una pieza metálica, por lo que se constituye como un sistema versátil para la realización de estas pruebas electroquímicas.

Palabras clave: sistema embebido, interfaz gráfica de usuario, pulido electroquímico.

## **ABSTRACT**

Development of an embedded computing system is presented in this document. This system has been specifically designed to perform electrochemical polishing tests in metallic conductive pieces. The process belongs to the non-conventional advanced machining techniques because the tool is not in direct contact with the workpiece.

The electrochemical polishing process is realized in an electrolytic medium, where two electrodes are biased: the tool and the workpiece. The electrical polarization produces removal of material in a micrometric scale, thus improving the surface finish of the workpiece (anode). The process is characterized by a inter electrode distance between 100 and 400  $\mu\text{m}$  and it is achieved by the gradual exploration of the workpiece by the biased tool electrode while the electrolyte is flowing in the inter electrode space.

The proposed system provides to the user of a manual or semi-automatic control for manipulating a three-axis mechanical movement device (cartesian axes  $x$ ,  $y$ ,  $z$ ). The trajectory of the electrode during the execution of the test, as well as the time duration of it can be defined. At the same time, the system monitors and records the most important variables of the process: polarization voltage and consumed current. Also, additional parameter can be introduced: physical characteristics of the workpiece (alloy type and dimensions), electrode (material), and electrolyte (chemical composition and concentration).

The control system was developed in Python programming language, which allows using different function libraries, generating graphical resources by means the Tkinter library, used to create a Graphical User Interface. In this sense, user can interact with the process and to observe different changes in the variables during its progress.

With the aim to validate this work, electrochemical polishing test were performed with different metallic alloys and using a saline solution as electrolyte. Such tests allowed to demonstrate that the embedded system is able to satisfactorily control the position of the mechanical system through synchronization of cartesian axes by stepping motors. The resolution of the stepping motors can be configured in full step, but also in half, octave



and sixteenth of step. Graphics of real time monitoring of process variables are stored and a test report with all process data is generated for further use.

Additionally, it was proved that the system can be used to electrochemical machining, by realizing the machining of a straight channel in a metallic piece. Thus, the system can be considered as a versatile device to perform different types of electrochemical processes.

Keywords: embedded system, graphical user interface, electrochemical polishing.

# ÍNDICE

DEDICATORIAS.....	viii
AGRADECIMIENTOS.....	ix
RESUMEN.....	x
ABSTRACT .....	xii
ÍNDICE.....	xiv
ÍNDICE DE TABLAS.....	xvi
ÍNDICE DE FIGURAS .....	xvii
1 INTRODUCCIÓN .....	1
2 PLANTEAMIENTO DEL PROBLEMA .....	4
2.1 Definición del problema .....	5
2.2 Objetivos de la investigación.....	6
2.2.1 Objetivo general .....	6
2.2.2 Objetivos específicos.....	6
2.3 Pregunta(s) de investigación.....	7
2.4 Justificación .....	7
2.5 Impactos.....	9
3 HIPÓTESIS O META DE INGENIERÍA .....	10
4 ESTADO DEL ARTE.....	11
4.1 Métodos de manufactura.....	11
4.1.1 Procesos basados en métodos mecánicos .....	12
4.1.2 Procesos basados en métodos térmicos .....	12
4.1.3 Procesos basados en métodos químicos .....	13
4.2 Pulido.....	14
4.2.1 Pulido mecánico .....	14

4.2.2	Pulido por electrólisis .....	15
4.3	Evolución de la tecnología para la manufactura no convencional.....	15
4.4	Interfaz de usuario .....	18
4.5	Raspberry Pi 2 B+.....	18
5	MÉTODO.....	22
5.1	Especificaciones de diseño .....	22
5.2	Diseño conceptual del sistema.....	23
5.3	Adecuación del sistema mecánico cartesiano .....	27
5.4	Controlador de motores a pasos.....	31
5.5	Diseño del hardware asociado al sistema embebido.....	35
5.6	Diseño de la interfaz de usuario.....	39
5.7	Ensamble del prototipo .....	49
6	RESULTADOS.....	51
6.1	Funcionamiento .....	51
6.2	Validación.....	59
7	CONCLUSIONES .....	68
8	REFERENCIAS .....	69
9	ANEXOS: script principal .....	73

## ÍNDICE DE TABLAS

Tabla 1: Ejemplo de maquinaria industrial .....	8
Tabla 2: Ejemplo de maquinaria comercial.....	8
Tabla 3: Secuencia a paso completo.....	33
Tabla 4: Secuencia a medio paso.....	33
Tabla 5: Lista de componentes requeridos para el prototipo de pulido electroquímico.	36

## ÍNDICE DE FIGURAS

Figura 1: Computador Raspberry Pi modelo B+ .....	20
Figura 2: Esquema básico para el proceso de pulido electroquímico .....	26
Figura 3: Adaptación del sistema de ejes coordenados para el proceso de pulido electroquímico .....	28
Figura 4A) Esquema del soporte, Figura 4B) Soporte de la herramienta.....	29
Figura 5: Modulo interruptor de fin de carrera.....	29
Figura 6: Motor a pasos NEMA 23 .....	31
Figura 7: Conexión interna de motor bipolar .....	32
Figura 8: Conexión interna de motor a pasos unipolar.....	32
Figura 9: Controlador prediseñado con el uso de los C.I. L297 y L298 .....	34
Figura 10: Modulo controlador TB 6560 (Shenzhen Global Technology Co., Ltd., 2015) .....	35
Figura 11: Diagrama de conexión para adquisición de datos .....	37
Figura 12: Circuito de control para el sistema mecánico. ....	38
Figura 13. Diagrama de caso de usos para el software de control.....	39
Figura 14: Secuencia de funcionamiento del proceso. ....	42
Figura 15: Diagrama de flujo para la ventana principal .....	43
Figura 16: Diagrama de flujo para el registro de un nuevo proceso. ....	45
Figura 17: Diagrama de flujo para la ventana con las características del sistema.....	45
Figura 18: Diagrama de flujo de la ventana de pulido electroquímico. ....	46
Figura 19: Diagrama de flujo para el movimiento mecánico. ....	48
Figura 20. Vista general del prototipo para pulido electroquímico .....	49
Figura 21:Diagrama a bloques del funcionamiento del sistema.....	50
Figura 22: Archivos del programa y ubicación .....	51
Figura 23: Ventana principal del programa.....	52
Figura 24: menú de opciones para botón Archivo.....	53
Figura 25: opciones para botón de ayuda .....	53
Figura 26:Ventana de registro de un nuevo proceso .....	54
Figura 27: Ventana para el registro de parámetros.....	55
Figura 28: Ventana final para control y monitoreo del proceso. ....	57



Figura 29: opción de grafica de resultados .....	58
Figura 30: Ventana de búsqueda para pruebas realizadas .....	58
Figura 31: Prueba con acero 10-12.....	60
Figura 32: Pulido obtenido en acero 10-12. ....	60
Figura 33: Mediciones en acero tratado .....	61
Figura 34: Área y pieza de acero tratado sin pulido .....	61
Figura 35: Pulido en acero tratado.....	61
Figura 36: Grafica de mediciones en acero 106-12.....	62
Figura 37: Pulido en pieza de acero 106-12 con malformaciones.....	62
Figura 38: Grafica de pulido tipo escalera en acero 10-12.....	63
Figura 39: Pulido con gran desprendimiento en acero 10-12.....	63
Figura 40: Acero HS-LA con superficie plana.....	64
Figura 41: Grafica de acero HS-LA al maquinar en línea recta .....	65
Figura 42: Línea formada por el proceso durante 15 min .....	65
Figura 43: Grafica de validación para el eje X.....	66
Figura 44: Línea formada por el proceso en el eje X .....	66
Figura 45: Grafica para línea de continuidad .....	67
Figura 46: Línea de continuación de la primera prueba .....	67

# 1 INTRODUCCIÓN

La manufactura se describe como el proceso de fabricar o producir algo con las manos; su raíz etimológica proviene del latín *manus factus*, traducido como mano y hacer, respectivamente, por lo que se considera así a todo aquello que es producto del trabajo manual (Grover, 2007). Esto hace referencia a que los primeros artefactos que el hombre ideó y construyó se hacían con las manos, o apoyados solamente con herramientas rudimentarias.

Con el paso del tiempo, los procesos de manufactura se fueron modificando hasta hacer uso de distintas herramientas, cada vez más elaboradas, para dar lugar a lo que en su conjunto se conoce como manufactura convencional. Esta consiste en la utilización de una herramienta de corte que se encuentra en contacto con la pieza con la que se desea trabajar, removiendo el material excedente mediante procesos de abrasión o fricción hasta que éste obtiene la forma deseada (Schey, 2002). Los procesos de fresado o torneado, son ejemplos comunes de la manufactura convencional, que se han utilizado durante mucho tiempo para la fabricación de piezas para la industria.

En épocas más recientes (los últimos 50 años), el desarrollo de la manufactura ha tenido grandes avances al crear e innovar las diversas herramientas ya existentes, haciéndolas cada vez mejores y especializadas para diferentes propósitos, acuñando un nuevo término: la manufactura no convencional, que se caracteriza por el uso de diversas técnicas donde comúnmente no existe un contacto directo entre la herramienta y la pieza de trabajo, por lo que el mecanismo de remoción de material ya no obedece a un proceso de fricción y abrasión, sino que depende de la naturaleza de otros procesos físico-químicos, como la electricidad, la luz o el calor (Grover, 2007).

Los procesos no convencionales permiten el maquinado de piezas que usualmente son difíciles de procesar mediante la manufactura convencional. Entre ellos, se encuentra el maquinado electroquímico (ECM, por las siglas en inglés de *electrochemical machining*), que es un proceso de manufactura no convencional que utiliza el fenómeno electroquímico de disolución anódica como mecanismo para remover material en piezas conductoras de electricidad (Pon Selvan, et al., 2017). Una variación de este proceso es la base del trabajo que se documenta en esta tesis.

Los avances de la tecnología para mejorar todos aquellos procesos de manufactura, tanto como la convencional como la no convencional, han hecho que el tema de la precisión genere un gran interés en el campo de estudio para los científicos e ingenieros, dando lugar a la manufactura avanzada. Esto implica la aparición de procesos automatizados por sistemas electrónicos y de cómputo, donde una computadora tiene el control de todo el proceso para que el resultado del producto sea lo más fiel a la idea con la cual fue concebida la pieza a realizar (Kosak, et al., 1997). Los procesos de manufactura avanzada requieren de un sistema de control para monitorear los diversos parámetros que serán utilizados para hacerlo eficiente, rápido y preciso.

La tecnología de control numérico es uno de los iconos de la manufactura avanzada, que se empezó a desarrollar a mediados del siglo pasado, como una evolución de los procesos convencionales, como lo mostró el MIT (*Massachusetts Institute of Technology*) en el año de 1953, al utilizar una fresadora de tres ejes, en el cual interpolaban diversas trayectorias para generar un movimiento tridimensional, con el uso de tarjetas perforadas y válvulas de vacío. Posteriormente, en 1960, comenzaron a realizar pruebas de adaptación al reemplazar las válvulas de vacío por transistores (Lamikiz Mentxaka, 2010).

Estos sistemas se adaptaron posteriormente a los procesos no convencionales, como se puede apreciar en el trabajo del Dr. Chaides, quien en su documento de tesis doctoral hizo uso de un sistema de Control Numérico, realizando un ajuste y combinación de técnicas de programación para el ajuste de software de control para realizar sus pruebas de maquinado mediante  $\mu$ EDM (micro manufactura por electro-descargas). De esta forma mostró que se requiere de un sistema de gran precisión que pueda manipular la distancia que existe entre la pieza y la herramienta de trabajo, en el orden de algunos micrómetros. Se plantea también que, al hacer uso de un software tanto comercial como de libre distribución, se deben de modificar los ficheros de trabajo requeridos para que pueda tenerse un mayor control del proceso dependiendo de cada caso de maquinado (Chaides Zuñiga, 2012).

Posteriormente, los Ing. Estrada y León realizaron el diseño construcción de una máquina para el proceso de maquinado electroquímico, en el cual ellos mostraron las ventajas y desventajas que se obtienen al utilizar una fuente de 75 Amperes para la corriente de

maquinado, haciendo uso de un sistema CNC (Control Numérico Computarizado) con PLC (por sus siglas en inglés *Programmable Logic Controller*) funcionando como sistema de control (Estrada Hernandez & León Bravo, 2014).

En las referencias presentadas en los párrafos precedentes, los autores se enfocan principalmente al proceso de mecanizado, sin profundizar en el tema del desarrollo de software, puesto que en sus proyectos es necesario usar software comercial. Es por lo que en el presente trabajo de investigación se propone aplicar los conceptos de ingeniería en computación para crear un sistema de control e instrumentación basado en software para su uso en el proceso de pulido electroquímico (ECP por sus siglas en inglés *Electro Chemical Polishing*), que es una variante del proceso ECM, utilizado para mejorar el acabado superficial de los componentes metálicos.

En este documento, se hace énfasis en el diseño de la herramienta de software, que se utilizará en conjunto con un sistema de cómputo embebido basado en Raspberry Pi 2, por lo que también se desarrolla el diseño de la instrumentación electrónica asociada al proceso.

## 2 PLANTEAMIENTO DEL PROBLEMA

Para el desarrollo de este proyecto se considera la integración de un prototipo para efectuar pruebas de pulido electroquímico. La intención es que, mediante una interfaz de usuario, este tenga el control del movimiento de un sistema cartesiano motorizado, tanto en modo manual como automático. En el eje  $z$  (altura) del sistema cartesiano se coloca un electrodo metálico como herramienta de maquinado, que actúa sobre una pieza de trabajo, la cual tiene como característica principal el ser un metal con dureza relativamente elevada, pero conductor de la electricidad. Ambos elementos deben estar separados por una distancia inter-electrodos (denominada usualmente como *gap*, por su tecnicismo en inglés) y sumergidos o bañados por una solución electrolítica (agua con cloruro de sodio).

El electrodo es polarizado mediante una fuente de poder de corriente directa, al mismo tiempo que un sistema de adquisición de datos se encuentra monitoreando el voltaje de polarización y la corriente consumida durante el proceso de pulido electroquímico.

Para el desarrollo de la interfaz gráfica mencionada, se consideran los siguientes requerimientos del sistema donde el usuario puede realizar las siguientes acciones:

1. Manipular un joystick tipo palanca, o por medio de determinados botones del teclado, realizar los movimientos del sistema mecánico.
2. Definir una trayectoria en función de figuras básicas predefinidas en el sistema.
3. Monitorear la posición del electrodo con respecto a los ejes cartesianos ( $x$ ,  $y$ ,  $z$ ).
4. Observar el tiempo de ejecución del proceso (inicio, duración y termino)
5. Presentar los valores de la medición de voltaje de polarización y corriente de pulido en tiempo real.
6. Introducir el tipo de material utilizado en herramienta y pieza de trabajo; electrolito y porcentaje de la disolución; información extra como comentario para dicha prueba
7. Almacenar los datos obtenidos para su posterior tratamiento (gráficas, rendimiento)
8. Adecuar el sistema mecánico, añadiendo sensores que permitirán trabajar en un área efectiva, creando un sistema integrado, de propósito específico.



De esta forma se describe en general el desarrollo del proyecto, pero también se debe de tomar en cuenta el funcionamiento físico del prototipo, puesto que al ser un sistema mecánico puede tener ciertos fallos en algún momento, por lo que durante la adecuación del sistema se considera la implementación de funciones de ajuste y seguridad.

## **2.1 Definición del problema**

Para este proyecto se considera la implementación de un sistema de cómputo embebido a un proceso de pulido electroquímico, requiriéndose de la adecuación de un sistema mecánico cartesiano de propósito general, para ser adaptado a dicho proceso. Para esto se requiere la integración de un prototipo, considerando: a) la interfaz gráfica de usuario para el control del proceso, instalada en una computadora de placa deducida Raspberry Pi, b) el diseño de la instrumentación del proceso, c) la integración de los elementos de software y hardware en un solo sistema y d) la validación del funcionamiento del prototipo.

El sistema de control para los ejes coordenados permitirá la manipulación de la herramienta de trabajo, otorgando la precisión requerida durante el proceso de pulido electroquímico. Con esto, al mismo tiempo se aumenta la calidad de manufactura al omitir el factor de error humano (Arévalo Mocayo & Hermosa Ocampo, 2014).

Para llevar un monitoreo de los resultados obtenidos, y que estos posteriormente puedan ser comparados al usar algunas variaciones de los mismos parámetros, se hace necesario el registro de las variables más importantes del proceso, a saber: voltaje de polarización del electrodo, corriente consumida durante el proceso de pulido, cálculo de la distancia inter-electrodos y tiempo de ejecución del maquinado. Para ello se generarán archivos que pueden ser consultados por el usuario de manera posterior a un experimento, convirtiéndose en una herramienta auxiliar para la comprensión de los fenómenos asociados al proceso durante el desarrollo de pruebas experimentales posteriores.

Para la realización de pruebas de validación del sistema desarrollado se efectúan pruebas preliminares, donde se obtienen diversos parámetros de funcionamiento del sistema mecánico para realizar una calibración, cerciorando que las distancias con las cuales se

esté trabajando sean correctas y no se obtengan mediciones erróneas dado que este proyecto pretende otorgar precisión y fiabilidad.

Posteriormente, y una vez desarrollada la interfaz de usuario donde se permite controlar de forma manual o automática con los patrones de movimiento preestablecidos para posicionar la herramienta de trabajo, se realizaron pruebas del pulido por disolución anódica del material, obteniendo los parámetros mencionados previamente, con lo cual estos mismos definirán la calidad del resultado obtenido al comparar diversas muestras.

## **2.2 Objetivos de la investigación**

### **2.2.1 Objetivo general**

Desarrollar un sistema computarizado que permita la manipulación de la herramienta de trabajo, utilizando un sistema mecánico cartesiano para realizar pruebas de pulido electroquímico en materiales metálicos que permitan un mejor conocimiento del proceso, mediante la flexibilización de su operación con una interfaz gráfica.

### **2.2.2 Objetivos específicos**

1. Integrar los componentes necesarios para contar con un prototipo del proceso de pulido electroquímico.
2. Desarrollar una interfaz donde el usuario pueda monitorear los valores de *gap*, corriente consumida durante el proceso y voltaje de polarización empleado en los electrodos.
3. Diseñar una interfaz gráfica para el control del movimiento del sistema mecánico por el usuario.
4. Integrar los componentes de software y hardware para crear un sistema capaz de realizar pruebas de pulido electroquímico.
5. Validar el funcionamiento del prototipo para realizar pruebas de pulido electroquímico

### **2.3 Pregunta(s) de investigación**

¿Qué lenguaje de programación será el más adecuado para programar la interfaz de control y monitoreo de un proceso de pulido electroquímico?

¿Dependiendo del lenguaje de programación se tendrá alguna mejora o desventaja con respecto al control del sistema?

¿Será eficiente el funcionamiento del prototipo al realizar pruebas de pulido electroquímico?

### **2.4 Justificación**

Este proyecto pretende abordar, como área de oportunidad, la aplicación de las herramientas de cómputo en la manufactura, particularmente para el proceso de pulido electroquímico, permitiendo replicar un sistema de maquinado de grado industrial, al ser un sistema desarrollado con equipamiento de bajo costo en comparación con los equipos utilizados en la manufactura a gran escala. El propósito de ello es contar con una herramienta con fines didácticos y de investigación.

Es importante mencionar que este trabajo forma parte de las actividades y entregables del proyecto “Desarrollo de un prototipo para maquinado electroquímico de metales utilizando voltaje pulsado de alta frecuencia”, apoyado por la Secretaría de Educación Pública, mediante el convenio “DSA/103.5/16/14424”, que se desarrolla en el Laboratorio de Investigación de Posgrado del Centro Universitario UAEM Atlacomulco.

Adicionalmente, se pretende introducir la manufactura avanzada en piezas metálicas de alta dureza a un sector más amplio de la población y que cuente con conocimientos básicos en las áreas de Ingeniería en Sistemas, Mecánica, Química, Electrónica o interés dentro de la misma área para la recreación de este proyecto para su uso y creación de nuevo conocimiento.

De esta forma se podrá hacer más accesible esta tecnología puesto que la maquinaria, recursos y personal empleados por la industria deben de ser especializados por lo que se



requiere de una gran inversión capital, por tal motivo este tipo de procesos se investigan mayormente por instituciones o sectores de la industria con gran poder adquisitivo.

Actualmente el adquirir un sistema mecánico como lo son impresoras 3D, cortadoras por láser entre otros, es un gasto que va desde los 350 USD dependiendo de los materiales de construcción y dimensiones del dispositivo por lo que se puede hacer una adaptación de estos para replicar el funcionamiento aquí propuesto.

*Tabla 1: Ejemplo de maquinaria industrial*

Marca	Tipo	Ejes	Área de trabajo	Dispositivo	Precio (aprox.)
HURO	Fresadora	4	14x38"		12,000 USD.
TREE	Fresadora	2	10x42"		32,000 USD

*Tabla 2: Ejemplo de maquinaria comercial.*

Marca	Tipo	Ejes	Área de trabajo	Dispositivo	Precio (Aprox.)
CNC	Router	4	8x10x18"		910 USD
CNC	Cortador Laser	3	9x7x3"		450 USD

Los costos de un sistema para manufactura ya sea convencional o avanzada se encuentran en un amplio rango de precios; es por ello que se plantea hacer uso de un prototipo comercial preconstruido para adaptarlo a una función específica como lo es el proceso de pulido electroquímico, donde se puede diseñar y ajustar a los requerimientos necesarios para su funcionamiento, en un entorno experimental.

## **2.5 Impactos**

### **Científico**

Al ser un proyecto aplicado en el proceso de manufactura no convencional se obtendrán diferentes datos recopilados al momento de realizar las pruebas, posteriormente las estadísticas podrán ser compartidas en la documentación de este proyecto de investigación.

### **Tecnológico**

Puesto que el presente proyecto pretende introducir el uso de un computador de placa reducida en conjunto con dispositivos electrónicos y un sistema de control, para la aplicación de técnicas de pulido, se podrá considerar como un sistema embebido de aplicación específica.



### **3 HIPÓTESIS O META DE INGENIERÍA**

Con el uso de diferentes herramientas computacionales, tanto de software como de hardware, se propone desarrollar un sistema de cómputo embebido para el monitoreo y control de un proceso de pulido electroquímico en piezas metálicas de alta dureza. Los principales parámetros por monitorear son: voltaje de polarización del electrodo y corriente consumida durante el proceso. Adicionalmente, se permiten incorporar como datos adicionales, otras variables relevantes, tales como: las dimensiones de la herramienta y la pieza de trabajo y el tipo de electrolito. Eso permitirá implementar un prototipo útil para el proceso, reproducible, portable para la ejecución de dichas pruebas, mediante el uso de una computadora de placa reducida, como componente principal del sistema.

## **4 ESTADO DEL ARTE**

En este apartado se presenta un análisis de la literatura técnica y científica relevante para la comprensión del problema a resolver, principalmente en lo relacionado a la aplicación de sistemas de cómputo y control del proceso de pulido electroquímico.

También se desarrollan las bases teóricas de los conceptos que se abordan en el trabajo de tesis, tales como la manufactura y los procesos no convencionales de manufactura.

### **4.1 Métodos de manufactura**

Los procesos de manufactura se desarrollan día tras día, con el fin de crear nuevos y diversos productos de acuerdo con las especificaciones que la industria moderna exige. Dependiendo de la forma y las características físicas o químicas de los materiales, se debe optar por aplicar alguno de estos métodos de manufactura, que mejor se adapte a las características del material y la precisión deseada. Por ejemplo, para fresado de materiales de alta dureza se requiere que la herramienta de trabajo tenga una dureza aún mayor para poder lograr un desbastado del material; de otro modo, el proceso de maquinado no es efectivo, al sufrir la herramienta un desgaste excesivo. Es por ello que el maquinado no convencional es una buena alternativa al utilizar métodos electromecánicos, químicos, entre otros, en los que el desprendimiento del material no es dependiente de la dureza del material a mecanizar (Zamora Vázquez, 2009).

La manufactura convencional se define por ser un proceso donde existe un desgaste, tanto en la pieza de trabajo como también en la herramienta de corte; este desgaste se debe a que la herramienta se encuentra en contacto directo con la pieza, para remover material de esta mediante mecanismos de fricción. Por este motivo, la herramienta tiene que ser de un material con una dureza considerablemente superior a la dureza del material de la pieza con la cual se desee trabajar, para que la remoción de material pueda llevarse a cabo de una manera eficiente. Se puede subdividir en dos tipos de procesos: por corte y por abrasión mecánica, siendo algunos de los procesos por corte: torneado, fresado, taladrado o recorte; mientras que los procesos de abrasión más comunes son el barrenado, pulido y cepillado (El Hofy, 2005).

En contraparte, la manufactura no convencional es la evolución de los métodos convencionales, y se caracteriza por no requerir de un contacto directo entre la herramienta y la pieza de trabajo durante el maquinado del material, por lo que se mejora significativamente la calidad del terminado en la pieza de trabajo, también denominado acabado superficial (Grover, 2007). Algunos de estos procesos, precisamente por no existir contacto entre material y herramienta, no producen desgaste en esta última, mejorando significativamente la eficiencia del proceso. De manera similar, se subdivide en las categorías de maquinado: mecánico, térmico, y químico, que se explican a continuación.

#### **4.1.1 Procesos basados en métodos mecánicos**

Se hace uso de procesos de erosión a alta velocidad mediante la aplicación de abrasivos o fluidos. Algunas de estas técnicas son (El Hofy, 2005):

1. WJM (Water Jet Machining/Cutting): Maquinado o corte por chorro de agua. Se hace fluir una corriente de agua a través de una fina boquilla que direcciona el chorro de agua a muy alta presión hacia la pieza que se desea realizar un corte.
2. AWJM (Abrasive Water Jet Machining): Adición de abrasivos en corte por agua. Es básicamente el mismo proceso de corte por chorro de agua, pero se añade algún material de tipo abrasivo.
3. USM (Ultrasonic Machining): Maquinado ultrasónico. Hace uso de diversos abrasivos compuestos en una pasta, posteriormente esta se usa junto con la herramienta de trabajo que vibra a altas frecuencias.

#### **4.1.2 Procesos basados en métodos térmicos**

En este tipo de procesos se hace uso de la energía térmica en una pequeña cantidad, para un área en específico donde se desea cortar o formar la pieza con la que se está trabajando. Las diferentes técnicas aplicadas con este método son (Grover, 2007):

1. EBM (Electron Beam Machining): Maquinado por haz de electrones. Este proceso hace uso de una ráfaga a alta velocidad de electrones que se concentran en la

superficie de la pieza de trabajo donde se desea remover material mediante fusión y vaporización

2. EDM (Electro Discharge Machining): El maquinado por electro descargas de plasma realiza una remoción de material generando arcos eléctricos con altas temperaturas en lugares específicos que funden y evaporan el metal (Soto C., 2008).
3. LBM (Laser Beam Machining): Maquinado por láser. Utiliza un haz de luz láser de alta energía para vaporizar las partículas del material en una zona puntual. El maquinado de piezas complejas se hace haciendo desplazarse el haz láser por determinadas trayectorias.

#### **4.1.3 Procesos basados en métodos químicos**

Se aplica donde los materiales son mayormente susceptibles a ciertas reacciones químicas. Algunos de estos procesos hacen uso de ácidos, bases, sales, o alguna otra sustancia que permita remover material en una zona deseada. Por mencionar algunos de estos procesos se enlistan los siguientes (Schey, 2002):

1. CHM (Chemical Milling): Perforación química. Hace uso de diversas composiciones químicas las cuales crean un conducto a través de la pieza de trabajo
2. ECM (Electrochemical Machining): Maquinado electroquímico. El cual usa el principio de disolución anódica que desprende material metálico entre dos electrodos con la ayuda de un electrolito.
3. ECP (Electrochemical Polishing): Pulido electroquímico. Es un derivado de ECM el cual es utilizado principalmente para dar un mejor acabado a técnicas de maquinado convencional. También es empleado para limpieza de superficies.
4. PCM (Photo Chemical Machining): Maquinado fotoquímico. La pieza de trabajo se cubre de un material foto-resistivo o fotosensible que se ataca con un proceso de revelado por iluminación.

## **4.2 Pulido**

Los procesos de maquinado convencional remueven material de la pieza de trabajo mediante una herramienta de mayor dureza, generando irregularidades en las piezas con las cuales se trabajó, debido a la abrasión mecánica que se produce durante la remoción del material excedente, por lo que muchas veces se requiere de un tratamiento adicional, que corrija estas imperfecciones y mejore el acabado superficial o rugosidad en la superficie de la pieza (Dinesh Chopra, 2001).

Para realizar lo anterior existen diversos métodos, que se eligen de acuerdo con el material y acabado deseado, y que son: pulido mecánico y pulido por electrolisis, los cuales se detallan a continuación (Encarnacion Vera, 2013):

### **4.2.1 Pulido mecánico**

El pulido mecánico es un proceso que usualmente requiere de dos etapas: grueso y fino, como se refiere comúnmente; es decir, que el proceso se lleva a cabo en etapas secuenciales de desbaste que van mejorando paulatinamente la rugosidad de una superficie hasta lograr el acabado superficial deseado.

Principalmente este tipo de pulido es usado en materiales en los cuales la pieza no es de un material duro y se lleva a cabo con el uso de abrasivos, que se pueden encontrar en presentaciones de pastas o polvos. Es común que los abrasivos se encuentren adheridos a alguna superficie de apoyo, para facilitar su uso (comúnmente formando papel lija), pero también se pueden utilizar directamente sobre la superficie, empleando otro material para su aplicación, como paños o estopas. Algunos abrasivos para pulido mecánico son

1. Carburo de silicio (SiC), el cual es un polvo para pulir materiales metálicos duros.
2. Óxido de cerio (CeO), siendo el material más utilizado para el pulido de materiales metálicos de poca dureza como lo es el cobre y el aluminio.
3. Dicromato de Amonio ((NH<sub>4</sub>)<sub>2</sub>Cr<sub>2</sub>O<sub>7</sub>) al ser un abrasivo con ácido fluorhídrico (HF), es utilizado en aleaciones de Titanio y Circonio.

#### **4.2.2 Pulido por electrólisis**

Por otra parte, el electro-pulido, pulido por electrólisis o pulido electroquímico, es un tipo de pulido no convencional utilizado ampliamente en la metalografía (el estudio visual de las características de los metales), donde los materiales metálicos de gran dureza son difíciles de pulir mediante el proceso convencional. El uso de este método se basa en las siguientes ventajas:

1. Produce un desbaste mecánico y pulido en el mismo lapso.
2. El tipo de alimentación de corriente afecta el tipo de resultado obtenido, por lo que pueden utilizarse diferentes fuentes para realizar diferentes acabados.
3. Rectifica la superficie al preferir los picos de la superficie con algún nivel de rugosidad; es decir, es un proceso muy selectivo.

El pulido electroquímico es un método simple y económico para mejorar la calidad superficial de las piezas terminadas, que consiste en la disolución anódica del material, permitiendo remover inclusiones, tanto metálicas como no metálicas de la superficie, así como rugosidades, por lo que se puede mejorar la resistencia a la corrosión de las piezas, dándoles valor agregado (Yesid Peña, et al., 2010).

El proceso de pulido electroquímico tradicionalmente utiliza electrolitos ácidos para lograr los perfiles adecuados en la superficie de los metales avanzados como titanio o aceros inoxidable, con la adición de especies químicas agresivas y en altas concentraciones, para remover la capa de óxido en diversos materiales (Taylor & Inman, 2014), (Zhao, et al., 2003). Sin embargo, al ser una variante del proceso de electro maquinado, que hace uso de electrolitos salinos y bases, se pueden emplear estas sustancias para realizar el pulido electroquímico de metales como aceros al carbón o aceros de baja aleación.

### **4.3 Evolución de la tecnología para la manufactura no convencional**

Los sistemas mecánicos son de gran ayuda para la manufactura moderna puesto que muchos de los procesos requieren de una gran precisión al crear diversos componentes cada vez más pequeños, llevándolos a medidas milimétricas (mm) o micrométricas ( $\mu\text{m}$ )

donde el ser humano no tiene la capacidad de recrear estas tareas de forma repetitiva, con herramientas manuales.

Los sistemas mecánicos controlados por ordenador son dispositivos ampliamente utilizados en el ámbito de manufactura. Con ello los mecanismos se mejoran aún más, facilitando tareas que se pueden automatizar como lo son impresiones 3D, corte por láser, diseños de graficas e impresión de planos entre algunas otras aplicaciones (Hermoso Fernández, 2012).

Pero para llegar a este punto se pasaron por diversas etapas, como fue en 1808 siendo el comerciante francés Joseph Marie Charles quien realizara la invención de un telar programable, por medio de tarjetas perforadas. Sumándose Elmer Ambrose Sperry al desarrollar el sistema conocido como girocompás. Estas invenciones crearon una pauta para la automatización de tareas repetitivas, donde además se debe de descartar el factor humano, ya que una persona normal no podría realizar estas tareas siempre de la misma forma y con la precisión requerida. Esto fue demostrado por el ingeniero John Parson, que en 1947 inventó el sistema DIGITON para la fabricación de hélices de helicóptero. (Vargas Soto, 2010).

Posteriormente, la automatización de procesos dio un paso importante cuando el tecnológico de Massachusetts, en el año de 1953 creó una maquina CNC que permitía controlar una fresadora de 3 ejes. (Lamikiz Mentxaka, 2010). De esta forma se comenzaron a automatizar los procesos de maquinado convencional llegando a escalas industriales y sentando las bases para la manufactura moderna.

En los procesos no convencionales de manufactura, los sistemas de control numérico para el posicionamiento de la herramienta de trabajo también han sido de utilidad. Tal es el caso del trabajo del Dr. Chaides, quien utilizó el proceso de micro maquinado por electrodescargas ( $\mu$ EDM) para realizar micro canales en un acero estructural tipo A36, demostrando que se puede reproducir este proceso de manufactura no convencional con materiales de fácil adquisición. Por esto, empleó una computadora con conexión paralela para definir la trayectoria del electrodo por medio del software libre LinuxCNC, donde el mismo tuvo una serie de ajustes en su programación para que funcionara como una

aplicación específica, ya que originalmente este software está orientado a herramientas y fresado con control numérico (Chaidés, et al., 2009).

Para controlar el espaciamiento (*gap*) que existe entre la herramienta y la pieza de trabajo, se utilizó la placa de desarrollo de Arduino®, específicamente el modelo UNO, que permitía controlar en tiempo real el *gap* al ser un electrodo un alambre de bronce de 25 mm de diámetro para WEDM (del inglés *Wire Electrical Discharge Machining*), puesto que previamente se habían realizado pruebas con minas de grafito de 0.5 mm y estas mismas tendían a quebrarse al hacer contacto sobre la superficie de la pieza de trabajo, y con el electrodo de bronce se produce un sobre corte de aproximadamente 10 %. De este trabajo se deduce que es necesario realizar un monitoreo constante del espaciamiento para reducir estos resultados de sobre corte, o incluso pérdida de la herramienta de trabajo, como también el realizar pruebas en diferentes materiales, puesto que solo se obtuvieron resultados del metal tipo acero A36 (Chaidés Zuñiga, 2012).

Otro proceso de manufactura no convencional fue aplicado por los Ing. Estrada y León, al hacer uso del proceso de ECM para realizar moldes en placas con un área máxima de trabajo de 50×50×1.2 mm (Estrada Hernández & León Bravo, 2014). En sus pruebas para obtener los mejores resultados de funcionamiento de su sistema, hacen uso de un electrolito a base de *NaCl*, que se caracteriza por ser uno de los electrolitos que funciona bien con casi cualquier tipo de material conductor como ya ha sido demostrado también por otros autores (Granda Gutiérrez, et al., 2015).

En la experimentación realizada por Estrada y León, sus resultados muestran que los mejores parámetros de funcionamiento son: una concentración de 300g/l de sal disuelta en agua, una corriente de 75A y un *gap* de 6 mm, que permitía un flujo constante del electrolito. Pero es necesario recubrir la herramienta de trabajo con algún material no conductor que forme una película protectora que creará el molde y recubrirá las áreas donde no se necesita que el proceso de ECM se lleve a cabo (Estrada Hernández & León Bravo, 2014).



#### **4.4 Interfaz de usuario**

Es necesario desarrollar un programa de propósito específico para el proceso ECP que se detalla en ese trabajo, al crear una interfaz que pueda manipular de forma simple los movimientos del electrodo, debido a que los diferentes programas de libre distribución o comerciales que se han identificado hasta el momento se encuentran enfocados a procesos de mecanizado convencional.

Del ya expuesto trabajo del Dr. Chaides, donde fue necesario modificar la estructura del programa principal al realizar diversos cambios del software LinuxCNC (Chaides, et al., 2009) o del Ing. León con el uso de software comercial *MACH 3 DEMO* (Estrada Hernandez & León Bravo, 2014), se puede notar que ambos programas necesitan realizar la programación del código G de forma manual, al ser sistemas específicos que requieren de parámetros definidos para realizar una correcta ejecución del proceso que se desee llevar a cabo, EDM o ECM, respectivamente.

En el presente trabajo, si bien es un derivado de ECM, el proceso ECP tiene sus particularidades, por lo que se hace necesario considerar una interfaz diseñada, también específicamente para dicho proceso, contribuyendo así al uso de herramientas adicionales para este proceso de manufactura.

Una mejora al sistema del Dr. Chaides es que, mientras que este hace de una computadora de sobremesa para el desarrollo de la interfaz, la propuesta de la presente tesis es hacer uso de un computador de placa reducida, minimizando el espacio necesario, pero además integrar todo el sistema mediante la aplicación del concepto de cómputo embebido.

#### **4.5 Raspberry Pi 2 B+**

El elemento central de sistema de cómputo embebido que se propone en esta tesis es la computadora de placa reducida Raspberry®, donde se podrá almacenar y ejecutar el programa de control, al contar con pines que permitan manipular el sistema de ejes coordinados como interactuar con los dispositivos de seguridad del sistema y monitoreo de los valores adquiridos por medio de la instrumentación electrónica.

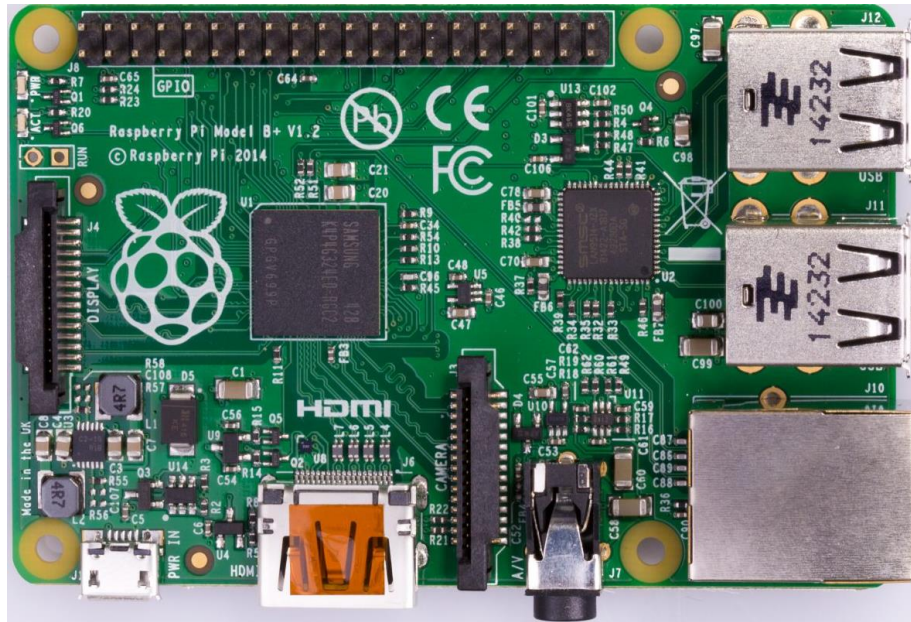
La interacción de este sistema con el prototipo ECP se realizará mediante una interfaz gráfica. Al ser un computador basado en software y hardware libre, posee un sistema operativo (S.O.) basado en Linux, llamado Raspbian, desarrollado por la misma empresa (Raspberry Pi Org, 2018). Debido a la característica de ser un S. O. de propósito general podrá cumplir con los requerimientos mínimos necesarios para el desarrollo de la interfaz de usuario y del software de monitoreo.

La placa Raspberry Pi, que se aprecia en la Figura 1, cuenta con las siguientes características (Raspberry Pi Org, 2018):

1. Sistema de alimentación por medio de un puerto micro USB el cual funciona con una alimentación de 5V y corriente de 500 mA ~ 2 A, por lo cual es posible alimentarlo con cualquier cargador que cubra estas características.
2. Puertos USB 2.0 que permiten tener una conexión con diversos periféricos con la misma conexión.
3. Un puerto de red con conector RJ-45 para poder conectar el computador en línea si es que así se desea.
4. Un conector para video tipo HDMI para visualizar el contenido que se esté procesando en ese momento en un monitor.
5. Un puerto para microSD que permite almacenar el sistema operativo con capacidad de lectura de hasta 64 GB (en su versión B+)
6. Procesador Broadcom BCM2836 con arquitectura ARM v7 @ 700 MHz
7. Memoria RAM de 512 MB @400MHz

Para la configuración e instalación del S.O, se puede realizar mediante la descargar y copia de los ficheros que se tienen en la página del fabricante por lo cual solo hay que seguir las instrucciones disponibles en dicho sitio.

El sistema operativo tiene precargado diversos programas de software libre como lo es Python, que se utiliza para programación multi paradigma que permitirá crear la estructura del programa base de control por medio de módulos de desarrollo.



*Figura 1: Computador Raspberry Pi modelo B+*

Para el vaciado de datos se hace uso de una interfaz que sea programada en conjunto con Python, la cual será la biblioteca grafica Tkinter (Grayson, 2000). Para ello se adaptará un conjunto de scripts que permitirá crear las diferentes bases para la interacción con el usuario, como lo son ventanas, botones, cuadros de texto entre otros *widjets*, los cuales son funciones predefinidas de dicha biblioteca.

Esta computadora de placa reducida ha sido implementada en diversos proyectos que tienen como objetivo controlar y manipular un sistema que ajuste el posicionamiento de algún sistema mecánico como lo son impresoras 3D o corte por láser, como también el control domótico de un sistema inteligente, sistema telemétrico, servidor o clúster, sistema multimedia, entre otros muchos proyectos que se desee realizar a cabo con base en este computadora (Raspberry Pi Org, 2018)

De igual forma se suman diversos proyectos que se basan en la construcción, modificación o mejora de diferentes máquinas de control numérico computacional *CNC*, grabadoras laser, que son construidos mediante sistemas discretos o directamente con placas programables que permiten desarrollar un software para el control de cada acción a realizar.

Muestra de ello es el prototipo de una fresadora desarrollado por el Ing. Riquelme que es utilizado para la fabricación de placas de circuito impreso PCB (de las siglas en inglés Printed Circuit Board) en el cual implemento la tarjeta de desarrollo Arduino, modelo UNO rev. 3. (Miguel, 2014).

Adicionalmente, existen diversos proyectos que se basan en el computador Raspberry para implementar el control de un sistema de ejes coordenados, en línea se encuentra una gran cantidad de estos, como ejemplo, Xiang describe paso a paso cómo construir una cortadora laser reutilizando componentes de grabadoras o reproductores de video (Zhai, 2017)

Como también lo demostró Catarino, al implementar esta tarjeta para controlar la remoción de material en un proceso de maquinado electroquímico, donde la ya mencionada computadora se encarga de controlar el movimiento que tendría un electrodo, con la ayuda de actuadores y drivers conectados a los pines GPIO, mismos que se encargan de gestionar la ruta preestablecida por un usuario a través de una interfaz de usuario (Catarino Aguilar, 2017).

## **5 MÉTODO**

En este apartado se presenta el proceso de integración de un prototipo de pulido electroquímico, a partir de una plataforma didáctica de movimiento cartesiano, a la que se le han incorporado los elementos necesarios para trabajar como un sistema de posicionamiento en tres ejes; dos de los ejes controlan la posición de la pieza de trabajo (horizontal y vertical), en tanto que el otro eje (desplazamiento arriba-abajo) controla el movimiento de un electrodo polarizado con corriente directa.

Se incluye también una descripción de la integración del sistema mecánico, que en conjunto con motores a pasos permite el movimiento del electrodo para recorrer la superficie de la pieza a pulir, con diferentes trayectorias.

Además, se describe la instrumentación utilizada para adquirir los datos del proceso, mediante el uso de sensores electrónicos.

Posteriormente se define la estructura del desarrollo de la programación del sistema (software), que determinarán las características de funcionamiento para manipular el prototipo por medio de una interfaz, dando al usuario un fácil control a su aplicativo el cual será implementado para poder reproducir experimentos de pulido electroquímico.

### **5.1 Especificaciones de diseño**

El planteamiento de los requerimientos mínimos se describe en esta sección, recopilando información que permita definir las necesidades para describir lo que el sistema debe de realizar, de igual forma contar con las especificaciones que requiere el prototipo, consiguiendo tener un sistema unificado y de esta forma contar con las herramientas necesarias para su desarrollo.

Tomando en cuenta el análisis de trabajos previos, se pueden definir las siguientes consideraciones, para ser aplicadas en el presente trabajo:

1. Monitorear el voltaje y corriente del proceso
2. Calcular la distancia gap aproximada que existe entre la pieza y herramienta de trabajo con lo cual se tendrá un buen funcionamiento del proceso electroquímico.

3. Hacer uso de un electrolito a base de cloruro de sodio (*NaCl*) con una proporción del 30% sobre litro.
4. Recubrir las áreas de la herramienta de trabajo para reducir sobre cortes o perturbaciones no deseadas.
5. Tener una variedad de trayectorias para que la herramienta se desplace sobre la pieza de trabajo, previo a que el usuario haya indicado sus dimensiones.
6. Realizar pruebas en una variedad de materiales para asegurar el correcto funcionamiento del sistema.

## **5.2 Diseño conceptual del sistema**

Para el diseño del prototipo se hace un análisis del funcionamiento del proceso de pulido electroquímico, puesto que es donde se verá reflejado el funcionamiento del sistema. Se puede observar que, para la ejecución de pruebas de maquinado no convencional, la gran mayoría de sus variantes hacen uso de energía eléctrica de corriente directa para remover el material de la pieza hacia la herramienta de trabajo de acuerdo con la ley de Faraday, la cual establece que la masa desprendida del electrodo positivo en una celda electrolítica es proporcional a la cantidad de corriente eléctrica que circula por dicha celda (Granda Gutiérrez, et al., 2015).

De esta forma se definen los componentes que cumplen con la premisa anterior y los cuales serán implementados para efectuar de forma satisfactoria el proceso de pulido electroquímico, describiéndose de la siguiente forma:

### **Fuente de alimentación**

Este aparato electrónico es la fuente que proporciona la energía eléctrica, mediante el voltaje de polarización y la corriente consumida entre los electrodos; la corriente fluye a través de los electrodos produciendo un proceso de electrolisis, método indispensable para la ejecución de mecanizado avanzado (Grover, 2007).

Para este proyecto se utilizó una fuente de corriente directa de laboratorio, con voltaje ajustable y protección de corriente, marca BK Precision, modelo 1760A. Esta fuente permite modificar las condiciones de polarización de la celda electrolítica, para realizar diferentes ensayos de pulido o incluso de maquinado electroquímico en corriente directa.

## **Electrodos**

Los electrodos se conforman, por una parte, por el material a procesar, y por la otra por el electrodo de referencia. Estos elementos son definidos también como pieza y herramienta de trabajo, respectivamente. Son un conjunto de piezas metálicas, mismos que al ser de un material conductor promueven el flujo de la energía eléctrica desde la fuente de polarización, puesto que van conectados directamente a las terminales de salida de la fuente.

En este arreglo de electrodos, el ánodo corresponde a la pieza de trabajo, mientras que el cátodo es la herramienta, que en conjunto con la solución electrolítica implementada crean un intercambio de electrones y iones, lo que da origen al desprendimiento de material desde la pieza de trabajo (Catarino Aguilar, 2017). Durante el proceso, los átomos de la pieza de trabajo se desprenden como iones que fluyen al electrolito.

Para el desarrollo de este proyecto, se utilizó acero al carbón como pieza de trabajo, mientras que como electrodos se usaron tubos de cobre y agujas de acero inoxidable.

## **Electrolito**

Se trata de un compuesto o disolución acuosa a base de ácidos, bases o sales que tienen una gran cantidad de aniones y cationes que aumentan la capacidad de conducción de la corriente eléctrica. El agua por sí misma es un aislante eléctrico, pero al contener estos electrolitos se transforma en un medio conductor.

Se han estudiado diversas composiciones de electrolitos y aquella que tiene mejores resultados con casi cualquier tipo de electrodos es el Cloruro de Sodio (*NaCl*), como lo demuestran distintos autores, por ejemplo, Estrada y León en sus pruebas (Estrada

Hernandez & León Bravo, 2014), al igual que Granda y colaboradores (Granda Gutiérrez, et al., 2015); estos autores concuerdan con la mezcla de sal y agua, seguidamente del Bromuro de Sodio (*NaBr*) como una posible alternativa.

### **Bomba de agua**

Es un motor eléctrico que tiene una adaptación para inducir movimiento en un medio acuoso, generalmente mediante el uso de impulsores. Aplicada a este prototipo, la bomba permitirá generar un flujo del electrolito a través de la herramienta de trabajo, lo que beneficia durante el proceso, al remover de la superficie de la pieza de trabajo los diversos lodos que se forman durante la reacción electroquímica, como lo demostró el autor Catarino, donde en algunas de sus pruebas se obtenían malformaciones a causa de algunas pequeñas obstrucciones en el electrodo debido a estos lodos, que ocasionaban distintos problemas al realizar el proceso de maquinado (Catarino Aguilar, 2017).

### **Sistema mecánico de ejes coordenados**

El dispositivo mecánico utilizado para el desarrollo de este proyecto es una plataforma didáctica de la marca Zen Tool Works (Zen ToolWorks, Inc., 2018). Una de sus principales características por la que fue seleccionado, es que está construido principalmente de placas de *Nylamid*, un material a base de plástico, lo cual es una ventaja ya que para llevar a cabo las pruebas de pulido electroquímico se hará uso principalmente de un electrolito que es una disolución de  $H_2O + NaCl$  el cual genera una gran corrosión a los materiales metálicos.

Adicionalmente permite realizar movimientos mecánicos en uno o más ejes con la ayuda de motores a pasos (paso a paso, *PaP*), puesto que se puede controlar el sentido de giro, velocidad, fuerza de empuje (torque) y precisión de movimiento por medio de secuencias generadas en las bobinas de dichos motores.



Este tipo de dispositivos son utilizados en diversas aplicaciones que tengan que ver con procesos de maquinado como lo son CNC, cortadoras laser, impresión en 3D entre otros (Andrade Fuentes, 2012).

### Integración de los elementos básicos

A partir de los elementos descritos, se genera una visión de lo que es el concepto de funcionamiento del proceso de pulido electroquímico, como se puede apreciar en la Figura 2, donde se propone el diseño y la conjunción de los componentes principales para el montaje del prototipo propuesto para este proyecto. Cabe destacar que el flujo del electrolito en el modo de pulido electroquímico se realiza de manera lateral al movimiento del electrodo, en tanto que, en otros procesos, como el maquinado electroquímico, el electrolito se hace fluir por el interior del electrodo, que generalmente es hueco.

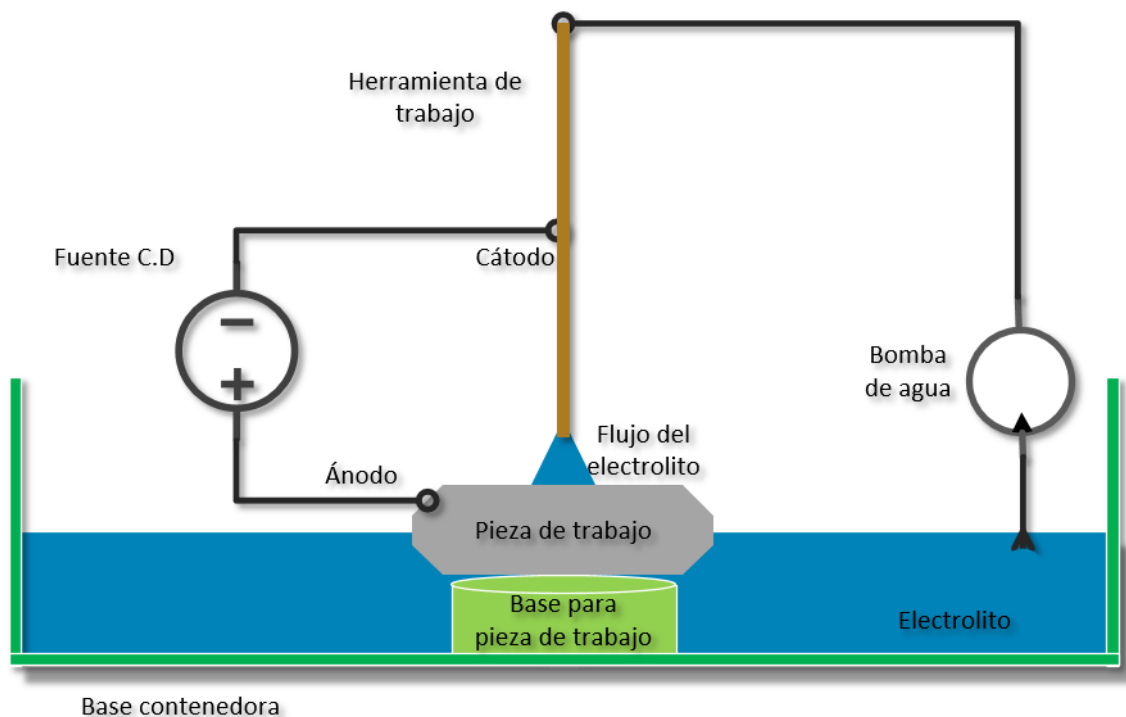


Figura 2: Esquema básico para el proceso de pulido electroquímico

Con lo anterior se definirán en la siguiente sección los dispositivos que se adicionaron para adecuar el sistema mecánico de forma que sea posible realizar pruebas de pulido con un voltaje de 5 V y una corriente que dependerá de la distancia inter electrodos (*gap*) con la que se trabaje (desde los 40 $\mu$ m hasta 300 $\mu$ m), como también del material de los electrodos con la propuesta de utilizar el electrodo de cobre y acero quirúrgico para la herramienta de trabajo (terminal negativa) y para la pieza de trabajo, la implementación de los aceros; 10-12, 10-15, 10-18, 106-12, HSLA, a fin de crear una base de conocimiento para el pulido no convencional y concretamente validar el funcionamiento del prototipo propuesto en este proyecto.

Posteriormente se analizarán los requerimientos de sistema para desarrollar la interfaz que permitirá manipular el movimiento del sistema mecánico y en general gestionar el funcionamiento del prototipo, mismo que estará gobernado por el usuario que realizará la ejecución de pruebas puntualiza de mejor forma cada punto de funcionamiento en el apartado 5.6.

### **5.3 Adecuación del sistema mecánico cartesiano**

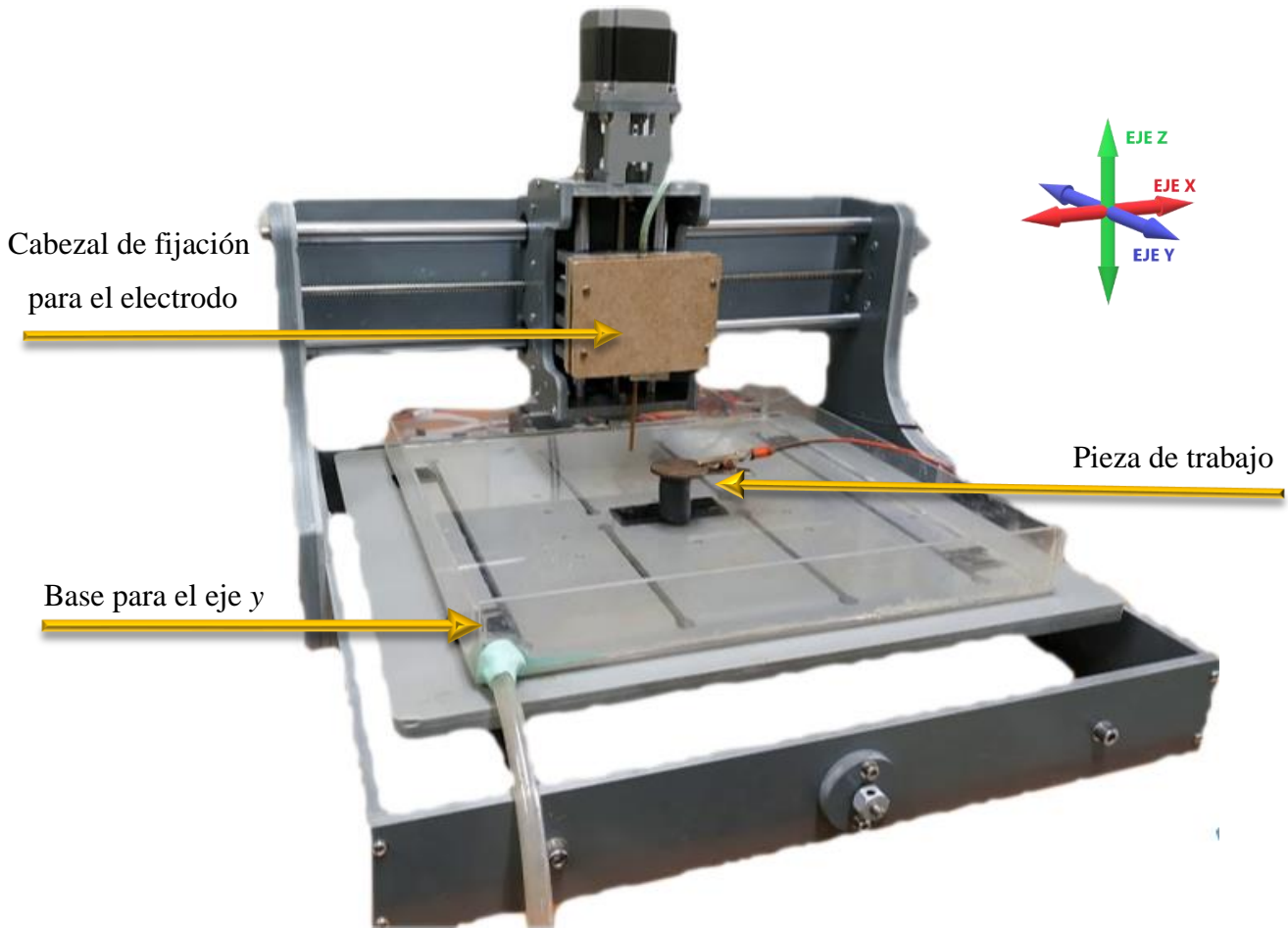
Los sistemas mecánicos empleados para los procesos de maquinado no convencional son un conjunto de componentes para generar un movimiento en trayectorias bien definidas, con una fuerza y velocidad específica, con ayuda de un controlador que le permite determinar la acción de acuerdo con el trabajo que se necesite realizar (Díaz del Castillo Rodríguez, 2008).

El sistema mecánico de ejes coordenados cuenta con un espacio efectivo de trabajo de 12 $\times$ 12 $\times$ 6 pulgadas. Dicho sistema se aprecia en la Figura 3, en el cual se indican algunos de los ajustes necesarios para adecuar a este dispositivo de forma que cumpliera con las especificaciones para el mecanizado avanzado para pulido electroquímico.

Las adaptaciones realizadas, como la adición de algunos componentes se mencionan en el siguiente listado:

### Cabezal para electrodo

Se creó un soporte para fijar en el cabezal de movimiento del eje z, puesto que la estructura de fijación es para un motor. La incorporación de este elemento se muestra en la Figura 3. Consiste en un soporte con dos placas que permitirán fijar al electrodo reduciendo perturbaciones que podrían tener como consecuencia alteraciones en los resultados obtenidos.



*Figura 3: Adaptación del sistema de ejes coordenados para el proceso de pulido electroquímico*

Las dimensiones con las que se diseñó el soporte del electrodo son de 10 cm de ancho por 7.5 cm de alto, lo que aprovecha la estructura del eje z como también de las cuatro

perforaciones de 3/8 de pulgada con las que ya cuenta Figura 4A, lo que permite empotrar al electrodo de una forma adecuada Figura 4B.

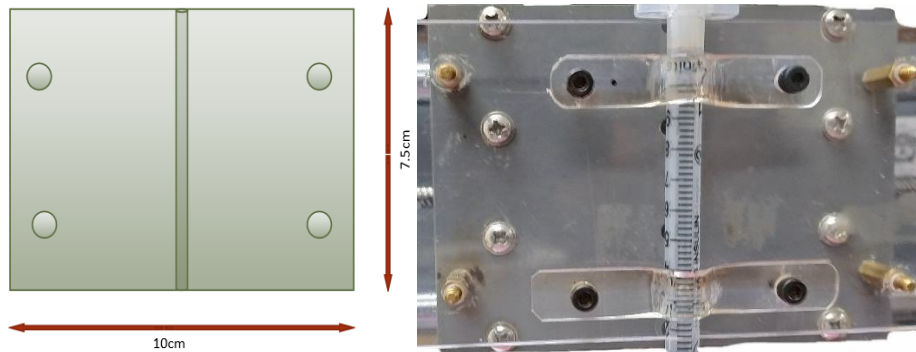


Figura 4A) Esquema del soporte, Figura 4B) Soporte de la herramienta

### Finales de carrera

También conocidos como *end-stop* o *limit-switch* (Figura 5), son dispositivos que envían un pulso de activación (1 lógico) cuando es presionado, mientras que en su posición natural, mantiene un nivel de 0V (0 lógico); estos módulos se adicionaron para delimitar y asegurar el espacio de trabajo, utilizando dos de ellos para cada eje dando un total de seis, mismos que son conectados al computador el cual decidirá las funciones que se cumplirán dependiendo del interruptor activado.

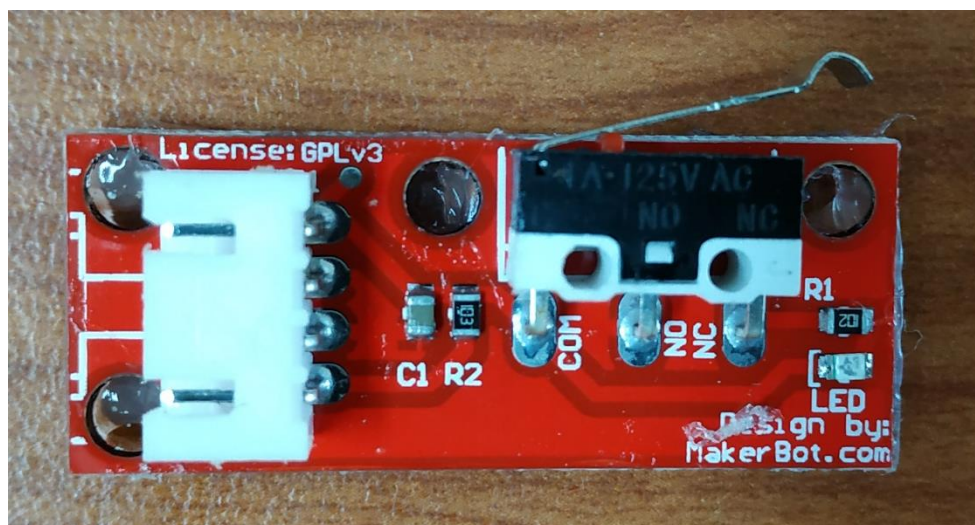


Figura 5: Modulo interruptor de fin de carrera

### **Base contenedora**

Este recipiente funge como depósito del electrolito y el lodo generado tras pruebas de pulido electroquímico después de ser utilizado por la herramienta y pieza de trabajo. Adicionalmente no puede exceder los 3.5cm de altura ya que es la distancia de separación que existe entre el punto más bajo del cabezal (eje z) con la base (eje y), por lo que, si la herramienta es trasladada a un punto que este fuera del recipiente, el cabezal podría mover o arrastrar el recipiente causando algún problema con la prueba como también el vaciar el contenido del contenedor y afectar el sistema electrónico.

De esta forma la altura máxima del recipiente puede ser igual o menor a 3.5cm, por lo que la profundidad y ancho del recipiente pueden ser equivalentes al espacio efectivo de trabajo, quedando un recipiente con un tamaño máximo de 12×12×1.4'' pulgadas.

### **Sistema de bombeo**

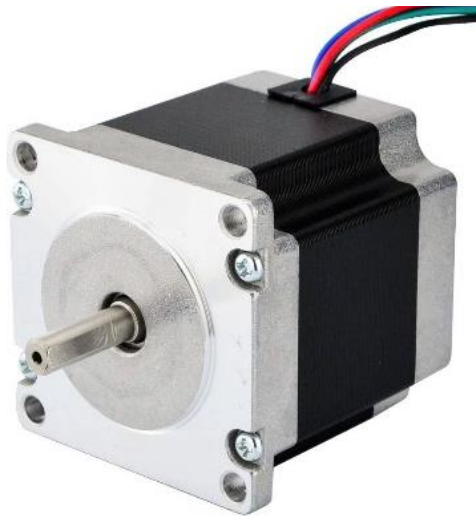
Es un conjunto de dispositivos que permiten enviar el electrolito hacia el electrodo hueco y remover el material generado durante el maquinado al existir un flujo constante. Para ello se implementó una bomba de agua de tipo centrifugadora con un voltaje de funcionamiento de 12 V; al ser una bomba compacta, construida en fibra de vidrio y sumergible, la cual tiene un flujo de hasta 800 litros/hr., dependiendo del voltaje de entrada, por lo cual se puede asegurar su buen funcionamiento aun con el electrolito, ya que este puede causar corrosión por su composición.

De igual forma se utiliza una manguera de cánula nasal, puesto que su flexibilidad permite manipularla evitando pliegues que interrumpan el flujo del electrolito como también que exista alguna fuga que pueda afectar el movimiento del eje x o afectar el funcionamiento del dispositivo. Esta manguera tiene un diámetro externo de 4.72 mm y diámetro interno de 3.65 mm, misma que va conectada directamente desde la bomba hacia el electrodo de trabajo.

## 5.4 Controlador de motores a pasos

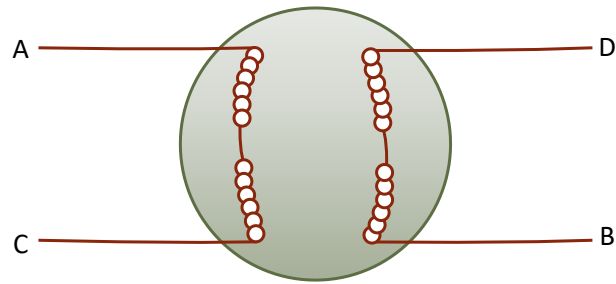
El sistema cartesiano didáctico incluye motores a pasos, que son los de mayor uso para este tipo de sistemas al otorgar una buena precisión de los movimientos al controlar la posición y la dirección de giro, dependiendo de la secuencia de trabajo, como de la configuración del motor.

El sistema mecánico cuenta con tres motores a pasos (Figura 6), de clasificación NEMA 23, modelo 67BYGH56-601A, con un voltaje de trabajo de 12~24V y una corriente máxima de trabajo de 3A, el ángulo de paso es de  $1.8^\circ$ , equivalente a 200 pasos por vuelta completa y la distancia de avance es de 8 mm por revolución, ya traducido al movimiento lineal del sistema mecánico.



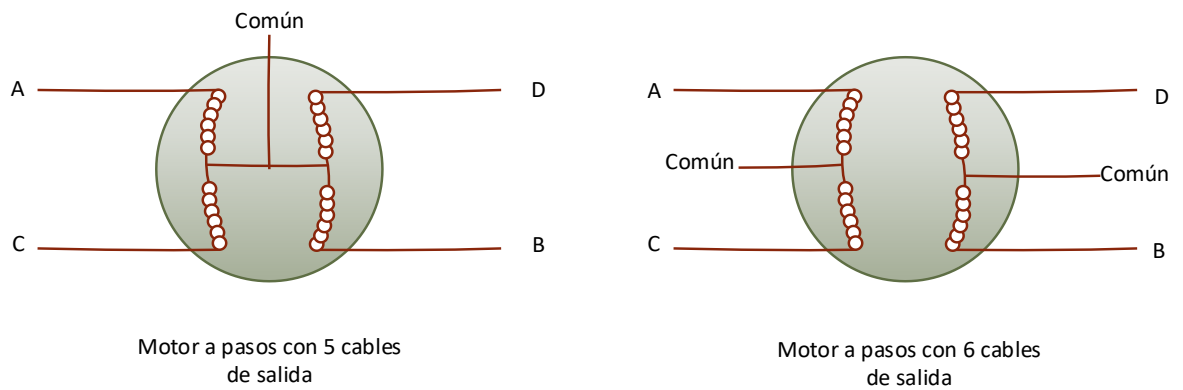
*Figura 6: Motor a pasos NEMA 23*

Es necesario conocer el funcionamiento de este tipo de componentes ya que existen diferentes configuraciones con las que estos motores trabajan. Usualmente se dan en dos casos: motor bipolar, el más común es el de cuatro cables de conexión (Figura 7) y motor unipolar con 5 y 6 hilos (Figura 8). Con lo cual los motores unipolares pueden usarse como bipolares si sus puntos de conexión *común* se pueden desprejar de la configuración.



Motor a pasos con 4 cables de salida

*Figura 7: Conexión interna de motor bipolar*



Motor a pasos con 5 cables de salida

Motor a pasos con 6 cables de salida

*Figura 8: Conexión interna de motor a pasos unipolar*

Una vez que se conoce su configuración se debe de comprender como generar el movimiento de rotación del eje, puesto que el orden de la secuencia que se les dé a cada conexión cambiara la posición y el sentido de giro. Describiendo la sucesión que se debe de seguir para ejecutar las transiciones de funcionamiento de la Tabla 3, se energiza una de las terminales de cada bobina para hacer girar el eje del motor dependiendo del orden que se tenga, pasa de igual manera para la configuración de medio paso solo que, para generar el movimiento de esta forma se crean paradas intermedias en las bobinas lo que da una mejor resolución pero se reduce la fuerza de torque.

Tabla 3: Secuencia a paso completo.

PASO	A	B	C	D
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

Tabla 4: Secuencia a medio paso

PASO	A	B	C	D
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

Puesto que las secuencias siempre llevan el mismo orden, es posible construir un circuito que se encargue de estos cambios, por lo cual se le puede llamar como controlador, también conocido como *driver*, el cual recibe las instrucciones de movimiento mediante combinaciones lógicas configurando el sentido de giro (sentido horario o anti-horario), resolución (paso completo o medio paso) como también la velocidad (tiempo de cambio entre cada secuencia) con la combinación de tres conexiones en su entrada.

En el diagrama de la Figura 9 se muestra un arreglo para controlar un motor a pasos, contando con 6 entradas de control (STMicroelectronic, 2001). Donde:

- $cw/\overline{ccw}$ . *clock wise/counter clock wise*: Esta entrada define el sentido de giro (horario o contra-reloj) que tendrá el motor al momento de funcionar.



- $\overline{clock}$ . Define la entrada para la frecuencia de trabajo en  $Hz$ . Es activado cuando el cambio de estado del tren de pulsos hace una transición de 1 lógico a 0 lógico.
- $half/full$ . Entrada que define el tipo de resolución que tendrá el motor, puesto que al estar en *cero* permitiría tener exactamente 200 pasos ( $360^\circ/1.8^\circ$ ) en configuración a paso completo y al establecer el funcionamiento a medio paso se permite tener pasos “intermedios” como se menciona en la Tabla 1.2 obteniendo un resultado de 400 pasos por vuelta.
- $\overline{reset}$ . Restablece la posición inicial del motor en el estado *Home* del C.I como  $ABCD = 0101$
- $enable$ . Habilita el funcionamiento del controlador mediante el cambio de un pulso lógico a 1 y se deshabilita cuando este pulso cambia a 0 lógico.

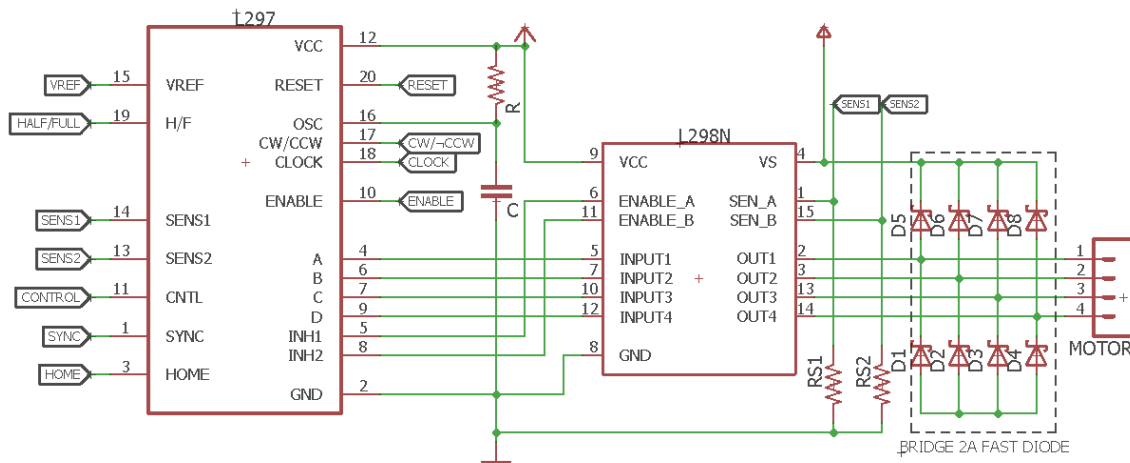


Figura 9: Controlador prediseñado con el uso de los C.I. L297 y L298

De igual forma se puede hacer uso de controladores prefabricados: *módulos* que tengan un rango de funcionamiento igual o mayor al necesario para cada motor que se desee controlar. De esta forma se pueden implementar circuitos o sistemas programados que gobiernen el funcionamiento y movimiento de los motores, como se puede dar en el procesamiento de un computador el cual envíe los comandos al controlador para establecer el movimiento que realizara el sistema mecánico.

Por ello se propone implementar en este prototipo utilizando el controlador TB6560, el cual tiene un voltaje de funcionamiento que va desde los 10 V hasta los 24 V, una corriente de 300 mA hasta 3 A para el motor que se desea controlar; esta variación es posible gracias a la modificación por medio de los interruptores 1,2 y 3. De igual forma se puede variar la resolución de paso, es decir, mejorar la resolución de movimiento llegando hasta 3200 pasos (1/16 de paso) por revolución, tomando en cuenta los motores que se utilizan en el sistema mecánico.

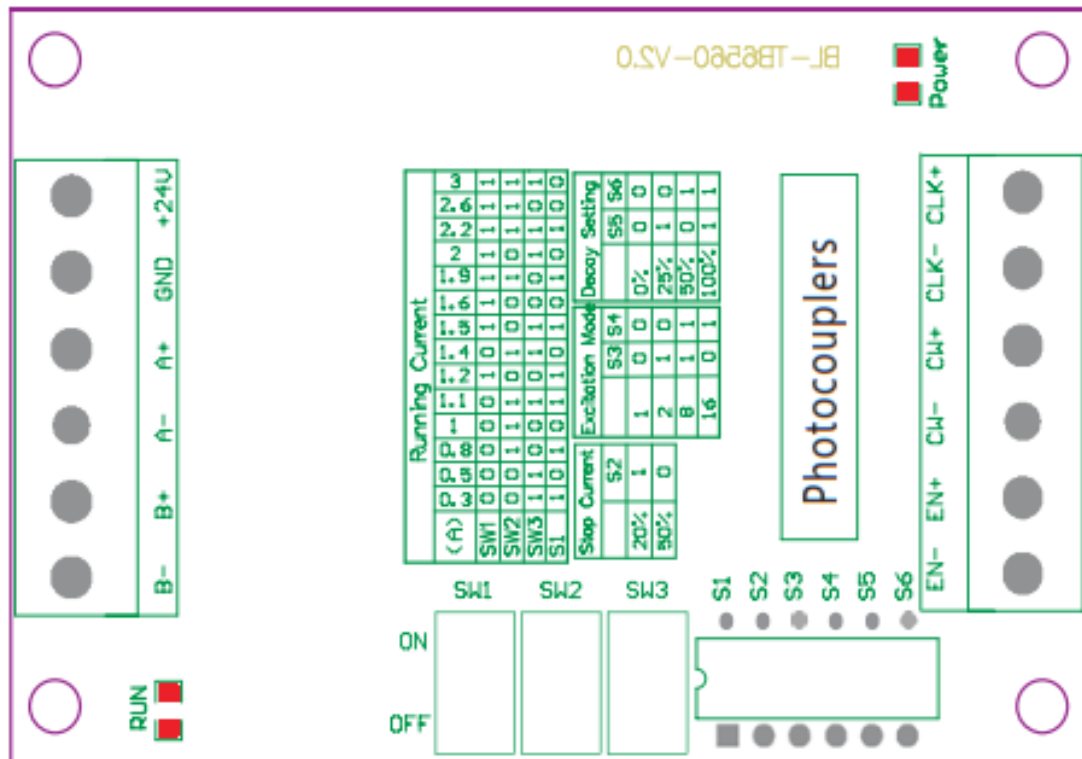


Figura 10: Modulo controlador TB 6560 (Shenzhen Global Technology Co., Ltd., 2015)

## 5.5 Diseño del hardware asociado al sistema embebido

Para el diseño del prototipo se hace uso de los dispositivos de la Tabla 5, toma como base la computadora ya mencionada Raspberry Pi® B+ en su versión 2, de la cual se tienen 26 pines GPIOs que podrán ser utilizados para controlar y monitorear las pruebas de ECP mediante una interfaz de control desarrollada y descrita en el apartado 5.6

De esta forma serán designados tres GPIOs para controlar un eje de movimiento del sistema mecánico mediante los módulos TB6560, con lo cual tres pines se encargarán del encendido y apagado del controlador (EN+), sentido de giro (CW+) y la velocidad del giro (CLK+) respectivamente mediante pulsos digitales (0 y 1 lógico), dando como resultado un total de 9 pines para esta acción.

*Tabla 5: Lista de componentes requeridos para el prototipo de pulido electroquímico*

Cantidad	Componentes
1	Computador de placa reducida Raspberry Pi 2 B+
1	Mando de control (Joystick) Genius Max Fighter F-16U
3	Modulo driver de accionamiento para motores TB6560
6	Interruptor de fin de carrera
1	Bomba sumergible centrifuga, CD 12V, 22W
1	Sensor de corriente ACS712 30A
1	Circuito integrado tipo ADC-MCP3204 de 12bits
2	Fuentes de poder BK PRECISION 1760A

Para el apartado de monitoreo se cuenta con una desventaja, puesto que la computadora únicamente cuenta con E/S de tipo digital, lo que no permite realizar una medición en el formato analógico, que es como se entregan sus respectivas mediciones de estas variables, sin embargo cuenta con los protocolos SPI y  $I^2C$  (del inglés Serial Peripheral Interface e Inter-Integrated Circuit, respectivamente), que son dos protocolos de comunicación utilizados para comunicar diferentes periféricos con cuatro líneas de conexión para SPI y solo dos líneas para  $I^2C$  (Leens, 2009).

Es por ello que se utiliza el circuito integrado ADC MCP3204 para adquirir la lectura de las variables analógicas de voltaje y corriente, siendo un conversor de tipo analógico/digital (ADC, por su siglas Analog to Digital Converter) de aproximación sucesiva el cual se comunicara con la placa del computador mediante la interfaz SPI

(Microchip Technology Inc., 2002). En la Figura 11 se pueden observar las conexiones que se utilizarán para comunicar el C.I con la Raspberry.

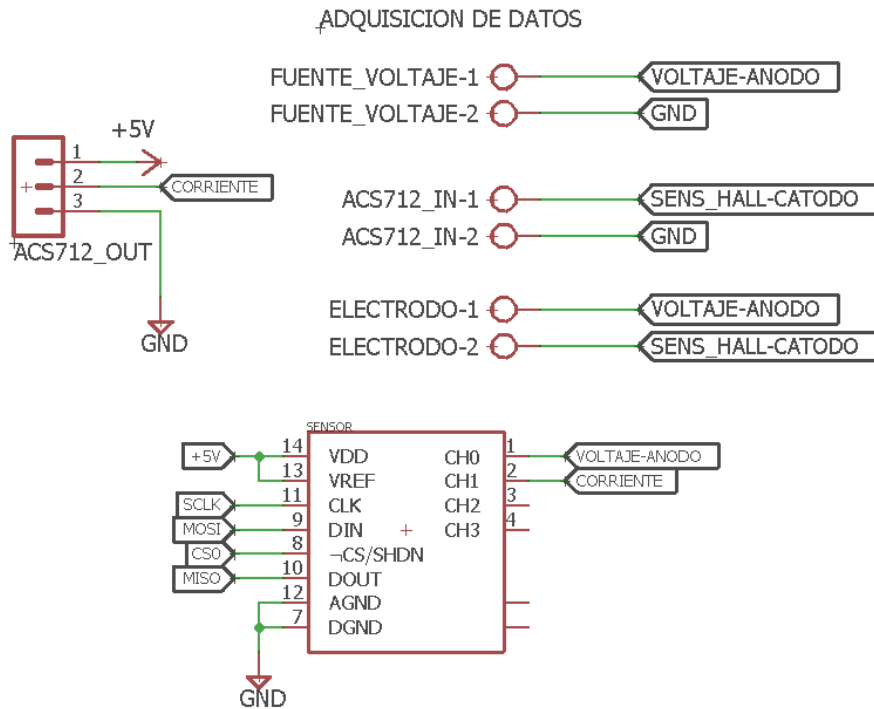


Figura 11: Diagrama de conexión para adquisición de datos

Adicionalmente, para la medición de la corriente se usará el módulo ACS-712 que tiene una sensibilidad de 66mV/A y tiene un rango de funcionamiento de  $\pm 30$  A. Este sensor es de *efecto Hall*, el cual detecta el campo magnético producido por la corriente que circule donde se esté realizando la medición. La salida de la medición que se proporciona es un voltaje con un rango de 0V a 5V en función de la corriente obtenida.

De esta forma el integrado MCP será conectado a los pines GPIO 8, 9, 10 y 11 definidos como CS0, MISO, MOSI y SCLK para enviar la información obtenida por medio del protocolo SPI para que el computador se encargue de monitorear y almacenar los parámetros obtenidos.

Seguidamente, se designarán seis pines de la computadora para ser conectados con 6 módulos finales de carrera (dos por eje, posición positiva y negativa), permitiendo

delimitar el espacio de trabajo y de esta forma se asegurará el correcto funcionamiento del sistema mecánico.

Una vez definidas las conexiones y componentes con los que contara el sistema, se designan las posiciones de cada uno de los conectores a utilizar Figura 12, unificando el apartado de adquisición de datos con las conexiones de los finales de carrera en una sola placa que podrá ser montada sobre el computador, dando al usuario una mayor facilidad para conectar todos los componentes, teniendo un mejor orden y reduciendo la posibilidad de un fallo en la conexión de algún componente.

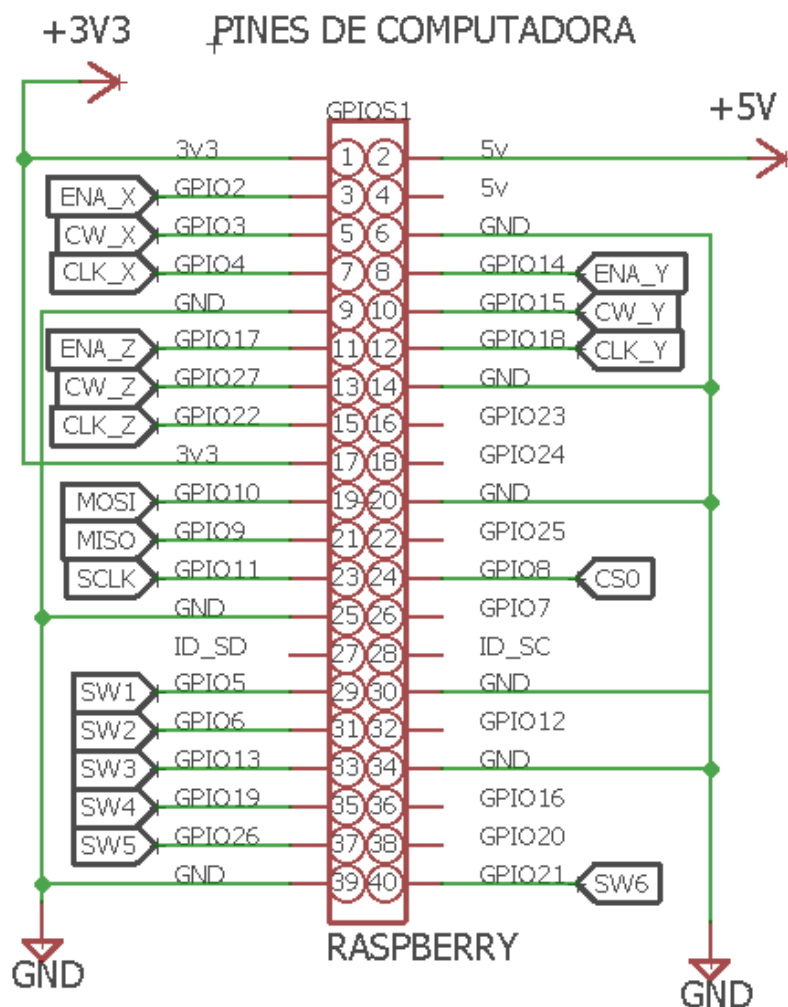


Figura 12: Circuito de control para el sistema mecánico.

## 5.6 Diseño de la interfaz de usuario

En este apartado se proponen la implementación de diversos métodos, funciones, módulos, bibliotecas de software y variables para que el prototipo funcione de una forma eficiente donde esta premisa podrá ser comprobada al momento de realizar las diversas pruebas de pulido electroquímico.

Para ello se analizan los requerimientos que debe cumplir el prototipo basándose en el diagrama de caso de uso Figura 13, donde se puede denotar que el actor manipula la interfaz a fin de controlar el movimiento del sistema mecánico, monitorear las variables de voltaje, corriente y distancia entre electrodos, como también la posibilidad de almacenar los registros de estos datos los cuales serán; resolución de cada eje, dimensiones del espacio, herramienta y pieza de trabajo, aleación de los electrodos, como también las características del electrolito y su concentración (Pressman, 2010).

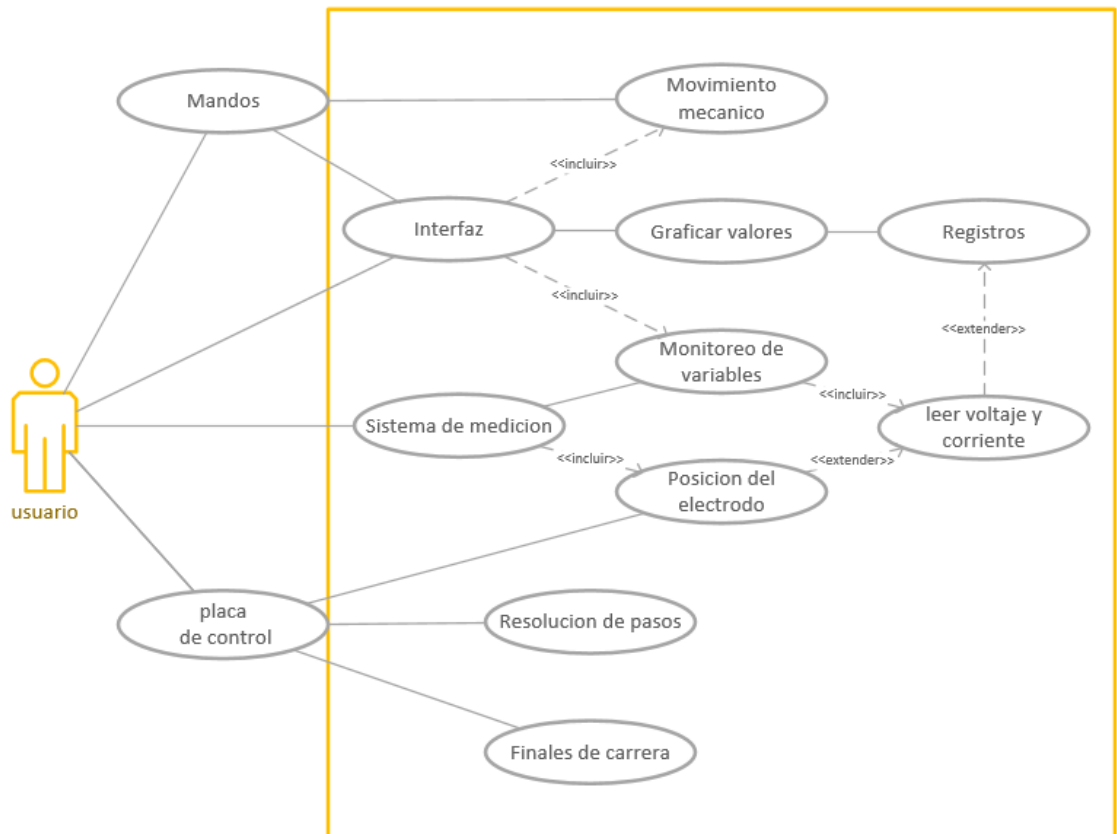


Figura 13. Diagrama de caso de usos para el software de control

Con base en las características de funcionamiento, se prosigue con la selección de las herramientas a implementar para el desarrollo y diseño de la interfaz de control, designando el lenguaje de programación Python en su versión 2.7 puesto que esta herramienta de programación, además de ser incluida de forma automática al ser instalado el sistema operativo Raspbian, tiene un bajo consumo de recursos de cómputo y una gran cantidad de documentación, lo que la convierte una opción apropiada para desarrollar el apartado de control.

Adicionalmente se incluirá la librería grafica de Tkinter en la versión 8.5, siendo compatible con la versión de Python antes mencionada, misma que se encargará de crear el contenido de la interfaz, como lo son ventanas, botones, etiquetas, entre otros de forma que pueda ser utilizados para que el usuario pueda interactuar con ellos y así manipular el sistema de control.

Una vez establecidas las herramientas con las cuales se diseñará y codificará la interfaz de control, se procede a dividir cada apartado para mejorar el orden y gestión del proceso.

- Principal (menú): es la primera ventana que el usuario visualizará al ejecutar el programa, en esta se podrá acceder al menú con diferentes acciones, como lo es ejecutar un nuevo proceso, abrir y graficar una prueba existente, ajustar los parámetros y abrir un menú de ayuda.
- Nuevo proceso: Esta ventana será desplegada después de seleccionar esta acción donde se podrá ingresar las diferentes características con las que se trabajará, con el fin de tener un registro de todos los parámetros involucrados al momento de efectuar una nueva prueba.
- Pulido: En esta ventana el usuario tendrá la opción de manipular el sistema mecánico de dos formas; manual que hace uso de un joystick o con los botones del teclado, como también dentro de la interfaz para posicionar el electrodo de una forma conveniente y movimiento automático, en la cual dentro del programa se programarán rutas preestablecidas que utilizarán las características introducidas en el apartado anterior. Al mismo tiempo se podrá visualizar las variables de pulido, gap y tiempo de ejecución de cada prueba.

- Graficar: Como su nombre lo dice, se graficará el resultado obtenido de las mediciones de voltaje y corriente de una prueba previamente realizada, a fin de analizar el resultado obtenido dependiendo de las propiedades de la herramienta, pieza de trabajo, electrolito, gap y corriente consumida.
- Ayuda: Este apartado servirá de apoyo para indicar al usuario el funcionamiento de cada sección, así mismo se indicarán algunas propiedades de cada acción.

Posteriormente se caracterizan las funciones, métodos, módulos y sentencias que serán utilizadas, entre otros recursos que puedan ser de apoyo para el desarrollo de la codificación del programa. Por consiguiente, para describir las acciones y características que se podrá realizar en la barra de menú, se enlistan las diferentes funciones que se podrán ejecutar directamente de la ventana principal

- Nuevo: Este botón despliega una ventana para ingresar el nombre y ubicación del archivo con el cual se va a trabajar. La entrada del nombre se define por “ecp\_fecha\_hora” donde la fecha y hora se obtienen del S.O, para establecer así el día y la hora de su ejecución. De igual manera, la ubicación de la carpeta se establece por la ruta “/home/pi /ecp/ajustes”, donde cualquiera de estas dos entradas puede ser modificada por el usuario.
- Graficar: Con este botón se muestra una nueva ventana en la cual se busca el archivo de una prueba de pulido previamente ejecutada. Esta opción permite agilizar la creación de un nuevo proceso con características similares a experimento ya realizado o la posibilidad de graficar estos los resultados obtenidos de dicha prueba.
- Salir: Esta función abre una ventana de confirmación para cerrar el programa principal, si se confirma la salida, la interfaz se cierra, pero si es cancelado solo se cierra la ventana de advertencia.
- Para el segundo submenú llamado ayuda, se despliega únicamente dos opciones definidas como: atajos y acerca de. A continuación, se describen sus características de funcionamiento.



- Atajos: Esta sección permite que el usuario conozca la combinación de teclas que se para las distintas funciones establecidas en el programa y con ello acceder a cierta opción de una forma más rápida.
- Acerca de: Por último, esta opción muestra al usuario la versión del S.O en la cual se esté ejecutando este proyecto, el nombre de usuario que esté utilizando la interfaz, la versión de del proyecto y características de este.

De esta forma se puede proponer la estructura de funcionamiento del programa en la Figura 14, donde se puede observar las diversas acciones que puede ejecutar el usuario, como un diagrama de secuencia de los eventos programados.

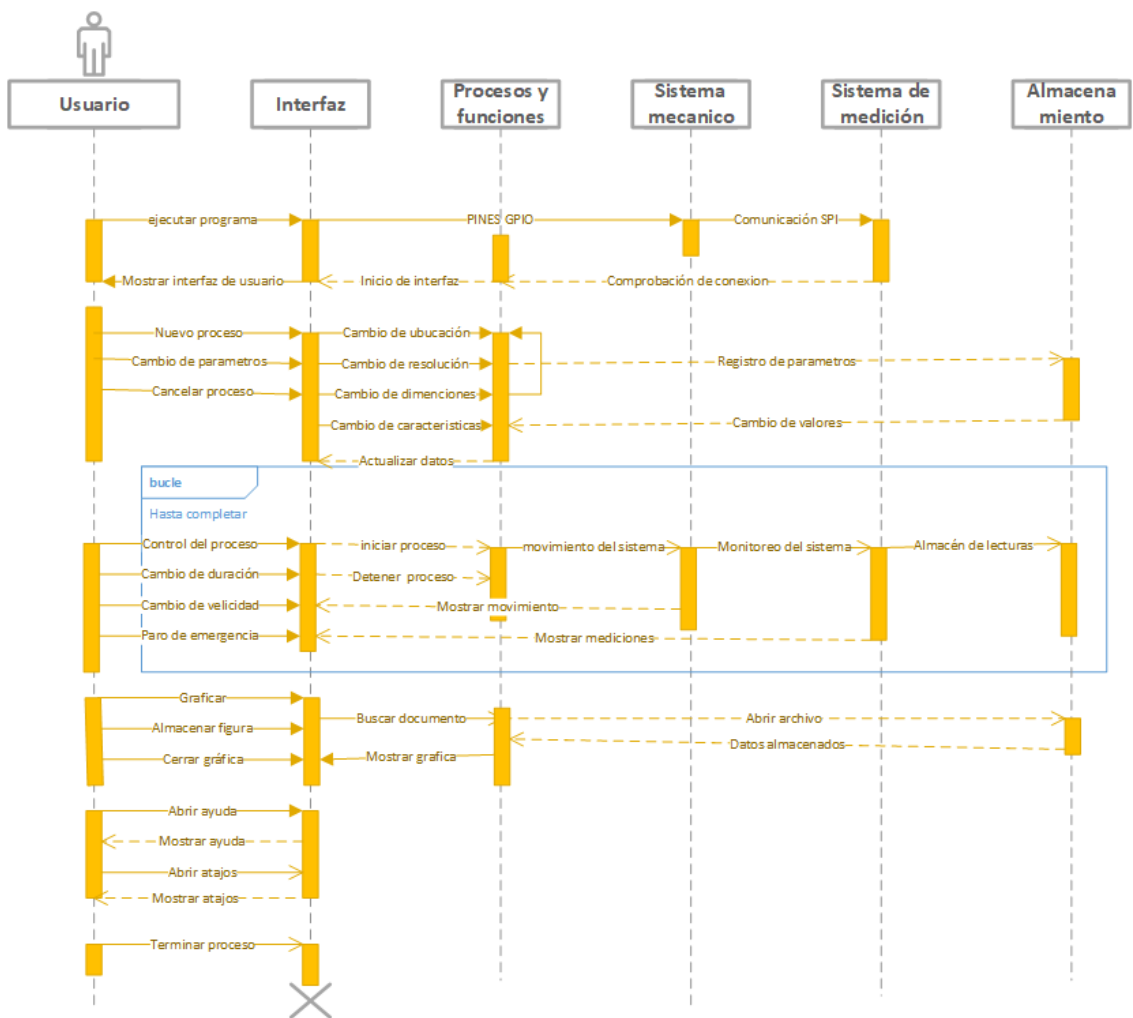


Figura 14: Secuencia de funcionamiento del proceso.

Para profundizar en las características y funcionamiento del software desarrollado, en la Figura 15 se muestra el diagrama de flujo de su funcionamiento, una vez que se está en la ventana principal. Puesto que al iniciar por primera vez el programa es necesario inicializar, todas las librerías, variables, funciones y métodos del programa los cuáles son las encargadas del funcionamiento de la interfaz, comunicación con el apartado de medición y control del movimiento para el sistema mecánico.

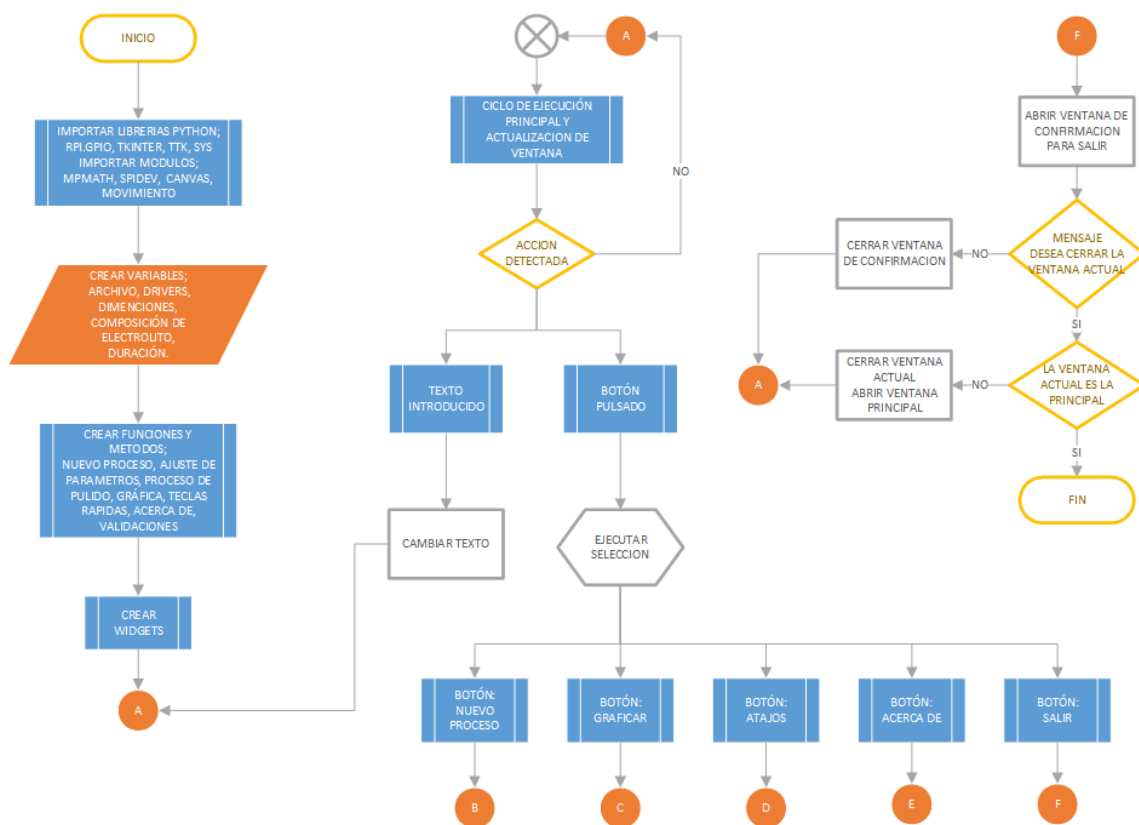


Figura 15: Diagrama de flujo para la ventana principal

El sistema tiene la posibilidad de almacenar los diversos parámetros del prototipo evitando pérdidas de datos al momento de realizar una prueba de pulido electroquímico. En el caso de que se inicialice la ejecución del proyecto de forma correcta, el programa principal podrá ser visualizado y se mantendrá a la espera de que el usuario ejecute alguna acción al presionar algún botón o combinación de teclas rápidas.

Mostrando el posible funcionamiento para la recreación de experimentos de pulido electroquímico, se describe de la siguiente manera: una vez en la ventana principal se podrán realizar las acciones previamente descritas, de dos formas diferentes:

- a) Desde el botón de nuevo proceso: con ello es posible configurar en el sistema las diferentes características de los componentes, útil cuando se pretende realizar una prueba de la que no se tenga registro previo.
- b) Mediante la combinación de las teclas rápidas *CTRL+R*: de esta forma se accederá a la ventana de pulido, donde se monitorean las variables y se define el movimiento del sistema mecánico. En este formato se carga la configuración preestablecida de todo el prototipo.

Posteriormente, para obtener los resultados de algún experimento, solo basta con presionar el botón de graficar, lo que lanza una ventana para buscar el archivo dentro del gestor de documentos y una vez seleccionado se abre una ventana que con la gráfica de las mediciones obtenidas.

Para la realización de nueva prueba, en la Figura 16 se define el funcionamiento que seguirá el programa para que el usuario defina el nombre y ruta donde se almacenan las lecturas de medición que se obtienen en esa prueba y las propiedades (Figura 17) con las que el sistema funciona. En esta ventana se introducirán las características que tendrán los electrodos (dimensiones y aleación), espacio de trabajo (tamaño y delimitación con finales de carrera), propiedades del electrolito (composición química y concentración) como también el movimiento del sistema mecánico (selección de resolución de cada eje y mando de control)

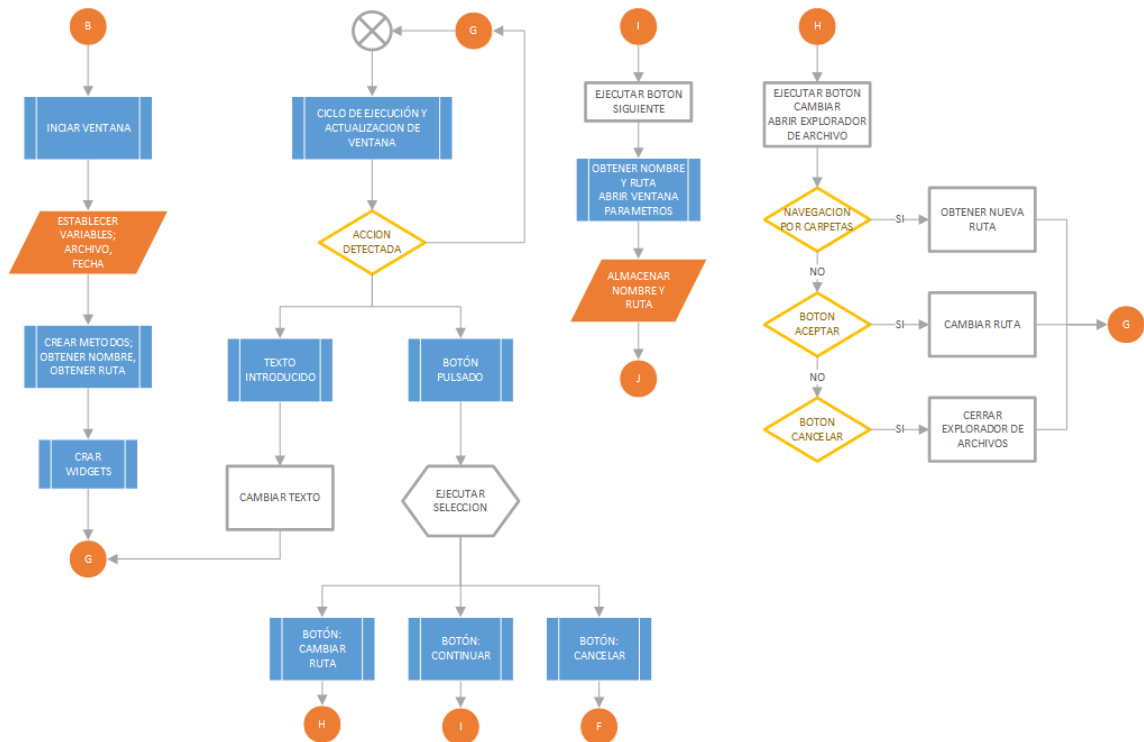


Figura 16: Diagrama de flujo para el registro de un nuevo proceso.

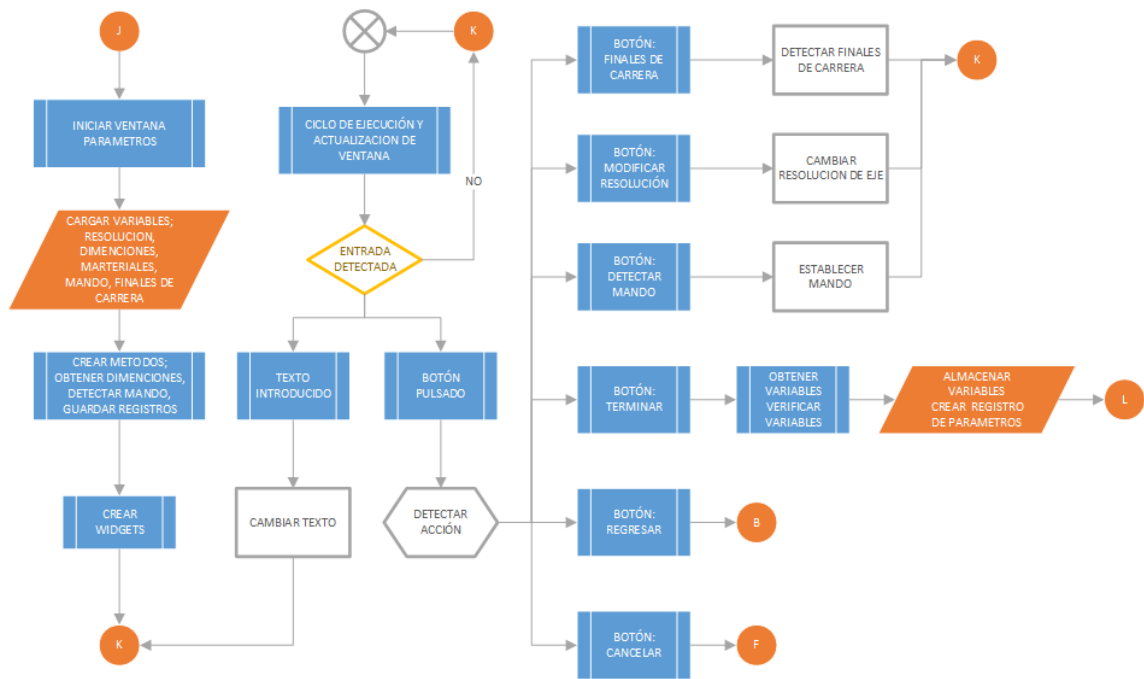


Figura 17: Diagrama de flujo para la ventana con las características del sistema.

Por último, una vez que se hayan introducido todas las propiedades del sistema y presionado el botón de finalizar, la ventana donde se controlar y monitorear el proceso de pulido electroquímico será mostrada, siguiendo el proceso indicado por el diagrama de flujo de la Figura 18. Aquí se podrá manipular el sistema de tres formas; teclado físico, botones dentro de la interfaz y con un joystick, de esta forma el usuario podrá controlar la posición del sistema mecánico de la forma que más se le haga cómoda.

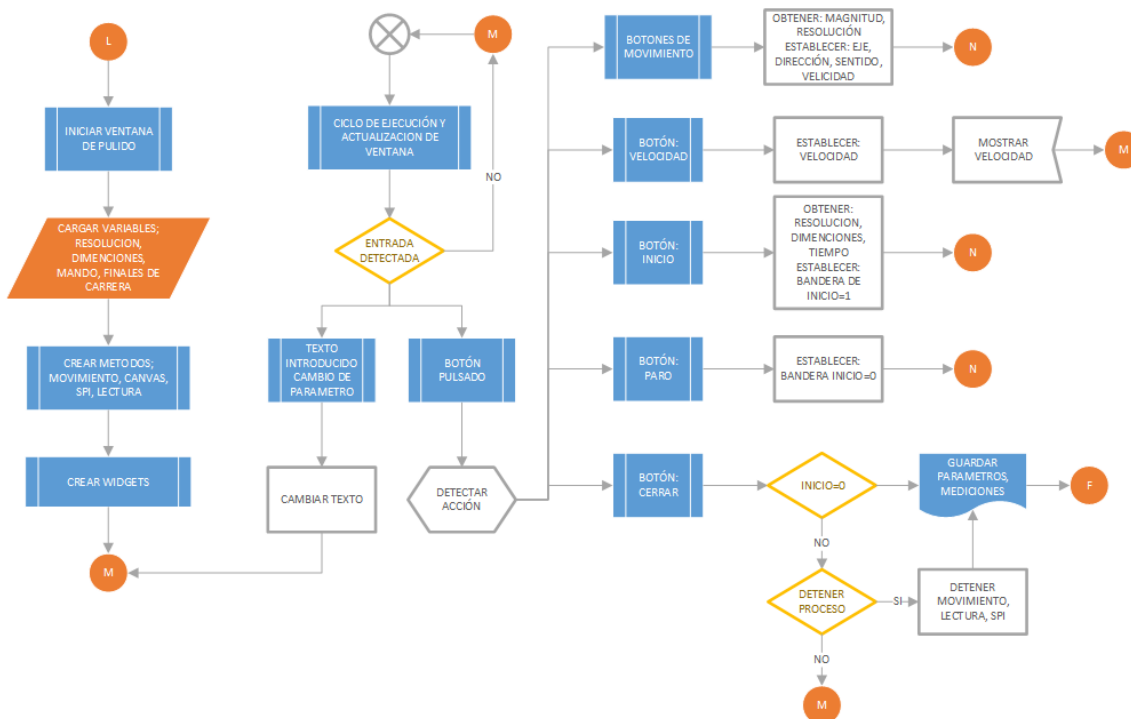


Figura 18: Diagrama de flujo de la ventana de pulido electroquímico.

Al mismo tiempo será posible modificar la velocidad de movimiento para todo el sistema y los tipos de movimientos que se efectuaran de forma automática son los siguientes (Figura 19):

- **Movimiento en ZIGZAG.** Se podrá desplazar la herramienta sobre la pieza de trabajo de forma horizontal (eje  $x$ ) hasta la posición  $x+$ , una vez llegado al límite de la pieza (determinado por las dimensiones del apartado de parámetros).

El movimiento que determinará el avance del eje y será provisto por el tamaño de la herramienta de trabajo, de esta forma el cabezal avanzará por el eje  $y$ , hasta alcanzar el límite del diámetro del electrodo y continuará con el movimiento hacia la posición  $x-$ , repitiendo este proceso hasta llegar al límite contrario por donde se comenzó el proceso, una vez llegado a este punto el movimiento regresara sobre sus pasos una y otra vez hasta cumplir con el tiempo de ejecución determinado en la interfaz.

- **CARACOL.** Como su nombre lo dice, se intentará reproducir la forma del caparazón de este animal, al igual que el movimiento de zigzag se desplazará desde  $x-$  hacia  $x+$ , una vez llegado a este punto, se procederá con el cambio de dirección mediante el eje  $y$ , ahora punto  $y-$  para llegar a  $y+$ , una vez completado esto se repetirá el proceso, pero ahora de forma inversa, de  $x+$  a  $x-$ , y de  $y+$  hacia  $y-$ . Una vez que se llegue al centro de la pieza de trabajo, al igual que en el apartado anterior, se regresara hasta el punto inicial de partida.
- **HORIZONTAL.** Es un movimiento ejecutado únicamente por el eje  $x$ , con el fin de realizar pulidos en forma de canaleta.
- **VERTICAL.** Para este tipo de movimiento, el cabezal o eje  $z$  se moverá hacia abajo y la rapidez con que alcance el límite dependerá de la velocidad provista en la interfaz.
- **TRANSVERSAL.** Ejecutará un movimiento al igual que el de tipo horizontal con la variante que este será aplicado sobre el eje  $y$ .

Al momento que termine el tiempo de ejecución de dicha prueba, la ventana en la que se encuentre se cierra y regresa a la ventana principal. Estando en este punto es posible graficar los resultados obtenidos en la prueba realizada a lo largo de este proceso.

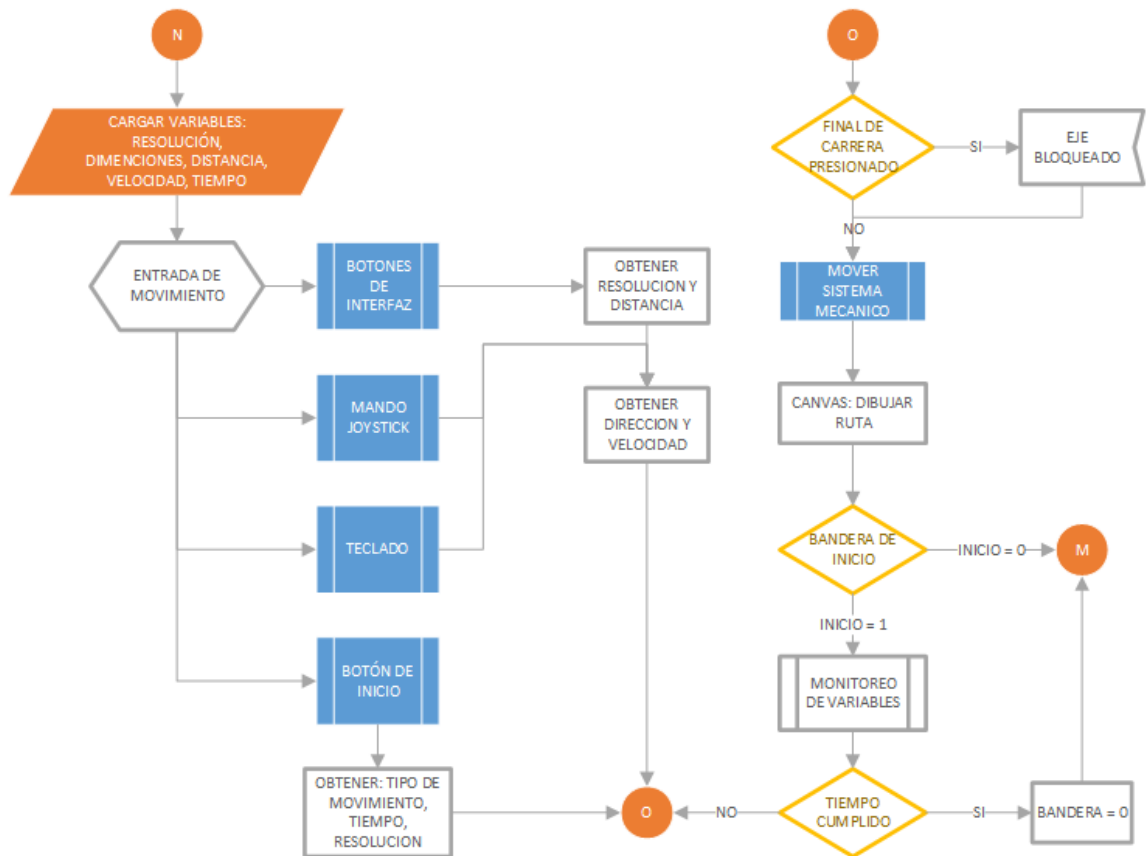
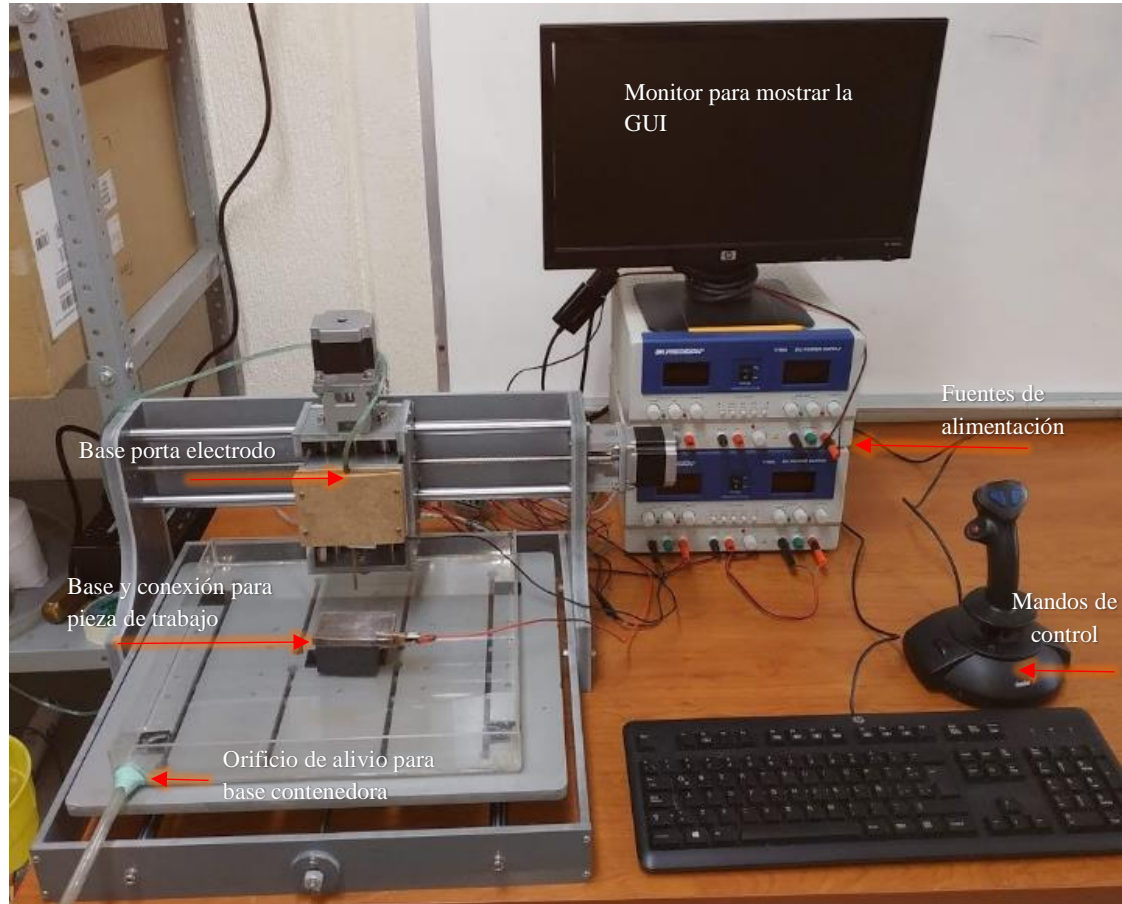


Figura 19: Diagrama de flujo para el movimiento mecánico.

En resumen, los diagramas anteriores permiten crear una estructura básica de las actividades a realizar para las funciones requeridas para que el sistema trabaje de forma que se cumplan los puntos mencionados durante el diseño de esta interfaz.

## 5.7 Ensamble del prototipo

En la Figura 20 se presenta una fotografía real de la implementación del prototipo implementado, destacando algunos los accesorios más importantes que lo conforman.



*Figura 20. Vista general del prototipo para pulido electroquímico*

De tal manera, se muestra en el diagrama de la Figura 21, la estructura y componentes que integran el prototipo del sistema, denotando cada elemento, de forma que pueda ser analizado para su correcta conexión entre cada dispositivo.

El sistema se compone principalmente del control que definirá las rutas del sistema mecánico, estos son los mandos de control. El sentido de cómo se moverá será dictado por la computadora, la cual recibirá la información y enviará al controlador correspondiente, la instrucción para efectuar el desplazamiento. Todo esto podrá ser visualizado en un monitor, en el cual también se verá una pequeña simulación del movimiento que fue



efectuado. Con ello será posible realizar las pruebas pertinentes con el conjunto del sistema mecánico, donde se puede ver que se adaptaron dos bases; porta electrodo y contenedor del electrolito.

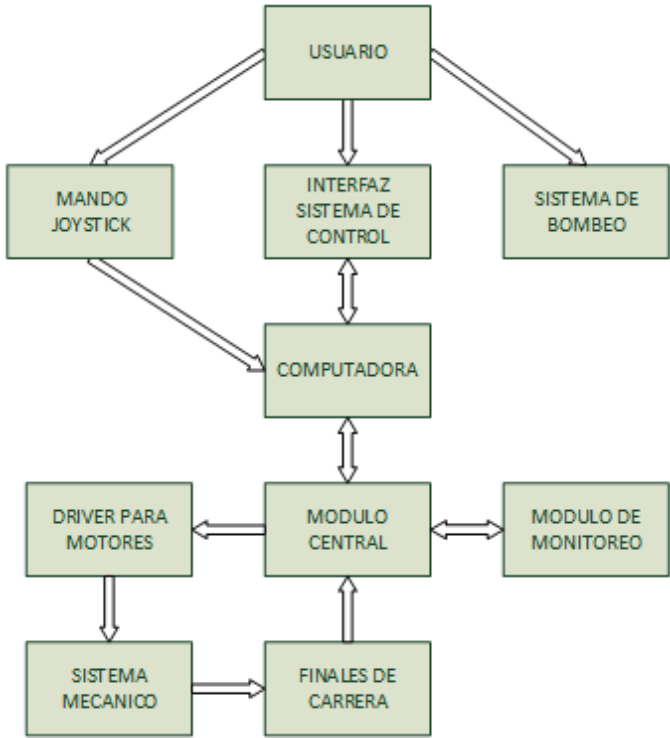


Figura 21: Diagrama a bloques del funcionamiento del sistema

## 6 RESULTADOS

En este apartado se muestra la validación del funcionamiento de la interfaz al aplicarse en pruebas de pulido electroquímico, donde es posible realizar un control del movimiento como también el monitoreo y almacenamiento de las variables de voltaje de polarización y corriente consumida.

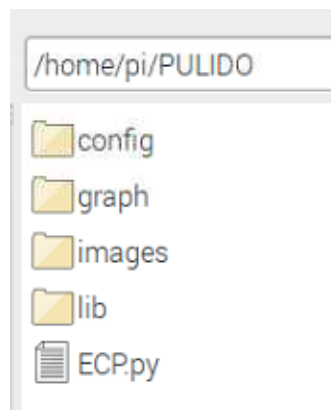
En primera instancia, se presenta el funcionamiento de toda la interfaz y cómo el usuario puede interactuar con la misma, mediante los menús y comandos disponibles. Posteriormente se describe cada posible acción que puede realizarse en cada una de las ventanas, con el fin de validar los objetivos de este proyecto.

Finalmente se muestran los resultados obtenidos de las pruebas de pulido electroquímico, aplicados con diversos materiales metálicos como electrodos, donde complementará la parte de control de movimiento del sistema mecánico y monitoreo de cada prueba dentro del prototipo.

### 6.1 Funcionamiento

El programa principal, almacenado en la memoria microSD de la tarjeta Raspberry Pi; este se encuentra ubicado en la siguiente ruta (/home/pi/Pulido), como se presenta en la Figura 22 y se ejecuta desde la consola de comandos mediante la secuencia:

```
Python ECP.py
```



*Figura 22: Archivos del programa y ubicación*

Una vez que se ponga en ejecución el programa principal, será mostrada la ventana de la Figura 23, en la cual el usuario podrá:

1. Abrir una ventana para la realización de una nueva prueba de pulido
2. Graficar los resultados obtenidos de una prueba previamente realizada
3. Cerrar la ventana principal
4. Mostrar combinaciones de teclas definidos como atajos
5. Mostrar una ventana que muestre las diferentes acciones que se puedan realizar y describir de forma detallada el funcionamiento de estas.



*Figura 23: Ventana principal del programa*

En el caso que se proceda a realizar una nueva prueba de pulido, se deben de realizar los siguientes pasos para su correcta ejecución y configuración: dirigirse al menú Archivo o en su defecto con la combinación de teclas *Ctrl+n* (ver Figura 24); posteriormente se abrirá la ventana de la Figura 26. En su defecto, si se quiere conocer los diferentes atajos o características del sistema se puede seleccionar la opción de ayuda Figura 25

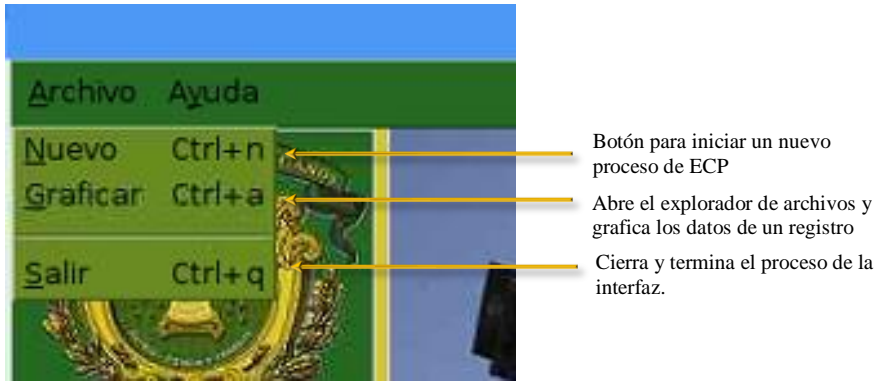
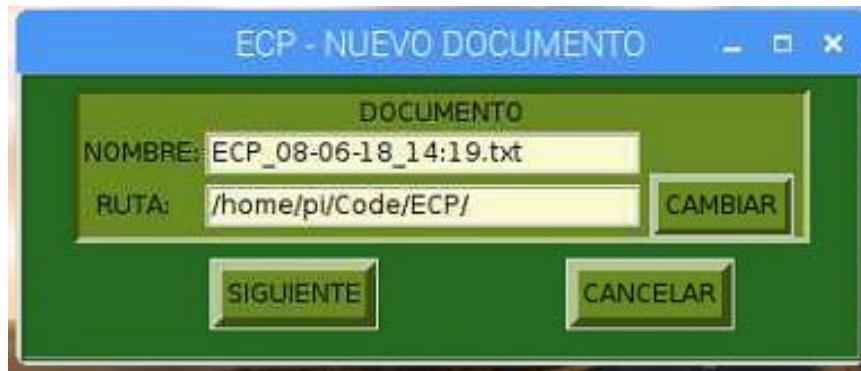


Figura 24: menú de opciones para botón Archivo



Figura 25: opciones para botón de ayuda

Siguiendo con la ejecución de un nuevo registro, se debe dar un nombre para el mismo y la ruta en el cual este es almacenado, teniendo la opción de cambiar cualquiera de estos dos parámetros. Como se puede observar en la Figura 26, por defecto el nombre del proceso se determina por las palabras clave ECP seguido de la fecha y hora de inicio de este, donde, al momento de escribir el documento solo mantiene separado por guiones cortos o bajos. Una vez que se tenga el nombre y la ruta deseada se debe presionar el botón de *siguiente* para continuar con el proceso. En su defecto, se puede regresar a la ventana principal presionando el botón de *cancelar*.



*Figura 26: Ventana de registro de un nuevo proceso*

Posteriormente, se muestra la ventana que contiene todos los parámetros con los que cuenta el prototipo (ver Figura 27). La función de esta ventana es indicar al sistema si se está haciendo uso de finales de carrera, la resolución a la que trabaja cada eje del sistema mecánico, las dimensiones de la herramienta y pieza de trabajo, como también si se hace uso de un mando para manipular el movimiento mecánico del sistema. Con ello, el software se ajusta para estas características de funcionamiento. Adicionalmente, estos valores se almacenan con el nombre y ubicación definido en el apartado anterior; de esta forma se tiene un registro de cada prueba realizada.

Como se puede observar en algunas palabras no se utilizaron acentos de pronunciación, esto fue para no generar algún error de ejecución si se requiere cambiar a otro tipo de entrada, según la compatibilidad que ofrecen las bibliotecas de Python. Además, cualquier parámetro que se ingrese es para definir las características con las que se está trabajando en la prueba que se realice en ese momento

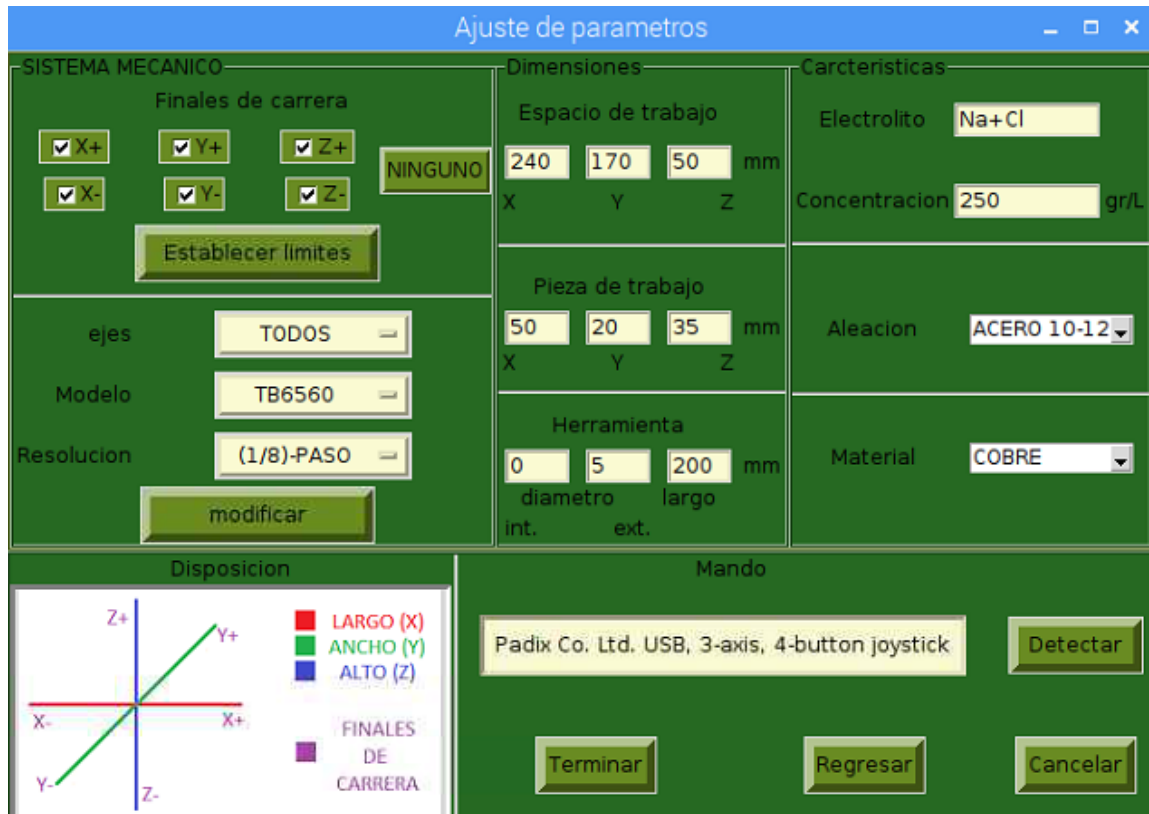


Figura 27: Ventana para el registro de parámetros.

De igual manera, en seguida se describe cada punto con el fin de dar al usuario un mejor entendimiento de este:

**Sistema mecánico:** aquí se define si se están usando los  *finales de carrera*  para asegurar el espacio de trabajo o en su defecto si no se hace uso de ellos. También es posible configurar el tipo de resolución con la que funciona cada eje teniendo cuatro opciones para ello: paso completo, medio paso, octavo y dieciseisavo de paso.

**Dimensiones:** Como su nombre lo indica, aquí se introducen los valores de las medidas del espacio de trabajo, pieza y herramienta, cada valor se introduce con la unidad de medida de milímetros *mm*.

**Disposición:** Es una guía que indica tanto la posición que tiene cada final de carrera, como también el cómo introducir los valores de las dimensiones en su respectivo apartado.

**Características:** Este apartado se utiliza para el registro del tipo de electrolito y la concentración utilizada específicamente en la prueba que se esté realizando en ese momento, al igual que el tipo de materiales utilizados, tanto en la herramienta, como en la pieza de trabajo.

**Mando:** En esta sección se adiciona la opción de control mediante el mando tipo joystick y con ello controlar el desplazamiento del electrodo, sin embargo, si no se encuentra conectado, solo se tiene la posibilidad de control mediante botones o dentro de la interfaz.

Cuando se termine con el registro de todos los datos se continúa el programa al presiona el botón *terminar* para iniciar con el proceso; sin embargo, aún se puede regresar al apartado anterior para modificar algún dato o en su defecto cancelar todo y regresar a la ventana de inicio.

Finalmente, en la ventana de la Figura 28 es en donde se puede realizar el control y monitoreo del proceso de pulido. Dentro de esta pantalla se tienen varias secciones que definen el funcionamiento del sistema, los cuales son: sistema mecánico, ajustes, monitoreo y controles.

Dentro de sistema mecánico es posible visualizar una simulación de la trayectoria que tendrá el electrodo. En la sección de ajustes se puede cambiar la velocidad o el tipo de movimiento que realiza el sistema, así como también la duración de ejecución de este, puesto que cada prueba será definida por el tiempo que el usuario establezca.

Para el apartado de monitoreo, se visualiza el voltaje de polarización y corriente consumida en la pieza de trabajo y esta será visualizada únicamente cuando se inicie el proceso y esto pasa después de presionar el botón *PLAY* situado en la sección de controles (abajo-derecha de la ventana). También es posible visualizar la hora de inicio y termino del proceso y la duración del mismo. O en su defecto, con el botón *STOP* el usuario podrá detener el proceso cuando este detecte un fallo durante la prueba que se esté llevando a cabo en ese momento.

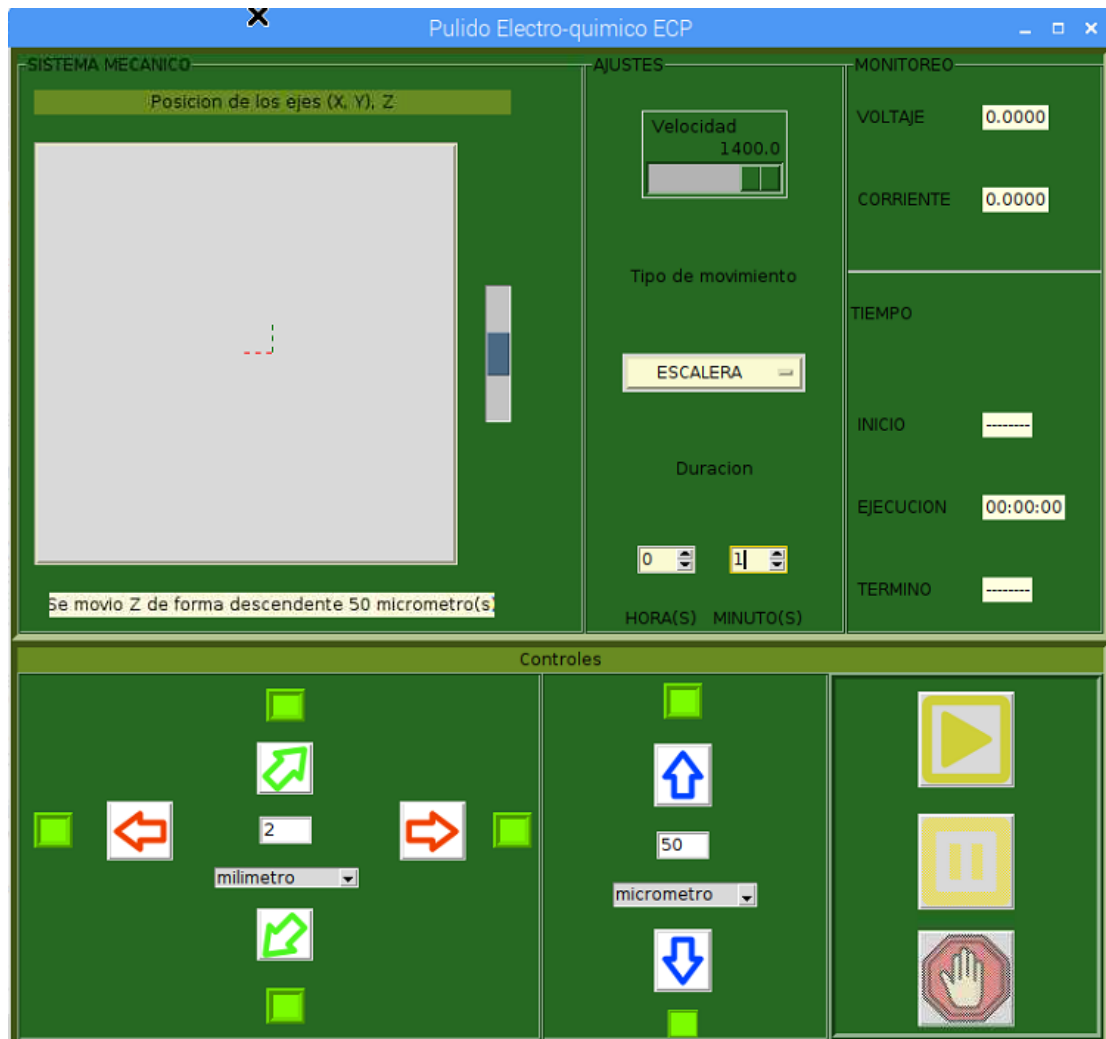


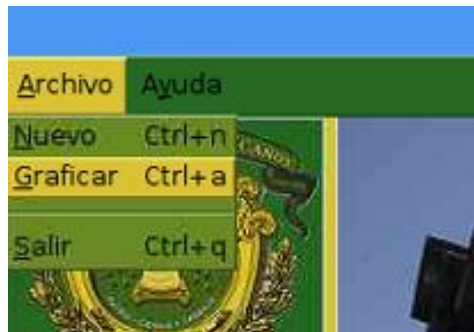
Figura 28: Ventana final para control y monitoreo del proceso.

En la sección de controles el usuario puede manipular la posición de inicio del electrodo, cambiando la resolución de movimiento en pasos, micrómetros y milímetros, para determinar de una forma más conveniente dicha ubicación, teniendo en cuenta que también se podrá controlar por medio del teclado con las teclas ( $\leftarrow$ ,  $\rightarrow$ ,  $\uparrow$ ,  $\downarrow$ , w, s); dos de estas para los ejes X, Y, Z respectivamente. Sin embargo, la resolución de paso, tanto de estos como del mando se establece en  $100\ \mu\text{m}$  por paso (cada tecleo), de tal forma que pueda mantenerse presionado sin verse afectado por las iteraciones de este.

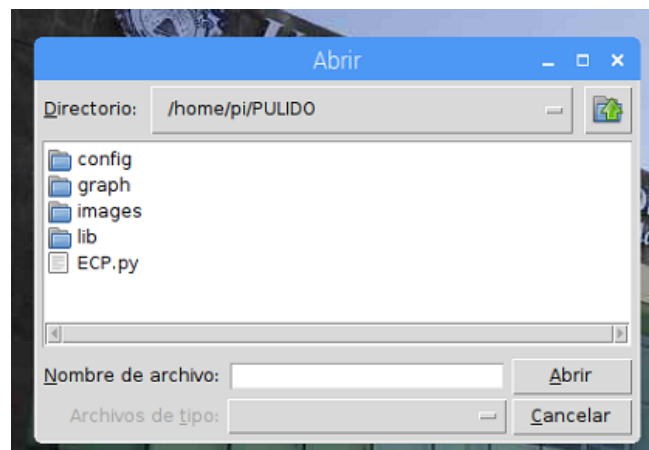


Así mismo se podrá variar la velocidad de movimiento, la cual está definida en un rango de 1 hasta 1400 Hertz, donde la distancia a alcanzar se verá afectada dependiendo de la resolución establecida en el apartado de parámetros.

Una vez que se termine de efectuar el proceso de pulido del apartado anterior, se pueden graficar los resultados obtenidos de las mediciones realizadas. Para ello basta con ir al menú Archivo > Graficar, como se exhibe en la Figura 29; aquí se abre el gestor de archivos que permite buscar el documento de dicha prueba, tal como se observa en la Figura 30.



*Figura 29: opción de grafica de resultados*



*Figura 30: Ventana de búsqueda para pruebas realizadas*

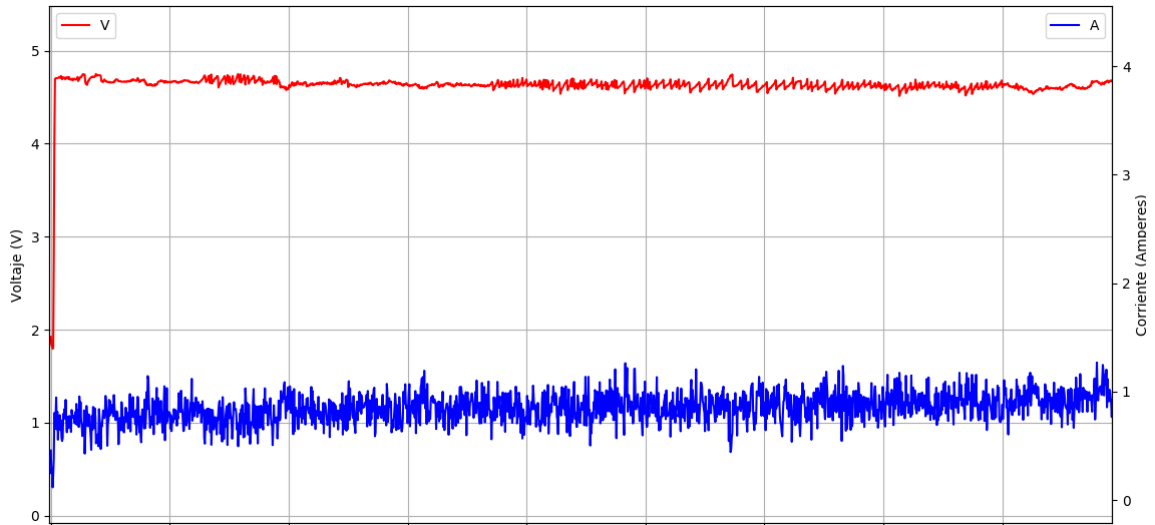
## 6.2 Validación

En esta sección se muestran los resultados de funcionamiento que se obtuvieron mediante la aplicación del prototipo en pruebas de pulido electroquímico. Las pruebas que se realizaron fueron efectuadas con dos tipos de herramientas: cobre y acero inoxidable; el primero tiene un diámetro externo de 2 milímetros y diámetro interno de 1 milímetro, mientras que el segundo corresponde a una jeringa hipodérmica de acero inoxidable tipo 21G×32mm (0.8×40mm), donde su punta fue recortada para tener un borde liso al igual que el primer electrodo (se eliminó la punta de la aguja).

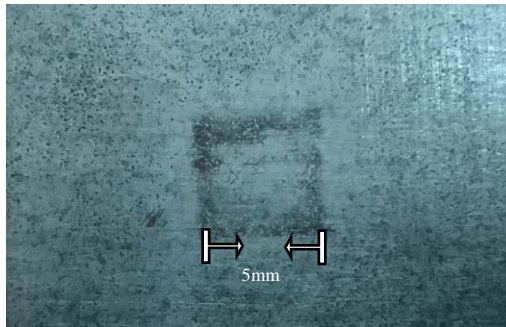
El material de las piezas donde se realizaron las pruebas es de cuatro tipos: acero tratado, acero 106-12 y acero 10-12, para el caso del electrodo de cobre y acero HSLA, para el segundo electrodo. La velocidad con la que se desplaza el electrodo únicamente en la cuarta prueba se estableció en 20% la velocidad de movimiento y un 50% para el resto.

Adicionalmente, en cada prueba se utilizó el electrolito a base de *NaCl*, con una concentración de 250 gr/L, ya que se ha demostrado que este tipo de electrolito a esa concentración es el que mejor trabaja con diversos tipos de metales (Granda Gutiérrez, et al., 2015) y (Estrada Hernandez & León Bravo, 2014).

Para la primera prueba, se efectuó un pulido con movimiento en escalera en un área de 5 mm durante 15 minutos. En la Figura 31 se muestra la lectura de la corriente y voltaje consumido dando como resultado un cuadrado casi sin bordes redondeados en el acero 10-12, tal como se presenta en la Figura 32, con una profundidad de pulido de aproximadamente 200  $\mu\text{m}$ .

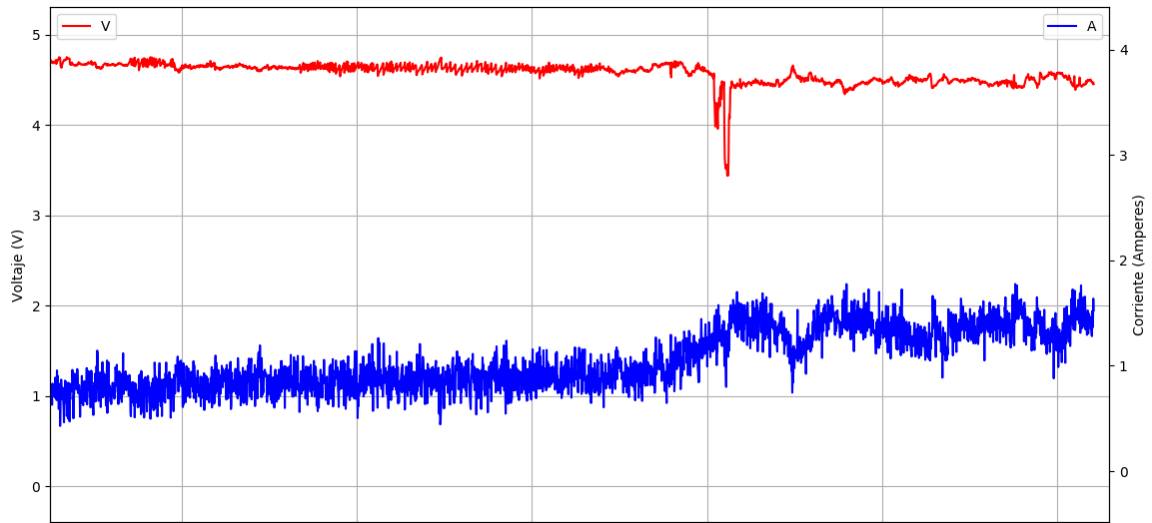


*Figura 31: Prueba con acero 10-12.*



*Figura 32: Pulido obtenido en acero 10-12.*

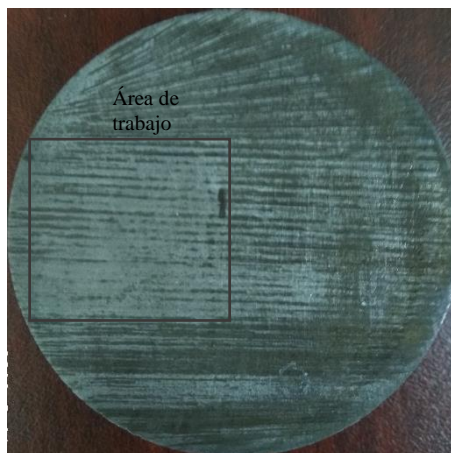
En la siguiente prueba, se muestran los resultados de pulido en un área de 2 centímetros cuadrados aplicados durante 10 minutos en la pieza de acero tratado con la técnica de movimiento en escalera. Como se puede observar en la Figura 33, se obtuvo una mayor fluctuación de la lectura realizada; esto sucede porque la pieza tiene pequeñas irregularidades en su superficie, como se presenta en la Figura 34, lo que pudo haber causado dichas variaciones. Posteriormente, en la Figura 35, se muestra una reducción considerable de las irregularidades después de realizar el pulido en esta pieza. Dado que la pieza en un principio contaba con irregularidades, la diferencia puede notarse en gran medida mediante el tacto.



*Figura 33: Mediciones en acero tratado*

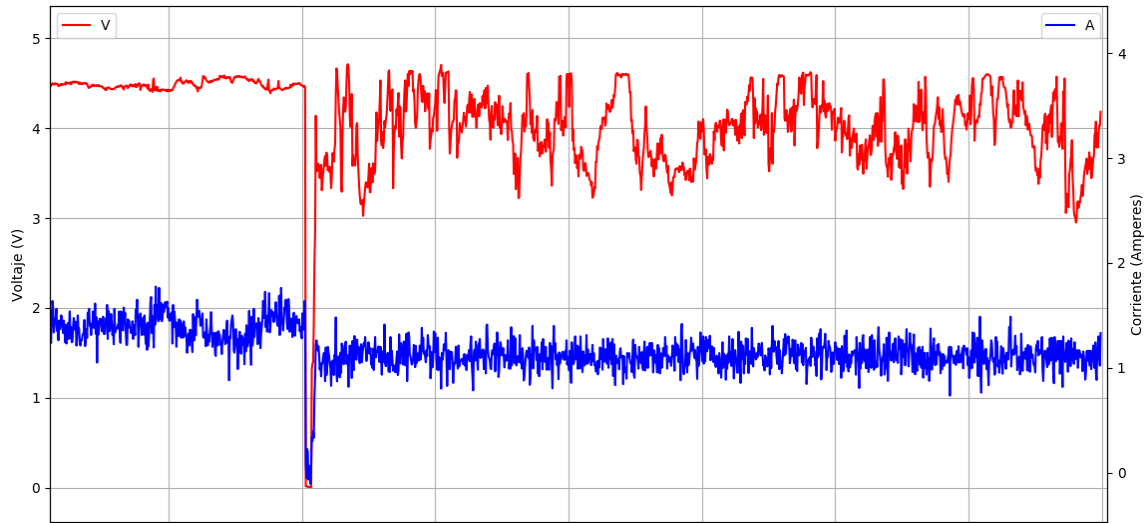


*Figura 34: Área y pieza de acero tratado sin pulido*

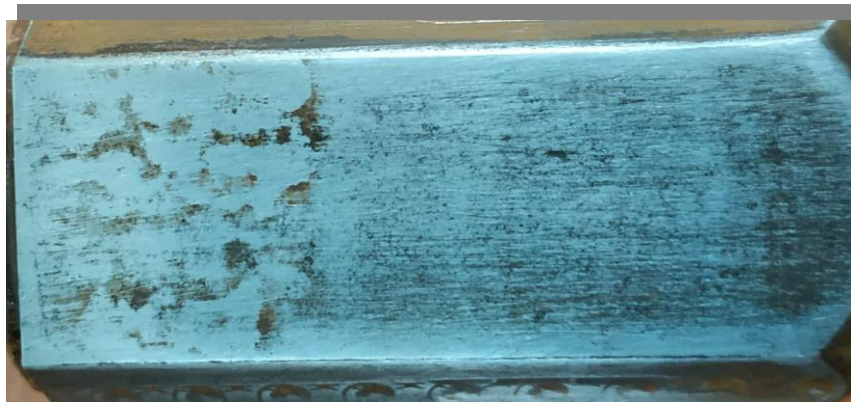


*Figura 35: Pulido en acero tratado.*

A continuación, se muestra la aplicación de una prueba en una pieza de tipo acero 106-12 durante un tiempo de 30 minutos, de la cual se puede observar que tuvo malformaciones al momento de realizar el pulido, y esto puede ser corroborado por la gran variación que de voltaje y corriente Figura 36. Adicionalmente se pueden observar puntos negros en la pieza lo que indica un posible contacto físico entre la herramienta y pieza de trabajo, como se presenta en la Figura 37.



*Figura 36: Grafica de mediciones en acero 106-12*

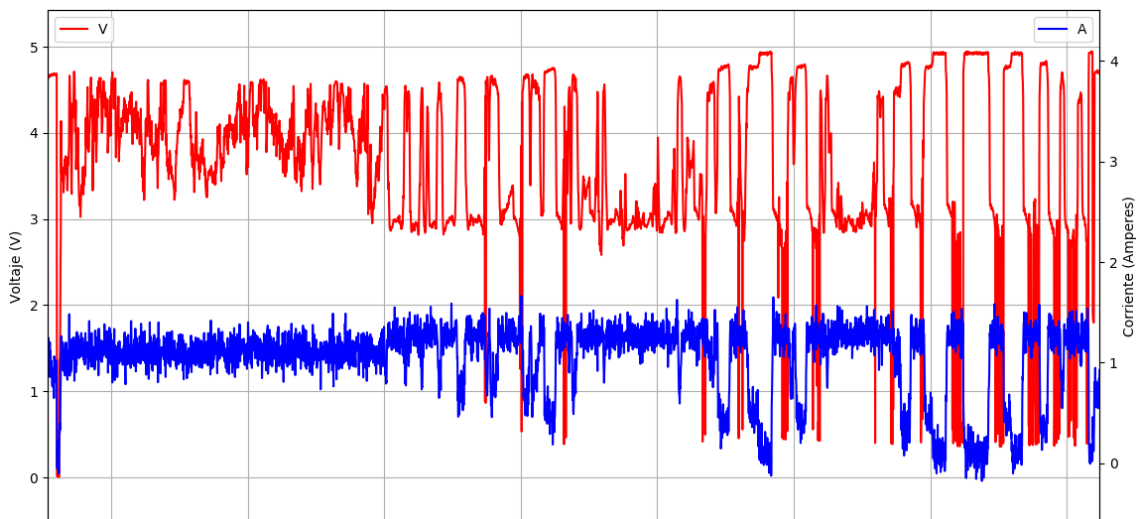


*Figura 37: Pulido en pieza de acero 106-12 con malformaciones*

Continuando con las pruebas, se realizaron algunas modificaciones para obtener diversos resultados del prototipo; por ello, se cambió la trayectoria que sigue el electrodo, así como

la velocidad de avance, por lo que en esta ocasión se implementó el movimiento de tipo caracol, y una velocidad de avance del 20%, obteniendo la siguiente grafica Figura 38. En la pieza Figura 39, se puede observar una gran cantidad de remoción en la parte izquierda de la pieza, por el contrario, en la parte derecha se obtuvo un resultado que al tacto se puede describir como granulado.

La razón más probable por lo que haya ocurrido esto es debido a que se redujo la velocidad de desplazamiento aunado una posible capa de recubrimiento que solo se vio afectada en algunas zonas de su superficie. En este modo, el proceso pasa de ser un pulido electroquímico a un proceso de maquinado, puesto que la cantidad de material removido es muy significativo, y se aprecia visualmente.



*Figura 38: Grafica de pulido tipo escalera en acero 10-12*



*Figura 39: Pulido con gran desprendimiento en acero 10-12*

Posteriormente se realizó el cambio del electrodo, siendo aplicado a la pieza de acero HS-LA, que se presenta en la Figura 40, en la que, al igual que las pruebas previamente realizadas, se ejecutaron bajo las siguientes características:

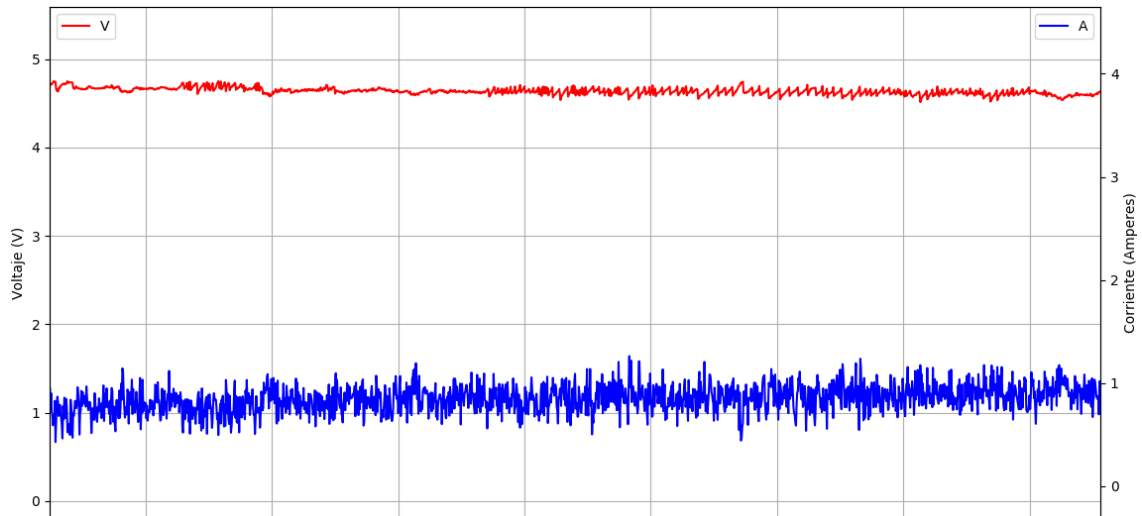
- Electrolito a base de cloruro de sodio con 25% de concentración
- Velocidad de movimiento del 50%
- Resolución de paso en 1/16 (3200 pasos/rev)

En esta etapa únicamente se realizaron pruebas con movimientos lineales en un solo eje, para verificar la profundidad y sobre corte que existe tras hacer un barrido en un mismo punto de forma iterativa.



*Figura 40:Acero HS-LA con superficie plana*

Para la primera prueba con estos ajustes se realizó un movimiento lineal en el eje Y durante 15 minutos. Como se puede apreciar en la gráfica de la Figura 41, se tiene una lectura de valores correcta; adicionalmente, se puede observar que el resultado es un maquinado de un canal recto, como se muestra en la Figura 42. La distancia recorrida es de 10 mm y apenas se puede observar un sobre corte, tanto en los límites como de forma transversal al proceso.



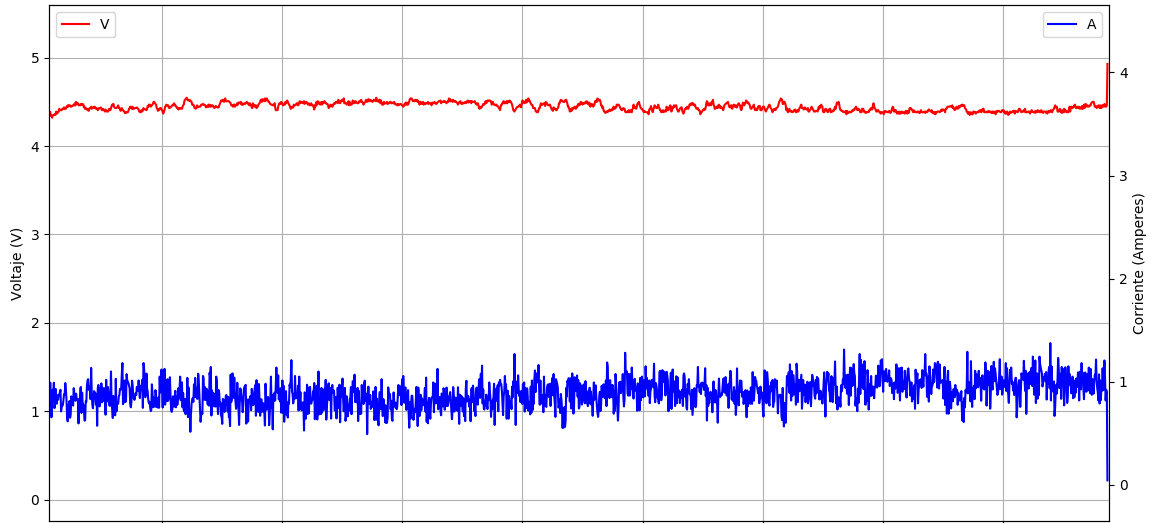
*Figura 41: Gráfica de acero HS-LA al maquinarse en línea recta*



*Figura 42: Línea formada por el proceso durante 15 min*

En el caso de la segunda prueba con el acero HSLA, se efectuó el movimiento en el eje X; esto se plantea de la siguiente forma esperando que los resultados obtenidos fueran lo más parecidos a la primera prueba. Comparando la segunda gráfica, mostrada en la Figura 43, donde los valores de esta son muy similares a los que se obtuvieron previamente en la Figura 41. En caso similar, la línea generada cuenta con características similares a las del primer resultado, indicando que el proceso es repetible.





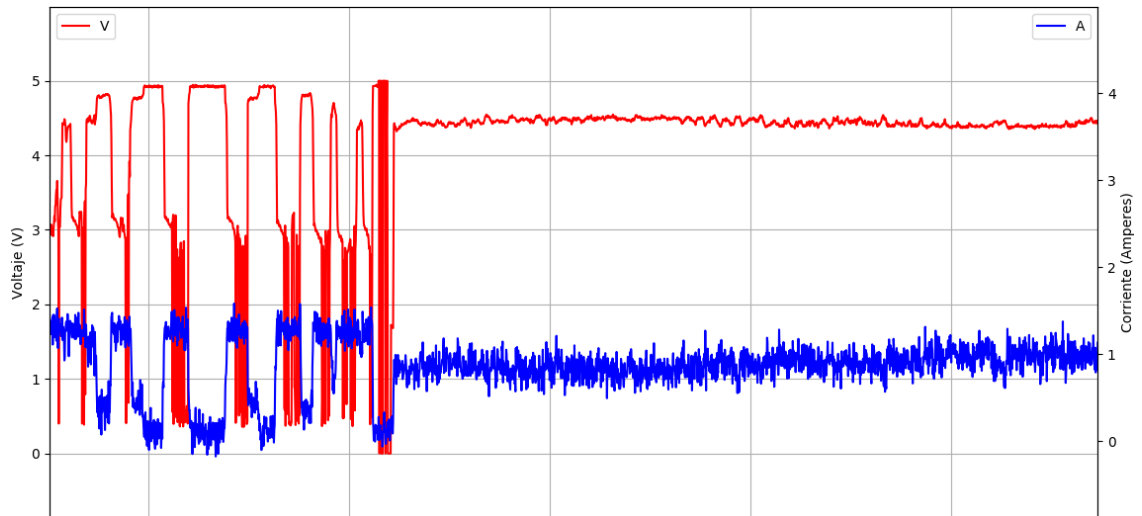
*Figura 43: Grafica de validación para el eje X*



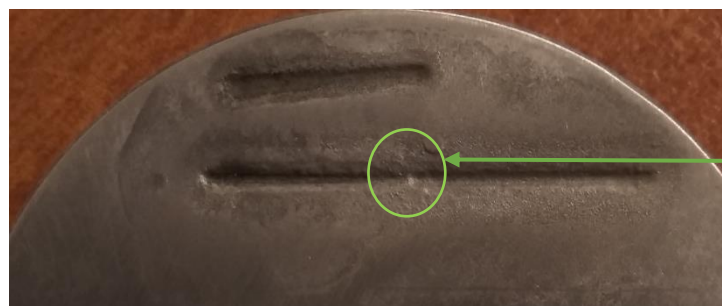
*Figura 44: Línea formada por el proceso en el eje X*

Finalmente, se realizó una prueba para determinar si es posible dar continuidad a un proceso ya efectuado, es decir “extender” el rango de este. En la Figura 46, apenas es apreciable la continuación de la tercera prueba en el mismo material. Mientras tanto, en la Figura 45 se presentan los resultados de la adquisición de las señales de voltaje y corriente, que son consistentes con las dos pruebas anteriores.

Las pruebas efectuadas en el acero HS-LA, indican la versatilidad del prototipo para trabajar en modo de pulido electroquímico, pero también de maquinado electroquímico, por lo que potencialmente pueden efectuarse ambos procesos en el mismo equipo.



*Figura 45: Grafica para línea de continuidad*



Punto de unión  
de las pruebas  
uno y tres

*Figura 46: Línea de continuación de la primera prueba*

## 7 CONCLUSIONES

Las pruebas efectuadas en este proyecto confirman que la construcción de un sistema embebido, con base en una computadora de placa reducida, permite realizar de forma adecuada ensayos de pulido electroquímico y maquinado electroquímico, dando al usuario la posibilidad de manipular el sistema mecánico de una forma intuitiva. El monitoreo de variables se efectúa de una buena forma, con la posibilidad de graficar estos resultados para que posteriormente puedan tener un tratamiento específico.

El sistema se construyó con una placa de control que permitiera mejorar la precisión de desplazamiento, dado que el sistema mecánico en un principio se planteaba para que funcionara con el módulo controlador con base en los integrados L297 Y L298, pero a comparación de este que solo tiene dos cambios de resolución (paso completo y medio paso), el driver TB6560 puede funcionar de una mejor manera con una configuración especial, mejorando la calidad de las trayectorias al ser pasos “más pequeños”, por lo que el sistema produce menos vibraciones, obteniendo un mejor resultado en las pruebas.

La codificación del apartado de movimiento permitió que la resolución mínima por paso fuese de hasta  $2.5 \mu\text{m}$ , con la configuración de 1/16 de paso; es decir, que se tendrían 3200 pasos por revolución, lo que a su vez implica que la distancia recorrida en una revolución es de 8 mm. Sin embargo, las capacidades tanto del motor como del módulo controlador, solo se podía asegurar su funcionamiento a partir de los  $20 \mu\text{m}$ .

El desarrollo de la interfaz permitió que el usuario cuente con la posibilidad de gestionar, controlar y monitorear el sistema de forma que el usuario pueda interactuar con la misma de forma intuitiva y con pocos ajustes. Adicionalmente se cuenta con la posibilidad de almacenar los parámetros de cada prueba, reduciendo la pérdida de información de cada prueba ejecutada, ya que posteriormente esta información puede ser comparada con algún otro resultado, o simplemente comprobar los mismos.

## 8 REFERENCIAS

Andrade Fuentes, O. A., 2012. *Manual de prácticas para el torno de control numérico CNC del taller de mecánica*. Veracruz(Xalapa): Tesis de Licenciatura.

Arévalo Mocayo, D. R. & Hermosa Ocampo, D. C., 2014. *Diseño y construcción de una prototipadora CNC que realiza el ruteo de pistas y el taladro de circuitos impresos utilizando procesamiento de imágenes en labview*. Quito(Sangolquí): Tesis de Licenciatura.

Catarino Aguilar, O., 2017. *Sistema asistido por computadora para la remoción controlada de material en un proceso de maquinado electroquímico*. Estado de México(Atlacomulco): Tesis de Maestría.

Chaides Zuñiga, O., 2012. *Diseño, Implementación y validación de sistema para micromanufactura por uEDM.* México(Monterrey): Tesis de Doctorado.

Chaides, O. y otros, 2009. Diseño y prueba de un sistema de control de espaciamiento y potencia para micro-EDM. *SciELO*, 3(2), pp. 37-45.

Díaz del Castillo Rodríguez, F., 2008. *Programación automática de máquinas CNC*, Cuautitlán Izcalli: Facultad de Estudios Superiores Cuautitlán.

Dinesh Chopra, B., 2001. *Copper chemical-mechanical polishing process using a fixed abrasive polishing pad and copper layer chemical-mechanical polishing solution specifically adapted for chemical-mechanical polishing with a fixed abrasive pad*. U.S, Patent No. US 6,276,996 B1.

El Hofy, H. A.-G., 2005. *Advance machining processes, Non traditional and hybrid machining processes*. Alexandría: Mc Graw-Hill.

Encarnacion Vera, J. V., 2013. *Dotación de infraestructura básica en el laboratorio para realizar metalografía de campo mediante réplicas*. Guayaqui: Tesis de Licenciatura.

Estrada Hernandez, E. F. & León Bravo, D. A., 2014. *Diseño y construcción de una máquina para mecanizado electroquímico de metales con una corriente de 75 Amperios*,

para el laboratorio de procesos de manufactura del DECEM.. Sangolquí: Tesis de Licenciatura.

Granda Gutiérrez, E. E. y otros, 2015. Remoción de material en diversos electrolitos durante el maquinado electroquímico de aceros de alta resistencia.. *Memorias del congreso internacional de Ingeniería electrónica*, 7-9 Octubre, Volumen XXXVII, pp. 25-30.

Grayson, J. E., 2000. *Python and Tkinter Programming*. Segunda ed. New York: Manning Publications Co.

Grover, M. P., 2007. *Fundamentos de manufactura moderna*. Tercera ed. D. F.(Delegación Álvaro Obregón): Mc-Graw Hill Interamericana.

Hermoso Fernández, J., 2012. *Unidad de control para fresadora CNC con cambio automático de herramienta*. Cataluña: Tesis de Licenciatura.

Kosak, J., Budzynski, A. F. & Domanowski, P., 1997. Computer simulation electrochemical shaping (ECM-CNC) using a universal tool electrode. *Journal of material processing technology*, 12 Junio, Issue 76, pp. 161-164.

Lamikiz Mentxaka, A., 2010. *Fabricación asistida por ordenador*, Bilbao: Departamento de Ingeniería Mecánica.

Leens, F., 2009. An Introduction to I2C and SPI Protocols. *IEEE Instrumentation & Measurement Magazine*, 12(1), pp. 8-13.

Microchip Technology Inc., 2002. *MCP3204/3208*, Arizona: Microchip,

Miguel, R. G., 2014. *Diseño y fabricacion de una fresadora CNC de 3 ejes para el mecanizado de PCB con plataformas de desarrollo abiertas*. Catagena(Murcia): Tesis de Licenciatura.

Ogata, K., 2010. *Ingeniería de control moderna*. Quinta ed. Madrid: Pearson.

Pon Selvan, C., Reddy Madara, S. & Sampath, S. S., 2017. Review of the Current State of Research and Development in Electro. *American International Journal of Research in Science, Technology, Engineering & Mathematics*, 17(304), pp. 14 - 26.

Pressman, R. S., 2010. *Ingeniería del software. Un Enfoque práctico*. Tercera ed. D.F.(Delegación Álvaro Obregón): Mc Graw-Hill Interamericana.

Raspberry Pi Org, 2018. *Raspberry pi, Teach, Learn, and Make with raspberry pi*. [En línea]

Available at: <https://www.raspberrypi.org/>  
[Último acceso: 18 Septiembre 2018].

Schey, J. A., 2002. *Procesos de manufactura*. Tercera ed. D. F.(Delegación Cuauhtémoc): Mac Graw-Hill Interamericana.

Shenzhen Global Technology Co., Ltd., 2015. *Module: TB6560*, Shenzhen: Shenzhen Global Technology Co., Ltd..

Soto C., R., 2008. *EDM: Maquinado por descarga Eléctrica*, Sonora: Centro de Integración para la Industria Automotriz y Aeronáutica de Sonora, A.C..

STMicroelectronic, 2001. *L297*, Plan-les-Ouates: STMicroelectronics.

Taylor, E. J. & Inman, M., 2014. Electrochemical surface finishing. *The Electrochemical Society Interface*, pp. 57 - 61.

Vargas Soto, J. E., 2010. *Historia del control automatico*, Veracruz: Asociación Mexicana de Mecatrónica.

Yesid Peña, D., Vásquez, C. & Estupiñan, H., 2010. Pulido electroquímico de dispositivos metálicos en la industria de los implantes biomédicos. *Puente*, 4(2), pp. 1 - 5.

Zamora Vázquez, A., 2009. *Diseño e implementación de una fuente de voltaje con aplicación a micromanufactura por electroerosión*. Mexico(Ciudad de México): Tesis de Licenciatura.

Zen ToolWorks, Inc., 2018. *ZEN Toolworks*. [En línea]  
Available at: <https://www.zencnc.com>  
[Último acceso: 10 Septiembre 2018].

Zhai, X., 2017. *Soloelectronicos*. [En línea]  
Available at: <http://soloelectronicos.com/2017/04/12/cnc-basado-en-raspberry-pi/>

Zhao, H., Van Humbeeck, J., Sohier, J. & de Scheerder, I., 2003. Electrochemical polishing of 316L Stainless Steel Slotted Tube Coronary Stents: an investigation of material removal and surface roughness. *Progress in Biomedical Research*, 8(2), pp. 70 - 81.

## 9 ANEXOS:

### 9.1 script principal

```
0001 #!/usr/bin/env python
0002 #-*- coding:utf-8 -*-#
0003
0004 import datetime, getpass, os, pygame, Queue, re, spidev, sys, time, threading
0005 import RPi.GPIO as GPIO
0006 import mpmath as mp
0007
0008 from lib import grafica
0009 from lib import canv as cv
0010 from lib import mov as mv
0011 from lib import datos as db
0012 from ttk import *
0013
0014 try:
0015     from Tkinter import *
0016
0017 except ImportError:
0018     raise ImportError, "Verificar paquete de Tkinter"
0019
0020 from tkFileDialog import *
0021 from tkMessageBox import *
0022 #VARIABLES
0023 master = None
0024 carp = os.getcwd()
0025 carpeta = carp + os.sep + "ECP" + os.sep
0026 archivo = 'pruebapulido'
0027 fechaReg = 'fecha de ejecucion'
0028 mediciones= os.sep + 'tmp' +os.sep + 'pruebapulidoBD.csv'
0029 modeloX = 'TB6560'
0030 modeloY = 'TB6560'
0031 modeloZ = 'TB6560'
0032 resolX = '(1/8)-PASO'
0033 resolY = '(1/8)-PASO'
0034 resolZ = '(1/8)-PASO'
0035 mando = '----- Buscar -----'
0036 ls1 = 0
0037 ls2 = 0
0038 ls3 = 0
0039 ls4 = 0
0040 ls5 = 0
0041 ls6 = 0
0042 espTrabX = 340
0043 espTrabY = 340
0044 espTrabZ = 30
0045 dPzTrabX = 10
0046 dPzTrabY = 10
0047 dPzTrabZ = 5
0048 dIntHerr = 0
0049 dExtHerr = 1
0050 largHerr = 100
0051 electrolito = 'Na+Cl'
0052 disolucion = 250
0053 tipoMetal=['ACERO_10-12', 'ACERO_10-15', 'ACERO_10-18', 'ACERO_106-12', 'ACERO_HS-LA',
'ACERO_TRATADO']
0054 materialElect=['COBRE', 'BRONCE', 'LATON', 'ACERO_QUIRURGICO']
0055 runHilo = True
0056 dat = True
0057 tipoMov = ''
0058 tipoPul = 'ESCALERA'
```



```

0059     GAP=0
0060     avancGap = 0
0061     esc = 'pasos'
0062     velocidad = 0
0063     horas = 0
0064     minutos = 15
0065     pause = 1
0066     hora_ini = None
0067     hora_fin = None
0068     cronoSS = 0
0069     cronoMM = 0
0070     cronoHH = 0
0071     voltaje = 0.0
0072     corriente = 0.0
0073     sentido = ''
0074     eje = ''
0075     magnitud = None
0076     escala = None
0077     #FUNCIONES
0078     def queueLoop():
0079
0080         try:
0081
0082             while True:
0083
0084                 function, args, kwargs = cola.get_nowait()
0085                 function(*args, **kwargs)
0086
0087             except:
0088                 pass
0089             root.after(1000, queueLoop)
0090
0091     def exitOnClose():
0092         global master
0093
0094         if askokcancel("CERRAR", "Realmente desea cerrar la ventana actual"):
0095             master.destroy()
0096
0097             if master != 'root':
0098                 root.deiconify()
0099
0100     def cancelar():
0101         global master
0102
0103         if askokcancel("CANCELAR", "Desea cancelar la operacion actual"):
0104             master.destroy()
0105
0106             if master != 'root':
0107                 root.deiconify()
0108
0109     def movAuto():
0110         global pause, runHilo
0111
0112         if runHilo:
0113             print 'mov'
0114             mv.playPause(pause)
0115             runHilo = False
0116
0117     def registros():
0118         global archivo, carpeta, fechaReg
0119         global mando, tipoPul, horas, minutos, GAP, avancGap, esc, velocidad
0120         global modeloX, modeloY, modeloZ, resolX, resolY, resolZ
0121         global electrolito, disolucion
0122         global espTrabX, espTrabY, espTrabZ, dPzTrabX, dPzTrabY, dPzTrabZ
0123         global dIntHerr, dExtHerr, largHerr, ls1, ls2, ls3, ls4, ls5, ls6
0124         global tipoMetal, materialElect
0125         t = datetime.time(horas, minutos)
0126         listVar = [
0127             'joystick: ',mando,'\n',

```

```

0128         'pulido: ', tipoPul, '\n',
0129         'fecha de registro: ', fechaReg, '\n',
0130         'duracion (HH:MM:SS): ', t, '\n',
0131         'GAP inicial: ', GAP, '\n',
0132         'avance entre capas: ', avancGap, ' ', esc, '\n',
0133         'velocidad: ', velocidad, '\n',
0134         'electrolito: ', electrolito, ' saturacion: ', disolucion, '
gr/L\n',
0135         '\tMATERIAL\npieza de trabajo: ', tipoMetal, '\n',
0136         'electrodo: ', materialElect, '\n',
0137         'driver (X, Y, Z): ', modeloX, ' ', modeloY, ' ', modeloZ, '\n',
0138         'resolucion (X, Y, Z): ', resolX, ' ', resolY, ' ', resolZ,
'\n',
0139         '\tDIMENSIONES\nespacio de trabajo(X, Y, Z)\n',
0140         espTrabX, ' ', espTrabY, ' ', espTrabZ, '\n',
0141         'pieza a trabajar(X, Y, Z)\n',
0142         dPzTrabX, ' ', dPzTrabY, ' ', dPzTrabZ, '\n',
0143         'electrodo (diametro; interno, externo, largo)\n',
0144         dIntHerr, ' ', dExtHerr, ' ', largHerr, '\n',
0145         'finales de carrera\nx+: ', ls1, 'x-: ', ls2, '\n',
0146         'y+: ', ls3, 'y-: ', ls4, '\n',
0147         'z+: ', ls5, 'z-: ', ls6,]
0148     crear = open ((carpeta+archivo+'.txt'), "w")
0149
0150     for var in listVar:
0151         crear.write(str(var))
0152     crear.close()
0153
0154     def rapido():
0155         root.withdraw()
0156         pulido()
0157
0158     def nuevo():
0159
0160         def ECP():
0161             global carpeta, archivo
0162             archivo=nombre.get().replace('-', '').replace(':', '')
0163             carpeta=ruta.get()
0164             winNvo.destroy()
0165             parametros()
0166
0167         def cambiar():
0168             cam = askdirectory()
0169             if cam is None:
0170                 return
0171             nRut = os.path.split(cam)[0]+os.sep+os.path.split(cam)[1]+os.sep
0172             ruta.set(nRut)
0173         root.withdraw()
0174         winNvo = Toplevel(root)
0175         winNvo.geometry("400x150+400+50")
0176         winNvo.wm_title("ECP - NUEVO DOCUMENTO")
0177         global carpeta, master, fechaReg
0178         nombre, ruta = StringVar(), StringVar()
0179
0180         master = winNvo
0181         fecha = time.strftime("%d-%m-%y")
0182         hora = time.strftime("%H:%M")
0183         fechaReg = fecha + "_" + hora
0184         arc = "ECP_" + fechaReg
0185
0186         nombre.set(arc)
0187         ruta.set(carpeta)
0188
0189         frameWin=Frame(winNvo, height=250, relief=RIDGE, bd=2, bg='#266922')
0190
0191         frameDoc=Frame(frameWin, relief=SUNKEN, bd=3, width=600, bg='OliveDrab4')
0192         Label(frameDoc, text='DOCUMENTO', bg='OliveDrab4').grid(columnspan=3)
0193         Label(frameDoc, text='NOMBRE:', bg='OliveDrab4').grid(row=1)
0194         nom=Entry(frameDoc, width=25, textvariable=nombre, bg='lightGoldenrodYellow',

```

```

highlightcolor='gold').grid(row=1, column=1)
0195     Label(frameDoc, text='RUTA: ', bg='OliveDrab4').grid(row=2)
0196     rut=Entry(frameDoc, width=25, textvariable=ruta, bg='lightGoldenrodYellow',
highlightcolor='gold').grid(row=2, column=1)
0197
0198     btnCamb=Button(frameDoc, text="CAMBIAR", command=cambiar, bd=3, width=5,
bg='OliveDrab4', activebackground='gold', highlightcolor='gold').grid(row=2, column=2, padx=5)
0199
0200     for i in range(3):
0201         frameDoc.columnconfigure(i, weight=3)
0202         frameDoc.rowconfigure(i, weight=2)
0203
0204     frameDoc.grid(columnspan=5, padx=5, pady=5)
0205
0206     btnSig = Button(frameWin, text="SIGUIENTE", command=ECP, bd=5, width=6,
bg='OliveDrab4', activebackground='gold', highlightcolor='gold')
0207     btnSig.grid(row=2, column=1, pady=2)
0208     btnCanc = Button(frameWin, text="CANCELAR", command=cancelar, bg='OliveDrab4',
activebackground='gold', highlightcolor='gold', bd=5, width=6)
0209     btnCanc.grid(row=2, column=3, pady=2, sticky=W)
0210
0211     for i in range(4):
0212         frameWin.columnconfigure(i, weight=4)
0213     frameWin.pack(side=LEFT, expand=1, fill=BOTH)
0214     winNvo.protocol("WM_DELETE_WINDOW", exitOnClose)
0215
0216     def guardar():
0217         file_path = savefilename()
0218         print "guardando archivo"
0219
0220     def salir():
0221
0222         if askyesno("Salir", "Realmente desea salir"):
0223             root.destroy()
0224             GPIO.cleanup()
0225
0226     def parametros():
0227
0228         def toggleLim():
0229
0230             if btnSel.config('text')[-1] == 'TODOS':
0231                 ls1 = sw1.set(1)
0232                 ls2 = sw2.set(1)
0233                 ls3 = sw3.set(1)
0234                 ls4 = sw4.set(1)
0235                 ls5 = sw5.set(1)
0236                 ls6 = sw6.set(1)
0237                 btnSel.config(text='NINGUNO')
0238
0239             else:
0240                 btnSel.config(text='TODOS')
0241                 ls1 = sw1.set(0)
0242                 ls2 = sw2.set(0)
0243                 ls3 = sw3.set(0)
0244                 ls4 = sw4.set(0)
0245                 ls5 = sw5.set(0)
0246                 ls6 = sw6.set(0)
0247
0248         def Lim():
0249             global ls1, ls2, ls3, ls4, ls5, ls6
0250             ls1 = sw1.get()
0251             ls2 = sw2.get()
0252             ls3 = sw3.get()
0253             ls4 = sw4.get()
0254             ls5 = sw5.get()
0255             ls6 = sw6.get()
0256
0257         def establecer():
0258             global modeloX, modeloY, modeloZ

```

```

0259         global resolX, resolY, resolZ
0260         eje = ej.get()
0261
0262     if eje == 'TODOS':
0263         modeloX = mod.get()
0264         modeloY = mod.get()
0265         modeloZ = mod.get()
0266         resolX = res.get()
0267         resolY = res.get()
0268         resolZ = res.get()
0269
0270     elif eje == 'X':
0271         modeloX = mod.get()
0272         resolX = res.get()
0273
0274     elif eje == 'Y':
0275         modeloY = mod.get()
0276         resolY = res.get()
0277
0278     elif eje == 'Z':
0279         modeloZ = mod.get()
0280         resolZ = res.get()
0281
0282 def detectar():
0283     lblJoy['bg']=['lightGoldenrodYellow']
0284     pygame.init()
0285     pygame.joystick.init()
0286
0287     try:
0288         j = pygame.joystick.Joystick(0) # Instancia Joystick
0289         j.init()
0290         joy.set(j.get_name())
0291
0292     except pygame.error:
0293         joy.set('joystick no encontrado')
0294
0295 def selectMetal(event):
0296     global tipoMetal
0297     tipoMetal = metal.get()
0298
0299 def selectElect(event):
0300     global materialElect
0301     materialElect = matElect.get()
0302
0303 def terminar():
0304     global mando
0305     global modeloX, modeloY, modeloZ
0306     global resolX, resolY, resolZ
0307     espTrabX = baseX.get()
0308     espTrabY = baseY.get()
0309     espTrabZ = baseZ.get()
0310     dPzTrabX = pzX.get()
0311     dPzTrabY = pzY.get()
0312     dPzTrabZ = pzZ.get()
0313     dIntHerr = herrX.get()
0314     dExtHerr = herrY.get()
0315     largHerr = herrZ.get()
0316     electrolito = disol.get()
0317     disolucion = conc.get()
0318     tipoMetal = metal.get()
0319     materialElect = matElect.get()
0320     mando = joy.get()
0321     mv.setEjes(modeloX, modeloY, modeloZ, resolX, resolY, resolZ)
0322     mv.setMedPz(dPzTrabX, dPzTrabY, dPzTrabZ)
0323     mv.setMedHerram(dIntHerr, dExtHerr, largHerr)
0324     pulido()
0325     winParam.destroy()
0326
0327 def regresar():

```

```

0328         nuevo()
0329         winParam.destroy()
0330     winParam = Toplevel(root)
0331     winParam.geometry("680x460+500+50")
0332     winParam.resizable(width=0, height=0)
0333     winParam.wm_title("Ajuste de parametros")
0334     global archivo, mando, master
0335     global duraHoras, duraMin, velocidad, avancGap
0336     global ls1, ls2, ls3, ls4, ls5, ls6
0337     global espTrabX , espTrabY, espTrabZ
0338     global dPzTrabX, dPzTrabY, dPzTrabZ
0339     global dIntHerr, dExtHerr, largHerr
0340     global electrolito, disolucion
0341     global tipoMetal
0342     global materialElect
0343     master = winParam
0344     ej, mod, res = StringVar(), StringVar(), StringVar()
0345     sw1, sw2 = IntVar(), IntVar()
0346     sw3, sw4 = IntVar(), IntVar()
0347     sw5, sw6 = IntVar(), IntVar()
0348     baseX, baseY, baseZ = IntVar(), IntVar(), IntVar()
0349     pzX, pzY, pzZ = IntVar(), IntVar(), IntVar()
0350     herrX, herrY, herrZ = IntVar(), IntVar(), IntVar()
0351     joy, disol, conc = StringVar(), StringVar(), IntVar()
0352     metal, matElect = StringVar(), StringVar()
0353     listEjes = ['X', 'Y', 'Z', 'TODOS']
0354     listModDriver = ['TB6560', 'L297']
0355     listRes = ['COMPLETO', '(1/2)-MEDIO', '(1/8)-PASO', '(1/16)-PASO']
0356     ej.set(listEjes[3])
0357     mod.set(listModDriver[0])
0358     res.set(listRes[3])
0359     limitSw = [
0360         ('X+', sw1, 0, 1), ('X-', sw2, 0, 2),
0361         ('Y+', sw3, 1, 1), ('Y-', sw4, 1, 2),
0362         ('Z+', sw5, 2, 1), ('Z-', sw6, 2, 2)]
0363     baseX.set(espTrabX), baseY.set(espTrabY), baseZ.set(espTrabZ)
0364     pzX.set(dPzTrabX), pzY.set(dPzTrabY), pzZ.set(dPzTrabZ)
0365     herrX.set(dIntHerr), herrY.set(dExtHerr), herrZ.set(largHerr)
0366     disol.set(electrolito), conc.set(disolucion)
0367
0368     #GUI
0369     framWin=Frame(winParam, relief=RIDGE, bd=2, bg='OliveDrab4')
0370     framSisMec=LabelFrame(framWin, bg='#266922', relief=RIDGE, bd=2, text='SISTEMA
MECANICO')
0371     Label(framSisMec, text='Finales de carrera', bg='#266922').grid(columnspan=4)
0372
0373     for text, val, y_pos, x_pos in limitSw:
0374         sw = Checkbutton(framSisMec, text=text, variable=val, offvalue=0,
onvalue=1, bg='OliveDrab4', activebackground='gold', highlightcolor='gold')
0375         sw.grid(row=x_pos, column=y_pos)
0376         btnSel = Button(framSisMec, text='TODOS', command=toggleLim, width=5, height=1,
bg='OliveDrab4', activebackground='gold', highlightcolor='gold',bd=2)
0377         btnSel.grid(row=1, column=3, rowspan=2, sticky=(W, E))
0378         btnLim = Button(framSisMec, text="Establecer limites", command=Lim,
bg='OliveDrab4', activebackground='gold', highlightcolor='gold',bd=5)
0379         btnLim.grid(row=3, column=1, columnspan=2, sticky=(W, E))
0380         Separator(framSisMec, orient=HORIZONTAL).grid(row=4, columnspan=4, sticky=(W, E))
0381         Label(framSisMec, text='ejes', bg='#266922').grid(row=5, column=0, sticky=E)
0382         menuEjes = OptionMenu(framSisMec, ej, *listEjes)
0383         menuEjes.config(bg='lightGoldenrodYellow', activebackground='gold',
highlightcolor='gold', takefocus=True, width=10)
0384         menuEjes.grid(row=5, column=1, columnspan=3)
0385         Label(framSisMec, text='Modelo', bg='#266922').grid(row=6, column=0, sticky=E)
0386         menuDriver = OptionMenu(framSisMec, mod, *listModDriver)
0387         menuDriver.config(bg="lightGoldenrodYellow", activebackground='gold',
highlightcolor='gold', takefocus=True, width=10)
0388         menuDriver.grid(row=6, column=1, columnspan=3)
0389         Label(framSisMec, text='Resolucion', bg='#266922').grid(row=7, column=0, sticky=E)
0390         menuResol = OptionMenu(framSisMec, res, *listRes)

```

```

0391         menuResol.config(bg="lightGoldenrodYellow",             activebackground='gold',
highlightcolor='gold', takefocus=True, width=10)
0392     menuResol.grid(row=7, column=1, columnspan=3)
0393     btnAx = Button(framSisMec, text="modificar", command=establecer, bg='OliveDrab4',
activebackground='gold', highlightcolor='gold', bd=5)
0394     btnAx.grid(row=8, column=1, columnspan=2, sticky=(E, W), padx=3)
0395
0396     for i in range(8):
0397         framSisMec.rowconfigure(i, weight=2)
0398
0399         if i < 3:
0400             framSisMec.columnconfigure(i, weight=3)
0401     framSisMec.pack(side=LEFT, expand=1, fill=BOTH)
0402     framDim=LabelFrame(framWin, bg='#266922', relief=RIDGE, bd=2, text='Dimensiones')
0403     Label(framDim, text='Espacio de trabajo', bg='#266922').grid(row=0, column=0,
columnspan=3, pady=10)
0404     dimBaseX = Entry(framDim, width=4, textvariable=baseX, bg='lightGoldenrodYellow',
highlightcolor='gold').grid(row=1)
0405     dimBaseY = Entry(framDim, width=4, textvariable=baseY, bg='lightGoldenrodYellow',
highlightcolor='gold').grid(row=1, column=1)
0406     dimBaseZ = Entry(framDim, width=4, textvariable=baseZ, bg='lightGoldenrodYellow',
highlightcolor='gold').grid(row=1, column=2)
0407     Label(framDim, text='mm', bg='#266922').grid(row=1, column=3)
0408     Label(framDim, text='X\tY\tZ', bg='#266922').grid(row=2, column=0, columnspan=3,
sticky=(E, W), pady=5)
0409     Separator(framDim, orient=HORIZONTAL).grid(row=3, columnspan=4, sticky=(W, E),
pady=10)
0410     Label(framDim, text='Pieza de trabajo', bg='#266922').grid(row=4, columnspan=3,
pady=5)
0411     dimPzX = Entry(framDim, width=4, textvariable=pzX, bg='lightGoldenrodYellow',
highlightcolor='gold').grid(row=5)
0412     dimPzY = Entry(framDim, width=4, textvariable=pzY, bg='lightGoldenrodYellow',
highlightcolor='gold').grid(row=5, column=1)
0413     dimPzZ = Entry(framDim, width=4, textvariable=pzZ, bg='lightGoldenrodYellow',
highlightcolor='gold').grid(row=5, column=2)
0414     Label(framDim, text='mm', bg='#266922').grid(row=5, column=3)
0415     Label(framDim, text='X\tY\tZ', bg='#266922').grid(row=6, column=0, columnspan=3,
sticky=(E, W), pady=2)
0416     Separator(framDim, orient=HORIZONTAL).grid(row=7, columnspan=4, sticky=(W, E),
pady=5)
0417     Label(framDim, text='Herramienta', bg='#266922').grid(row=8, column=0,
columnspan=3, pady=5)
0418     dimHerrX = Entry(framDim, width=4, textvariable=herrX, bg='lightGoldenrodYellow',
highlightcolor='gold').grid(row=9)
0419     dimHerrY = Entry(framDim, width=4, textvariable=herrY, bg='lightGoldenrodYellow',
highlightcolor='gold').grid(row=9, column=1)
0420     dimHerrZ = Entry(framDim, width=4, textvariable=herrZ, bg='lightGoldenrodYellow',
highlightcolor='gold').grid(row=9, column=2)
0421     Label(framDim, text='mm', bg='#266922').grid(row=9, column=3)
0422     Label(framDim, text='diametro\t largo', bg='#266922').grid(row=10, column=0,
columnspan=3)
0423     Label(framDim, text='int.\text.', bg='#266922').grid(row=11, column=0,
columnspan=2, sticky=N)
0424
0425     for i in range(11):
0426         framDim.rowconfigure(i, weight=2)
0427
0428         if i < 4:
0429             framDim.columnconfigure(i, weight=3)
0430     framDim.pack(side=LEFT, expand=1, fill=BOTH)
0431     framCar=LabelFrame(framWin, bg='#266922', relief=RIDGE, bd=2,
text='Carcteristicas')
0432     Label(framCar, text='Electrolito', bg='#266922').grid(row=0, column=0)
0433     tipoDisol = Entry(framCar, width=10, textvariable=disol,
bg='lightGoldenrodYellow', highlightcolor='gold').grid(row=0, column=1)
0434     Label(framCar, text='Concentracion', bg='#266922').grid(row=1, column=0)
0435     concentracion = Entry(framCar, width=10, textvariable=conc,
bg='lightGoldenrodYellow', highlightcolor='gold').grid(row=1, column=1)
0436     Label(framCar, text='gr/L', bg='#266922').grid(row=1, column=2)

```

```

0437 Separator(framCar, orient=HORIZONTAL).grid(row=2, colspan=4, sticky=(N, E, W))
0438 Label(framCar, text='Aleacion', bg='#266922').grid(row=3, column=0)
0439 metal = Combobox(framCar)
0440 metal.config(width=10, textvariable=metal)
0441 metal.grid(row=3, column=1, colspan=2)
0442 metal['values']=tipoMetal
0443 metal.bind('<<ComboboxSelected>>', selectMetal)
0444 metal.current(0)
0445 Separator(framCar, orient=HORIZONTAL).grid(row=4, colspan=4, sticky=(W, E))
0446 Label(framCar, text='Material', bg='#266922').grid(row=5, column=0)
0447 tipoElect = Combobox(framCar)
0448 tipoElect.config(width=10, textvariable=matElect)
0449 tipoElect['values']=materialElect
0450 tipoElect.current(0)
0451 tipoElect.bind('<<ComboboxSelected>>', selectElect)
0452 tipoElect.grid(row=5, column=1, colspan=2)
0453
0454 for i in range(7):
0455     framCar.rowconfigure(i, weight=2)
0456
0457     if i < 3:
0458         framCar.columnconfigure(i, weight=3)
0459 framCar.pack(side=LEFT, expand=1, fill=BOTH)
0460 framWin.pack(expand=1, fill=BOTH)
0461 framBot=Frame(winParam, relief=RIDGE, bd=2, bg='#266922')
0462 Label(framBot, text='Disposicion', bg='#266922').grid()
0463 imagen = PhotoImage(file='imagenes/posEjes.png')
0464 lblDisp = Label(framBot, image=imagen, relief=SUNKEN, borderwidth=3)
0465 lblDisp.image = imagen
0466 lblDisp.grid(row=1, rowspan=2, sticky=(N, E, W, S))
0467 Separator(framBot).grid(row=0, rowspan=3, column=1, sticky=(N, S))
0468 Label(framBot, text='Mando', bg='#266922').grid(row=0, column=2, colspan=2)
0469 lblJoy=Label(framBot, textvariable=joy, relief=SUNKEN, width=35, height=2,
borderwidth=3, bg='OliveDrab4')
0470 lblJoy.grid(row=1, column=2, colspan=2, sticky=(W), padx=10)
0471 btnDet = Button(framBot, text="Detectar", command=detectar, bg='OliveDrab4',
activebackground='gold', highlightcolor='gold', bd=5, width=6, height=1)
0472 btnDet.grid(row=1, column=4, padx=5)
0473 btnTerm = Button(framBot, text="Terminar", command=terminar, bg='OliveDrab4',
activebackground="gold", highlightcolor='gold', bd=5, width=5)
0474 btnTerm.grid(row=2, column=2)
0475 btnAtras = Button(framBot, text="Regresar", command=regresar, bg='OliveDrab4',
activebackground='gold', highlightcolor='gold', bd=5, width=5)
0476 btnAtras.grid(row=2, column=3)
0477 btnCanc = Button(framBot, text="Cancelar", command=cancelar, bg='OliveDrab4',
activebackground='gold', highlightcolor='gold', bd=5, width=5)
0478 btnCanc.grid(row=2, column=4)
0479
0480 for i in range(4):
0481     framBot.columnconfigure(i, weight=3)
0482     framBot.rowconfigure(i, weight=2)
0483
0484     if i < 3: framBot.rowconfigure(i, weight=2)
0485 framBot.pack(expand=1, fill=BOTH)
0486 winParam.protocol("WM_DELETE_WINDOW", exitOnClose)
0487
0488 def pulido():
0489
0490     def setEscXY(event):
0491         txtMsj.set('La resolucio del plano XY cambio: '+ escalaXY.get())
0492
0493     def setEscZ(event):
0494         txtMsj.set('La resolucio del plano Z cambio: '+ escalaZ.get())
0495
0496     def setGAP(event):
0497         txtMsj.set('La resolucio del GAP cambio: '+ escalaGap.get())
0498
0499     def btnSetGap():
0500         global avancGap, esc, GAP

```

```

0501         GAP = GAPini.get()
0502         m = float(magnitudGap.get())
0503         if m >= 0:
0504             avancGap = m
0505             esc = escalaGap.get()
0506             mv.setAvancCapa(avancGap, esc)
0507         txtMsj.set('El avance (GAP) por cada capa es de: '+ str(m) + ' ' + esc +
'(s)')
0508
0509     def btnXPos():
0510         global sentido, eje, magnitud, escala
0511         m = float(magnitudXY.get())
0512         if m > 0:
0513             magnitud = m
0514             sentido = 0
0515             eje = 'X'
0516             escala = escalaXY.get()
0517             cola.put(mv.manual(sentido, eje, magnitud, escala))
0518             cola.put(cv.xp())
0519             cola.put(txtMsj.set('Se movio X de forma positiva ' + str(magnitud)
+ ' ' + escala + '(s)'))
0520         else:
0521             txtMsj.set("introduzca un numero positivo en XY")
0522
0523     def btnXNeg():
0524         m = float(magnitudXY.get())
0525         if m > 0:
0526             magnitud = m
0527             sentido = 1
0528             eje = 'X'
0529             escala = escalaXY.get()
0530             cola.put(mv.manual(sentido, eje, magnitud, escala))
0531             cola.put(cv.xl())
0532             txtMsj.set('Se movio X de forma negativa ' + str(magnitud) + ' '
+ escala + '(s)')
0533         else:
0534             txtMsj.set("introduzca un numero positivo en XY")
0535
0536     def btnYPos():
0537         m = float(magnitudXY.get())
0538         if m > 0:
0539             magnitud = m
0540             sentido = 1
0541             eje = 'Y'
0542             escala = escalaXY.get()
0543             cola.put(mv.manual(sentido, eje, magnitud, escala))
0544             cv.yf()
0545             cola.put(txtMsj.set('Se movio Y de forma positiva '
+ str(magnitud) + ' ' + escala + '(s)'))
0546         else:
0547             cola.put(txtMsj.set("introduzca un numero positivo en XY"))
0548
0549     def btnYNeg():
0550         m = float(magnitudXY.get())
0551         if m > 0:
0552             magnitud = m
0553             sentido = 0
0554             eje = 'Y'
0555             escala = escalaXY.get()
0556             cola.put(mv.manual(sentido, eje, magnitud, escala))
0557             cola.put(cv.yl())
0558             txtMsj.set('Se movio Y de forma negativa ' + str(magnitud) + ' ' +
escala + '(s)')
0559         else:
0560             txtMsj.set("introduzca un numero positivo en XY")
0561
0562     def btnZPos():
0563         m = float(magnitudZ.get())
0564         if m > 0:

```



```

0566             magnitud = m
0567             sentido = 0
0568             eje = 'Z'
0569             escala = escalaZ.get()
0570             cola.put(mv.manual(sentido, eje, magnitud, escala))
0571             sepp=pos.get()
0572             sepp-=1
0573             gapElect['value']=sepp
0574             pos.set(sepp)
0575             txtMsj.set('Se movio Z de forma ascendente ' + str(magnitud) + '
'+ escala +'(s)')
0576         else:
0577             txtMsj.set("introduzca un numero positivo en Z")
0578
0579     def btnZNeg():
0580         m = float(magnitudZ.get())
0581         if m > 0:
0582             magnitud = m
0583             sentido = 1
0584             eje = 'Z'
0585             escala = escalaZ.get()
0586             cola.put(mv.manual(sentido, eje, magnitud, escala))
0587             sepp=pos.get()
0588             sepp+=1
0589             gapElect['value']=sepp
0590             pos.set(sepp)
0591             txtMsj.set('Se movio Z de forma descendente ' + str(magnitud) + '
'+ escala +'(s)')
0592         else:
0593             txtMsj.set("introduzca un numero positivo en Z")
0594
0595     def cronometro():
0596         global horas, minutos, dat
0597         global pause, cronoSS, cronoMM, cronoHH
0598         d.lectura()
0599         v=d.getVolt()
0600         c=d.getCurr()
0601         if pause==0:
0602             cronoSS += 1
0603             if cronoSS>59:
0604                 cronoMM +=1
0605                 cronoSS = 0
0606             elif cronoMM>59:
0607                 cronoHH +=1
0608                 cronoMM = 0
0609             elif cronoHH > 23:
0610                 cronoHH = 0
0611             crono = datetime.time(cronoHH, cronoMM, cronoSS)
0612             termino = datetime.time(horas, minutos, 0)
0613             if crono <= termino:
0614                 txtTimTrans.set(crono)
0615                 cola.put(txtVolts.set(v))
0616                 cola.put(txtCurr.set(c))
0617                 cola.put(lblVarT_T.after(1000, cronometro))#890 con for a
10
0618             elif crono > termino:
0619                 pause=1
0620                 mv.termino()
0621                 print "pulido terminado"
0622                 detener()
0623                 cronoHH = 0
0624                 cronoMM = 0
0625                 cronoSS = 0
0626         else:
0627             pausar()
0628             return
0629
0630     def iniciar():
0631         global pause

```

```

0632         pause = 0
0633         hora=txtHoraIni.get()
0634         if hora == '-----':
0635             hora_ini = time.strftime("%H:%M:%S")
0636             txtMsj.set("El proceso se ha iniciado")
0637             txtHoraIni.set(hora_ini)
0638         else:
0639             txtMsj.set("Proceso reanudado")
0640         btnXPlus['state']=['disable']
0641         btnXLess['state']=['disable']
0642         btnYPlus['state']=['disable']
0643         btnYLess['state']=['disable']
0644         btnZPlus['state']=['disable']
0645         btnZLess['state']=['disable']
0646         btnPlay['state']=['disable']
0647         btnPause['state']=['normal']
0648         btnStop['state']=['normal']
0649         hiloPlay.start()
0650         cronometro()
0651
0652     def pausar():
0653         global pause
0654         pause = 1
0655         mv.playPause(pause)
0656         txtMsj.set("Proceso pausado")
0657
0658         hora_fin = time.strftime("%H:%M:%S")
0659         txtHoraTerm.set(hora_fin)
0660         btnXPlus['state']=['normal']
0661         btnXLess['state']=['normal']
0662         btnYPlus['state']=['normal']
0663         btnYLess['state']=['normal']
0664         btnZPlus['state']=['normal']
0665         btnZLess['state']=['normal']
0666
0667         btnPlay['state']=['normal']
0668         btnPause['state']=['disable']
0669         btnStop['state']=['normal']
0670
0671     def detener():
0672         global mediciones
0673         txtMsj.set("Proceso terminado")
0674         hora_fin = time.strftime("%H:%M:%S")
0675         txtHoraTerm.set(hora_fin)
0676         btnStop['state']=['disabled']
0677         btnPause['state']=['disabled']
0678         btnPlay['state']=['disabled']
0679         registros()
0680         d.insertarDatos()
0681         winPul.destroy()
0682         root.deiconify()
0683
0684     def tiempo():
0685         global horas, minutos
0686         horas = hour.get()
0687         minutos = minut.get()
0688         if horas > 0 or minutos > 0:
0689             btnPlay['state']=['normal']
0690             mv.setTiempo(horas, minutos)
0691             t = datetime.time(horas, minutos)
0692             txtMsj.set('el tiempo de ejecucion se cambio a (HH:MM)' + str(t))
0693         else:
0694             btnPlay['state']=['disable']
0695             txtMsj.set('el tiempo de ejecucion debe ser mayor de cero')
0696
0697     def camVel(vel):
0698         global velocidad
0699         velocidad = vel
0700         mv.setVelocidad(vel)

```

```

0701
0702     def camPulido(pul):
0703         txtMsj.set('El tipo de pulido se cambio a ' + pul)
0704         mv.setPul(pul)
0705
0706         winPul = Toplevel(root)
0707         winPul.geometry("850x750+230+0")
0708         winPul.resizable(width=True, height=True)
0709         winPul.wm_title("Pulido Electro-quimico ECP")
0710         global master
0711         global ls1, ls2, ls3, ls4, ls5, ls6
0712         global archivo, mediciones, GAP
0713         nombreTabla=archivo.lower().replace('-', '').replace(':', '').replace('.', '').replace(' ', '')
0714         mediciones='/home/pi/Code/ECP/' + nombreTabla +
'BD.csv'#/tmp/' + nombreTabla + 'BD.csv'
0715         db.setDir(nombreTabla, mediciones)
0716         cola = Queue.Queue()
0717         hiloPlay = threading.Thread(target=movAuto, name='Automatico')
0718         master=winPul
0719         txtMsj=StringVar()
0720         magnitudXY=StringVar()
0721         escalaXY=StringVar()
0722         magnitudZ=StringVar()
0723         escalaZ=StringVar()
0724         GAPini=IntVar()
0725         magnitudGap=IntVar()
0726         escalaGap=StringVar()
0727         txtVolts=StringVar()
0728         txtCurr=StringVar()
0729         txtHoraIni=StringVar()
0730         txtTimTrans=StringVar()
0731         txtHoraTerm=StringVar()
0732         pos = IntVar()
0733         minim = IntVar()
0734         maxim = IntVar()
0735         vel = IntVar()
0736         hour = IntVar()
0737         minut = IntVar()
0738         mov = StringVar()
0739         pos.set(50)
0740         minim.set(0)
0741         maxim.set(100)
0742         txtMsj.set('esperando acciones')
0743         txtVolts.set('0.0000')
0744         txtCurr.set('0.0000')
0745         txtHoraIni.set('-----')
0746         txtTimTrans.set('00:00:00')
0747         txtHoraTerm.set('-----')
0748         vel.set(700)
0749         hour.set(0)
0750         minut.set(15)
0751         magnitudGap.set('0')
0752         magnitudXY.set('0')
0753         magnitudZ.set('0')
0754         listMov = ['ESCALERA/ZIG-ZAG', 'CARACOL/ESPIRAL', 'RECTA HORIZONTAL(X)', 'RECTA
VERTICAL(Y)', 'PERIMETRO CUADRADO']
0755         mov.set(listMov[0])
0756         frame=Frame(winPul, relief=SUNKEN, bd=5, bg='OliveDrab4')
0757         sisMecFrame=LabelFrame(frame, bg='#266922', relief=RIDGE, bd=2, text='SISTEMA
MECANICO')
0758         Label(sisMecFrame, text='Posicion de los ejes (X, Y), Z', bg='OliveDrab4',
width=25).grid(row=0, column=0, columnspan=2, padx=10, pady=10, sticky=(W, E))
0759         frameEjes=Frame(sisMecFrame, relief=GROOVE, bd=3, bg='light yellow')
0760         cv.winCan(frameEjes, 300, 300)
0761         cv.setPos(150,150)
0762         frameEjes.grid(row=1, column=0, padx=10, pady=10)
0763         gapElect= Progressbar(sisMecFrame, orient='vertical', length=100,
mode='indeterminate')
0764         gapElect['value']=pos.get()

```

```

0765         gapElect['maximum']=maxim.get()
0766         gapElect.grid(row=1, column=1, padx=10)
0767         lblMens=Label( sisMecFrame, textvariable=txtMsj, bg='light yellow', width=50)
0768         lblMens.grid(row=2, column=0, columnspan=2, padx=10, pady=10)
0769         sisMecFrame.pack(side=LEFT, fill=BOTH, expand=1)
0770         frameMov=LabelFrame(frame, bg='#266922', relief=RIDGE, bd=2, text="AJUSTES")
0771         Scale(frameMov, bg='#266922', label='VELOCIDAD', variable=vel, bd=2, width=20,
from_=1, to=1400, command=camVel, resolution=0.1, orient=HORIZONTAL, activebackground='gold',
showvalue=0).grid(row=0, column=0, columnspan=2, padx=10)
0772         velTeclado = Entry(frameMov, width=4, textvariable=vel, highlightcolor='gold')
0773         velTeclado.grid(row=0,column=2, columnspan=2)
0774         Label(frameMov, text='GAP inicial\tAvance por capa', bg='#266922').grid(row=1,
columnspan=4)
0775         gapDef = Entry(frameMov, width=4, textvariable=GAPini, highlightcolor='gold')
0776         gapDef.grid(row=2,column=0)
0777         Label(frameMov, text='um', bg='#266922').grid(row=2, column=1)
0778         movGap=Entry(frameMov, width=4, textvariable=magnitudGap, highlightcolor='gold')
0779         movGap.grid(row=2,column=2, columnspan=2)
0780         tipoGAP=Combobox(frameMov, height=2)
0781         tipoGAP.config(width=10, textvariable=escalaGap, state='readonly')
0782         tipoGAP.grid(row=3, column=0, columnspan=4)
0783         tipoGAP['values']=['paso', 'micrometro']
0784         tipoGAP.bind('<<ComboboxSelected>>', setGAP)
0785         tipoGAP.current(0)
0786         btnGap=Button(frameMov, text='CAMBIAR', command=btnSetGap, bg='OliveDrab4')
0787         btnGap.grid(row=4, column=1, columnspan=2)
0788         Label(frameMov, text='Movimiento', bg='#266922').grid(row=5,columnspan=4,
padx=10)
0789         menuResol=OptionMenu(frameMov, mov, *listMov, command=camPulido)
0790         menuResol.config(width=17,bg='lightGoldenrodYellow', activebackground='gold',
highlightcolor='gold', takefocus=True)
0791         menuResol.grid(row=6, columnspan=4)
0792         Label(frameMov, text='Duracion', bg='#266922').grid(row=7, columnspan=4, padx=10)
0793         hrs = Spinbox(frameMov, textvariable=hour, command=tiempo)
0794         hrs.grid(row=8, column=1)
0795         hrs.configure(from_=0, to=23, increment=1, highlightcolor='gold',
bg='lightGoldenrodYellow', activebackground='gold', width=3)
0796         mins = Spinbox(frameMov, textvariable=minut, command=tiempo)
0797         mins.grid(row=8, column=2)
0798         mins.configure(from_=0, to=59, increment=1, highlightcolor='gold',
bg='lightGoldenrodYellow', activebackground='gold', width=3)
0799         Label(frameMov, text='HORA(S)\tMINUTO(S)', bg='#266922').grid(row=9,
columnspan=4, padx=5)
0800
0801         for i in range(9):
0802             frameMov.rowconfigure(i, weight=2)
0803
0804             if i < 4:
0805                 frameMov.columnconfigure(i, weight=4)
0806         frameMov.pack(side=LEFT, fill=BOTH, expand=1)
0807         monitFrame=LabelFrame(frame, bg='#266922', relief=RIDGE, bd=2, text="MONITOREO")
0808         Label(monitFrame, text="VOLTAJE", bg='#266922').grid(row=0,column=0, sticky=W,
padx=5)
0809         Label(monitFrame, textvariable=txtVolts, bg='light yellow').grid(row=0,column=1,
sticky=W)
0810         Label(monitFrame, text="CORRIENTE", bg='#266922').grid(row=1,column=0, sticky=W,
padx=5)
0811         Label(monitFrame, textvariable=txtCurr, bg='light yellow').grid(row=1,column=1,
sticky=W)
0812         Separator(monitFrame, orient=HORIZONTAL).grid(row=2, sticky=(W, E), columnspan=2)
0813         Label(monitFrame, text="INICIO", bg='#266922').grid(row=3,column=0, sticky=W,
padx=5)
0814         Label(monitFrame, textvariable=txtHoraIni, bg='light
yellow').grid(row=3,column=1, sticky=W)
0815         Label(monitFrame, text="EJECUCION", bg='#266922').grid(row=4,column=0, sticky=W,
padx=5)
0816         lblVarT_I=Label(monitFrame, textvariable=txtTimTrans, bg='light yellow')
0817         lblVarT_I.grid(row=4,column=1, sticky=W)
0818         Label(monitFrame, text="TERMINO", bg='#266922').grid(row=5,column=0, sticky=W,

```

```

padx=5)
0819     Label(monitFrame,                textvariable=txtHoraTerm,                bg='light
yellow').grid(row=5,column=1, sticky=W)
0820     monitFrame.columnconfigure(0, weight=2)
0821     monitFrame.columnconfigure(1, weight=3)
0822     monitFrame.pack(side=LEFT, fill=BOTH, expand=1)
0823
0824     for i in range(7):
0825         monitFrame.rowconfigure(i, weight=2)
0826     frame.pack(fill=BOTH, expand=1)
0827     bottomframe=Frame(winPul, bd=5, relief=SUNKEN, bg='OliveDrab4')
0828     Label(bottomframe, text="Controles", bg='OliveDrab4').pack(fill=X)
0829     cntlXYframe=Frame(bottomframe, bd=2, relief=RIDGE, bg='#266922')
0830     Label(cntlXYframe, text='tecla:\nDERECHA', bg='#266922').grid(row=5, column=8)
0831     imgXPlus=PhotoImage(file="images/derecha.png")
0832     btnXPlus=Button(cntlXYframe, image=imgXPlus, command=btnXPos)
0833     btnXPlus.image=imgXPlus
0834     btnXPlus.grid(row=4, column=8)
0835
0836     if ls1:
0837         ledSw1 = Label(cntlXYframe, relief=RIDGE, bd=5, state=DISABLED,
activebackground='red', width=2, height=1, bg='lawn green')
0838         ledSw1.grid(row=4, column=10, padx=5)
0839         Label(cntlXYframe, text='tecla:\nIZQUIERDA', bg='#266922').grid(row=5, column=2)
0840         imgXLess=PhotoImage(file="images/izquierda.png")
0841         btnXLess=Button(cntlXYframe, image=imgXLess, command=btnXNeg)
0842         btnXLess.image=imgXLess
0843         btnXLess.grid(row=4, column=2)
0844
0845     if ls2:
0846         ledSw2=Label(cntlXYframe, relief=RIDGE, bd=5, state=DISABLED,
activebackground='red', width=2, height=1, bg='lawn green')
0847         ledSw2.grid(row=4, column=0, padx=5)
0848         movEje=Entry(cntlXYframe, width=4, textvariable=magnitudXY,
highlightcolor='gold')
0849         movEje.grid(row=4, column=5, padx=5)
0850         tipoMov_XY= Combobox(cntlXYframe, height=2)
0851         tipoMov_XY.config(width=10, textvariable=escalaXY, state='readonly')
0852         tipoMov_XY.grid(row=6,column=4, columnspan=3, pady=5)
0853         tipoMov_XY['values']=['paso', 'micrometro', 'milimetro', 'centimetro']
0854         tipoMov_XY.bind('<<ComboboxSelected>>', setEscXY)
0855         tipoMov_XY.current(2)
0856
0857         Label(cntlXYframe, text='tecla:\nARRIBA', bg='#266922').grid(row=2, column=6)
0858         imgYLess=PhotoImage(file="images/atras.png")
0859         btnYLess=Button(cntlXYframe, image=imgYLess, command=btnYPos)
0860         btnYLess.image=imgYLess
0861         btnYLess.grid(row=2,column=5, pady=5, padx=5)
0862
0863     if ls4:
0864         ledSw4=Label(cntlXYframe, relief=RIDGE, bd=5, state=DISABLED,
activebackground='red', width=2, height=1, bg='lawn green')
0865         ledSw4.grid(row=0,column=5, pady=10)
0866         Label(cntlXYframe, text='tecla:\nABAJO', bg='#266922').grid(row=8, column=6)
0867         imgYPlus=PhotoImage(file="images/delante.png")
0868         btnYPlus=Button(cntlXYframe, image=imgYPlus, command=btnYNeg)
0869         btnYPlus.image=imgYPlus
0870         btnYPlus.grid(row=8,column=5, padx=5, pady=10, sticky=N)
0871
0872     if ls3:
0873         ledSw3=Label(cntlXYframe, relief=RIDGE, bd=5, state=DISABLED,
activebackground='red', width=2, height=1, bg='lawn green')
0874         ledSw3.grid(row=10,column=5, pady=10, sticky=N)
0875
0876     for i in range(10):
0877         cntlXYframe.rowconfigure(i, weight=2)
0878         cntlXYframe.columnconfigure(i, weight=2)
0879     cntlXYframe.pack(side=LEFT, fill=BOTH, expand=1)
0880

```

```

0881         cntlZframe=Frame(bottomframe, bd=2, relief=RIDGE, bg='#266922')
0882         Label(cntlZframe, text='tecla:\nW', bg='#266922').grid(row=2, column=2)
0883         imgZPlus=PhotoImage(file="images/arriba.png")
0884         btnZPlus=Button(cntlZframe,image=imgZPlus, command=btnZPos)
0885         btnZPlus.image=imgZPlus
0886         btnZPlus.grid(row=2,column=1, padx=10, pady=5)
0887
0888         if ls5:
0889             ledSw5=Label(cntlZframe, relief=RIDGE, bd=5, state=DISABLED,
0890 activebackground='red', width=2, height=1, bg='lawn green')
0891             ledSw5.grid(row=0,column=1, padx=10, pady=5)
0892
0893         movEje=Entry(cntlZframe, width=4, textvariable=magnitudZ, highlightcolor='gold')
0894         movEje.grid(row=4,column=1, pady=5)
0895         tipoMovZ=Combobox(cntlZframe, height=2)
0896         tipoMovZ.config(width=10, textvariable=escalaZ, state='readonly')
0897         tipoMovZ.grid(row=6,columnspan=3, pady=5)
0898         tipoMovZ['values']=['paso', 'micrometro', 'milimetro']
0899         tipoMovZ.bind('<<ComboboxSelected>>', setEscZ)
0900         tipoMovZ.current(1)
0901
0902         Label(cntlZframe, text='tecla:\nS', bg='#266922').grid(row=8, column=2)
0903         imgZLess=PhotoImage(file="images/abajo.png")
0904         btnZLess=Button(cntlZframe, image=imgZLess, command=btnZNeg)
0905         btnZLess.image=imgZLess
0906         btnZLess.grid(row=8,column=1, padx=10, pady=5, sticky=N)
0907
0908         if ls6:
0909             ledSw6=Label(cntlZframe, relief=RIDGE, bd=2,
0910 state=DISABLED,activebackground='red', width=2, height=1, bg='lawn green')
0911             ledSw6.grid(row=10,column=1, padx=10, sticky=N)
0912
0913         for i in range(10):
0914             cntlZframe.rowconfigure(i, weight=2)
0915             if i < 3:
0916                 cntlZframe.columnconfigure(i, weight=2)
0917                 cntlZframe.pack(side=LEFT, fill=BOTH, expand=1)
0918                 cntlPlayframe = Frame(bottomframe, bd=5, relief=RIDGE,
0919 bg='#266922')
0920                 imgPlay=PhotoImage(file="images/play.png")
0921                 btnPlay=Button(cntlPlayframe, command=iniciar, image=imgPlay)
0922                 btnPlay.image=imgPlay
0923                 btnPlay.grid(row=0, column=0, padx=10, pady=5)
0924                 imgPause=PhotoImage(file="images/pause.png")
0925                 btnPause=Button(cntlPlayframe, command=pausar, image=imgPause, state=DISABLED)
0926                 btnPause.image=imgPause
0927                 btnPause.grid(row=1, column=0, padx=10, pady=5)
0928                 imgStop=PhotoImage(file="images/stop.png")
0929                 btnStop=Button(cntlPlayframe, command=detener, image=imgStop, state=DISABLED)
0930                 btnStop.image=imgStop
0931                 btnStop.grid(row=2, column=0, padx=10, pady=5)
0932                 cntlPlayframe.columnconfigure(0, weight=3)
0933
0934         for i in range(2):
0935             cntlPlayframe.rowconfigure(i, weight=3)
0936             cntlPlayframe.pack(side=LEFT, fill=BOTH, expand=1)
0937             bottomframe.pack(fill=BOTH, expand=1)
0938             winPul.bind('<Key-Left>', lambda event: mv.manual(1,'X', 100, 'micrometro'))
0939             winPul.bind('<Key-Right>', lambda event: mv.manual(0,'X', 100, 'micrometro'))
0940             winPul.bind('<Key-Down>', lambda event: mv.manual(0,'Y', 100, 'micrometro'))
0941             winPul.bind('<Key-Up>', lambda event: mv.manual(1,'Y', 100, 'micrometro'))
0942             winPul.bind('<Key-w>', lambda event: mv.manual(0,'Z', 100, 'micrometro'))
0943             nPul.bind('<Key-W>', lambda event: mv.manual(0,'Z', 100, 'micrometro'))
0944             winPul.bind('<Key-s>', lambda event: mv.manual(1,'Z', 100, 'micrometro'))
0945             winPul.bind('<Key-S>', lambda event: mv.manual(1,'Z', 100, 'micrometro'))
0946             winPul.protocol("WM_DELETE_WINDOW", exitOnClose)
0947
0948         def hotkeys():
0949             winHotK = Toplevel(root)

```

```

0948         winHotK.geometry("800x600+300+50")
0949         winHotK.wm_title("Teclas rapidas")
0950         winHotK.protocol("WM_DELETE_WINDOW", exitOnClose)
0951
0952     def acerca():
0953         winAb = Toplevel(root)
0954         winAb.geometry("400x200+800+50")
0955         winAb.wm_title("Acerca de...")
0956         winAb.protocol("WM_DELETE_WINDOW", exitOnClose)
0957
0958     def plots():
0959         doc = askopenfilename()
0960         grafica.graff(doc)
0961
0962     try:
0963         root = Tk()
0964         master=root
0965         root.geometry("950x600+180+80")
0966         root.resizable(width=False, height=False)
0967         root.wm_title("Pulido electroquimico")
0968
0969         menuBarra = Menu(root, bg='#266922', activebackground='gold')
0970         menuArchivo = Menu(menuBarra, tearoff=0, bg='OliveDrab4', activebackground='gold')
0971         menuBarra.add_cascade(label="Archivo", underline=0, menu=menuArchivo)
0972         menuArchivo.add_command(label="Nuevo", underline=0, command=nuevo,
0973         accelerator="Ctrl+n")
0974         menuArchivo.add_command(label="Graficar", underline=0, command=plots,
0975         accelerator="Ctrl+a")
0976         menuArchivo.add_separator()
0977         menuArchivo.add_command(label="Salir", underline=0, command=salir,
0978         accelerator="Ctrl+q")
0979         menuAyuda = Menu(menuBarra, tearoff=0, bg='OliveDrab4', activebackground='gold')
0980         menuBarra.add_cascade(label="Ayuda", underline=1, menu=menuAyuda)
0981         menuAyuda.add_command(label="Atajos", underline=0, command=hotkeys,
0982         accelerator="Ctrl+h")
0983         menuAyuda.add_command(label="Acerca de...", underline=0, command=acerca,
0984         accelerator="Ctrl+e")
0985         root.config(menu=menuBarra)
0986
0987         frameMarco=Frame(root, bg='#266922')
0988         btnAbrir = Button(frameMarco, width=6, height=3, text="INICIO\nRAPIDO", bd=2,
0989         command=rapido, bg='OliveDrab4', activebackground='gold', highlightcolor='gold')
0990         btnAbrir.grid(row=0, padx=10, pady=3)
0991         btnGraf = Button(frameMarco, width=6, height=3, text="ABRIR\nY\nGRAFICAR", bd=2,
0992         command=plots, bg='OliveDrab4', activebackground='gold', highlightcolor='gold')
0993         btnGraf.grid(row=1, padx=10, pady=3)
0994         btnSalir = Button(frameMarco, width=6, height=3, text="SALIR", bd=2,
0995         command=salir, bg='OliveDrab4', activebackground='gold', highlightcolor='gold')
0996         btnSalir.grid(row=2, padx=10, pady=3)
0997         frameMarco.pack(side=LEFT, anchor=NE, fill=Y)
0998         frameSep=Frame(root, bg='gold3', width=5)
0999         frameSep.pack(side=LEFT, fill=Y, expand=1)
1000         fondo=PhotoImage(file="images/bg.png")
1001         lblFondo = Label(root, image=fondo).pack(side=LEFT)
1002
1003         root.bind('<Control-n>', lambda event: nuevo())
1004         root.bind('<Control-N>', lambda event: nuevo())
1005         root.bind('<Control-f>', lambda event: rapido())
1006         root.bind('<Control-F>', lambda event: rapido())
1007         root.bind('<Control-g>', lambda event: plots())
1008         root.bind('<Control-G>', lambda event: plots())
1009         root.bind('<Control-q>', lambda event: salir())
1010         root.bind('<Control-Q>', lambda event: salir())
1011         root.bind('<Control-h>', lambda event: hotkeys())
1012         root.bind('<Control-H>', lambda event: hotkeys())
1013         root.bind('<Control-e>', lambda event: acerca())
1014         root.bind('<Control-E>', lambda event: acerca())
1015
1016         root.protocol("WM_DELETE_WINDOW", exitOnClose)

```

```
1009
1010     queueLoop()
1011     root.mainloop()
1012
1013 except KeyboardInterrupt:
1014     raise KeyboardInterrupt, "INTERRUPCION CTRL + C"
1015
1016 finally:
1017     GPIO.cleanup()
1018
```



## 9.2 Diagrama electrónico del sistema

