



**UNIVERSIDAD TÉCNICA DE COTOPAXI**  
**FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS**  
**CARRERA DE INGENIERÍA ELECTROMECAÁNICA**

**PROPUESTA TECNOLÓGICA**

**“DESARROLLO DE UN ENTORNO VIRTUAL 3D PARA EL CONTROL DE NIVEL DE AGUA”**

Proyecto de Titulación presentado previo a la obtención del Título de Ingenieros en  
Electromecánica.

**Autores:**

Intriago Pazmiño Jackson Fernando  
Toapanta Sifaz Jonathan Francisco

**Tutor:**

Ing. M. Sc. Luigi Orlando Freire Martínez

**LATACUNGA – ECUADOR**

**2021**



## DECLARACIÓN DE AUTORÍA

Nosotros, Intriago Pazmiño Jackson Fernando y Toapanta Sifaz Jonathan Francisco declaramos ser autores del presente proyecto de investigación: **“DESARROLLO DE UN ENTORNO VIRTUAL 3D PARA EL CONTROL DE NIVEL DE AGUA”**, siendo el Ing. M. Sc. Luigi Orlando Freire Martínez tutor del presente trabajo; y eximimos expresamente a la Universidad técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certificamos que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de nuestra exclusiva responsabilidad.

---

Intriago Pazmiño Jackson Fernando

C.I. 172173332-5

---

Toapanta Sifaz Jonathan Francisco

C.I. 172691274-2



## AVAL DEL TUTOR DE TITULACIÓN

En calidad de tutor del Trabajo de Investigación sobre el título:

**“DESARROLLO DE UN ENTORNO VIRTUAL 3D PARA EL CONTROL DE NIVEL DE AGUA”**, de Intriago Pazmiño Jackson Fernando, Toapanta Sifaz Jonathan Francisco, de la Carrera de Ingeniería Electromecánica, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnico suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Consejo Directivo de la Facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

Latacunga, Agosto 2021

-----  
El Tutor

Ing. M. Sc. Luigi Orlando Freire Martínez

CI: 0502529589



## APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la Facultad de Ciencias de la Ingeniería y Aplicadas de la Carrera de Ingeniería Electromecánica; por cuanto el o los postulantes: Intriago Pazmiño Jackson Fernando, Toapanta Sifaz Jonathan Francisco, el título de proyecto de titulación: **“DESARROLLO DE UN ENTORNO VIRTUAL 3D PARA EL CONTROL DE NIVEL DE AGUA”**, han considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación del Proyecto.

Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

Latacunga, Agosto 2021

Para constancia firman:

---

**Lector 1 (Presidente)**

Ing. M. Sc. Freire Andrade Verónica Paulina

CI: 050205622-9

---

**Lector 2**

Ing. M. Sc. Corrales Bastidas Byron Paúl

CI:050234776-8

---

**Lector 3**

Ing. M. Sc. Porrás Reyes Jeffeson Alberto

CI:070440044-9





## AVAL DE IMPLEMENTACIÓN

En calidad de Director de la Carrera de Ingeniería Electromecánica de la Universidad Técnica de Cotopaxi certifico que mediante el proyecto de propuesta tecnológica: “**DESARROLLO DE UN ENTORNO VIRTUAL 3D PARA EL CONTROL DE NIVEL DE AGUA**”, de los estudiantes; **INTRIAGO PAZMIÑO JACKSON FERNANDO** y **TOAPANTA SIFAZ JONATHAN FRANCISCO** realizan la entrega de una tarjeta de adquisición de datos, para el Laboratorio de Investigación de Automatización y Control de la Carrera de Ingeniería Electromecánica de la Universidad Técnica de Cotopaxi.

De acuerdo a lo anterior se hace constar que el proyecto de propuesta tecnológica se encuentra en las condiciones adecuadas.

Latacunga, Agosto 2021

.....  
**DIRECTOR DE LA CARRERA**

Ing. Cristian Fabián Gallardo Molina

C.C. 050284769-2



## AGRADECIMIENTO

Quiero agradecer a Dios, a mis padres Eduardo y Nelly ya que con esfuerzo y sacrificio me ayudaron a sobresalir en mi carrera universitaria, y a mi hermana Kerly que ha sido el pilar fundamental para llegar a cumplir este gran sueño.

A mis profesores por haberme mostrado lo linda e importante que es mi carrera y por haberme enseñado siempre el camino del esfuerzo y los buenos valores.

Gracias a todos los que de una u otra manera me ayudaron día a día a crecer como persona y como un futuro Ingeniero Electromecánico.

Jackson Fernando



## AGRADECIMIENTO

Quiero agradecer primero a Dios, por la fortaleza que les dio a mis padres Alonso y Roció ya que con esfuerzo y sacrificio me ayudaron a sobresalir brindándome ánimos en mi carrera universitaria, y a mi tío Efraín que ha sido el apoyo fundamental para llegar a cumplir este gran sueño. Con su ejemplo diario de una persona con buenos valores.

A mis profesores por haberme brindado de su conocimiento para entender lo importante que es mi carrera y por haberme enseñado siempre el camino del esfuerzo y los buenos valores.

Gracias a todos los que de una u otra manera me ayudaron día a día a crecer como persona y como un futuro Ingeniero Electromecánico.

Jonathan Francisco



## DEDICATORIA

Dedico esta tesis, a mis padres, por todo su amor, trabajo en todos estos años, por su educación, consejos, paciencia, por su confianza he logrado llegar hasta aquí y convertirme en lo que soy.

Para mi familia y amigos que me han conocido durante mi carrera universitaria, con su apoyo, aliento y confianza, han realizado sus propios esfuerzos para lograr mis metas académicas.

Jackson Fernando



## DEDICATORIA

Dedico esta tesis, a mis padres, por todo su amor, trabajo en todos estos años, por su educación, consejos, paciencia, por su confianza he logrado llegar hasta aquí y convertirme en lo que soy.

Para aquella persona que a pesar de no formar parte directa de mi familia no ha dudado en brindarme su apoyo su cariño y su amor.

Para mi familia y amigos que me han conocido durante mi carrera universitaria, con su apoyo, aliento y confianza, han realizado sus propios esfuerzos para lograr mis metas académicas.

Jonathan Francisco



**UNIVERSIDAD TÉCNICA DE COTOPAXI**

**FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS**

**TITULO: “DESARROLLO DE UN ENTORNO VIRTUAL 3D PARA EL CONTROL DE NIVEL DE AGUA”**

**Autores:**

**Intriago Pazmiño Jackson Fernando**

**Toapanta Sifaz Jonathan Francisco**

**RESUMEN**

La realización de procesos de control de nivel en la industria, en la actualidad la integración de tecnologías de realidad virtual en sistemas de entrenamiento de control de proceso ha marcado un camino exitoso dentro de la automatización industrial, de esa manera el proyecto contempla la implementación de un sistema virtual 3D en el área de automatización, convirtiéndose en una solución de capacitación práctica, en temas relacionados con la automatización, control e instrumentación, permitiendo la gestión de una planta para control de nivel simulado en un entorno virtual desarrollado en una plataforma 3D, reduciendo el riesgo de espacio físico inadecuado y costo en su implementación; el diseño del entorno virtual se desarrollará mediante el software motor gráfico Unity esto permitirá visualizar el comportamiento de una planta de control de nivel de un sistema real utilizado para el control de entrada y salida de datos en un sistema de control, gracias a la implementación de una tarjeta DAQ permitiendo el acondicionamiento de señales de entrada y salida digitales-analogicas, regulado por un algoritmo de control PID, los elementos de observación final serán dos válvulas de control, un tanque de llenado, teniendo un comportamiento real gracias a la implementación de la ecuación de Bernoulli, logrando así un sistema hardware in the loop. El sistema está diseñado para ser compatible con diferentes tipos de controladores. La propuesta pretende ser una herramienta virtual útil.

**Palabras claves:** Habilidades, proyecto, sistema virtual, Unity, control, controladores.



**UNIVERSIDAD TÉCNICA DE COTOPAXI**  
**FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS**  
**TITLE: DEVELOPMENT OF A 3D VIRTUAL ENVIRONMENT FOR WATER  
LEVEL CONTROL**

**Authors:**

**Intriago Pazmiño Jackson Fernando**

**Toapanta Sifaz Jonathan Francisco**

**ABSTRACT**

The limited practical skills to perform level control processes in the industry creates limitations of practical knowledge, currently the integration of virtual reality technologies in process control training systems has marked a successful path within industrial automation, thus the project contemplates the implementation of a 3D virtual system in the area of automation, becoming a practical training solution, on issues related to automation, control and instrumentation, allowing the management of a plant for level control simulated in a virtual environment developed in a 3D platform, reducing the risk of inadequate physical space and cost in its implementation; the design of the virtual environment will be developed using the graphic engine software Unity this will allow to visualize the behavior of a plant level control of a real system used by students as training in the control of input and output data in a control system, thanks to the implementation of a DAQ card allowing the conditioning of input and output signals, regulated by a PID control algorithm, the final observation elements will be two control valves, a filling tank, having a real behavior thanks to the implementation of the Bernoulli equation, thus achieving a hardware system in the loop. The system is designed to be compatible with different types of controllers. The proposal is intended to be a useful virtual tool.

**Keywords:** Skills, project, virtual system, Unity, control, controllers.

**INDICE**

DECLARACIÓN DE AUTORÍA .....	i
AVAL DEL TUTOR DE TITULACIÓN .....	ii
APROBACIÓN DEL TRIBUNAL DE TITULACIÓN.....	iii
AVAL DE IMPLEMENTACIÓN.....	iv
AGRADECIMIENTO .....	v
AGRADECIMIENTO .....	vi
DEDICATORIA.....	vii
DEDICATORIA.....	viii
RESUMEN .....	ix
ABSTRACT .....	x
1. INFORMACIÓN BÁSICA.....	1
2. DISEÑO INVESTIGATIVO DE LA PROPUESTA TECNOLÓGICA .....	2
2.1. Título de la propuesta Tecnológica.....	2
2.2. Tipo de alcance .....	2
2.3. Sinopsis de la propuesta tecnológica .....	2
2.4. Objeto de estudio y campo de acción .....	2
2.4.1. Objeto de estudio .....	2
2.4.2. Campo de acción .....	2
2.5. Situación problemática y problema .....	3
2.5.1. Situación problemática .....	3
2.5.2. Diagrama de Ishikawua .....	3
2.5.3. Problema.....	4
2.5.4. Hipótesis o formulación de pregunta científica.....	4
2.5.5. Categorización de las variables de investigación (variables independiente y dependiente). .....	4
2.6. Objetivos .....	4



2.6.1.	Objetivo general .....	4
2.6.2.	Objetivos específicos.....	4
2.7.	Descripción de las actividades y tareas propuestas con los objetivos establecidos .....	5
3.	MARCO TEÓRICO .....	7
3.1.	Análisis de los trabajos precedentes .....	7
3.2.	Realidad virtual.....	8
3.3.	Realidad virtual en la educación.....	9
3.4.	Tipos de realidad virtual .....	9
3.4.1.	Escritorio .....	9
3.4.2.	Proyectado .....	9
3.4.3.	Semi-inmersivo.....	10
3.4.4.	Inmersivo .....	10
3.5.	Hardware in the loop.....	10
3.6.	Sistema CAD: .....	10
3.7.	Técnicas de modelado.....	11
3.8.	Texturizado .....	12
3.9.	Mapeado UV .....	12
3.10.	Motor gráfico Unity3D.....	13
3.11.	Automatización. ....	14
3.12.	Pirámide de automatización .....	14
3.13.	Control automático de procesos .....	15
3.14.	Microcontroladores .....	15
3.15.	Sistema de señales entrada y salida.....	15
3.17.	Tarjeta de Adquisición de Datos (DAQ).....	17
3.18.	Sistemas PLC .....	17
3.18.1.	PLC S7 300.....	18
3.19.	Sistemas de control.....	18

3.19.1.	Sistema de control .....	18
3.19.2.	Clasificación de los sistemas de control .....	19
3.19.3.	Tipos de señal .....	19
3.19.4.	Tipo de control.....	19
3.19.5.	Estrategias de control.....	20
3.19.6.	Control en bucle cerrado.....	20
3.20.	Tipos de compensadores o controladores clásicos .....	21
3.21.	Controlador PID .....	21
3.22.	Métodos de sintonía de controladores PID .....	22
3.22.1.	Métodos de sintonía para técnicas de primer orden más lapso muerto (FODT) .....	22
4.	METODOLOGÍA .....	24
4.1.	Variables .....	25
4.2.	Implementación del entorno virtual .....	26
4.3.	Diseño y selección de componente .....	26
4.4.	Componentes que interviene en el sistema de control .....	26
4.5.	Diseño de la planta modelado en 3D .....	27
4.6.	Desarrollo de la aplicación.....	29
4.7.	Diseño del ambiente en el software Unity 3D .....	30
4.8.	Montaje de la escena en Unity3D .....	30
4.9.	Selección de texturas.....	31
4.10.	Implementación de puntos de colisión .....	31
4.11.	Implementación de textos.....	32
4.12.	Introducción de sonido de la bomba y agua .....	32
4.13.	Programación del movimiento de la capsula de movimiento. ....	33
4.14.	Desarrollo del avatar. ....	33
4.15.	Desarrollo del menú de entrada.....	34
4.16.	Comunicación entre Unity3D y DAQ.....	34

4.17.	Programación sistema de salida y entradas de datos al software Unity .....	35
4.18.	Simulación del comportamiento de la planta. ....	35
4.19.	Visualización de las formas de ondas de sintonización del lazo del control del sistema industrial de nivel.....	37
4.19.1.	Tarjeta de adquisición de datos .....	37
4.20.	Diagrama eléctrico de la DAQ .....	37
4.21.	Diagrama de un control PID.....	39
4.22.	Programación PLC Simatic S7-300 .....	40
4.23.	Identificación del sistema y sintonización del PID .....	40
4.23.1.	Método de Sintonía Lambda.....	40
4.23.2.	Sintonización por método Lambda.....	41
5.1.	: Configuración del teclado.....	43
5.2.	Diagrama y diseño de la DAQ .....	43
5.3.	Presentación menú de inicio del programa .....	44
5.4.	Interacción del avatar con la planta de control .....	44
5.5.	Visualización de la pantalla HMI .....	45
5.6.	Curvas de respuestas del proceso de nivel.....	45
5.7.	Prueba de control externo ON/OFF. ....	45
5.8.	Prueba de control con PID .....	46
5.9.	Método de obtención de nuestro modelo matemático. ....	47
5.10.	Formas de ondas sintonización de PID. ....	48
5.11.	Forma de onda PWM rectificadas a analógica .....	49
5.12.	Verificación de funcionamiento del sistema mediante la medición de voltaje. ....	50
5.13.	Visualización de variación del chorro de agua llenado de tanque. ....	53
5.14.	Módulo para sistema de control .....	54
6.	PRESUPUESTO Y ANÁLISIS DE IMPACTOS.....	55
6.1.	Análisis de Costos.....	55

6.1.1.	Factibilidad económica.....	55
6.1.2.	Impacto Práctico .....	56
6.1.3.	Impacto tecnológico .....	56
7.	CONCLUSIONES Y RECOMENDACIONES.....	57
7.1.	Conclusiones.....	57
7.2.	Recomendaciones .....	57
8.	REFERENCIAS .....	58

**INDICE DE FIGURAS.**

Fig.2.1. Diagrama de Ishikawa ..... 3

Figura 3.1. Pirámide de automatización. .... 14

Figura 3.4. Partes de un sistema DAQ..... 17

Figura 3.5. Diagrama de bloques de un PLC..... 18

Figura 3.6. Sistema de control ..... 18

Figura 3.7. Sistema de control ..... 19

Figura 3.8 Bucle abierto ..... 19

Figura 3.9. Sistema de control ..... 20

Figura 3.10. Sistema de control ..... 20

Figura 3.11. Sistema de control ..... 22

Figura 3.12 Estructura de lazo simple ..... 22

Figura 4.1. Diagrama de procesos. .... 24

Figura.4.2. Necesidades del sistema controlado..... 26

Figura 4.3. Diseño estructural del balcón..... 27

Figura 4.4. Diseño estructural de la cubierta..... 28

Figura 4.5. Diseño de la bomba de agua..... 28

Figura 4.6. Diseño final de la planta..... 29

Figura 4.7. Diagrama de flujo..... 29

Figura 4.8. Importación del diseño 3D a Unity. .... 30

Figura 4.9.Ventana de trabajo Unity . .... 30

Figura 4.10. Textura de materiales e iluminación. .... 31

Figura 4.11. Puntos colaiders. .... 31

Figura 4.12. Textos y labels..... 32

Figura 4.13. audio del agua ..... 32

Figura 4.15. Programación player primera persona. .... 33

Figura 4.16. Programación del diseño del avatar. .... 33

Figura 4.17. Diseño del menú de entrada. ....	34
Figura 4.18. Comunicación Serial. ....	34
Figura 4.19. Comunicación Serial. ....	35
Figura 4.20. Comunicación Serial. ....	37
Figura 4.21. Esquema diseño entradas y salidas placa DAQ. ....	38
Figura 4.22. Diagrama PID.....	39
Figura 4.22. Diagrama PID.....	39
Figura 4.23. Programación TIA PORTAL.....	40
Figura 5.1. Módulo para el sistema de control. ....	43
Figura 5.2. Menú de entrada.....	44
Figura 5.3. Interacción del avatar con el entorno. ....	44
Figura 5.4. Visualización de la pantalla HMI.....	45
Figura 5.5. Formas de ondas en unity.....	45
Figura 5.6. Visualización del control ON/OFF .....	46
Figura 5.7. Control PID .....	46
Figura 5.7. Acondicionamiento de señales eléctricas.....	46
Figura 5.8. Modelo matemático del sistema de control de nivel.....	48
Figura 5.9. PID Tuner sintonía PID.....	48
Figura 5.10. Visualización de las ondas de sintonía de PID primera prueba .....	49
Figura 5.11. Visualización de las ondas de sintonía de PID final. ....	49
Figura 5.12. Visualización del filtrado de onda PWM a analógico.....	50
Figura 5.13. Tanque de nivel de agua vacío. ....	50
Figura 5.14. Voltaje de medición tanque vacío. ....	51
Figura 5.15. Tanque de nivel medida 100 cm. ....	51
Figura 5.16. Voltaje de medición a 100 cm de capacidad del tanque. ....	52
Figura 5.17. Tanque de nivel medida 200 cm. ....	52
Figura 5.18. Medida de voltaje referencia máxima del tanque. ....	53

Figura 5.19. Volumen del chorro de llenado de tanque.....	53
Figura 5.20. Volumen de chorro de llenado del tanque.....	54
Figura 5.21. Módulo para el sistema de control. ....	54

**INDICE DE TABLAS**

Tabla 2.1. Sistema de tareas en base a los objetivos .....	5
Tabla.2: Partes del sistema de automatización .....	27
Tabla 3. Constantes obtenidas para el proceso control nivel.....	42
Tabla 4. Datos históricos de nuestro sistema virtual. ....	47



## **1. INFORMACIÓN BÁSICA**

### **PROPUESTO POR:**

Intriago Pazmiño Jackson Fernando

Toapanta Sifaz Jonathan Francisco

### **TEMA APROBADO:**

Desarrollo de un entorno virtual 3D para el control de nivel de agua.

### **CARRERA:**

Ingeniería Electromecánica.

### **DIRECTOR DE LA PROPUESTA TECNOLÓGICA:**

Ing. Luigi Orlando Freire Martínez. M. Sc.

### **EQUIPO DE TRABAJO:**

- Intriago Pazmiño Jackson Fernando
- Toapanta Sifaz Jonathan Francisco
- Ing. Luigi Orlando Freire Martínez. M. Sc. (Asesor Técnico)

### **LUGAR DE EJECUCIÓN:**

Región Sierra, Provincia Cotopaxi, Cantón Latacunga, Parroquia San Felipe.

### **TIEMPO DE DURACIÓN DE LA PROPUESTA:**

Período académico Abril 2021 – Agosto 2021.

### **FECHA DE ENTREGA:**

Agosto 2021

### **LÍNEA(S) Y SUBLINEAS DE INVESTIGACIÓN A LAS QUE SE ASOCIA LA**

### **PROPUESTA TECNOLÓGICA:**

#### **Línea de investigación de la Universidad Técnica de Cotopaxi:**

Procesos industriales

#### **Sub línea de investigación de la Carrera:**

Automatización, control y protecciones de sistemas electromecánicos.

### **TIPO DE PROPUESTA TECNOLÓGICA:**

Tecnología, innovación.

## **2. DISEÑO INVESTIGATIVO DE LA PROPUESTA TECNOLÓGICA**

### **2.1. Título de la propuesta Tecnológica**

Desarrollo de un entorno virtual 3D para el control de nivel de agua.

### **2.2. Tipo de alcance**

Se ha determinado que el alcance de la propuesta tecnológica planteada es interdisciplinar esto se debe a que el proyecto planteado propone realizar un entorno de aprendizaje donde se involucra la realidad virtual, control de procesos electromecánicos de manera donde la interacción del estudiante sea positiva en el aprendizaje. Esto se logrará mediante un diagrama PID permitirá el control y el monitoreo de señales digitales y analógicas a través de un periférico emulando un proceso industrial real posibilitando a los estudiantes y usuarios un acceso de manera interactiva con el proceso virtual.

### **2.1. Área del conocimiento**

07 Ingeniería, Industria y Construcción / 071 Ingeniería y profesiones afines / 07114 Electrónica y automatización.

### **2.3. Sinopsis de la propuesta tecnológica**

Se pretende llevar un sistema didáctico para el aprendizaje del control automático en el caso control de nivel para lo cual se implementará un entorno virtual que simule el comportamiento de una planta de control de nivel. Con este sistema el aprendizaje de esta técnica de control resulta efectivo y perdurable en los estudiantes por el alto impacto que genera en ellos, además la retención de información se incrementará significativamente cuando los estudiantes están expuestos a más de un medio ampliando su visión de aprendizaje y motivándolos a buscar más opciones de auto educación.

### **2.4. Objeto de estudio y campo de acción**

#### **2.4.1. Objeto de estudio**

Tanque de Nivel en Realidad Virtual.

#### **2.4.2. Campo de acción**

330000 Ciencias Tecnológicas / 3306 Ingeniería y Tecnología Eléctricas / 330602 Aplicaciones Eléctricas.

## 2.5.Situación problemática y problema

### 2.5.1. Situación problemática

La Realidad Virtual cumple un papel importante en la industria, permite el entrenamiento en procesos que se ejecutan en plantas industriales. Ante la limitación de recursos módulos prácticos para las aplicaciones industriales del mundo real, además que el avance vertiginoso de estas tecnologías de programación y control. En la actualidad las plataformas de realidad virtual acaparan la atención de la automatización industrial como herramientas complementarias para el entrenamiento en supervisión, control y mantenimiento de los procesos industriales.

En una fábrica industrial es necesario tener conocimientos en sistemas de automatización y control. Desde este punto de vista, la profesión de la ingeniería electromecánica ha formado a futuros profesionales en diferentes campos. Por ello, es necesario integrar puestos de trabajo con nuevas tecnologías y de bajo costo. Reducir el espacio es el desarrollo de la práctica.

En la actualidad, en el laboratorio de sistemas de automatización y control, representa un costo alto en módulos de aprendizaje, lo que no permite adquisición a gran escala por la falta de espacio físico en los laboratorios para poder realizar prácticas realizadas en el laboratorio, generalmente se realiza en grupos en la mayoría de los casos esto no permite un aprendizaje personalizado.

### 2.5.2. Diagrama de Ishikawa

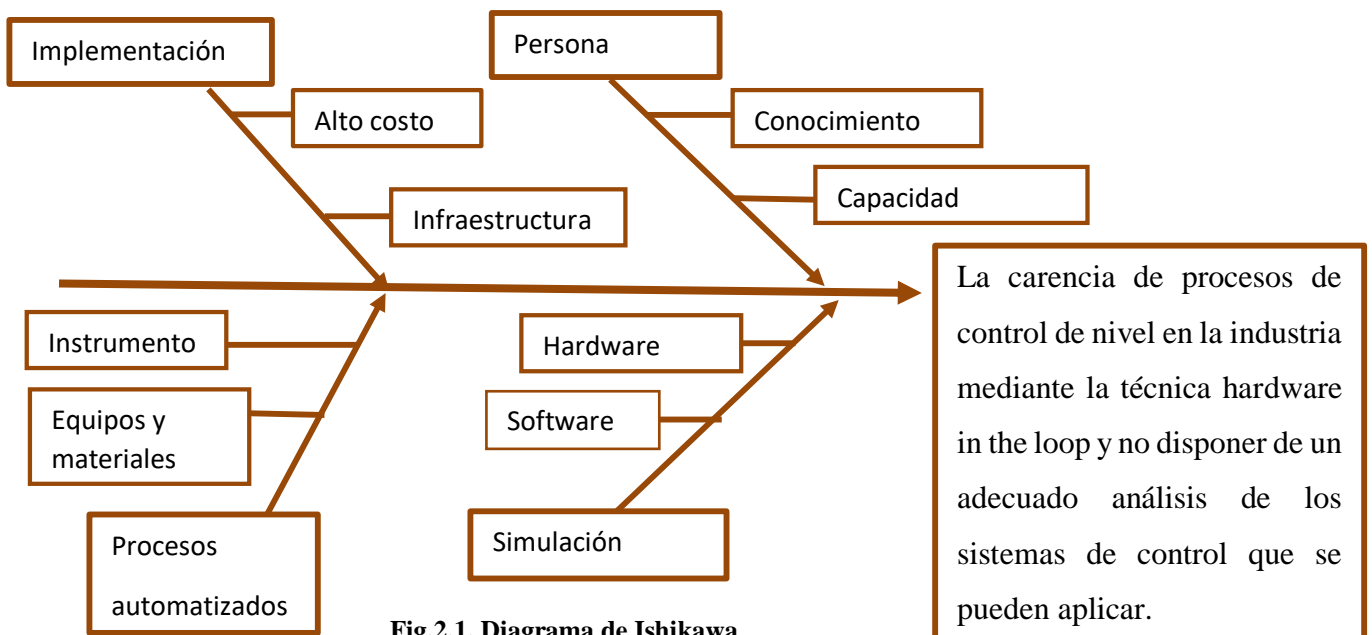


Fig.2.1. Diagrama de Ishikawa

### **2.5.3. Problema**

La carencia de procesos de control de nivel en la industria mediante la técnica hardware in the loop y no disponer de un adecuado análisis de los sistemas de control que se pueden aplicar.

### **2.5.4. Hipótesis o formulación de pregunta científica**

El desarrollo de un proceso de nivel 3D mediante la técnica hardware in the loop (HIL) permitirá analizar los sistemas de control que se pueden aplicar en procesos de control de nivel en la industria.

Categorización de las variables de investigación (variables independiente y dependiente).

En consecuencia, de la hipótesis planteada se identifican dos variables:

- **Variable Independiente:** Control de nivel.
- **Variable Dependiente:** Módulo DAQ planta de nivel.

## **2.6. Objetivos**

### **2.6.1. Objetivo general**

Desarrollar un proceso de nivel 3D mediante la técnica hardware in the loop (HIL), orientando al análisis de los sistemas de control de nivel en la industria.

### **2.6.2. Objetivos específicos**

- Realizar una búsqueda bibliográfica sobre entornos virtuales 3D para la implementación de sistemas de control industrial.
- Elaborar entorno virtual 3D enfocado al caso de estudio de nivel mediante el motor gráfico Unity 3D para la simulación del proceso del nivel y control en tiempo real.
- Diseñar una interfaz mediante DAQ para el manejo de señales eléctricas físicas, entradas y salidas.
- Realizar pruebas de funcionamiento para validar el correcto funcionamiento del sistema de entrenamiento hardware in the loop.

**2.7.Descripción de las actividades y tareas propuestas con los objetivos establecidos**

**Tabla 2.1. Sistema de tareas en base a los objetivos**

Objetivo	Actividades	Resultado de la actividad	Descripción de la actividad
Realizar una búsqueda bibliográfica sobre entornos virtuales 3D para la implementación de sistemas de control industrial.	<ul style="list-style-type: none"> <li>- Detalle de la información acerca de los ambientes virtuales y sus técnicas de aplicación al control de nivel.</li> <li>- Determinación de la versatilidad que proporciona la realidad virtual en la variable control de nivel.</li> <li>- Análisis y selección del software en busca del más idóneo y amigable para desarrollar un ambiente virtual.</li> </ul>	<p>Conocimiento de diferentes programas para el uso y simulación de realidad virtual</p> <p>Conocimiento de técnicas aplicables para el control de nivel</p> <p>Utilización de unity3D para realizar el ambiente virtual</p>	<p>Desarrollo del marco teórico.</p> <p>Esquema en anexos.</p> <p>Descargar el software a utilizar.</p>
Elaborar un entorno virtual 3D enfocado al caso de estudio de nivel mediante el motor gráfico Unity 3D para la simulación del proceso del nivel y control en tiempo real.	<ul style="list-style-type: none"> <li>- Análisis del entorno virtual en base a la creación 3D de una planta de control de nivel.</li> <li>- Análisis de la compatibilidad de Unity 3D con las diferentes tarjetas de adquisición de datos para el trabajo.</li> <li>-</li> </ul>	<p>Exportación del diseño CAD (fbx) de la estación de bombeo.</p> <p>Comunicación serial entre el entorno virtual 3D de Unity y arduino que funcionara como tarjeta de adquisición de datos.</p>	<p>Se importa el diseño CAD de la planta de control de nivel.</p> <p>Entorno simulado del proceso y conexión serial de la DAQ.</p>
Diseñar una interfaz mediante DAQ para el manejo de señales	<ul style="list-style-type: none"> <li>- Análisis de las variables físicas que se necesitan en el sistema para la interacción de los estudiantes.</li> </ul>	<p>Conocimiento de variables de entrada y salida (digitales y analógicas) así</p>	<p>Interacción del estudiante en un entorno virtual.</p>

<p>eléctricas físicas, entrada y salidas.</p>	<ul style="list-style-type: none"> <li>-Análisis compatibilidad del software con la tarjeta de adquisición de datos, interconexión con el control PID para realizar pruebas del diseño.</li> <li>- Detalle del control PID con todas las variables definidas.</li> </ul>	<p>como también señales de entrada de pulsadores</p> <p>Diseño del circuito para la tarjeta de adquisición de datos (arduino) como también componentes de transformación de datos digitales-analógicos</p>	<p>Diseño del circuito en Proteus para la adquisición de Unity y el programa de control.</p> <p>Diseño del control PID en arduino para el control de nivel del tanque.</p>
<p>Realizar pruebas de funcionamiento para validar el correcto funcionamiento del sistema de entrenamiento hardware in the loop.</p>	<ul style="list-style-type: none"> <li>- Análisis de la comunicación entre el módulo, DAQ y la programación.</li> <li>- Análisis de funcionabilidad del software en tiempo real y distintos errores.</li> <li>- Análisis del sistema de entrenamiento hardware in the loop.</li> </ul>	<p>Funcionalidad del módulo de sistemas de control finalizado.</p> <p>Guías de laboratorio.</p>	<p>Realización de pruebas.</p>

### 3. MARCO TEÓRICO

#### 3.1. Análisis de los trabajos precedentes

En mayo del 2016 en Salamanca se presentó un “Piloto de realidad aumentada y navegación peatonal de la propiedad territorial: boceto, ejecución y evaluación formativa” en la Universidad de Salamanca que trata a partir una dimensión pedagógica, este desarrollo también ha sido acelerado siendo fuertemente restringido con la ejecución de nuevas metodologías, materiales de propagación, y el mundo de conocimiento. Con el primero Aprendizaje Movil – m-learning-, etapa marcada por la portabilidad y la distancia del proceso de enseñanza-aprendizaje, deviniendo en un nuevo estadio, el Aprendizaje Situado –*u-learning*-, en donde la información aparecer en cualquier momento y lugar que se requiere, confluyendo campos del conocimiento.[1].

Luego investigando que en agosto del 2018 en la ciudad de Latacunga se presentó un sistema de control en cascada de la variable nivel para el aprendizaje de control de procesos en la Universidad de las Fuerzas Armadas. Se realizó una técnica de control en cascada para el versátil nivel en la estación ND-0704 del Laboratorio de Redes Industriales y Control de Procesos de la ESPE – Extensión Latacunga, mismo que será utilizado para el aprendizaje de control de procesos a través de un entorno de realidad virtual. Dicho entorno ha sido desarrollado en base al diagrama P&ID de la estación de control y pretende emular el funcionamiento de una planta industrial. El sistema didáctico permite el monitoreo y regulación del nivel del tanque de la estación de entrenamiento desde el entorno virtual implementado, además de la simulación del control en cascada del mismo proceso, característica que permite el acercamiento de los estudiantes al control de procesos de una manera interactiva y realista. Se hace uso de módulos periféricos como las gafas de realidad virtual Oculus Rift y el controlador gestual Leap Motion que permiten la interacción e inmersión en el entorno para complementar la experiencia de aprendizaje[2].

Se tiene que, en marzo del 2018 fue presentado en Quito el Desarrollo de un entorno de realidad virtual 3D con capacidad de inmersión que emule el lanzamiento y recepción de un raspador de tubería (PIG), orientado al entrenamiento dentro de la industria petrolera, con el objetivo de reforzar la capacitación de los operadores de la industria petrolera. En el trabajo actual, se ha desarrollado una plataforma de realidad virtual 3D con capacidades inmersivas, que puede simular el entorno y el funcionamiento del lanzamiento y recibir trampas del raspador de tubería pipeline inspection gauges (PIG), con el objetivo de reforzar la capacitación de los operadores

de la industria petrolera. Para ello, en primer lugar, se revisó y recopiló la información existente de varias trampas de lanzamiento y recepción de diferentes empresas operadoras de oleoductos nacionales, y se definieron los parámetros básicos y específicos de un modelo de trampa típica de entretenimiento virtual[3].

Y en mayo del 2019 en la ciudad de Quito se mostró el “Progreso de una aplicación de realidad virtual semi inmersiva para terapia recreacional” en la Universidad Politécnica Nacional, el presente trabajo de titulación, presenta una solución de software para brindar terapia recreacional a personas con parálisis física, mediante una aplicación de realidad virtual semi-inmersiva, con el objetivo de apoyar a la función integral del Niño (FINE). El aplicativo fue desarrollado con enfoques ágiles, como o son SCRUM y la metodología experimental iPlus[4]. La realidad virtual (VR) es una herramienta poderosa para probar y evaluar nuevos productos e ideas en los procesos de fabricación, además se ha sumergido completamente al usuario en una simulación experimental en la creación rápida de prototipos, fabricación. [5]

El desarrollo de una plataforma integrada de software y hardware para el control de subestaciones eléctricas mediante un entorno virtual. En el desarrollo de este artículo se verifica que cada subestación eléctrica 3D está integrada al centro de supervisión para la adquisición de datos y control de una compañía de energía eléctrica real.[6]

Mediante el esquema de tele-operación el operador está inmerso en un entorno de realidad aumentada y permite al mismo ejecutar tareas destinadas con el modelado 3D en entornos remotos.[7]

La animación de robots en entornos de realidad virtual se basa en principios mecánicos y la unidad de control, mediante la transferencia del diseño en CAD de un robot a un entorno de simulación virtual permite realizar simulaciones de movimientos en un entorno establecido. [8]

### **3.2. Realidad virtual**

Es difícil dar una definición clara de realidad virtual, porque según RAE, estos son términos opuestos.

- ✓ Realidad: existencia real y efectiva de algo.
- ✓ Virtual. Tiene la virtud de producir efectos, aunque no producirá efectos ahora, y suele ser contrario a la realidad.

Sin embargo, se puede considerar como definiciones bastante acertadas a las siguientes:

“La realidad virtual es una simulación generada por computadora de un entorno 3D que permite a los usuarios ver e interactuar con el contenido de ese entorno.”.



“La tecnología de realidad virtual proporciona un entorno inmersivo que permite a los usuarios tener experiencias modificadas de la realidad”.

Por tanto, podemos concluir que la realidad virtual es una simulación del mundo real con las mejoras que hacen posibles las tecnologías disponibles, tanto en la creación del entorno 3D como en la manipulación del usuario. [9].

### **3.3. Realidad virtual en la educación**

Las universidades y los centros educativos han observado un aumento en el uso de las tecnologías de la información y la comunicación en los procesos educativos y de aprendizaje que utilizan tecnologías basadas en inteligencia artificial como los asistentes de enseñanza inteligentes, los sistemas de gestión del aprendizaje y los videojuegos. De esta manera es posible la edificación de métodos de monitorización sutil para analizar el progreso en el aprendizaje de los estudiantes, también se podrían plantear diferentes escenarios donde los alumnos dispondrán de interfaces cerebrales para interactuar directamente con una máquina y un software que podrían realizar las tareas de un tutor virtual con conexión directa a las áreas de aprendizaje cerebrales [10].

Como resultado, existe una creciente necesidad de herramientas educativas y de formación innovadoras que promuevan y respondan a los intereses y objetivos de la sociedad. Esto representa un enfoque constructivo para alentar a los estudiantes a tomar la iniciativa en su proceso de aprendizaje. El sistema hipermedia es un modelo obvio para este enfoque. En distinto sentido, según la teoría doctrinal, existe una combinación de aprendizaje individual o aprendizaje cooperativo con otros. [11].

### **3.4. Tipos de realidad virtual**

Según el artículo del investigador Mohamad A [12] titulada “*Introduction to virtual reality technology*” manifiesta que existe 4 tipos de realidad virtual:

#### **3.4.1. Escritorio**

Entorno virtual 3D gráficamente expuesto en un computador de escritorio monitor, ordenador.

#### **3.4.2. Proyectado**

Entorno 3D proyectado sobre una pantalla. Permite a un solo usuario demostrar conceptos a un grupo de la gente. A CAVE™, donde se utilizan varias pantallas para rodear al usuario de imágenes, es la forma más avanzada de realidad virtual proyectada que se utiliza en la actualidad.

**3.4.3. Semi-inmersivo**

Los simuladores de vuelo, barcos y vehículos más avanzados son semi-inmersivos. La cabina, el puente o el asiento del conductor es un modelo físico, mientras que la vista del mundo exterior está generada por computadora (generalmente proyectada).

**3.4.4. Inmersivo**

Entorno 3D visto a través de una pantalla montada en la cabeza (HMD). En un sistema completamente inmersivo, el usuario se siente parte del entorno (experimenta una sensación de "presencia"). El usuario no tiene contacto visual con el mundo físico [12].

**3.5. Hardware in the loop.**

En referencia a la metodología HIL y al diseño de controladores embebidos se encuentran diversos trabajos que permiten tener un panorama de los alcances a nivel industrial e investigativo, además de las diferentes herramientas de hardware y software que actualmente se tienen al alcance y facilitan procesos como la adquisición, el procesamiento y el análisis de datos y señales permitiendo en un futuro la posible manifestación de esta metodología no solo en el campo de la automatización y el control, si no en otras áreas de la ingeniería [13].

**3.6. Sistema CAD:**

En general, el diseño asistido por computadora (CAD) puede entenderse como una "aplicación de la informática al proceso de diseño". Especificar una definición significa un sistema CAD, que es un sistema informático que automatiza el proceso de diseño para un tipo particular de entidad, como un sistema CAD, una aplicación de solo imagen que afecta un aspecto particular del proceso de diseño. La ayuda informática se puede utilizar para la mayoría de las actividades de proceso y el dibujo es el lugar más utilizado. El éxito de utilizar un sistema CAD es reducir el tiempo dedicado al ciclo de búsqueda. Puede observar rápidamente cambios en el modelo y cambios en el proyecto, principalmente gracias al uso de un sistema gráfico interactivo.

El sistema CAD se basa en la representación informática del modelo. Esto permite la documentación del proyecto y dibujos automatizados detallados, lo que le permite realizar simulaciones en su modelo utilizando técnicas digitales en lugar de la creación de prototipos. Los ciclos de diseño que utilizan sistemas CAD solo se ven afectados al incluir pasos de simulación entre la generación del modelo y el boceto. Este simple cambio puede ahorrarle mucho tiempo en el proceso de diseño al predecir cuándo se descubrirán ciertos defectos de diseño.

Estas herramientas se pueden dividir básicamente en programas de dibujo bidimensionales (2D)

y / o tridimensionales (3D). Las herramientas de dibujo 2D se basan en entidades geométricas vectoriales como puntos, líneas, arcos y polígonos que se pueden manipular a través de una interfaz gráfica. Las herramientas 3D agregan superficies y sólidos [14].

### 3.7. Técnicas de modelado

El modelado 3D utiliza una variedad de métodos y técnicas, según la complejidad requerida para el modelo y la multitud de técnicas para aprender sobre herramientas de modelado de software certificadas. Esto creó una comunidad de personas que intentaban crear una pequeña cantidad de modelos. Más detalladamente, aquí hay algunas técnicas importantes que necesita conocer para desarrollar su proyecto actual:

- **Modelo Rotoscopia:** Esta es una técnica que comienza con una foto o un boceto en tres diseños diferentes, incluidas las imágenes frontal, lateral y superior de los planos ZX, ZY y ZX utilizados como modelos para crear modelos 3D. Los objetos se realizan mediante la creación de planos y extrusiones. [15].
- **Modelado Solidify:** Técnica utilizada principalmente para añadir profundidad a los planos que componen el modelo creado. Estos son estándares incorporados en el software 3D que se utilizan para evitar que el tamaño de los archivos sea demasiado grande.
  - **Modelado de sólidos:** (*CSG Constricted Solid Geometry*) Geometría sólida estructurada. Este modelo representa la masa de un objeto, incluido el centro de gravedad del objeto, los materiales, la densidad, la textura y más. Debido a la estrecha gama de polígonos, se utiliza para entrenar modelos complejos generados por computadoras y aplicaciones industriales y médicas con la ayuda de primitivas.
- **Modelado de Contorno:** (*B-Rep - Boundary Representation*) Representación de fronteras. Esta técnica está destinada a representar solo el contorno del modelo, la forma conceptual del casco. Son mucho más fáciles de crear y modificar. Lo interesante de estos modelos es que solo el exterior del modelo es importante y el interior no está modelado por lo que el modelo no muestra un tamaño excesivo. [16].
- **Modelado Low Poly:** Esta es una técnica de modelo de baja calidad, generada con una pequeña cantidad de polígonos para optimizar recursos y utilizada para crear una sensación voluminosa bien definida en videojuegos texturizados de alta calidad. Con este enfoque, todos los proyectos se dirigen a dispositivos de bajos ingresos. Los

avances en tecnología dan a los dispositivos un estilo estético con gráficos acelerados por hardware de la misma manera que se crean los modelos de baja resolución. [17].

- **Modelado High Poly:** Una técnica de modelado de alto nivel que se utiliza principalmente para modelos orgánicos que requieren una gran cantidad de polígonos, caras y vértices debido a problemas de expresión física. Esta técnica se utiliza para proyectos en el mundo del cine, cortometrajes, animación e incluso juegos de consola con altos recursos materiales. [17].
- **Modelado por caja:** También conocido como modelo de caja. Este es un modelo gráfico de alta calidad creado fuera de una caja utilizando herramientas de software de edición de malla con extrusión, subdivisión, solidificación, biselado y más. Es una técnica que utiliza modificadores para facilitar el modelado y, en última instancia, presenta modelos de conteos de polígonos bajos a altos. [18].

### 3.8. Texturizado

El objeto 3D que termina en la etapa de modelado pasa por la viñeta y se agrega a la viñeta para crear más definición y realismo. Esto mejorará la imagen del modelo. La textura le permite combinar una variedad de materiales como colores, metales, vidrio y madera. Da más detalles a tus objetos, crea mapas normales y mejora visualmente tu modelo. No siempre es posible crear texturas con software de modelado. Puede utilizar herramientas externas que le permitan crear y editar imágenes digitales, o puede extraer fácilmente de fotos de alta calidad que solo extraen texturas.

Actualmente, los anuncios de medios visuales muestran objetos compuestos que contienen imágenes en el modelo en lugar de los objetos reales. Debido a la textura de este modelo, es difícil distinguir entre objetos 3D y objetos. Hay un proceso para ejecutar. Utilice sombras, mapas y materiales UV. [19].

### 3.9. Mapeado UV

El mapeado UV es utilizado para añadir texturas a partir de imágenes a los modelos 3D. Es una técnica aplicada para los modelos de formas arbitrarias y complejas como modelos orgánicos. La ventaja de usar mapeado UV es que se ajusta perfectamente a los modelos, el procedimiento es el de abrir la malla del modelo realizando cortes, así la malla se puede visualizar en 2D, a esta nueva imagen se le conoce como mapa de la malla, la nueva imagen es exportada a cualquier software de edición de imágenes como *Gimp*, *Photoshop*, incluso *Paint*, con el objetivo de pintar la textura siguiendo el mapa UV, así la imagen después del proceso de colorear se ajusta al modelo 3D, todo este proceso de pintar la textura necesariamente se realiza

de manera manual porque podemos tener un control total de cada pixel de la imagen. El mapa UV se traduce a la forma de asignar la parte de una imagen a un polígono en el modelo. Existe una correspondencia entre cada vértice del modelo cada coordenada 2D de la imagen textura [20].

### 3.10. Motor gráfico Unity3D

Unity3D es un potente motor 3D multiplataforma y un entorno de desarrollo fácil de usar. Lo suficientemente fácil para el principiante y lo suficientemente potente para el experto; Unity3D debería interesar a cualquiera que quiera crear fácilmente juegos y aplicaciones 3D para dispositivos móviles, computadoras de escritorio, internet y consolas [21].

*Unity Technologies* es una compañía de desarrollo de videojuegos, el cual ofrece una plataforma completa para crear nuevas tendencias y generar nuevas ideas para juegos, también se puede crear aplicaciones hermosas y atractivas en modos 2D, 3D, Realidad Virtual (VR) y RA (AR).

*Unity* es un software que cuenta con un poderoso motor de física que ayudan a los juegos a dar gran realismo, cuenta con un motor gráfico y un editor con todas las prestaciones y características que facilitan la creación de los videojuegos. Permite exportar los proyectos finales a cualquier plataforma, medio o dispositivo. Permite el desarrollo de aplicaciones con conexión a internet o generar modos de juegos en línea con diferentes usuarios a nivel mundial, se puede conectar consolas, dispositivos móviles, sistemas de entretenimiento para el hogar, *smart TV*, *smart watch*.

El software *Unity* como se menciona, ayuda a generar varios videojuegos realistas, *Unity* ofrece todo lo que se necesita para el desarrollar contenido de calidad, te ayuda a mantener una presentación en los video juegos que realices con gran productividad permitiendo conectarse con el público a nivel mundial. Dentro del software de desarrollo de videojuegos tiene varias herramientas y recursos que ayudan realizar trabajos de una forma más cómoda y rápido, entre las herramientas podemos mencionar *Unity Asset Store*, *Unity Cloud Build*, *Unity Analytics*, *Unity Ads*, *Unity Everyplay* y *Unity Certification*. La empresa *Unity* ha sido creada para usuarios de distintos niveles, como usuarios anónimos que pueden hacer uso del software con recursos limitados hasta usuarios registrados, entre los cuales se encuentran grandes fabricantes importantes [22].

Herramienta que permite crear videojuegos para diversas plataformas (PC, videoconsolas, móviles, etc.) mediante un editor visual y programación vía *scripting*, y pudiendo conseguir resultados totalmente profesionales. Se le conoce como un motor de desarrollo o motor de

juegos. El término motor de videojuego, *game engine*, hace referencia a un software el cual tiene una serie de rutinas de programación que permiten el diseño, la creación y el funcionamiento de un entorno interactivo; es decir, de un videojuego. Dentro de las funcionalidades típicas que tiene un motor de videojuegos, son las siguientes:

- Motor gráfico para renderizar gráficos 2D y 3D
- Motor físico que permita simular las leyes de la física
- Animaciones
- Sonidos
- Inteligencia Artificial
- Programación o scripting

### 3.11. Automatización.

La automatización de los procesos productivos es uno de los aspectos que más ha evolucionado en la industria desde sus comienzos. La integración de tecnologías clásicas como la mecánica y la electricidad con otras más modernas (electrónica, informática, telecomunicaciones, etc.) está haciendo posible esta evolución [23].

### 3.12. Pirámide de automatización

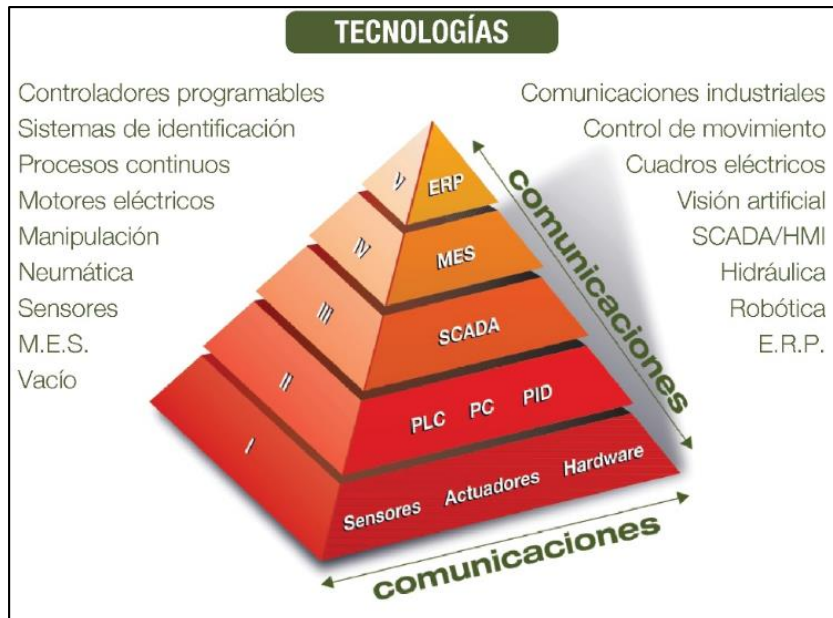


Figura 3.1. Pirámide de automatización.

Fuente: [23]

- El primer nivel o "nivel de campo" incluye los dispositivos físicos presentes en la industria, como los actuadores y sensores.
- El segundo nivel o "nivel de control" incluye los dispositivos controladores como ordenadores, PLCs, PIDs, etc.
- El "nivel de supervisión" (tercer nivel) corresponde a los sistemas de supervisión, control y adquisición de datos (SCADA).
- En un nivel superior o "nivel de planificación" se encuentran los sistemas de ejecución de la producción (MES).
- La cúspide de la pirámide ("nivel de gestión") la componen los sistemas de gestión integral de la empresa (ERP) [23].

### **3.13. Control automático de procesos**

El control automático de procesos es utilizado para mantener en un punto de consiga las variables que abarca un proceso estas variables pueden ser: temperatura, presión, nivel, etc. Los procesos en los cuales está involucrado el control automático son de naturaleza dinámica es decir siempre ocurren cambios en ellos y si no se realizan acciones pertinentes o correctivas en las variables importantes del proceso, los índices de producción y calidad no cumplirán con el diseño establecido [24].

### **3.14. Microcontroladores**

Un microcontrolador es un circuito integrado o "chip" (es decir, un dispositivo electrónico que integra en un solo encapsulado un gran número de componentes) que tiene la característica de ser programable. Es decir, que es capaz de ejecutar de forma autónoma una serie de instrucciones previamente definidas por nosotros. El microcontrolador sería el componente principal de la circuitería de procesamiento y control [25].

### **3.15. Sistema de señales entrada y salida**

El microcontrolador Arduino es una plataforma electrónica basada en hardware y software de código abierto, gratuita, multiplataforma y fácil de usar. Una placa de hardware gratuita que combina un microcontrolador reprogramable y un conjunto de pines medibles (conectados internamente a los pines de E / S del microcontrolador) que hacen que la conexión de sensores y actuadores sea muy simple y fácil.

El lenguaje de programación es gratuito. Un lenguaje de programación es un lenguaje artificial que puede representar comandos que se pueden ejecutar en una máquina (siguiendo reglas de sintaxis específicas). Este software generalmente requiere que se programe usted mismo utilizando lenguajes de programación estándar como: Python, C, Java, Php, entre otros [25].

Con este dispositivo en cada placa Arduino, el Arduino tiene un convertidor de USB a serie integrado que permite que Arduino se comunique directamente con la PC. [26].

### **3.16. Convertidores Digital – Analógico (DAC)**

Los convertidores DAC (*Digital-Analog Converter*), se encargan de convertir un código digital en un nivel de corriente o tensión analógica. Si el código contiene  $n$  bits entonces significa que existen  $2^n$  combinaciones binarias posibles; cada combinación corresponde a un nivel de tensión diferente a la salida. [27]

#### **3.16.1. MCP4725**

El MCP4725 es un DAC de 12 bits, es decir, un terminal que permite obtener una señal de tensión analógica desde un procesador digital como Arduino.

Cuando vimos las salidas analógicamente vimos que, en realidad, estas proporcionan una señal PWM entre GND y  $V_{cc}$ . Aunque apliquemos un filtro RC, no conseguiremos una auténtica señal analógica de tensión. En su lugar, un DAC ideal generaría una señal es la que la salida generada adopte cualquier valor intermedio entre GND y  $V_{cc}$ , generando una auténtica analógica.

Sin embargo, los DAC reales no son capaces de adoptar cualquier valor de tensión, si no que la salida está constituida en  $N$  etapas, por lo que proporcionan una señal analógica rectificadas de  $2^N$  niveles. El MCP4725 se controla por I2C por lo que es sencillo realizar su lectura. Dispone de dos posibles direcciones, que se elige mediante la conexión del pin ADDR.

También incorpora una memoria EEPROM que permite que mantenga el nivel de tensión incluso después de un corte de alimentación. La tensión de alimentación del MCP4725 es de 2.7V a 5.5. La tensión máxima que puede proporcionar es  $V_{cc}$ . Alimentado a una tensión de 5V sus 4096 niveles (12 bits) suponen una precisión de, aproximadamente, 1mV.



La corriente máxima que puede proporcionar es de 25mA. Es una intensidad reducida, pero, por otro lado, similar a la de una salida de Arduino. El tiempo de cambio de salida típico es de 6  $\mu$ s.

El MCP4725 puede ser útil en cualquier tipo de proyecto que requiera la generación de una auténtica señal analógica en lugar de una PWM como, por ejemplo, para alimentar dispositivos especialmente sensibles, calibración de sensores, sistemas de adquisición de datos, o generadores de funciones como ondas triangulares u ondas.

### 3.17. Tarjeta de Adquisición de Datos (DAQ)

“La adquisición de datos (Data Acquisition) es el proceso de medir con una PC un fenómeno eléctrico o físico como voltaje, corriente, temperatura, presión o sonido. Un sistema de este tipo debe constar de sensores, hardware de medición de adquisición de datos y una computadora con software programable. Comparados con los sistemas de medidas tradicionales, los sistemas DAQ basados en PC aprovechan la potencia del procesamiento, la productividad, la visualización y las habilidades de conectividad de las computadoras estándares en la industria proporcionando una solución de medidas más potente, flexible y rentable” [28].



**Figura 3.4. Partes de un sistema DAQ**

**Fuente:** [28]

### 3.18. Sistemas PLC

PLC: dispositivo electrónico que puede ser programado por personal no informático y destinado a realizar funciones de automatización lógica, combinaciones y secuencial en tiempo real en un entorno industrial.[29]

#### Diagramas de bloques



**Figura 3.5. Diagrama de bloques de un PLC**

**Fuente:** [29]

### 3.18.1. PLC S7 300

Este PLC, pertenece a una amplia gama de módulos empleados para tareas de automatización.

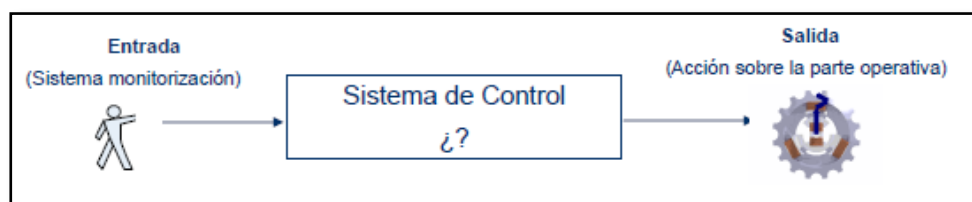
El S7-300 brinda la posibilidad de realizar estructuras descentralizadas e interconexiones por red.

Cuenta con módulos para ampliarse sin ningún inconveniente, en el caso de que aumenten las tareas, según las necesidades futuras. [29]

## 3.19. Sistemas de control

### 3.19.1. Sistema de control

- **Sistema:** Conjunto de componentes que interacción entre sí. Relaciona las entradas al sistema y las salidas del mismo.
- **Sistema de Control:** la salida es controlada para adoptar unos valores o cambios de forma preestablecida por la entrada.



**Figura 3.6. Sistema de control**

**Fuente:** [30]

Ejemplos:

- Control de temperatura (Entrada= $T^a$ , Salida=Voltaje)
- Control de una trayectoria en un robot (Entrada=posición, Salida=Voltaje).

### 3.19.2. Clasificación de los sistemas de control

- ✓ Tipo de señal
- ✓ Tipo de control
- ✓ Numero de entradas/salidas

### 3.19.3. Tipos de señal

#### Sistemas de control analógicos

Basados en señales de tipo continuo, con un margen de variación determinado.

Las señales típicas tratadas son tensiones o intensidades proporcionales a magnitudes físicas.

#### Sistemas de control digitales

Basados en señales binarias (todo/nada) que representan dos estados (bits  $\square$  0/1).

#### Sistemas de control híbridos

Sistemas que trabajan a la vez con señales digitales/analógicas.

Normalmente la unidad de control central está formada por un  $\mu$ P (digital), por lo que necesitan módulos de A/D, D/A.



Figura 3.7. Sistema de control

Fuente: [30]

### 3.19.4. Tipo de control

Sistema de control en bucle abierto:

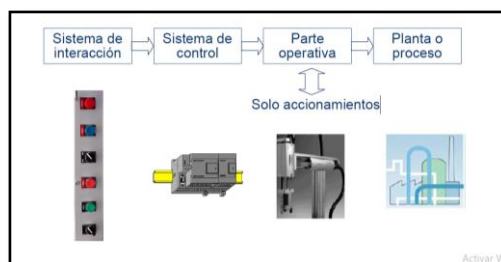


Figura 3.8 Bucle abierto

Fuente: [30]

Sistemas de control en bucle cerrado:

- Permiten corregir errores
- Permiten actuar frente a perturbaciones
- Más costos
- Ejemplo: mecanismo de piloto automático

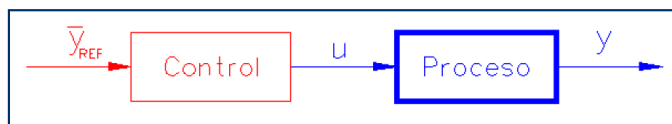
### 3.19.5. Estrategias de control

**Control en bucle abierto:**

- **Control todo-nada**

Encendido o apagado de un proceso o planta.

Calefacción o aire acondicionado tradicional de un vehículo, abrir o cerrar un grifo. Secuencias o acciones conmutadas por tiempo o por eventos.



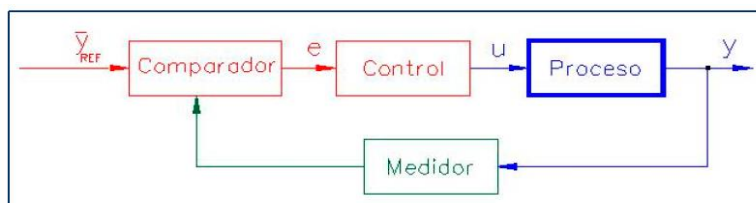
**Figura 3.9. Sistema de control**

Fuente: [30]

### 3.19.6. Control en bucle cerrado

Se utiliza un nuevo elemento en el control, el compensador o controlador.

Su función es calcular la acción de control a aplicar para corregir el error con el fin de que la salida tenga el valor esperado y marcado en la entrada, siempre asegurado la estabilidad del sistema.



**Figura 3.10. Sistema de control**

Fuente: [30]

Regulación automática (el compensador se denomina regulador)

Encendido/apagado de un proceso según el error con respecto a una consigna fija.

Servomecanismo o tracking

El objetivo del control es el seguimiento de una trayectoria predefinida.

### **3.20. Tipos de compensadores o controladores clásicos**

#### **Control proporcional (P)**

La señal de salida del compensador es proporcional a la señal de entrada al mismo (señal de error).

Hace el efecto de amplificador con una ganancia constante.

#### **Control integral (I)**

La señal de salida es proporcional a la integral de la señal error.

La salida es proporcional a la acumulación de los efectos de los errores pasados.

Hace desaparecer el error en régimen permanente.

Aumenta la estabilidad relativa.

#### **Control derivativo (D)**

La señal de salida es proporcional a la razón de cambio con el tiempo del error (derivativo).

Es insensible a errores constantes o que varían lentamente. No se usa solo.

La respuesta resulta más rápida.

### **3.21. Controlador PID**

El controlador PID es el algoritmo de control más común y se basa en una implementación de la idea de retroalimentación simple. Esto reduce la influencia de las turbulencias y, gracias a estas ventajas en el control del proceso, permite que el proceso siga los ajustes correctamente. El control PID se utiliza en más del 95% de los circuitos de control.

Los controladores PID han sobrevivido a muchos cambios de tecnología que van desde la neumática a los microprocesadores pasando por los tubos de vacío, los transistores, y los circuitos integrados. El microprocesador ha tenido una influencia crucial sobre el controlador PID. Prácticamente todos los controladores PID fabricados hoy en día se basan en microprocesadores.

Esto ha creado oportunidades para proporcionar características adicionales tales como sintonía automática, planificación de ganancia, adaptaciones continuas y diagnóstico, además se implementa de muchas formas diferentes, como un controlador único o como parte de un

paquete DDC (Control Digital Directo) o un sistema de control de procesos distribuido jerárquico. [31]

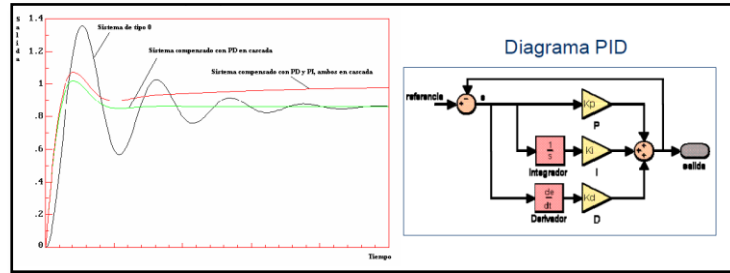


Figura 3.11. Sistema de control

Fuente: [31]

### 3.22. Métodos de sintonía de controladores PID

La figura 3.12 muestra la estructura de un lazo de control PID de una planta, si se puede obtener un modelo matemático de la planta, entonces es posible aplicar diversos métodos para determinar los parámetros de sintonía del controlador que cumplirán las especificaciones del sistema. Sin embargo, si el modelo matemático de la planta no puede obtenerse fácilmente, entonces no es posible un enfoque analítico para el diseño de un controlador PID. Para solucionar este problema se debe recurrir a enfoques experimentales para ajustar los controladores PID [32].

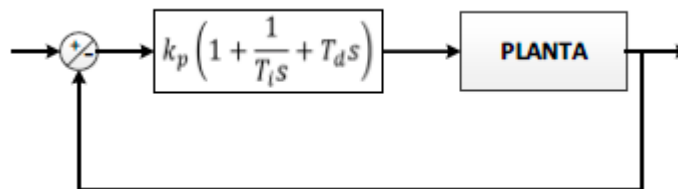


Figura 3.12 Estructura de lazo simple

#### 3.22.1. Métodos de sintonía para técnicas de primer orden más lapso muerto (FODT)

##### a) Método de sintonía Lambda

La sintonía de lambda es un caso especial de asignación de polos que se utiliza con frecuencia en la industria de procesos. El proceso se modela por el modelo FODT mostrado en la ecuación 1 [33].

$$P(s) = \frac{k_p}{1+sT} e^{-sL} \quad [1]$$

Para la obtención del diseño PID, se usa la forma interactuante del controlador mostrado en la ecuación 2.

$$C'(s) = K' \frac{(1+sT'_i)(1+sT'_d)}{sT'_i} \quad [2]$$

Se aproxima el retardo de tiempo, lo que da la función de transferencia del proceso, ecuación 3

$$P(s) = \frac{k_p}{1+sT} e^{-sl} \approx \frac{k_p(1-\frac{sL}{2})}{(1+sT)(1-\frac{sL}{2})} \quad [3]$$

El tiempo integral se escoge como  $sT'_i = T$  y el tiempo derivativo como  $sT'_d = L/2$ . Los ceros del controlador entonces se cancelarán los polos del proceso, y la función de transferencia del lazo se hace, ecuación 4.

$$G_1(s) = P(s)C'(s) \approx \frac{k_p k' (1-\frac{sL}{2})}{sT} \quad [4]$$

La ecuación característica se muestra en la ecuación 5.

$$s \left( T - k_p k' \frac{L}{2} \right) + k_p k' = 0 \quad [5]$$

Requiriendo que el polo en lazo cerrado sea  $s = 1/T_{cl}$  encontramos lo mostrado en la ecuación 6.

$$k_p k' = \frac{T}{\frac{L}{2} + T_{cl}} \quad [6]$$

Que da las reglas de sintonía mostradas en las ecuaciones 7, 8 y 9.

$$K' = \frac{1}{k_p} \frac{T}{\frac{L}{2} + T_{cl}} \quad [7]$$

$$T'_i = T \quad [8]$$

$$T'_d = \frac{L}{2} \quad [9]$$

Los parámetros para el controlador PID no interactuante son los mostrados en las ecuaciones 10, 11 y 12.

$$K = \frac{1}{k_p} \frac{\frac{L}{2} T}{\frac{L}{2} + T_{cl}} \quad [10]$$

$$T'_i = T \frac{L}{2} \quad [11]$$

$$T'_d = \frac{TL}{L+2T} \quad [12]$$

#### 4. METODOLOGÍA

Para llevar a cabo este proyecto se utilizó una metodología iterativa e incremental, este procedimiento permite programar un proyecto en diferentes bloques de tiempo denominados.

En consecuencia, este tipo de metodología permitirá dividir el proyecto en diferentes etapas, las cuales serán validadas en el momento de cada una.

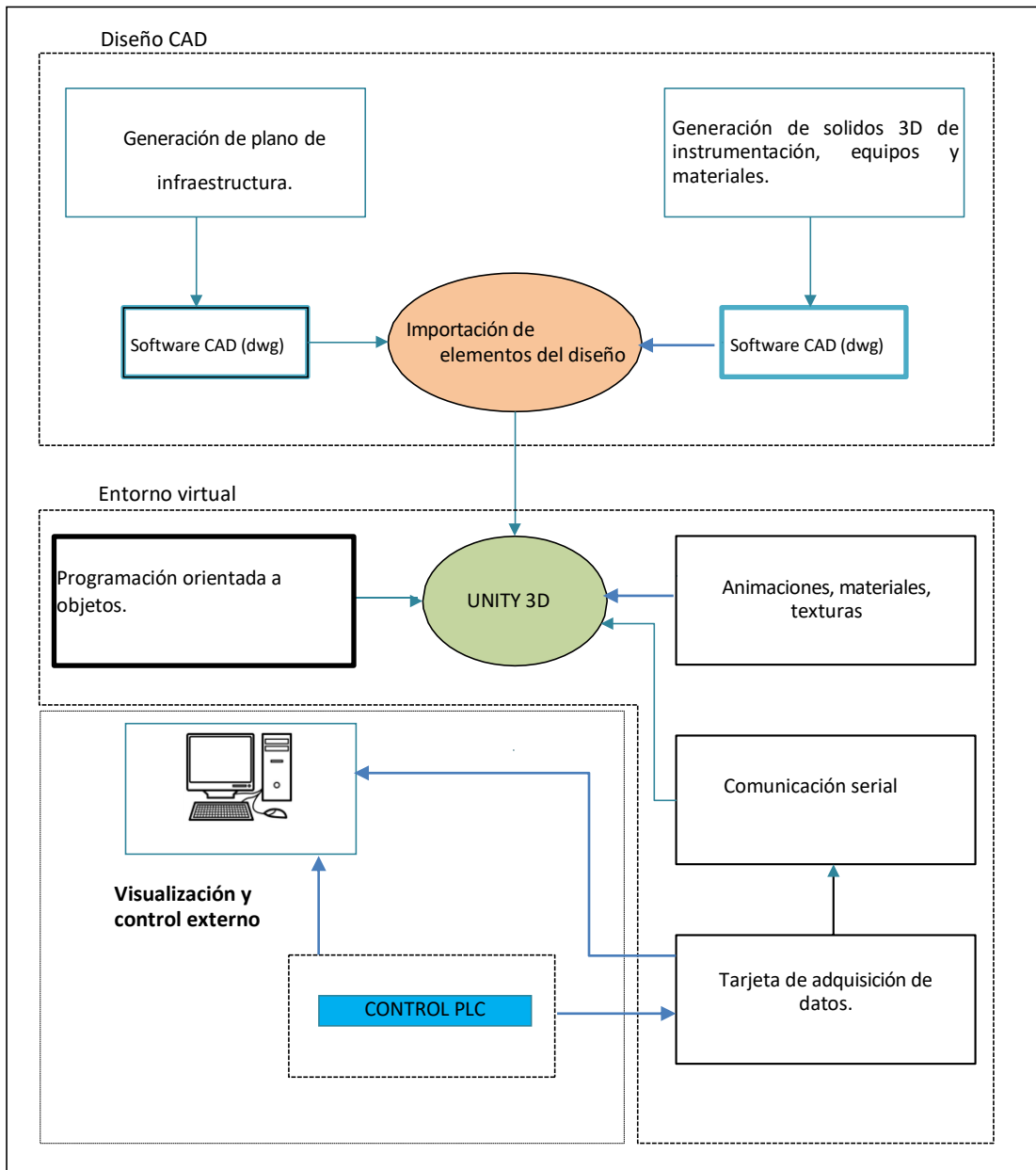


Figura 4.1. Diagrama de procesos.

Para la planificación, el proyecto se divide principalmente en fases, en las cuales una de ellas se divide en iteraciones:



**Etapa de formación.** - Incluirá la instalación de los materiales necesarios, la búsqueda de textos e indagación de herramientas, como la investigación de entornos virtuales industriales, tutorías instructivas o la ejecución de pruebas y errores.

**Etapa de desarrollo.** - Incluirá los parámetros del proyecto donde en cada etapa se debe crear un prototipo para probarlo hasta que se alcancen los objetivos principales del tema del proyecto de acuerdo con los experimentos del proyecto. El ensayo y error de la mejora continua.

El desarrollo del proyecto, que se dividió en cinco etapas, hay varias etapas en cada fase:

- **Análisis de los requisitos.** – Etapa de análisis de qué hacer, cómo hacerlo, qué herramientas, limitaciones y objetivos se quieren alcanzar.
- **Diseño de software y programación sistemas.** - Diseño del entorno virtual.
- **Implementación del entorno virtual.** - Desarrollo de la lógica de la aplicación.
- **Pruebas (ensayo y error).** - Etapa en la que se comprueba el correcto funcionamiento del módulo y de las funciones implementadas.
- **Revisión.** - Fase de integración de contenidos, la última fase consiste en la integración del material gráfico final y el mapeo final de las entradas y salidas de las señales digitales y analógicas requeridas.

En el caso de la revisión, las etapas finales y las conclusiones se refieren a una iteración específica del proyecto. Además, todos los comentarios son satisfactorios para el desarrollo del proyecto.

#### 4.1. Variables

**Tabla 4.1** Variable Independiente: Sistema de simulación

Concepto	Categoría	Consecuencia
Control de nivel.	Aumentar la cantidad de módulos.	Permitir un aprendizaje individual.
	Visualizar de mejor manera el proceso de control de nivel.	Mejora el interés del estudiante.
	Aumentar el campo aprendizaje.	Mejora mayor comprensión del proceso.

**Tabla 4.2.** Variable dependiente: Optimizar el método de aprendizaje del estudiante.

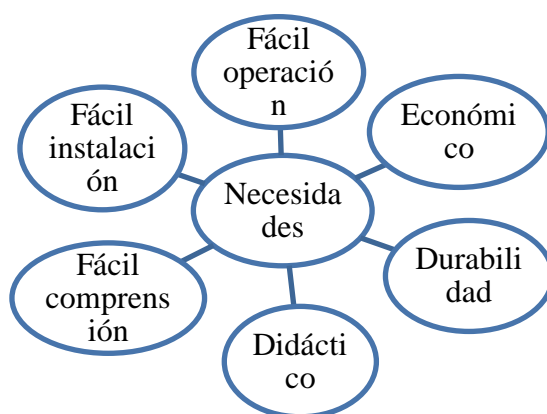
Concepto	Categoría	Indicadores	Técnicas	Instrumentos
Módulo DAQ planta de nivel.	Variador.	Simulación de control PID	Herramientas de Unity.	Unity.
	Electroválvulas	Control del paso del agua.	Catálogo	Controlador
	Diseño del tanque.	Características de material.	Modelado en 3d	Controlador
		Dimensiones	Modelado en 3d.	Sketchup
		Nivel máximo y mínimo.	Ecuación de Bernoulli	Unity.
	Controlador lógico programable	Control de actuadores.	Catálogo	Programación
		Contador.	Catálogo	Programación
		Control de señales eléctricas	PLCS7-300	Programación

#### 4.2. Implementación del entorno virtual

Para la creación del entorno virtual se han utilizado herramientas de modelado 3D (que nos permiten diseñar la planta en tres dimensiones) y el motor gráfico Unity3D para la animación y caracterización de los componentes del entorno virtual.

#### 4.3. Diseño y selección de componente

En el sistema de aprendizaje, el proceso de control de nivel se desarrollará en el sistema virtual según las necesidades de la universidad.



**Figura.4.2.** Necesidades del sistema controlado.

#### 4.4. Componentes que interviene en el sistema de control

Para analizar y comprender mejor las partes principales del sistema de control, se divide en diferentes partes según el tipo de sistema, como se muestra en la siguiente tabla

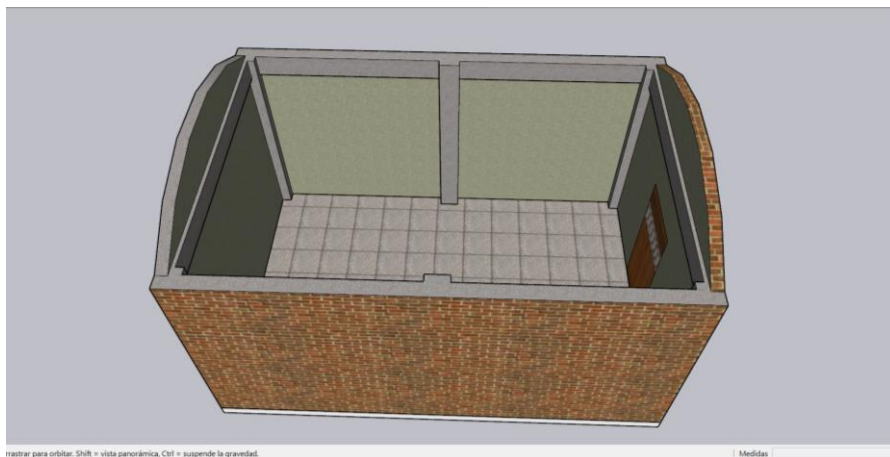
**Tabla.2: Partes del sistema de automatización**

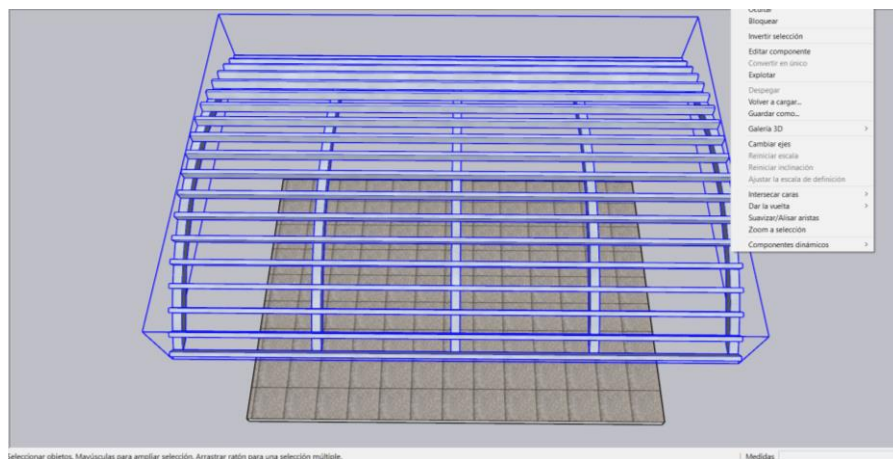
<b>Sistema</b>	<b>Partes del proceso.</b>
Diseño	Tanque, bombas, tuberías, válvulas.
Mapas uv	Modelado, Texturas, Puntos de rotación.
Actuador	Válvulas, bomba.
Pulsadores	Sistema de señales de entrada y salida.
Convertidores	Circuito convertidor de señal digital a analógicas.
Controladores	Controladores lógicos programable

Una vez determinadas las partes principales del sistema de control, comience con el diseño y selección de los materiales correspondientes requeridos para la simulación.

#### **4.5. Diseño de la planta modelado en 3D**

El proceso de diseño se desarrolla con la ayuda de un software de diseño CAD, el propósito es diseñar de forma rápida y precisa la estructura industrial en una forma tridimensional. Como sistema de formación basado en la realidad virtual, sus funciones y características de diseño deben ser las mismas que las reales.

**Figura 4.3. Diseño estructural del balcón.**

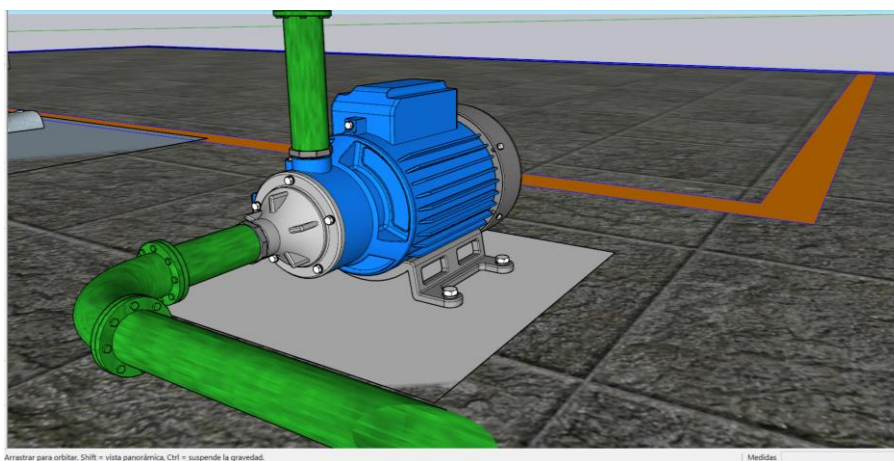


**Figura 4.4. Diseño estructural de la cubierta.**

Utilizando modelado CAD, se realiza el modelado en partes tales como el diseño de las bombas, válvulas, tuberías soportes, estructura, señalética.

Consiguiendo así del diseño único del sistema de control.

La forma de algunas estructuras es importante para la caja de control. Para dar a las personas un mejor sentido de la realidad.



**Figura 4.5. Diseño de la bomba de agua.**

El diseño de la fábrica virtual propuesta requiere equipos muy detallados, que pueden modelarse en otras herramientas CAD y luego importarse al programa de diseño para convertirlo en equipos personalizados, resaltando así la autenticidad de la fábrica. El diseño se toma del software Factory I / O, que es un diseño de procesamiento de control muy práctico.

Para el diseño de la planta se han utilizado modelos existentes en un catálogo web gratuito y de acceso abierto. La figura muestra el diseño preliminar con software de diseño CAD. Combine

todos los diseños para formar el diseño final de la fábrica con una entrada marcada y una red de seguridad.



Figura 4.6. Diseño final de la planta.

#### 4.6. Desarrollo de la aplicación

El proyecto contempla el DESARROLLO DE UN ENTORNO VIRTUAL 3D PARA EL CONTROL DE NIVEL DE AGUA, en el cual se describe en forma general el desarrollo de la aplicación: se va creando un proceso industrial de nivel, para eso se utilizará software de diseño 3D, para posteriormente se realizará mejoramientos del ambiente industrial para visualizarlo en el ambiente virtual de Unity aplicando diferentes herramientas que nos facilita el software Unity, como receptor de señales tenemos una DAQ nos permite realizar la interacción entre usuario y sistema virtual mediante controladores.

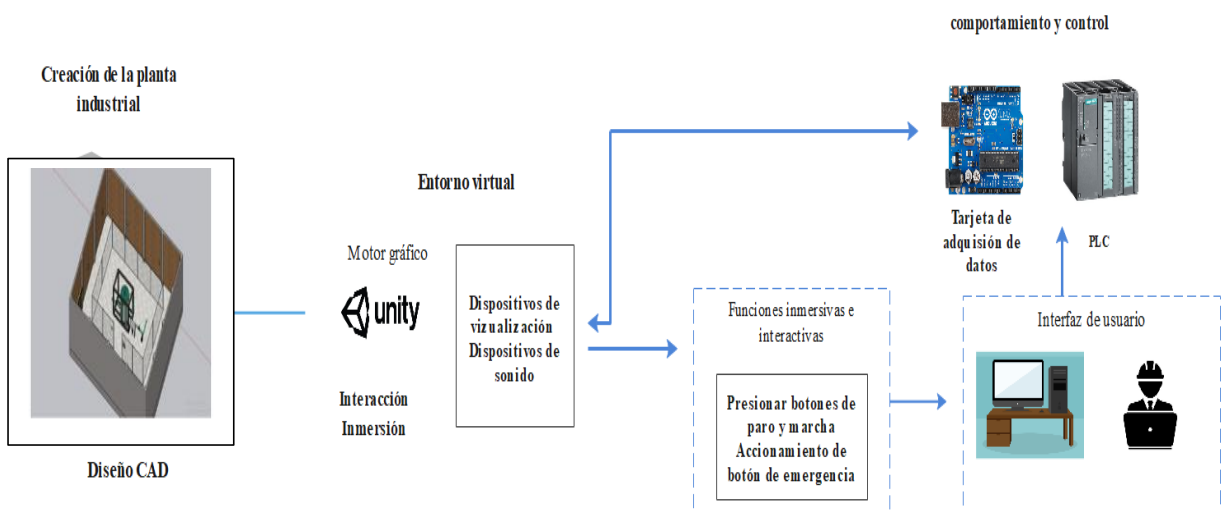


Figura 4.7. Diagrama de flujo

#### 4.7. Diseño del ambiente en el software Unity 3D

El modelo 3D se importa a Unity3D en formato \*.fbx, donde se pierde la textura, por lo que se deben agregar nuevamente texturas a todos los objetos para obtener un entorno virtual realista. Con el fin de mantener el diseño de la instalación para una mejor experiencia del usuario, no afecta la funcionalidad del entorno. Dando clic izquierdo en *Assets import package*.

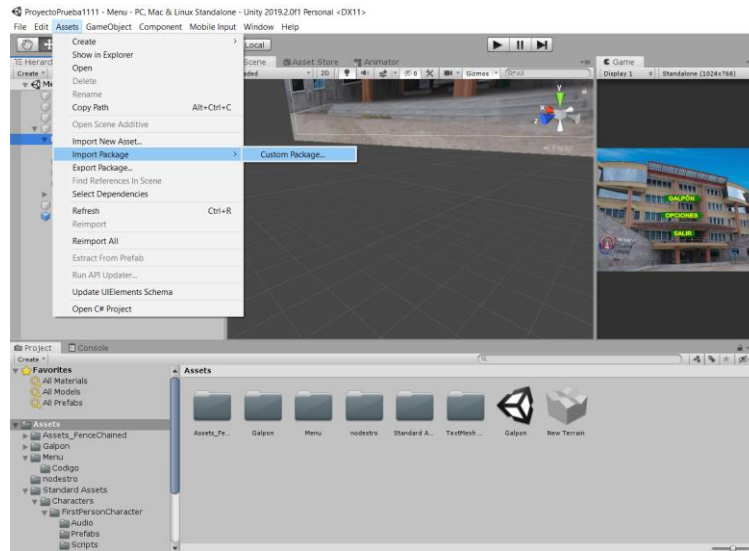


Figura 4.8. Importación del diseño 3D a Unity.

#### 4.8. Montaje de la escena en Unity3D

Se procede al montaje de la escena ya diseñada anteriormente exportada desde el software de diseño CAD en formato \*.fbx, para su posterior manejo tanto, como la programación, añadir textura, sonido, implementación de un avatar. Se puede visualizar dos ventanas la primera se encuentra al lado izquierdo nos permite modificar la escena implementando características como las ya mencionadas anteriormente la segunda ventana muestra una visualización de la escena como un ejecutable.

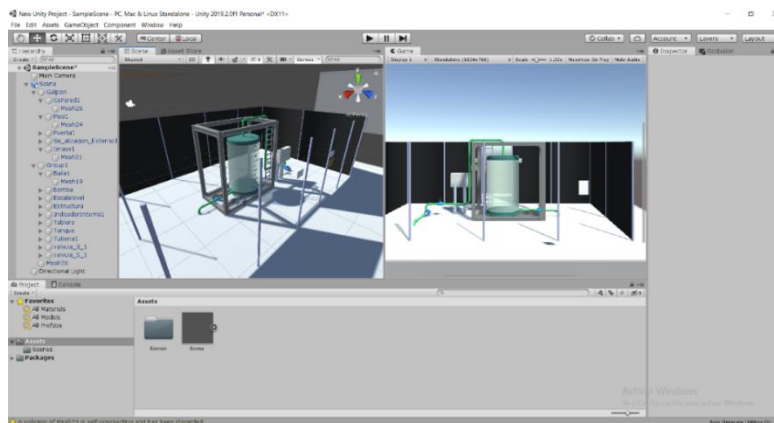


Figura 4.9. Ventana de trabajo Unity.



#### 4.9. Selección de texturas

Al momento de importar la escena en el programa de simulación mucho de los materiales no con lleva la misma textura, pierden detalles al momento de su importación para eso debemos proceder a crear materiales dando sus características:

Color, iluminación, textura, tipo de material, reflectividad, reflejos.

Se importó los siguientes materiales.

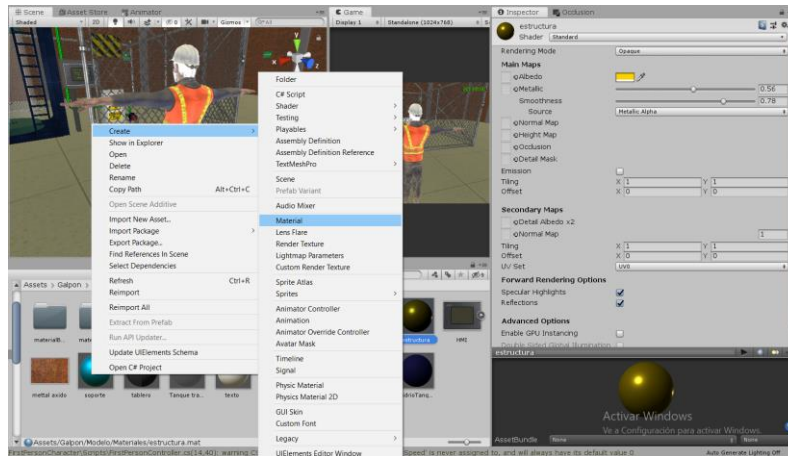


Figura 4.10. Textura de materiales e iluminación.

#### 4.10. Implementación de puntos de colisión

Cuando otros objetos rodean un sistema de partículas, el efecto suele ser más convincente si las partículas interactúan con dichos objetos permitiendo en Unity 3D cada interacción da como resultado fallas, las cuales son que los objetos al momento del contacto se dispersen o atraviesan los objetos, como consecuencia eso no da realismo de encontrarse en una planta de control usando la herramienta *box collider* generamos puntos de colisión para evitar los errores ya mencionados. Se dio características a los siguientes objetos como por ejemplo las estructuras de la planta, bombas, tuberías, cerramiento de malla, pantalla HMI.

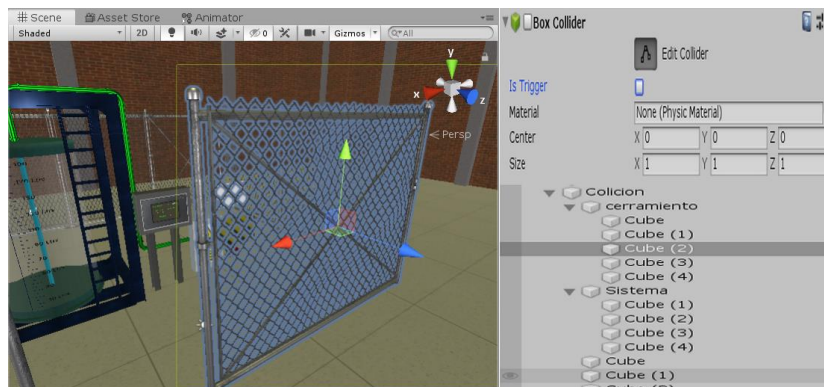


Figura 4.11. Puntos colaiders.

#### 4.11. Implementación de textos

El *control Text* muestra una pieza no-interactiva del texto al usuario. Este puede ser utilizado para proporcionales subtítulos o *labels* para otros controles o para mostrar instrucciones u otros textos, para nuestro diseño utilizamos también textos integrados nos permite el uso de textos implícitos.

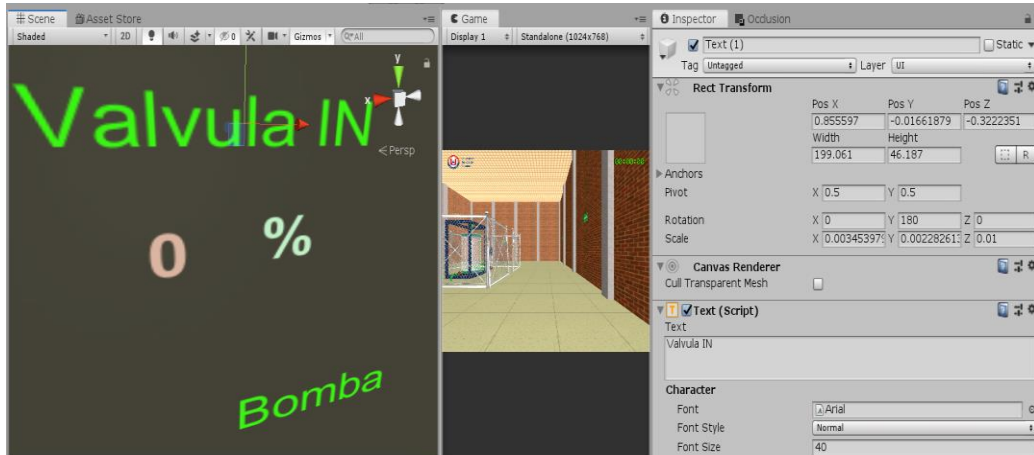


Figura 4.12. Textos y labels.

#### 4.12. Introducción de sonido de la bomba y agua

Para un mejor materialismo, mejor experiencia entre el usuario y el entorno virtual se procedió a la implementación de sonidos industriales los cuales fueron la integración de sonidos de una bomba en funcionamiento, el sonido del agua al momento de llenarse el tanque donde usuario al momento de activar cada uno de los actuadores se coordinará con el sonido.

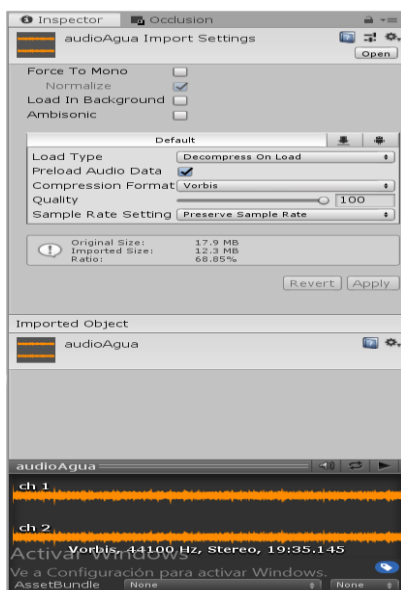


Figura 4.13. Audio del agua.

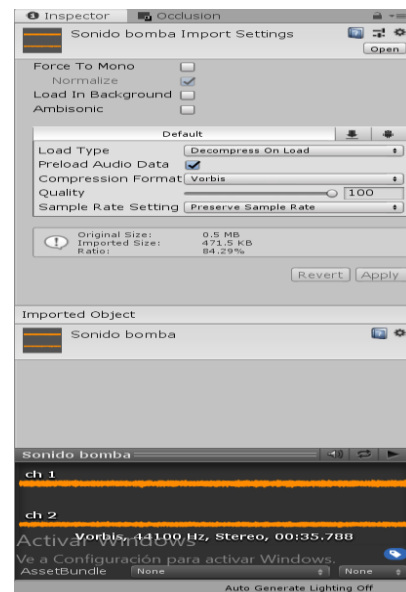


Figura 4.14. Audio de la bomba.



#### 4.13. Programación del movimiento de la capsula de movimiento.

Procedimos a crear la *colaider capsul* que sería nuestro player programando un scrip le daremos movimiento donde se podrá interactuar en primera persona como opción, en el espacio virtual donde ese encuentra nuestra planta de control, donde escucharemos visualizaremos las funciones de cada una de los componentes de la planta.

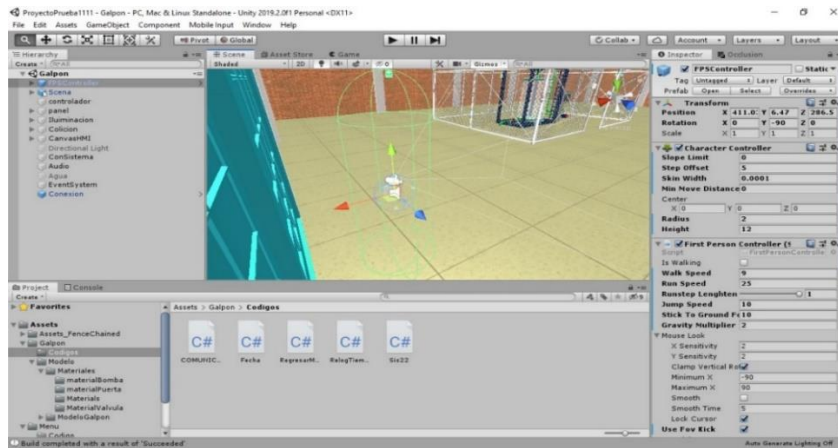


Figura 4.15. Programación player primera persona.

#### 4.14. Desarrollo del avatar.

Una de las características fundamentales de Unity 3D es su disposición de importar objetos personajes ya diseñados, programados con movimiento el cual nos ahorra tiempo al momento de realizar nuestro diseño en su totalidad, se importó un avatar ya modelado con sus características de un operador, la programación se dio las siguientes características las cuales puede realizar el movimiento, caminar, correr interactuar con el medio ambiente aquí se usa una herramienta fundamental llamada "User Gravity" permite implementar gravedad a nuestro avatar permanecerá fijo al desactivar esta opción el avatar saldría de la escena al momento de ejecutarla.

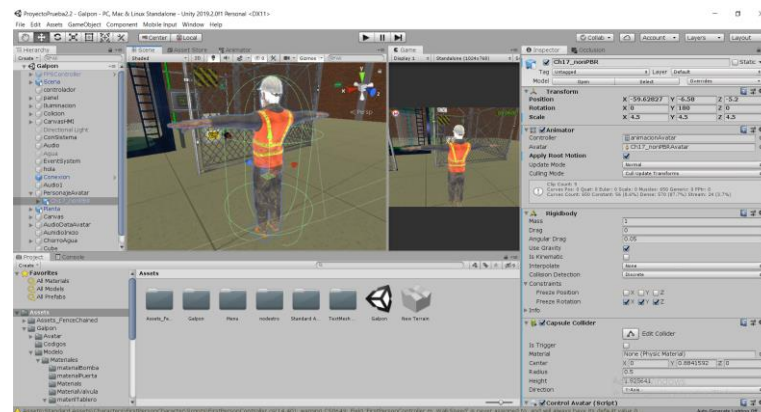


Figura 4.16. Programación del diseño del avatar.

#### 4.15. Desarrollo del menú de entrada

El menú de entrada fue diseñado con dos escenas principales, la primera nos permite el ingreso de la entrada al proceso la segunda permite entrar al menú de comunicación de enlace con la tarjeta de adquisición de datos. Unity nos permite una fácil importación de imágenes como se visualiza en el fondo del menú tenemos una foto de la parte exterior de la Universidad Técnica de Cotopaxi.



Figura 4.17. Diseño del menú de entrada.

#### 4.16. Comunicación entre Unity3D y DAQ

Es parte fundamental la comunicación serial entre Unity y la tarjeta de adquisición de datos en al cual nosotros usaremos un microcontrolador arduino, donde necesitamos realizar la comunicación mediante un scrip nosotros definiremos las diferentes variables que interfieren en la comunicación mostradas en la fig 4.18.

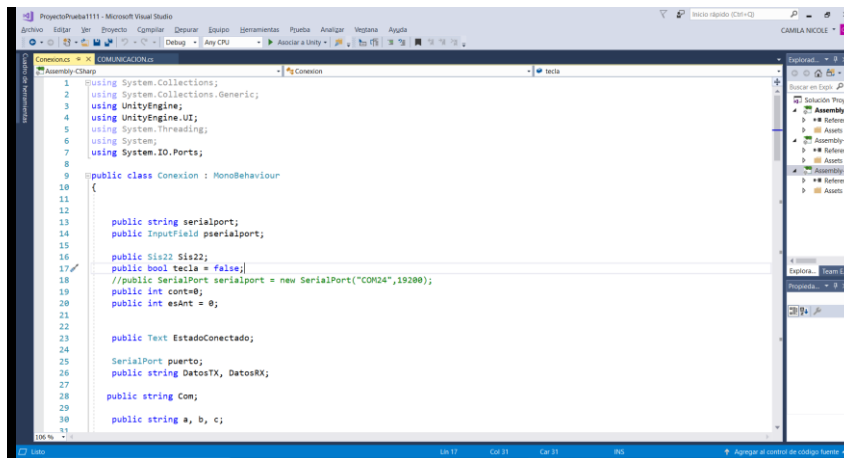


Figura 4.18. Comunicación Serial.

#### 4.17. Programación sistema de salida y entradas de datos al software Unity

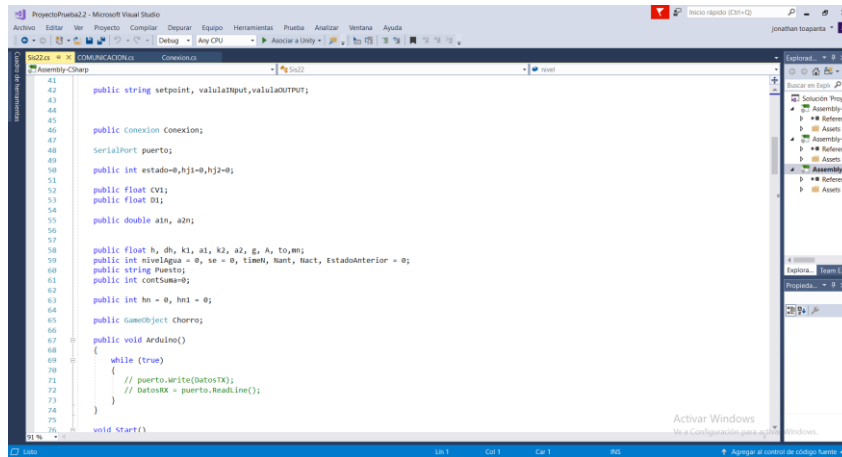


Figura 4.19. Comunicación Serial.

#### 4.18. Simulación del comportamiento de la planta.

El algoritmo de control que simula el comportamiento de la fábrica mediante el principio de Bernoulli, en este caso, se define de la siguiente manera:

$$\frac{dh}{dt} = \frac{q_{in} - q_{out}}{A} \quad (32)$$

$$q_{in} = k_1 a_1 \quad (33)$$

$$q_{out} = k_2 a_2 \sqrt{2gh} \quad (34)$$

Donde:

$h$ : nivel del tanque (0-200 cm)

$q_{in}$ : flujo de entrada en el tanque (0 – 5 V)

$q_{out}$ : flujo de salida en el tanque (0 – 5 V)

$A$ : área del tanque (1.5 m)

$k_1$ : constante de la válvula en la entrada del tanque (0.05)

$k_2$ : constante de la válvula en la salida del tanque (0.015)

$a_1$ : apertura de la válvula de entrada del tanque (0 – 100 %)

$a_2$ : apertura de la válvula de salida del tanque (0 – 100 %)

$g$ : gravedad (9.8 m/s<sup>2</sup>)

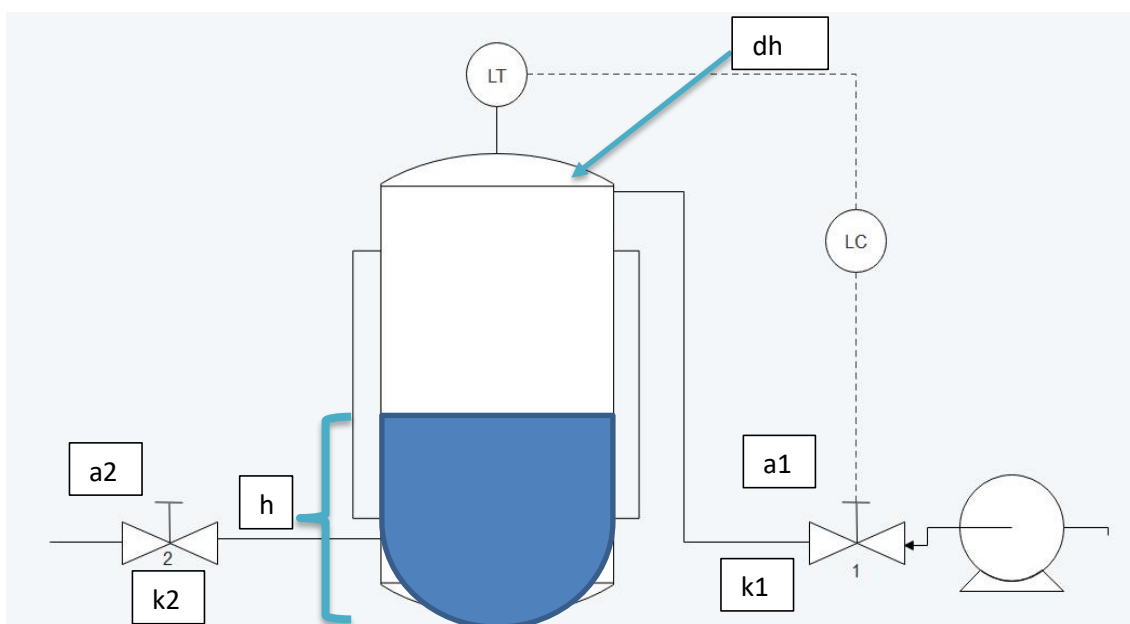
**Algoritmo del comportamiento del tanque**

$$\frac{dh}{dt} = \frac{q_{in} - q_{out}}{A}$$

Este nos permite realizar el comportamiento real de el llenado del tanque.

$$A = (k_1 * a_1 - k_2 * a_2 * \text{Mathf.Sqrt}(2 * g * h));$$

$$h = h + dh * t_0; \quad (35)$$



Donde:

dh: altura total del tanque.

h: la variación de la altura del nivel del líquido

$k_1$ : constante de la válvula en la entrada del tanque (0.05)

$k_2$ : constante de la válvula en la salida del tanque (0.015)

$a_1$ : apertura de la válvula de entrada del tanque (0 – 100 %)

$a_2$ : apertura de la válvula de salida del tanque (0 – 100 %)

$g$ : gravedad (9.8 m/s<sup>2</sup>)

Así es como ingresa nuestra ecuación de Bernoulli a nuestra programación para que el comportamiento de nuestra planta sea real donde nosotros tenemos las variables utilizadas en la ecuación de Bernoulli.

#### 4.19. Visualización de las formas de ondas de sintonización del lazo del control del sistema industrial de nivel

La verificación del funcionamiento del sistema de control se visualiza mediante la implementación del HMI a través de herramientas UI donde se visualiza las variables de entrada y salida.

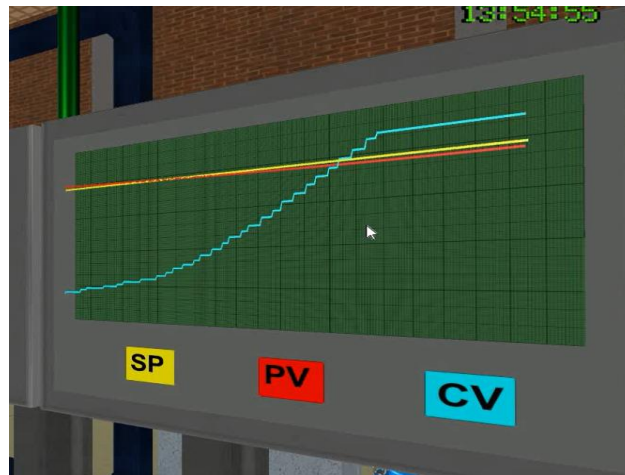


Figura 4.20. Comunicación Serial.

##### 4.19.1. Tarjeta de adquisición de datos

Esta tarjeta nos permitirá recibir datos de Unity3D como también almacenar datos analógicos para nosotros poder realizar el control mediante un PID utilizando un microcontrolador programado donde también le adaptamos un convertidor de señal digital analógico, porque para realizar el control usaremos un microcontrolador arduino el cual no tiene salidas analógicas.

Para aplicar un control PID veremos los parámetros básicos a tener en cuenta sobre el control proporcional integral veremos su aplicación en un sistema real, utilizando un microcontrolador programado.

La ecuación del PID:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (36)$$

#### 4.20. Diagrama eléctrico de la DAQ

Para realizar el control de nuestro sistema simulado necesitamos determinar el control de nuestras 3 variables las cuales nos permitirá el enlace, diferentes tipos de controles. Aquí detallaremos las entradas y puertos los cuales serán utilizados. Como partes importantes de nuestra DAQ tenemos los convertidores digitales analógicos para poder así tener señales las cuales podemos controlar dependiendo de diferentes necesidades del programador.

- Tenemos en el SP tanto externamente conectado a una bornera la cual nos permite realizar el control de una manera remota al sistema sea usando señales de un controlador como tambien mediante un potenciómetro
- El CV es el que nos permite el control de la apertura de nuestras valvulas de entrada como de salida
- PV es nuestro sensor el cual nos permite determinar el nivel del tanque el cual es de 0-200 litros representado respectivamente de 0 a 5 voltios

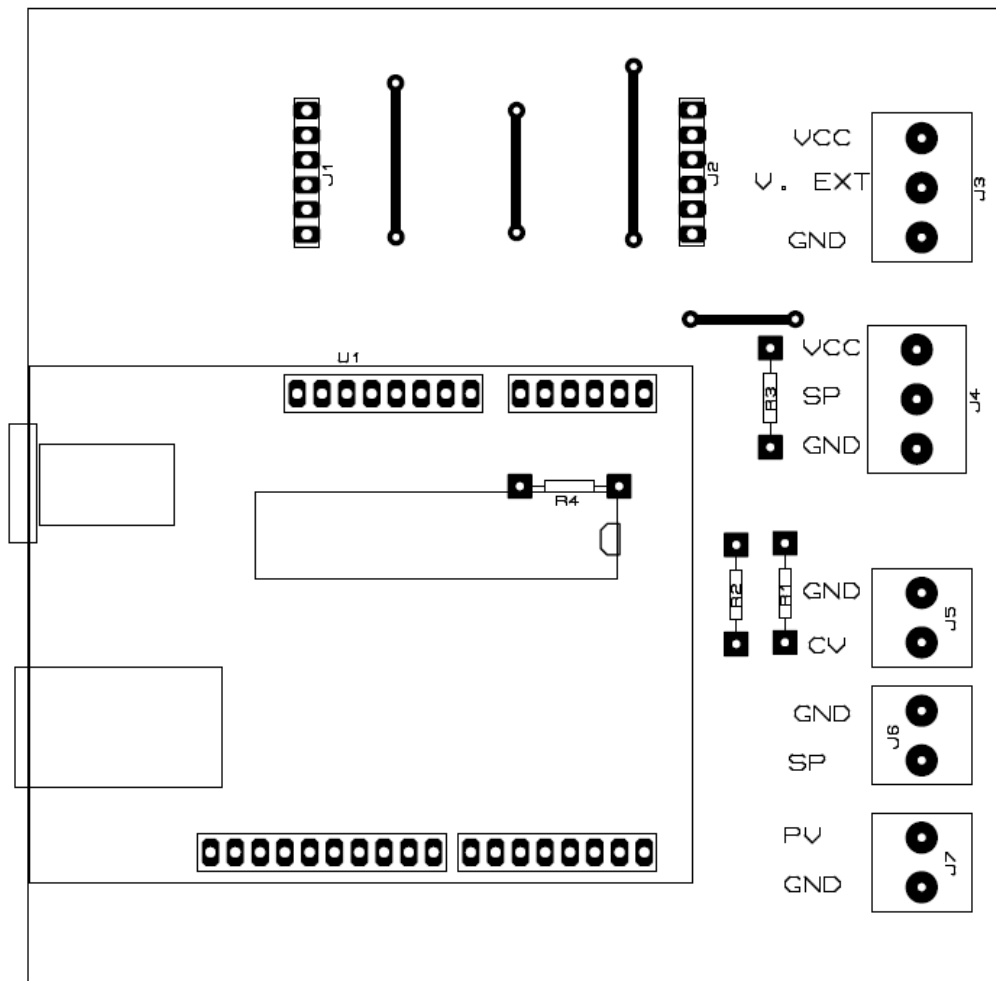


Figura 4.21. Esquema diseño entradas y salidas placa DAQ.

4.21. Diagrama de un control PID

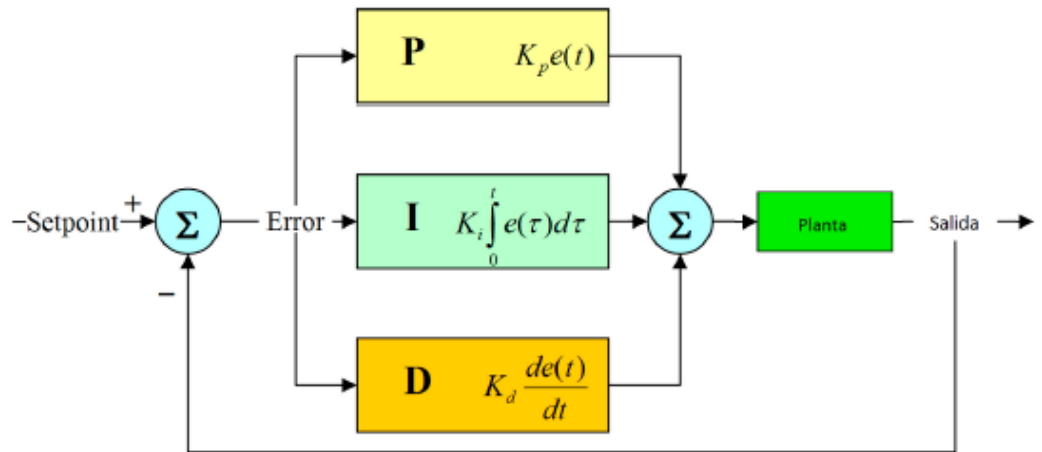


Figura 4.22. Diagrama PID

Dada la ecuación se define que:

$e(t)$  es el error de la señal.

$u(t)$  salida del controlador y entrada de control al proceso.

$K_p$  es la ganancia proporcional.

$T_i$  es la constante de tiempo integral.

$T_d$  es la constante de tiempo derivativa.

**Del diagrama de flujo determinamos lo siguiente:**

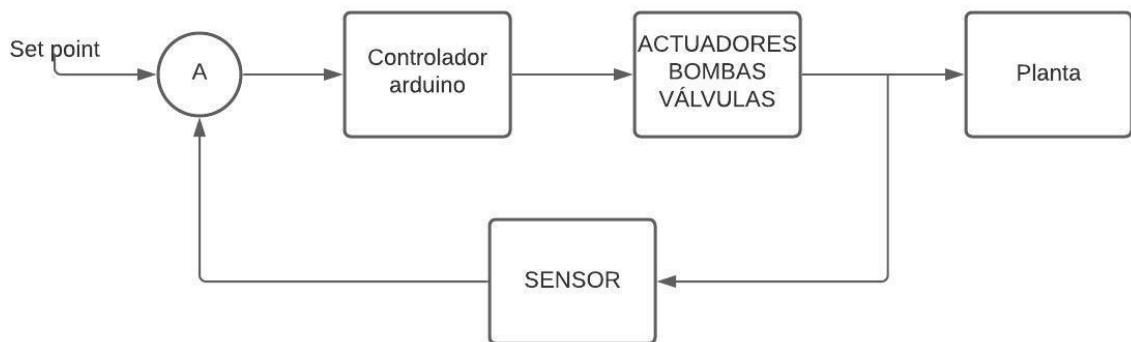


Figura 4.22. Diagrama PID

El programa anterior funciona bien, pero tiene limitaciones cuando se aplica a un sistema real. Para que se comporte como un PID a nivel industrial, se deben considerar otros parámetros. El algoritmo PID funciona mejor cuando se ejecuta a intervalos regulares. Si el concepto de tiempo

se incluye en el PID, los cálculos se pueden simplificar. Los PID para principiantes están diseñados para períodos irregulares. Esto puede plantear dos problemas:

- Hay un comportamiento incoherente en el PID, ya que a veces se ejecuta con regularidad y otras no.

Se requieren operaciones matemáticas adicionales para calcular los términos que corresponden a la parte derivada e integral del PID ya que ambos dependen del tiempo. Debe asegurarse de que la función realizada por el PID lo haga de forma regular. Basado en un tiempo de ejecución especificado, el PID decide si los cálculos deben ser realizados o devueltos por la función. Una vez que nos aseguramos de que el PID se ejecuta de forma regular, los cálculos de las partes integrales y derivadas se simplificarán

#### 4.22. Programación PLC Simatic S7-300

Aquí, tenemos un control PID para controlar tres variables, lo que nos permite controlar SP, CV, PV. Esto no solo nos permite manipular la válvula del depósito de agua y la entrada de la bomba, sino que también nos permite detectar la salida de la señal del sensor para poder controlar. Se recibe una señal en bits del sensor, la señal ingresa a un bloque de normalización y su salida ingresa a un bloque que escala la señal normalizada entre valores reales antes de enviarla al bloque PID. Cabe señalar que cuando el PLC se enciende y recibe señales de sensor, el bloque de normalización siempre estará activo.

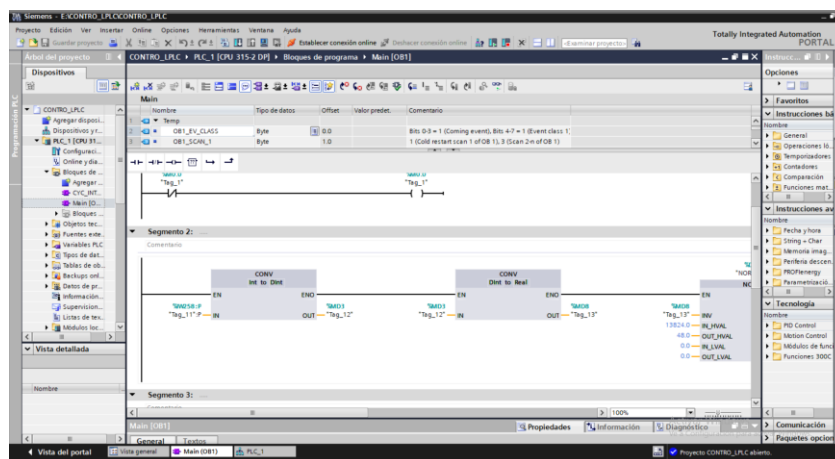


Figura 4.23. Programación TIA PORTAL

#### 4.23. Identificación del sistema y sintonización del PID

##### 4.23.1. Método de Sintonía Lambda



Este es un método simple, siempre que los parámetros de diseño se seleccionen correctamente, puede funcionar bien en ciertas situaciones. El método básico elimina un polo del proceso, lo que dará como resultado una mala respuesta a las perturbaciones de carga en un proceso dominado por constantes de tiempo.

Requerimientos: modelo de primer orden, estable o integral, con tiempo muerto.

Parámetros de diseño: constante de tiempo del lazo cerrado ( $\lambda$ ).

$$\lambda = \frac{\lambda}{T_0} \quad (36)$$

Control PID

$$PID_{(s)} = k\left(1 + \frac{1}{T_i s}\right) \quad (37)$$

#### 4.23.2. Sintonización por método Lambda.

El modelo matemático del proceso es el siguiente:

$$G(s) = \frac{0.02}{1+s} e^{-3.6s} \quad (38)$$

Donde:

$$T=1 ; L=3.6; K=0.02$$

Y definiendo la constante  $T_{cl}=1$ , y utilizando las fórmulas correspondientes, se obtiene.

$$k_c = \frac{1}{K} \left( \frac{\frac{L}{2} + T}{\frac{L}{2} + T_{cl}} \right) \quad (39)$$

$$= \frac{1}{0.02} \left( \frac{\frac{3.6}{2} + 1}{\frac{3.6}{2} + 1} \right)$$

$$= 5$$

$$T_i = T + \left(\frac{L}{2}\right) \quad (40)$$

$$= 1 + \left(\frac{3.6}{2}\right) = 1.8$$

$$T_d = \left(\frac{TL}{L+2T}\right) \quad (41)$$

$$= \left(\frac{1(3.6)}{3.6 + 2}\right)$$

$$= 0.643$$

Se procede a realizar una tabla con los valores obtenidos. (Tabla)

**Tabla 3. Constantes obtenidas para el proceso control nivel**

Constantes		Método de sintonía
	Auto calibre	Lamba
	Valor	Valor con sintonía
<b>Kc</b>	2	10.52
<b>Ti</b>	1	1
<b>Td</b>	0	0.001

## 5. ANALISIS DE RESULTADOS

### 5.1.: Configuración del teclado

Se realizó un sistema de control externo como un videojuego de Pc donde al presionar las distintas teclas el usuario puede desplazarse por la escena simulada en Unity 3D.

Instrucciones de las teclas animadas.

- **W** caminar hacia delante
- **S** caminar hacia atrás
- **A** caminar hacia la izquierda
- **D** caminar hacia la derecha
- **Esc** control del mouse
- **Shift** Seleccionar cursor mouse
- **→** girar la cámara izquierda
- **→** girar la cámara derecha
- **↑** cámara hacia arriba
- **↓** cámara hacia abajo

### 5.2. Diagrama y diseño de la DAQ

Empleando la metodología descrita en el capítulo 4, se logra obtener el diagrama de la tarjeta de adquisición de datos para sistemas de control finalizado, dando paso a la realización de las pruebas, para verificar su correcto funcionamiento y utilidad. Donde nos permite tener las señales de entrada y salida.

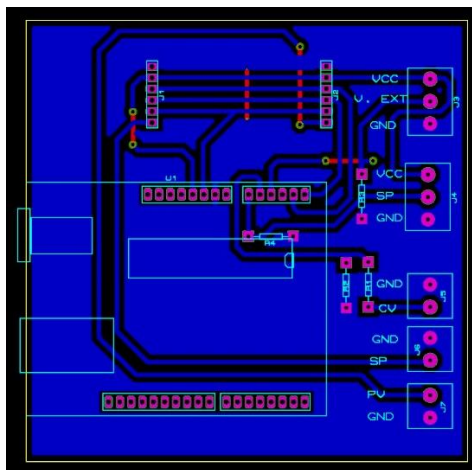


Figura 5.1. Módulo para el sistema de control.

### 5.3. Presentación menú de inicio del programa

El programa de aplicación de realidad virtual desarrollado se centra en el proceso de control a nivel industrial y la relación entre los usuarios en este caso. Además, se ubican letreros de seguridad donde se deben garantizar las explicaciones durante el entrenamiento. De esta forma, el entrenamiento para diferentes tipos de controles es completo y efectivo. En el diseño de la aplicación se desarrolló un menú de entrada que nos permite acceder a la comunicación serial del escenario y al puerto de la tarjeta de adquisición de datos para su posterior control. Como se muestra en la Figura 5.2 a continuación.



Figura 5.2. Menú de entrada

### 5.4. Interacción del avatar con la planta de control

El entorno se vuelve más interactivo al tener avatar que realizan tareas dentro de la aplicación y con la que es posible interactuar.

Aquí se muestra ya simulado la interacción de nuestro avatar con la planta el cual nos permitirá, recorrer libremente por el área donde podremos visualizar la pantalla HMI, el llenado y vaciado del tanque, visualizamos el control del tanque la cantidad de litros que existe.



Figura 5.3. Interacción del avatar con el entorno.

### 5.5. Visualización de la pantalla HMI

En la pantalla HMI se muestra el nivel del líquido en centímetros, el porcentaje de apertura de la válvula de entrada y válvula de salida controlando el valor de apertura como también podemos controlar nuestro SP interno o externo.



Figura 5.4. Visualización de la pantalla HMI

### 5.6. Curvas de respuestas del proceso de nivel.

El control PID se realizó para el control del nivel de líquido en el tanque manteniendo un nivel de agua constante y en caso de haber un consumo recompensar dicho consumo y mantenerlo siempre al nivel deseado el cual es monitoreado desde el panel de control que nos muestra las curvas de respuesta de las variables que actúa en este control.



Figura 5.5. Formas de ondas en unity.

### 5.7. Prueba de control externo ON/OFF.

El sistema de control por lazo cerrado es el más sencillo que podemos encontrar se trata del sistema ON/OFF. Como se muestra en la figura 5.6.



Figura 5.6. Visualización del control ON/OFF

### 5.8. Prueba de control con PID

La forma de conexión y la interfaz de programación para el control PID será el encargado de manipular la variación de control de nivel donde el SP puede ser manipulado según el rango requerido como también las variables de CV y PV. Usando un PLC-S7-300 para realizar el control.

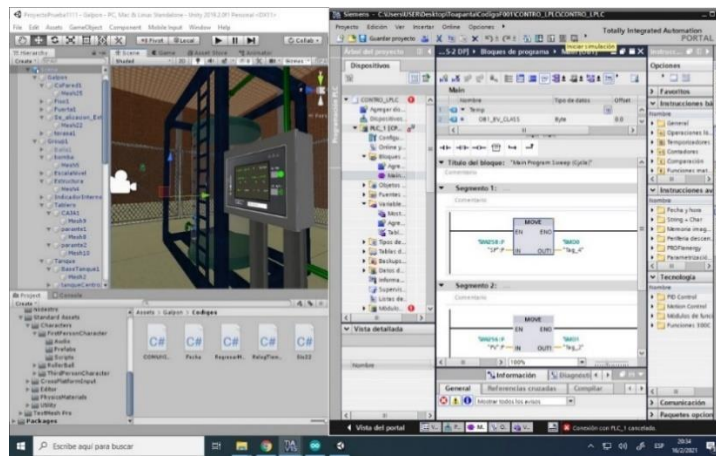


Figura 5.7. Control PID

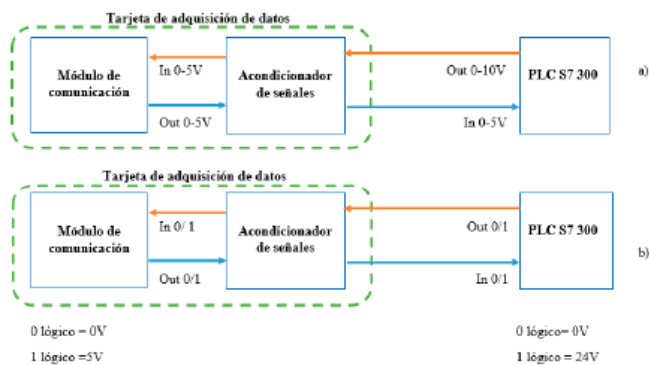


Figura 5.7. Acondicionamiento de señales eléctricas.

### 5.9. Método de obtención de nuestro modelo matemático.

Para la obtención de nuestro modelo matemático se realizó primero la toma de datos en Excel de los valores mostrados en la tabla 4 para poder realizar la sintonía de nuestro sistema de control PID utilizamos como herramienta el software Matlab para ingresar los datos de nuestro sistema, donde posteriormente usamos como herramienta *System identification toolbox*. Los datos completos de encuentran en el anexo 15.

**Tabla 4. Datos históricos de nuestro sistema virtual.**

PV			SP		
h(cm)	voltage	tiempo(s)	h(cm)	voltage	tiempo
0	0	0,000	0	0,000	0,000
1	0,025	0,210	1	0,025	0,210
2	0,050	0,420	2	0,050	0,420
3	0,075	0,630	3	0,075	0,630
4	0,099	0,840	4	0,100	0,840
5	0,124	1,050	5	0,125	1,050
6	0,149	1,259	6	0,150	1,259
7	0,174	1,469	7	0,175	1,469
8	0,199	1,679	8	0,199	1,679
9	0,224	1,889	9	0,224	1,889
10	0,249	2,099	10	0,249	2,099
11	0,273	2,309	11	0,274	2,309
12	0,298	2,519	12	0,299	2,519
13	0,323	2,729	13	0,324	2,729
14	0,348	2,939	14	0,349	2,939
15	0,373	3,149	15	0,374	3,149
16	0,398	3,358	16	0,399	3,358
17	0,423	3,568	17	0,424	3,568
18	0,447	3,778	18	0,449	3,778
19	0,472	3,988	19	0,474	3,988
20	0,497	4,198	20	0,499	4,198
21	0,522	4,408	21	0,524	4,408
22	0,547	4,618	22	0,548	4,618
23	0,572	4,828	23	0,573	4,828
24	0,597	5,038	24	0,598	5,038
25	0,622	5,248	25	0,623	5,248
26	0,646	5,457	26	0,648	5,457
27	0,671	5,667	27	0,673	5,667
28	0,696	5,877	28	0,698	5,877
29	0,721	6,087	29	0,723	6,087
30	0,746	6,297	30	0,748	6,297
31	0,771	6,507	31	0,773	6,507
32	0,796	6,717	32	0,798	6,717
33	0,820	6,927	33	0,823	6,927
34	0,845	7,137	34	0,848	7,137
35	0,870	7,347	35	0,873	7,347
36	0,895	7,556	36	0,897	7,556
37	0,920	7,766	37	0,922	7,766
38	0,945	7,976	38	0,947	7,976
39	0,970	8,186	39	0,972	8,186

El cual nos dio a conocer nuestro modelo matemático para poder nosotros obtener los datos de sintonía de nuestro control.

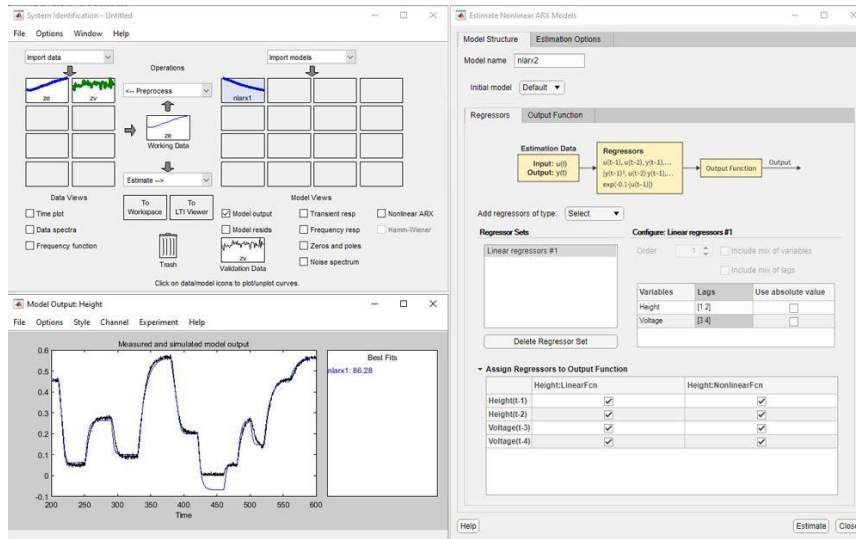


Figura 5.8. Modelo matemático del sistema de control de nivel.

Como resultado tenemos la función de transferencia de nuestro entorno calculado por nuestro software Matlab una vez recopilado los datos nos dio como resultado la siguiente función de transferencia.

$$G(s) = \frac{0.1}{1+s} e^{-0.12s} \quad (42)$$

Una vez obtenido el modelo matemático se dio a configurar con otra herramienta del sistema de Matlab el cual es el PID *tuner* el cual nos permite obtener una forma de onda de sintonía la cual nosotros obtuvimos usando el método de sintonía Lambda.

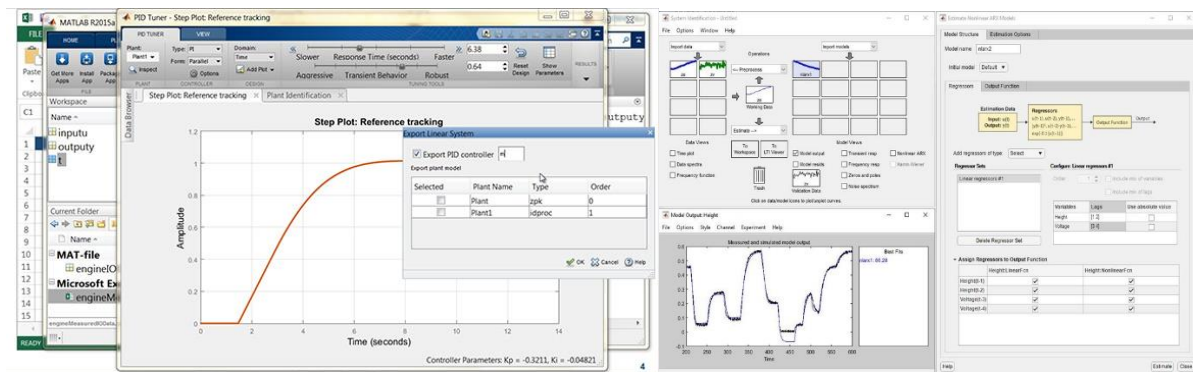


Figura 5.9. PID Tuner sintonía PID.

### 5.10. Formas de ondas sintonización de PID.

Para la sintonización del controlador PID se lo realizó mediante el método lambda de sintonía el cual nos permitió determinar las ondas de nuestras señales de control.



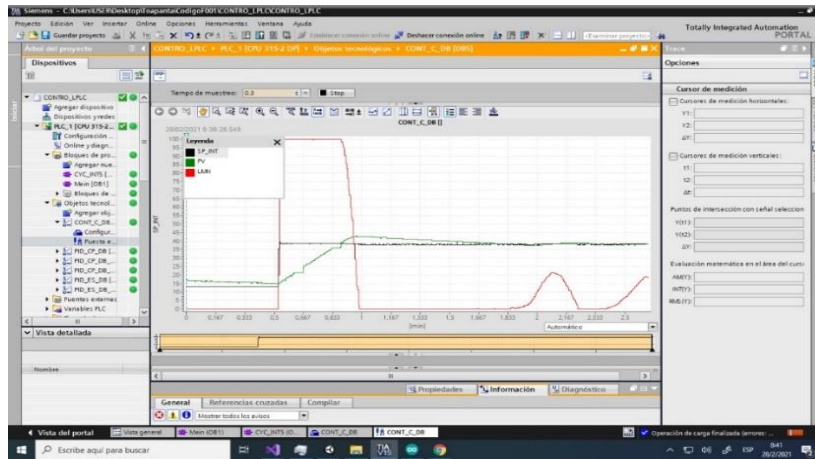


Figura 5.10. Visualización de las ondas de sintonía de PID primera prueba

Con la primera prueba de sintonía logramos visualizar varios sobre impulsos que tenía nuestro control realizado mediante el manejo de la herramienta PID tunin realizamos una segunda prueba la cual nos permitió visualizar de mejor manera la sintonización de nuestro control PID.

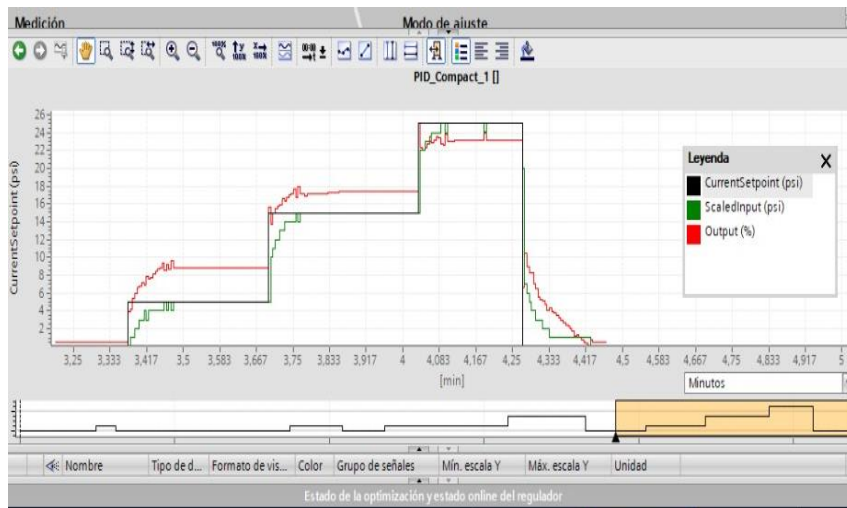


Figura 5.11. Visualización de las ondas de sintonía de PID final.

### 5.11. Forma de onda PWM rectificada a analógica

Para emitir las señales de control las cuales son SP, CV, PV debemos convertir la señal digital en analógico usando un convertidor digital analógico con la ayuda de un osciloscopio podemos captar las dos señales al mismo tiempo para visualizar el filtrado de la señal pwm a analógica.

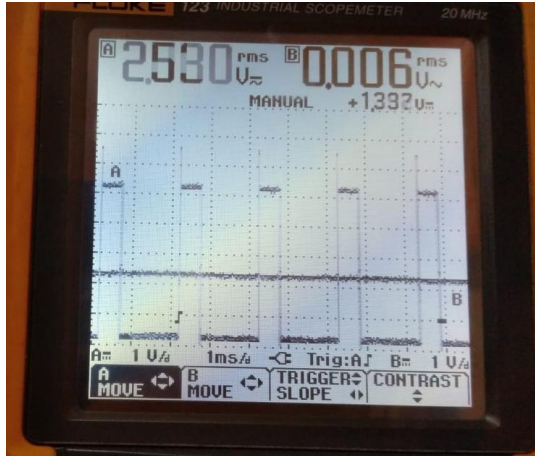


Figura 5.12. Visualización del filtrado de onda PWM a analógico

### 5.12. Verificación de funcionamiento del sistema mediante la medición de voltaje.

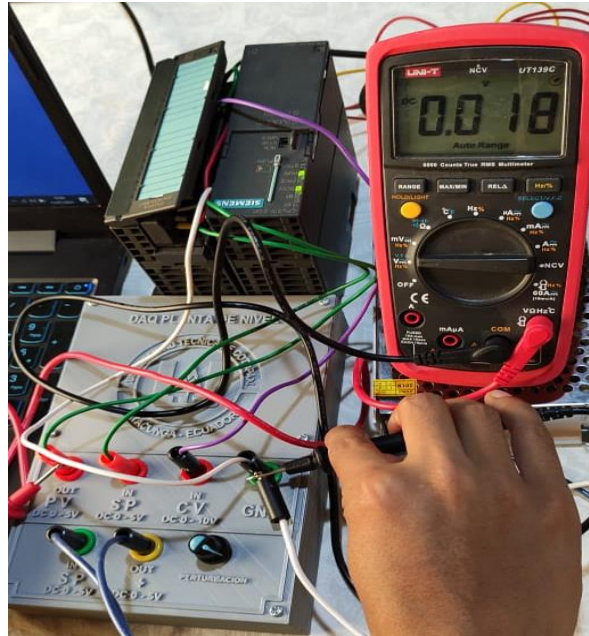
En el entorno virtual se encuentra nuestro tanque de nivel el cual nos da como valor de 0 a 200cm como dato digital, donde se convierte ese dato en dato analógico en un rango de 0 a 5 v para la comprobación se realizó tres medidas:

La primera medida se realizó con el tanque en su menor medida como se puede observar:



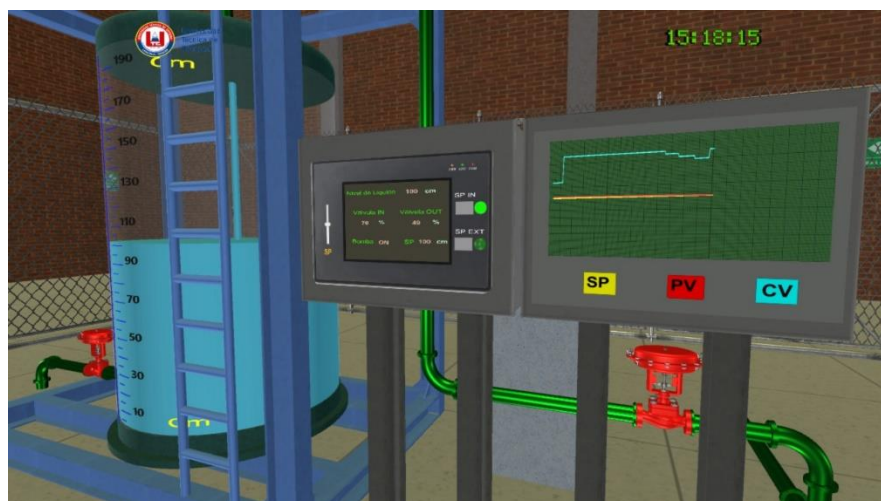
Figura 5.13. Tanque de nivel de agua vacío.

En la parte virtual podemos observar cómo nuestras ondas de señal del SP, PV, CV, se encuentran bajas ya que no se detecta el nivel del líquido se encuentra en cero. Al realizar la medición externa en nuestra tarjeta de adquisición de datos tenemos como resultado el siguiente voltaje:



**Figura 5.14. Voltaje de medición tanque vacío.**

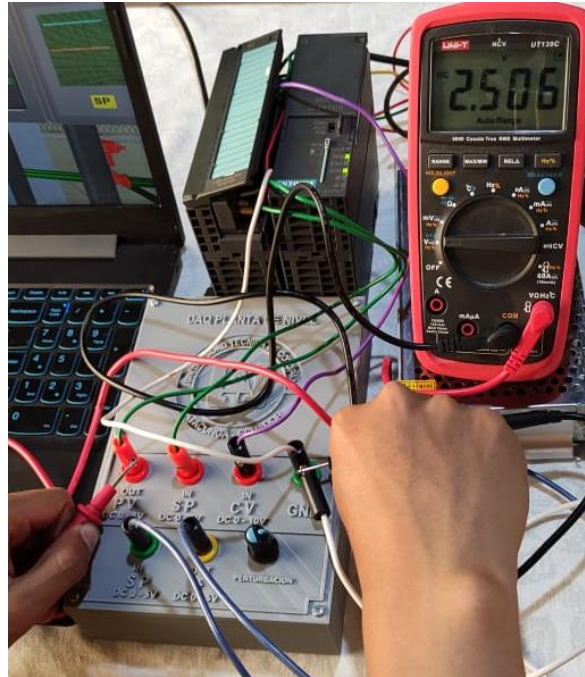
La segunda medida que se realizó en el tanque fue cuando su nivel medio el cual sería 100 cm:



**Figura 5.15. Tanque de nivel medida 100 cm.**

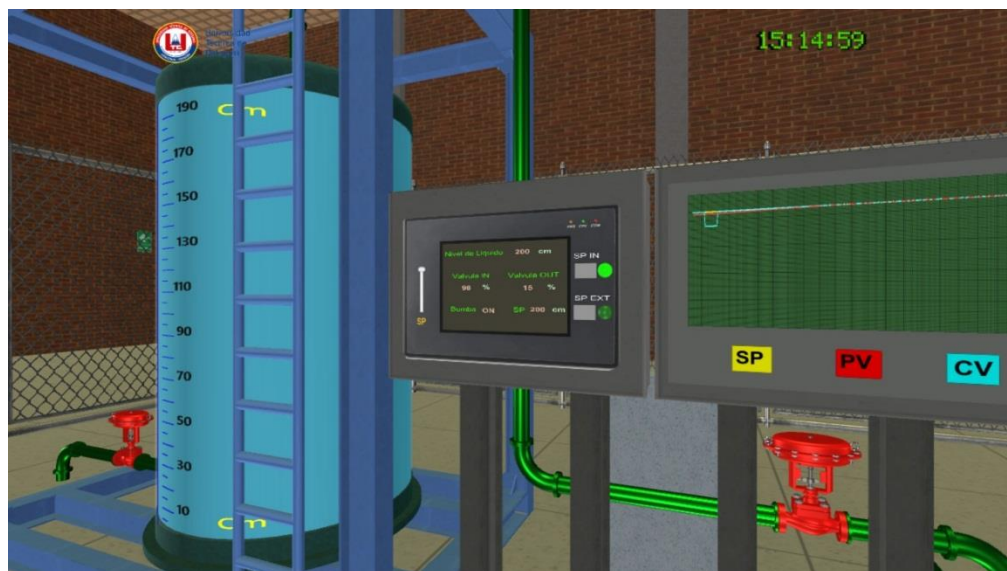
Se obtuvo como resultado en la parte virtual se pudo visualizar como las ondas de las señales del SP PV CV sufrieron una variación incremento como podemos ver en la pantalla HMI. Al realizar la medición externa se observa un incremento del voltaje inicial:





**Figura 5.16.** Voltaje de medición a 100 cm de capacidad del tanque.

Como medida de referencia se obtuvo con el nivel del líquido a su capacidad máxima la cual en valores sería 200cm:



**Figura 5.17.** Tanque de nivel medida 200 cm.

Aquí como resultado tenemos en parte virtual se visualiza las ondas de señales a su máximo nivel, así como en la pantalla se puede visualizar tanto el SP y el nivel del tanque en 200 cm y como resultado en la medición de voltaje externo se obtuvo:

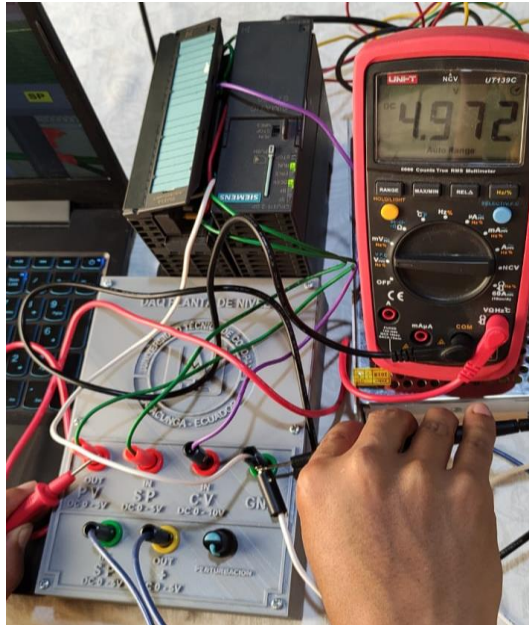


Figura 5.18. Medida de voltaje referencia máxima del tanque.

### 5.13. Visualización de variación del chorro de agua llenado de tanque.

Para mejorar el realismo de nuestro entorno virtual se acondiciono el volumen de chorro de agua del llenado del tanque dependiendo del control que se realiza a la apertura de la válvula de entrada o cargar.

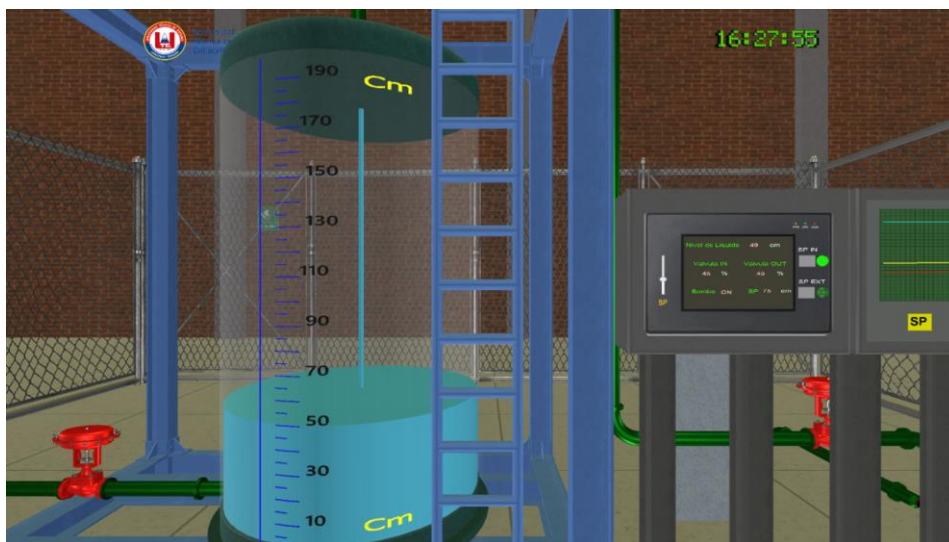
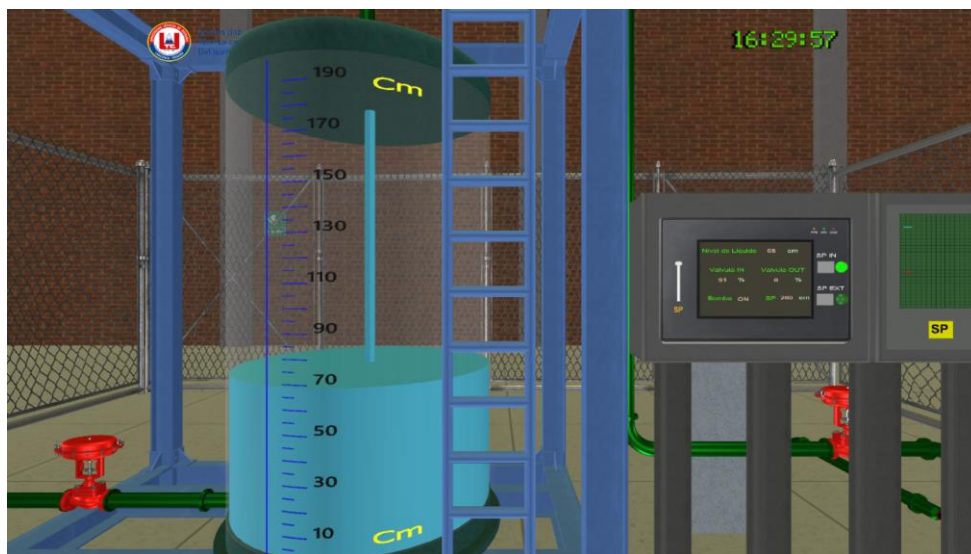


Figura 5.19. Volumen del chorro de llenado de tanque.



**Figura 5.20. Volumen de chorro de llenado del tanque.**

En las figuras anteriores podemos visualizar como el volumen del chorro del agua incrementa esto se debe a la variación de nuestro control PID el cual incrementa la apertura dependiendo de la necesidad de nuestro sistema.

#### **5.14. Módulo para sistema de control**

Empleando la metodología descrita en el capítulo 4, se logra obtener el módulo para sistemas de control finalizado, dando paso a la realización de las pruebas, para verificar su correcto funcionamiento y utilidad. A continuación, se representa el módulo.



**Figura 5.21. Módulo para el sistema de control.**

## 6. PRESUPUESTO Y ANÁLISIS DE IMPACTOS

El objetivo de este capítulo fue ordenar y analizar la información recolectada de carácter monetario en el desarrollo de un sistema didáctico simulado para el aprendizaje de control de procesos electromecánicos para el control de nivel se consideró varios aspectos como, la construcción, costo de material, egresos económicos por el precio reducido, y el análisis de costo y beneficio.

### 6.1. Análisis de Costos

#### Diseño

Se registra con varios controladores externos.

#### Simplicidad

Puede operarla cualquier persona es de simple uso, su mantenimiento es mínimo y las posibilidades de ruptura o fallas son casi nulas.

#### Materiales

- Arduino
- Integrado LM 358 N
- Resistencias
- Pulsadores
- borneras

#### 6.1.1. Factibilidad económica

<b>Mano de Obra</b>				
Descripción	Cantidad (personas)	Horas de trabajo	Costo (USD/Hora)	Valor Total
Programador	1	6	30,00	180
Ayudante	1	6	20,00	120
Imprevistos				20
<b>Total</b>				<b>\$320</b>
<b>Material indirecto</b>				
Descripción	Cantidad	Unida d	Valor Unitario (USD)	Valor Total
Transporte	12	Viajes	5,00	60,00
Imprevistos				10,00
<b>Total</b>				<b>\$70,00</b>

<b>Material Directo</b>				
Descripción	Cantidad	Unidad	Valor unitario	Total
Arduino UNO	2	UN	14,50	29
Resistencias	19	UN	0,20	3,80
Pulsadores	2	UN	0,25	0,50
Bornera	3	UN	0,25	0,75
Integrado LM358	3	UN	0,50	1,50
Impresión de placa	2	UN	30	60
Impresión de carcasa en 3D	2	UN	40	80
Imprevisto		UN		
<b>TOTAL</b>				<b>\$175,55</b>

### Costo total del proyecto

<b>Costo Total del Proyecto</b>	
<b>MD</b>	<b>\$ 175,55</b>
<b>MI</b>	<b>\$ 70</b>
<b>MO</b>	<b>\$ 320</b>
<b>TOTAL</b>	<b>\$ 565,55</b>

#### 6.1.2. Impacto Práctico

El módulo de entorno virtual sirve para realizar pruebas en sistemas de control de nivel en la Carrera de Ingeniería Electromecánica, consta de actuadores con los que se puede receptor la señal y transformarla por medio de una tarjeta de adquisición de datos; esto podría fortalecer la enseñanza teórico-práctica y mejorar los conocimientos adquiridos por los estudiantes.

#### 6.1.3. Impacto tecnológico

La finalidad del módulo se basa en la fomentación de utilizar nuevos equipos y dispositivos tecnológicos, en el que interviene la interacción entre el módulo, el software, el equipo de adquisición de datos, y el estudiante, este se adapta a los requerimientos del docente y estudiante, pues por medio de controladores externos, se puede recibir o emitir señales analógicas y digitales, emitiendo órdenes a actuadores o recibiendo señales de sensores para transformarlas y procesarlas según el requerimiento, incentivando al estudiante al desarrollo de entornos virtuales, que transformaran dichas señales según sea el requerimiento



## **7. CONCLUSIONES Y RECOMENDACIONES**

### **7.1. Conclusiones**

- Mediante la investigación de entornos virtuales acondicionados a procesos industriales nos permitió la selección de equipos, programas necesarios para el desarrollo de nuestro entorno virtual.
- El motor gráfico Unity 3D comprende una amplia gama de funciones que ayudan a acondicionar nuestro entorno virtual de una planta de control de nivel para visualizar su comportamiento en tiempo real.
- Por medio de la tarjeta de adquisición, se recepta o emite las señales, para controlar los actuadores de una forma práctica y eficaz disminuyendo tiempo, cableado, y elementos para su activación o acondicionamiento según se requiera.
- La integración de un control externo permite al usuario interactuar de manera que permite visualizar el comportamiento de nuestra planta de control de nivel, mediante el controlador externo PLC S7 300 que consta de un módulo de entradas y salidas digitales analógicas realizando el control del sistema adecuadamente enviando y recibiendo señales eléctricas de voltaje con un rango de 0 a 5 (V)

### **7.2. Recomendaciones**

- Para la estabilización de velocidad de comunicación se debe mantener la misma velocidad tanto en la tarjeta de adquisición de datos como en el software Unity.
- Se recomienda utilizar software de diseño con licencia gratuita, y en ingeniería, también ayudará a los estudiantes a utilizar varias formas predefinidas para llevar la inteligencia de los dibujos de ingeniería en 3D, para modelar diferentes tipos de procesos industriales en 3D.
- Con este módulo se puede seguir escalando la pirámide de automatización ya que es un sistema abierto adaptable a diferentes tipos de control.

**8. REFERENCIAS**

- [1] A. José, “Desarrollo de un entorno de realidad virtual 3d con capacidad de inmersión que emule el lanzamiento y recepción de un raspador de tubería (PIG), orientado al entrenamiento dentro de la industria petrolera”, *Desarro. Un Entorno Real. Virtual 3D Con Capacid. Inmersión Que Emule El Lanzamiento Y Recepción Un Raspador Tubería (Pig), Orientado Al Entren. Dentro La Ind. Pet.*, p. 244, 2018.
- [2] R. alexander Rosero Hidalgo, Pogo Mauricio, “Departamento de eléctrica y electrónica”, *Sist. Control en Cascada la Var. Niv. para el Aprendiz. Control procesos*, 2018.
- [3] J. C. Joo Nagata, “Modelo De Realidad Aumentada Y Navegación Peatonal Del Patrimonio Territorial : Diseño , Implementación Y Evaluación Educativa”, p. 475, 2016.
- [4] J. S. Benavides Cárdenas, “Desarrollo de una aplicación de realidad virtual Semi-inmersiva para terapia Recreacional”. 2019.
- [5] M. A. Abidi, B. Lyonnet, P. Chevaillier, R. Toscano, y P. Baert, *Simulation of manufacturing processes via Virtual Reality*, vol. 2, núm. March. 2018.
- [6] A. C. Silva, A. Cardoso, E. A. Lamounier Junior, y C. L. Barreto Junior, “Virtual reality for monitor and control of electrical substations”, *An. Acad. Bras. Cienc.*, vol. 93, núm. 1, pp. 1–17, 2021.
- [7] W. X. QUEVEDO PÉREZ, “Implementar Un Esquema De Tele-Operación Para Un Robot Manipulador Móvil Maniobrado a Través De Dispositivos Hápticos, Para Incrementar La Transparencia Del Sitio Remoto, a Través De Entornos De Realidad Virtual Y Realidad Aumentada, En El Sitio Local”, p. 165, 2016.
- [8] V. H. Andaluz *et al.*, “Unity3D virtual animation of robots with coupled and uncoupled mechanism”, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9768, núm. April 2017, pp. 89–101, 2016.
- [9] N. Narayana, “Public Perceptions Regarding Use of Virtual Reality in Health Care: A Social Media Content Analysis Using Facebook. *Journal of Medical Internet Research*”, 2017.

- [10] Sánchez, “Técnicas de la Inteligencia artificial aplicadas a la educación”, *Revista iberoamericana de inteligencia artificial*, 2016.
- [11] S. E. digital SAS., “Dispositivos para la realidad virtual que se pueden utilizar.”, 2015.
- [12] M. Kasim, “Introduction tu virtual Reality Technology”, vol. 1, p. 22, 2017.
- [13] J. C. Martinez y J. Andrade, “Implementación de controladores en Sistemas Retroalimentados usando electrónica embebida y simulación Hardware In The Loop”, p. 114, 2013.
- [14] C. P. de grado I. en Sistemas, “Simulador Sistema CAD”, 2016.
- [15] C. Morcillo, “Diseño Gráfico 3Dcon software libre.”, 2017.
- [16] C. González, “Fundamentos de Síntesis de Imagen 3D”, 2017.
- [17] C. Delgado, “Desarrollo de un personaje animado 2D a 3D.”, 2017.
- [18] Hernán., “Fundamentos Básicos del modelado 3D”, 2017.
- [19] M. Fernandez, “Modelado, texturizado y ajuste de malla. Madrid”, 2016.
- [20] Blender org, “wiki.Blender.org”, *¿Qué es el mapeo UV en el Model. 3D y cuál es su propósito?*, 2017.
- [21] I. Zamojc, “Introduction to Unity3D”, 2015.
- [22] Unity 3D, “El motor de creación del contenido líder en el mundo”, 2018.
- [23] I. Training, “Pirámide de automatización”, 2020.
- [24] C. Smith, “Control Automático de Procesos Teoría Y Práctica”, 2017.
- [25] C. D. E. I. Informática, “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO (DOMSYSTEM) DE SEGURIDAD Y CONTROL PARA MANTENER EL RESGUARDO DE BIENES Y EL CONFORT MEDIANTE UNA RED DE SENSORES UTILIZANDO COMUNICACIÓN WIRELESS BLUETOOTH”, 2015.
- [26] L. Diéguez, “COMUNICACIONES SERIE EN ARDUINO: UART, I2C Y SPI”, 2019.
- [27] R. S. Castro Rios, “Estudio E Implementación De Un Convertidor Analógico Digital Y Digital Analógico Sigma Delta”, núm. October, p. 223, 2015.
- [28] T. A. Oscar, *ARDUINO Curso práctico de formación*, Primera., vol. 6, núm. 06. 2016.

- [29] M. D. Fatriani, “SISTEMA DE MONITOREO EN LÍNEA DE LAS VARIABLES DE CONTROL: ULTRASONIDO, TEMPERATURA, AIRE Y CAUDAL DE AGUA DE LOS MOTORES DC DEL TREN ACABADOR DEL PROCESO DE LAMINACIÓN EN CALIENTE EN ACERÍAS NACIONALES DEL ECUADOR”, *Keefektifan Penerapan Pendekatan Keterampilan Proses dalam Pembelajaran IPA terhadap Minat Belajar Siswa Kelas IV SD Negeri Golo Yogyakarta*, vol. 16, núm. 2, pp. 39–55, 2015.
- [30] O. Ingenier, F. Torres, y C. Jara, “Tema 6. Autómatas programables I.”
- [31] C. Smith and A. Corripio, *Control automático de procesos: Teoría y práctica*. 1991.
- [32] K. Ogata, “Modern Control Engineering”, *Sist. Control en tiempo Discret.*, 2018.
- [32] T. H. KJ Astrom, “Control PID avanzado.”, *Metod. Model.*, 2018.

# 9. ANEXOS



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO.Ports;
using System.Threading;
using System;
using UnityStandardAssets.CrossPlatformInput;

public class Sis22 : MonoBehaviour
{
    // Start is called before the first frame update
    public GameObject nivel;
    public GameObject clip;
    public GameObject DesfogueSonido;
    public string CV;
    public string D;

    //SerialPort serialport = new SerialPort("COM24",19200);

    public string DatosTX, DatosRX;
    public string SP;
    public string PV;
    public Text textoNivel;
    public Text ValvulaOU;
    public Text ValvulaINP;
    public Text EstBomba;
    public string PuertoCom;
    public string setpoint, valulaINput,valulaOUTPUT;
    public Conexion Conexion;

    SerialPort puerto;

    public int estado=0,hj1=0,hj2=0;

    public float CV1;
    public float D1;

    public double a1n, a2n;

    public float h, dh, k1, a1, k2, a2, g, A, to,mn;
    public int nivelAgua = 0, se = 0, timeN, Nant, Nact, EstadoAnterior = 0;
    public string Puesto;
    public int contSuma=0;

    public int hn = 0, hn1 = 0;
    public GameObject Chorro;

    public void Arduino()
    {
        while (true)
        {
```



```
// puerto.Write(DatosTX);
// DatosRX = puerto.ReadLine();
}
}

void Start()
{
    LoadData();
    //clip = GameObject.FindGameObjectWithTag("clip_audio");
    // DesfogueSonido = GameObject.FindGameObjectWithTag("clip_audio1");
    k1 = 0.05f;
    k2 = 0.015f;
    g = 9.8f;
    A = 0.5f;
    to = 0.02f;
    clip.GetComponent<AudioSource>().Stop();
    puerto = new SerialPort(PuertoCom, 19200);
    puerto.Open();
    puerto.ReadTimeout = 80;
    puerto.WriteTimeout = 80;
    Thread sampleThread = new Thread(new ThreadStart(Arduino));
    sampleThread.IsBackground = true;
    sampleThread.Start();
    Chorro.gameObject.SetActive(false);
}

void Update()
{
    //puerto.BaseStream.Flush();
    try
    {
        DatosRX = puerto.ReadLine();
        string[] datosRecibidos = DatosRX.Split("a"[0]);
//Trama de datos recibidos
        CV = datosRecibidos[0].Replace(".", ",");
        D = datosRecibidos[1].Replace(".", ",");
    }
    catch (System.Exception ex)
    {
        ex = new System.Exception();
    }
}

// Update is called once per frame
void FixedUpdate()
{
    contSuma = contSuma + 1;
    if (contSuma>1000) {
    if (estado == 0) {
        estado = 1;
    }
}
}
```



```
if (puerto.IsOpen)
{
    if (EstadoAnterior != nivelAgua) {
        puerto.Write(nivelAgua + "," + 3 + "," + '\n');
        //print("hola como estas");
    }
    EstadoAnterior = nivelAgua;
}
DatosTX = "a" + SP + "a" + PV; //Trama de datos para enviar
float.TryParse(CV, out CV1);
float.TryParse(D, out D1);
//a1n = Math.Tanh((double)CV1);
//a2n = Math.Tanh((double)D1);
a1 = CV1;
a2 = D1;
print(a1);
print(a2);
ValvulaINP.text = ((int)(a1*20)).ToString();
ValvulaOU.text = ((int)(a2*50)).ToString();
if (a1 > 0)
{
    if (hn == 0)
    {
        EstBomba.text = "ON";
        clip.GetComponent<AudioSource>().Play();
        DesfogueSonido.GetComponent<AudioSource>().Stop();
        Chorro.gameObject.SetActive(true);
        hn = 1;
        hn1 = 0;
    }
}
else {
    EstBomba.text = "OFF";
}
if (a2 > 0)
{
    if (hn1 == 0)
    {
        clip.GetComponent<AudioSource>().Stop();
        DesfogueSonido.GetComponent<AudioSource>().Play();
        Chorro.gameObject.SetActive(false);
        hn1 = 1;
        hn = 0;
    }
}

if (((a1 == 0) && (a2 == 0)) || (h==0)) {
    clip.GetComponent<AudioSource>().Stop();
    DesfogueSonido.GetComponent<AudioSource>().Stop();
}
```





```
        hn = 0;
        hn1 = 0;
        Chorro.gameObject.SetActive(false);
    }

    dh = (k1 * a1 - k2 * a2 * Mathf.Sqrt(2 * g * h));
    h = h + dh * to;
    if (h <= 0)
    {
        h = 0;
        Nant = 0;
    }
    else if (h >= 10) {
        Nant = 10;
        h = 10;
    }
    nivel.transform.localScale = new Vector3(0.112f, 0.112f, ((0.16f - 0.001f) / 10)
* h + 0.001f);

    to = Time.deltaTime; //0.02
    mn = h * 20.0f;
    // print(mn);
    nivelAgua = (int)mn;
    textoNivel.text = nivelAgua.ToString();
    setpoint= nivelAgua.ToString(); ;
se = se + 1;
if (System.DateTime.Now.Second != timeN)
{
    Nact = nivelAgua - Nant;
    //print(Nact);
    se = 0;
    Nant = nivelAgua;
}
timeN = System.DateTime.Now.Second;

if ((h > 0) && (h < 0.005f)&&(a1>0))
{
    //clip.GetComponent<AudioSource>().Play();
}
else if (h == 10)
{
    clip.GetComponent<AudioSource>().Stop();
}
else if (h == 0) {
    clip.GetComponent<AudioSource>().Stop();
}
}
private void LoadData()
{
    PuertoCom = PlayerPrefs.GetString(Conexion.Com);
}
}
```



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.Threading;
using System;
using UnityStandardAssets.CrossPlatformInput;
using UnityEngine.SceneManagement;
public class MenuPrincipal : MonoBehaviour
{
    public static bool gameP;
    public static bool boolseguroP;

    public GameObject menuP, seguroP;
    //public double a1, a2;
    //public float hp = 0, h = 0, k1, k2, g, A, to;
    //public GameObject nivel;
    public float CV1;
    void Update()
    {
        if (Input.GetKeyUp(KeyCode.Escape))
        {
            SceneManager.LoadScene("Menu");
            Cursor.lockState = CursorLockMode.None;
            Cursor.visible = true;
            //SceneManager.LoadScene("Conexion");
        }
    }

    public void IniciarJuego(string nombre) {
        SceneManager.LoadScene(nombre);
    }

    public void CerrarJuego() {
        Application.Quit();
        Debug.Log("Salir");
    }

    public void SwitchPause() {
        if (gameP)
        {
            bntResume();
        }
        else
        {
            bntPause();
        }
    }

    void bntResume() {
        menuP.SetActive(false);
        Time.timeScale = 0;
        gameP = true;
    }
}
```

```
}  
    void bntPause() {  
menuP.SetActive(false);  
    Time.timeScale = 0;  
    gameP = true;  
    }  
    public void mPrincipal(string name)  
    {  
        SceneManager.LoadScene(name);  
    }  
    public void pane12() {  
        seguroP.SetActive(false);  
    }  
    void Start()  
    {  
    }  
}
```





```
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.Threading;
using System;
using System.IO.Ports;

public class Conexion : MonoBehaviour
{
    public string serialport;
    public InputField pserialport;

    public Sis22 Sis22;
    public bool tecla = false;
    //public SerialPort serialport = new SerialPort("COM24",19200);
    public int cont=0;
    public int esAnt = 0;

    public Text EstadoConectado;

    SerialPort puerto;
    public string DatosTX, DatosRX;

    public string Com;

    public string a, b, c;

    void PP(string porr)
    {
        serialport = porr;
    }

    public static Conexion conexion;

    private void Awake()
    {
        if (conexion == null) {
            conexion = this;
            DontDestroyOnLoad(gameObject);
        } else if (conexion!=this) {
            Destroy(gameObject);
        }
    }

    public void opciones() {
```



```
/*
    if (puerto.IsOpen)
    {
        EstadoConectado.text = "CONECTADO";
    }
    else
    {
        EstadoConectado.text = "DESCONECTADO";
    }
*/
cont = 0;
}

void Start()
{
    pserialport.onEndEdit.AddListener(PP);
}

public void enlazar()
{
    cont = cont + 1;
    if (cont==1) {
        SaveData();
        puerto = new SerialPort(serialport, 19200);
        puerto.Close();
        puerto.Open();
        puerto.ReadTimeout = 80;
        puerto.WriteTimeout = 80;

    }
    if (cont>=2) {
        puerto.Close();
        puerto.ReadTimeout = 0;
        puerto.WriteTimeout = 0;
        EstadoConectado.text = "DESCONECTADO";
        print("Desconectado");
        cont = 0;
    }
    print(cont);
}
```

```
if (puerto.IsOpen)
{
    EstadoConectado.text = "CONECTADO";
}
else
{
    EstadoConectado.text = "DESCONECTADO";
}

}

public void salirJuego() {
    puerto.Close();
}

private void SaveData() {
    PlayerPrefs.SetString(Com, serialport);
}
}
```





```

using System;
using UnityEngine;
using UnityStandardAssets.CrossPlatformInput;
using UnityStandardAssets.Utility;
using Random = UnityEngine.Random;
namespace UnityStandardAssets.Characters.FirstPerson
{
    [RequireComponent(typeof (CharacterController))]
    [RequireComponent(typeof (AudioSource))]
    public class FirstPersonController : MonoBehaviour
    {
        [SerializeField] private bool m_IsWalking;
        [SerializeField] private float m_WalkSpeed;
        [SerializeField] private float m_RunSpeed;
        [SerializeField] [Range(0f, 1f)] private float m_RunstepLenghten;
        [SerializeField] private float m_JumpSpeed;
        [SerializeField] private float m_StickToGroundForce;
        [SerializeField] private float m_GravityMultiplier;
        [SerializeField] private MouseLook m_MouseLook;
        [SerializeField] private bool m_UseFovKick;
        [SerializeField] private FOVKick m_FovKick = new FOVKick();
        [SerializeField] private bool m_UseHeadBob;
        [SerializeField] private CurveControlledBob m_HeadBob = new
CurveControlledBob();
        [SerializeField] private LerpControlledBob m_JumpBob = new LerpControlledBob();
        [SerializeField] private float m_StepInterval;
        [SerializeField] private AudioClip[] m_FootstepSounds;    // an array of
footstep sounds that will be randomly selected from.
        [SerializeField] private AudioClip m_JumpSound;           // the sound played
when character leaves the ground.
        [SerializeField] private AudioClip m_LandSound;           // the sound played
when character touches back on ground.

        private Camera m_Camera;
        private bool m_Jump;
        private float m_YRotation;
        private Vector2 m_Input;
        private Vector3 m_MoveDir = Vector3.zero;
        private CharacterController m_CharacterController;
        private CollisionFlags m_CollisionFlags;
        private bool m_PreviouslyGrounded;
        private Vector3 m_OriginalCameraPosition;
        private float m_StepCycle;
        private float m_NextStep;
        private bool m_Jumping;
        private AudioSource m_AudioSource;

        // Use this for initialization
        private void Start()
        {
            m_CharacterController = GetComponent<CharacterController>();
            m_Camera = Camera.main;
            m_OriginalCameraPosition = m_Camera.transform.localPosition;
            m_FovKick.Setup(m_Camera);

```



```
m_HeadBob.Setup(m_Camera, m_StepInterval);
    m_StepCycle = 0f;
    m_NextStep = m_StepCycle/2f;
    m_Jumping = false;
    m_AudioSource = GetComponent<AudioSource>();
    m_MouseLook.Init(transform , m_Camera.transform);
}

// Update is called once per frame
private void Update()
{
    RotateView();
    // the jump state needs to read here to make sure it is not missed
    if (!m_Jump)
    {
        m_Jump = CrossPlatformInputManager.GetButtonDown("Jump");
    }

    if (!m_PreviouslyGrounded && m_CharacterController.isGrounded)
    {
        StartCoroutine(m_JumpBob.DoBobCycle());
        PlayLandingSound();
        m_MoveDir.y = 0f;
        m_Jumping = false;
    }
    if (!m_CharacterController.isGrounded && !m_Jumping && m_PreviouslyGrounded)
    {
        m_MoveDir.y = 0f;
    }

    m_PreviouslyGrounded = m_CharacterController.isGrounded;
}

private void PlayLandingSound()
{
    m_AudioSource.clip = m_LandSound;
    m_AudioSource.Play();
    m_NextStep = m_StepCycle + .5f;
}

private void FixedUpdate()
{
    float speed;
    GetInput(out speed);
    // always move along the camera forward as it is the direction that it being
    aimed at
    Vector3 desiredMove = transform.forward*m_Input.y +
transform.right*m_Input.x;
```





```
// get a normal for the surface that is being touched to move along it
RaycastHit hitInfo;
Physics.SphereCast(transform.position, m_CharacterController.radius,
Vector3.down, out hitInfo,
                    m_CharacterController.height/2f, Physics.AllLayers,
QueryTriggerInteraction.Ignore);
desiredMove = Vector3.ProjectOnPlane(desiredMove,
hitInfo.normal).normalized;

m_MoveDir.x = desiredMove.x*speed;
m_MoveDir.z = desiredMove.z*speed;

if (m_CharacterController.isGrounded)
{
    m_MoveDir.y = -m_StickToGroundForce;

    if (m_Jump)
    {
        m_MoveDir.y = m_JumpSpeed;
        PlayJumpSound();
        m_Jump = false;
        m_Jumping = true;
    }
}
else
{
    m_MoveDir += Physics.gravity*m_GravityMultiplier*Time.fixedDeltaTime;
}
m_CollisionFlags =
m_CharacterController.Move(m_MoveDir*Time.fixedDeltaTime);

ProgressStepCycle(speed);
UpdateCameraPosition(speed);

m_MouseLook.UpdateCursorLock();
}

private void PlayJumpSound()
{
    m_AudioSource.clip = m_JumpSound;
    m_AudioSource.Play();
}

private void ProgressStepCycle(float speed)
{
    if (m_CharacterController.velocity.sqrMagnitude > 0 && (m_Input.x != 0 ||
m_Input.y != 0))
    {
        m_StepCycle += (m_CharacterController.velocity.magnitude +
(speed*(m_IsWalking ? 1f : m_RunstepLenghten)))*
Time.fixedDeltaTime;
    }
}
```



```
if (!(m_StepCycle > m_NextStep))
{
    return;
}

m_NextStep = m_StepCycle + m_StepInterval;

PlayFootStepAudio();
}

private void PlayFootStepAudio()
{
    if (!m_CharacterController.isGrounded)
    {
        return;
    }
    // pick & play a random footstep sound from the array,
    // excluding sound at index 0
    int n = Random.Range(1, m_FootstepSounds.Length);
    m_AudioSource.clip = m_FootstepSounds[n];
    m_AudioSource.PlayOneShot(m_AudioSource.clip);
    // move picked sound to index 0 so it's not picked next time
    m_FootstepSounds[n] = m_FootstepSounds[0];
    m_FootstepSounds[0] = m_AudioSource.clip;
}

private void UpdateCameraPosition(float speed)
{
    Vector3 newCameraPosition;
    if (!m_UseHeadBob)
    {
        return;
    }
    if (m_CharacterController.velocity.magnitude > 0 &&
m_CharacterController.isGrounded)
    {
        m_Camera.transform.localPosition =
            m_HeadBob.DoHeadBob(m_CharacterController.velocity.magnitude +
                (speed*(m_IsWalking ? 1f : m_RunstepLenghten)));
        newCameraPosition = m_Camera.transform.localPosition;
        newCameraPosition.y = m_Camera.transform.localPosition.y -
m_JumpBob.Offset();
    }
    else
    {
        newCameraPosition = m_Camera.transform.localPosition;
        newCameraPosition.y = m_OriginalCameraPosition.y - m_JumpBob.Offset();
    }
    m_Camera.transform.localPosition = newCameraPosition;
}

private void GetInput(out float speed)
{

```



```

// Read input
float horizontal = CrossPlatformInputManager.GetAxis("Horizontal");
float vertical = CrossPlatformInputManager.GetAxis("Vertical");

bool waswalking = m_IsWalking;
#if !MOBILE_INPUT
// On standalone builds, walk/run speed is modified by a key press.
// keep track of whether or not the character is walking or running
m_IsWalking = !Input.GetKey(KeyCode.LeftShift);
#endif

// set the desired speed to be walking or running
speed = m_IsWalking ? m_WalkSpeed : m_RunSpeed;
m_Input = new Vector2(horizontal, vertical);

// normalize input if it exceeds 1 in combined length:
if (m_Input.sqrMagnitude > 1)
{
    m_Input.Normalize();
}

// handle speed change to give an fov kick
// only if the player is going to a run, is running and the fovkick is to be
used
if (m_IsWalking != waswalking && m_UseFovKick &&
m_CharacterController.velocity.sqrMagnitude > 0)
{
    StopAllCoroutines();
    StartCoroutine(!m_IsWalking ? m_FovKick.FOVKickUp() :
m_FovKick.FOVKickDown());
}

private void RotateView()
{
    m_MouseLook.LookRotation (transform, m_Camera.transform);
}

private void OnControllerColliderHit(ControllerColliderHit hit)
{
    Rigidbody body = hit.collider.attachedRigidbody;
    //dont move the rigidbody if the character is on top of it
    if (m_CollisionFlags == CollisionFlags.Below)
    {
        return;
    }

    if (body == null || body.isKinematic)
    {
        return;
    }
    body.AddForceAtPosition(m_CharacterController.velocity*0.1f, hit.point,
ForceMode.Impulse);
}
}

```





```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class ControlAvatar : MonoBehaviour
{
    public float velocidadMovimiento = 5.0f;
    public float velocidadRotacion = 200.0f;
    private Animator anim;
    public float x, y;

    public GameObject AudioCaminar;
    public GameObject AudioInicio;

    public int ln=0,ln1=0;
    void Start()
    {
        anim = GetComponent<Animator>();
        Cursor.lockState = CursorLockMode.None;
        Cursor.visible = false;
    }
    // Update is called once per frame
    void Update()
    {
        x = Input.GetAxis("Horizontal");
        y = Input.GetAxis("Vertical");
        transform.Rotate(0, x * Time.deltaTime * velocidadRotacion, 0);
        transform.Translate(0, 0, y * Time.deltaTime * velocidadMovimiento);

        anim.SetFloat("VelX", x);
        anim.SetFloat("VelY", y);
        if (y == 1)
        {
            if (ln == 0)
            {
                AudioCaminar.GetComponent<AudioSource>().Play();
                ln = 1;
            }
        }
        else if (y == 0) {
            AudioCaminar.GetComponent<AudioSource>().Stop();
            ln = 0;
        }
        if (y == -1)
        {
            if (ln == 0)
            {
                AudioCaminar.GetComponent<AudioSource>().Play();
                ln = 1;
            }
        }
        else if (y == 0)
        {
            AudioCaminar.GetComponent<AudioSource>().Stop();
            ln = 0;
        }
    }
}
```



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class RelogTiempoReal : MonoBehaviour
{
    public Text TiempoActual1;
    public Text TiempoActual2;

    void Update()
    {
        TiempoActual1.text =
        System.DateTime.Now.Hour.ToString("00") + ":" +
        System.DateTime.Now.Minute.ToString("00") + ":" +
        System.DateTime.Now.Second.ToString("00");
        //System.DateTime.Now.tous.ToString("00");

        TiempoActual2.text =
        System.DateTime.Now.Hour.ToString("00") + ":" +
        System.DateTime.Now.Minute.ToString("00") + ":" +
        System.DateTime.Now.Second.ToString("00");

    }
}

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Fecha : MonoBehaviour
{
    public Text TiempoActual;

    void Update()
    {
        TiempoActual.text =
        System.DateTime.Now.Year.ToString("00") + "/" +
        System.DateTime.Now.Month.ToString("00") + "/" +
        System.DateTime.Now.Day.ToString("00");

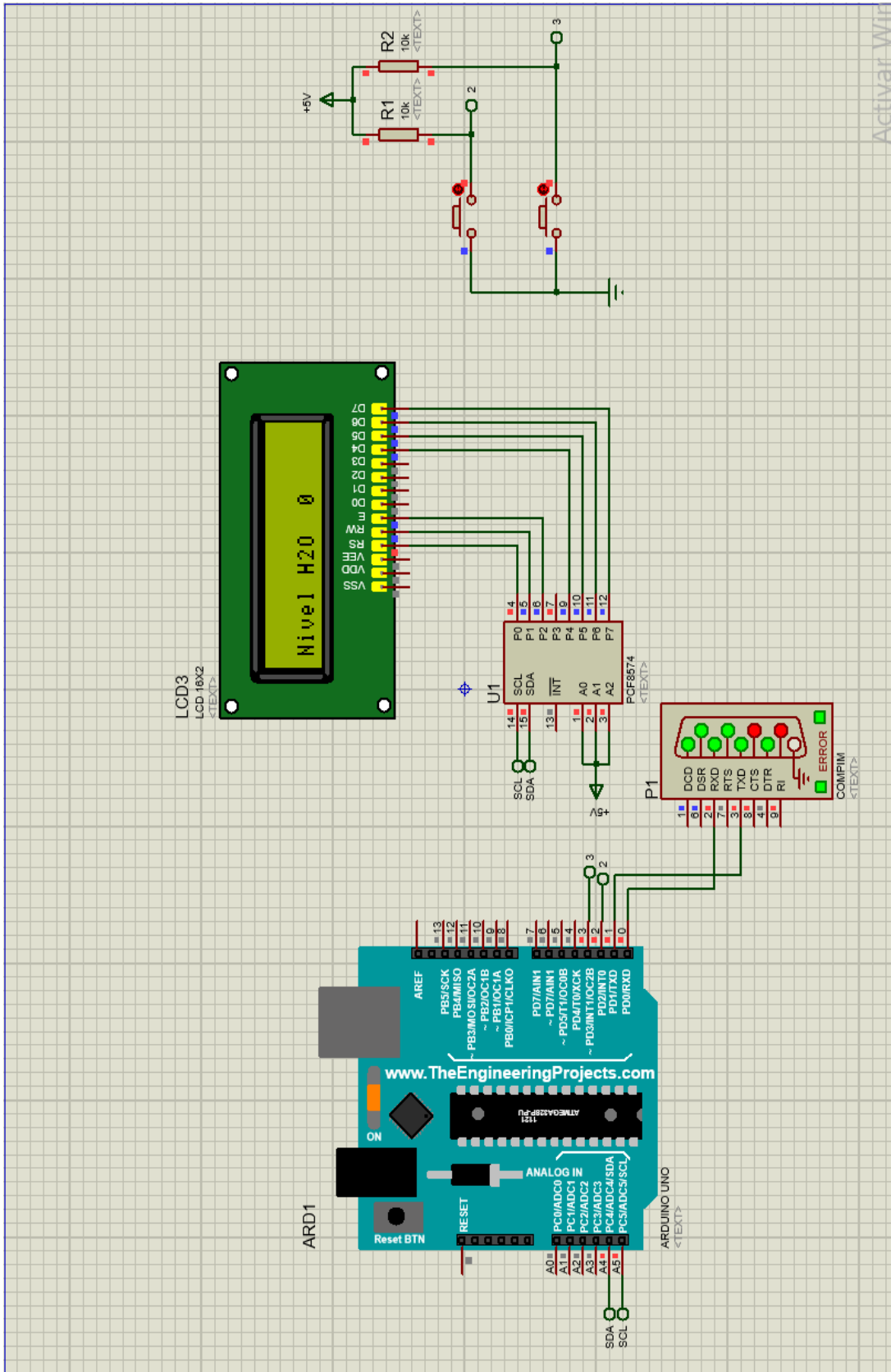
    }
}
```

**ANEXO 8****Código primera prueba de control del sistema simulado**

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
String str = "";
const char separator = ',';
const int dataLength = 4;
int data[] = {0, 0, };
int datahola = 0;
String nomData = "";
int a = 0, b = 0;
int pul1 = 2, pul2 = 3;
int dataA = 0, dataB = 0;
void setup() {
  Serial.begin(19200);
  pinMode(pul1, INPUT);
  pinMode(pul2, INPUT);
  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
}
long t1 = 0, t2 = 0, t3 = 3;
void loop() {
  int m = digitalRead(pul1);
  int n = digitalRead(pul2);

  if (m == LOW) {
    dataA = 1;
    dataB = 0;
  }
  if (n == LOW) {
    dataB = 1;
    dataA = 0;
  }
  t1 = millis();
  if (t1 > (t2 + 300)) {
    lcd.clear();

    if (dataA == 1) {
      lcd.setCursor(2, 0);
      lcd.print("Llenando");
    }
    if (dataB == 1) {
      lcd.setCursor(2, 0);
      lcd.print("Vacizando");
    }
  }
}
```







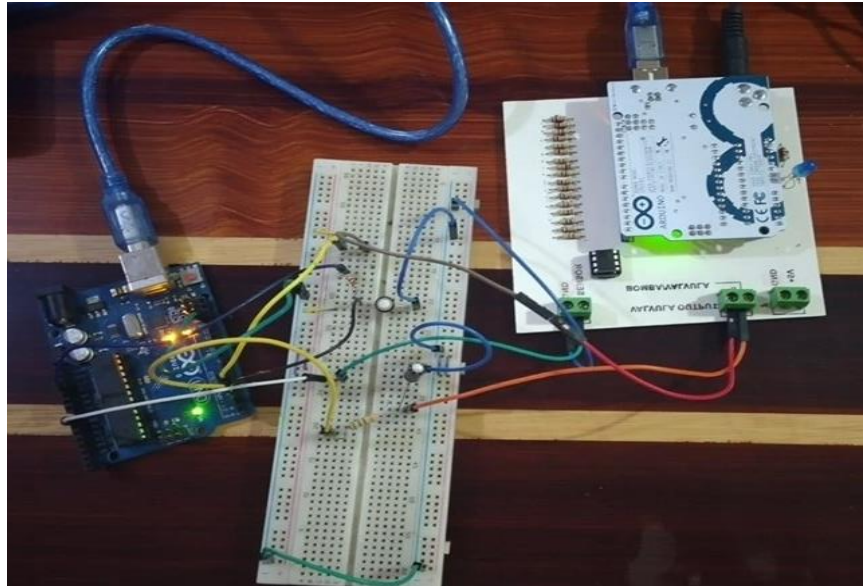
```
char c;
int Salida = 3;
int sensor = A0;
int Valvula = 4;
int pul = 2;
void setup() {
  Serial.begin(9600);
  pinMode(Salida, OUTPUT);
  pinMode(Valvula, OUTPUT);
  pinMode(pul, INPUT);
}
int pwmn = 0;
float dataSensor = 0, setpoint = 100;
void loop() {
  int am = analogRead(sensor);
  float a1 = am;
  float a2 = (a1 * 5) / 1024;
  dataSensor = a2;

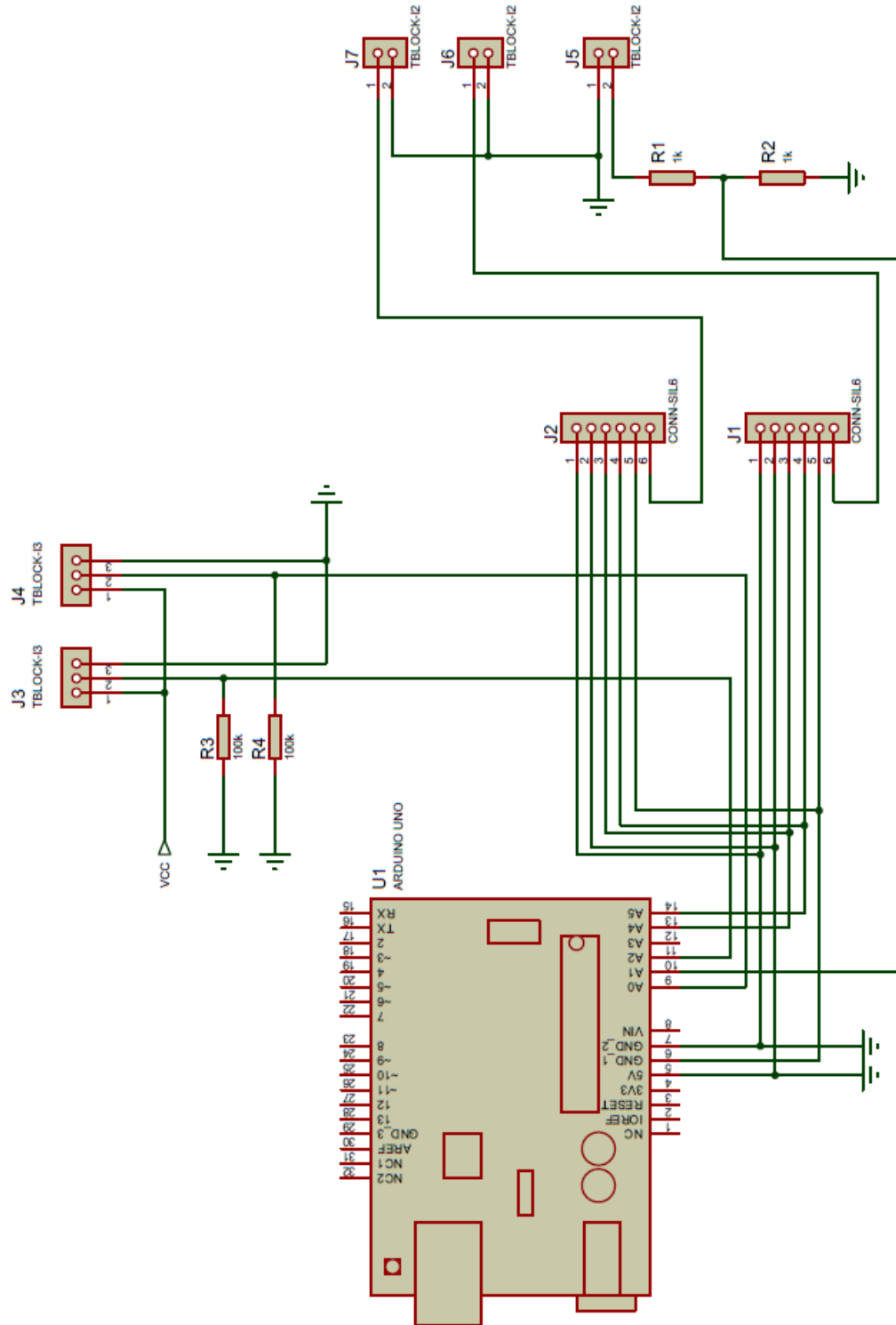
  if (digitalRead(pul) == HIGH) {
    digitalWrite(Valvula, LOW);
    if (dataSensor < ((setpoint / 40) + 0.12) )
    {
      pwmn = 255;
    } else {
      pwmn = 0;
    }
  } else {
    digitalWrite(Valvula, HIGH);
  }
  Serial.print(setpoint);
  Serial.print(" , ");
  Serial.print(dataSensor * 38.16);
  Serial.print(" , ");
  Serial.println(dataSensor);

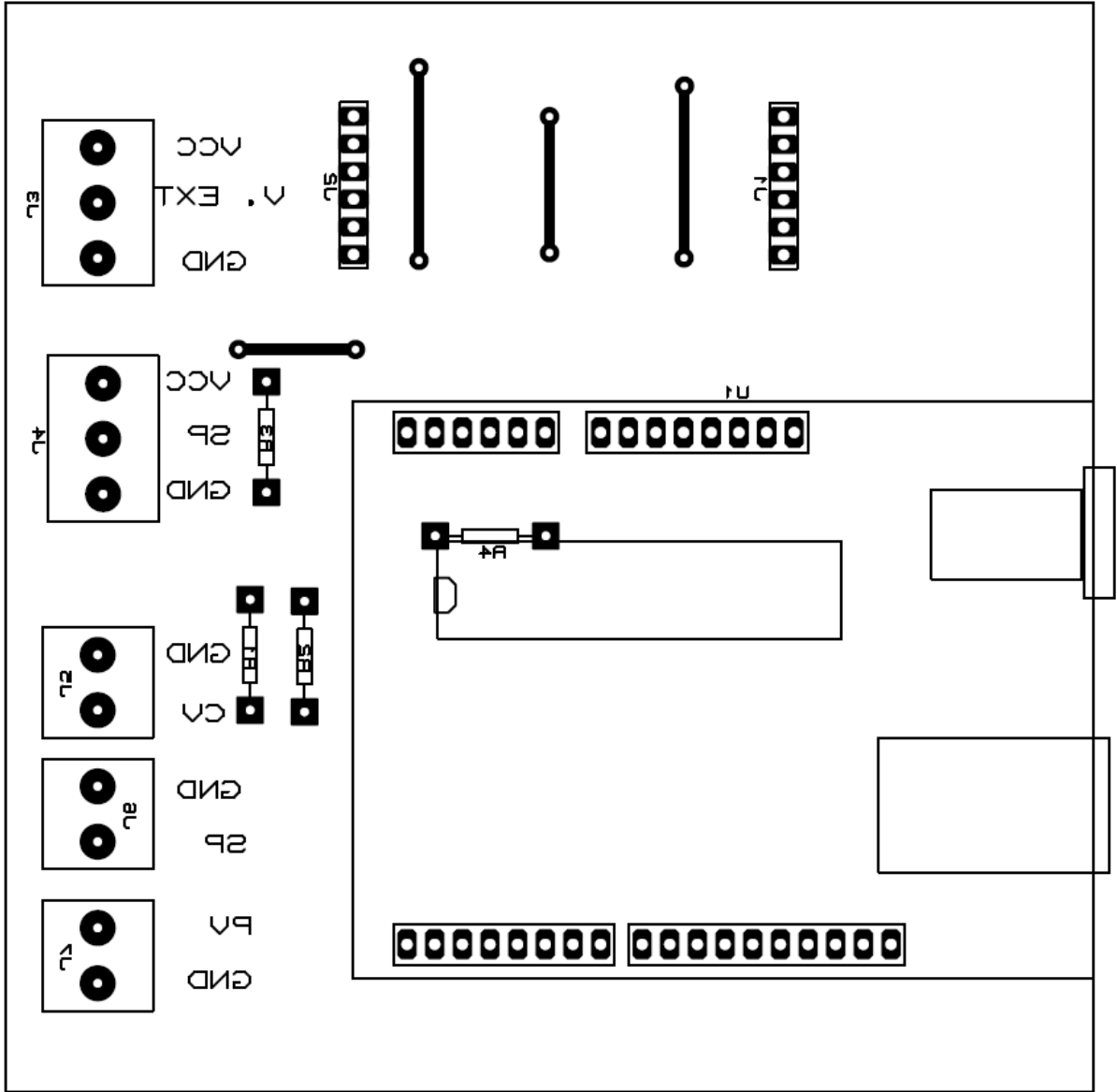
  //Serial.println(cv);

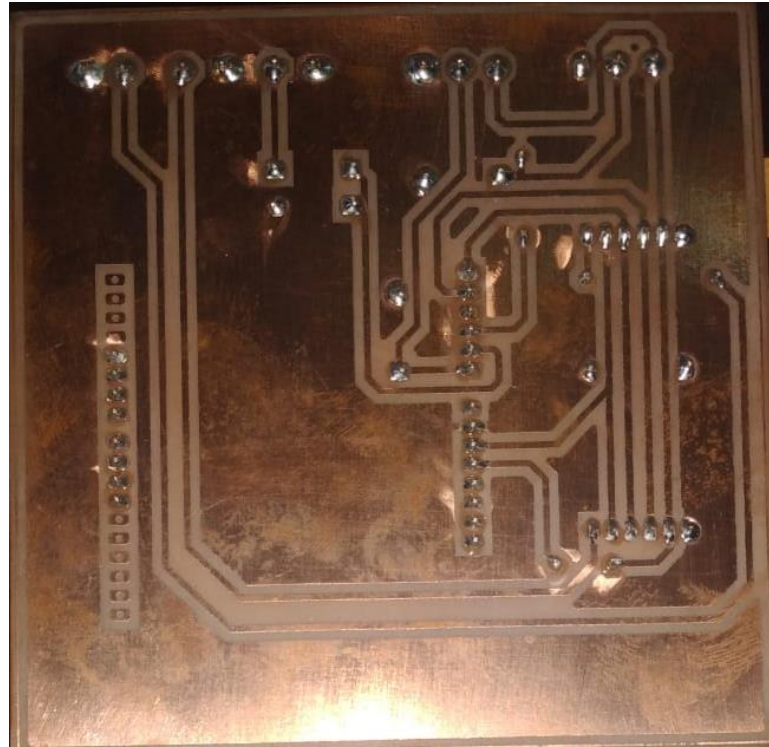
  analogWrite(Salida, pwmn);
  delay(20);
}
```

```
void setup() {  
  Serial.begin(9600);  
  pinMode(Salida, OUTPUT);  
  pinMode(Valvula, OUTPUT);  
  pinMode(pul, INPUT);  
}  
  
int pwmn = 0;  
float dataSensor = 0, setpoint = 50;  
  
void loop() {  
  int am = analogRead(sensor);  
  float a1 = am;  
  float a2 = (a1 * 5) / 1024;  
  dataSensor = a2;  
  
  El Sketch usa 4302 bytes (13%) del espacio de memoria.
```

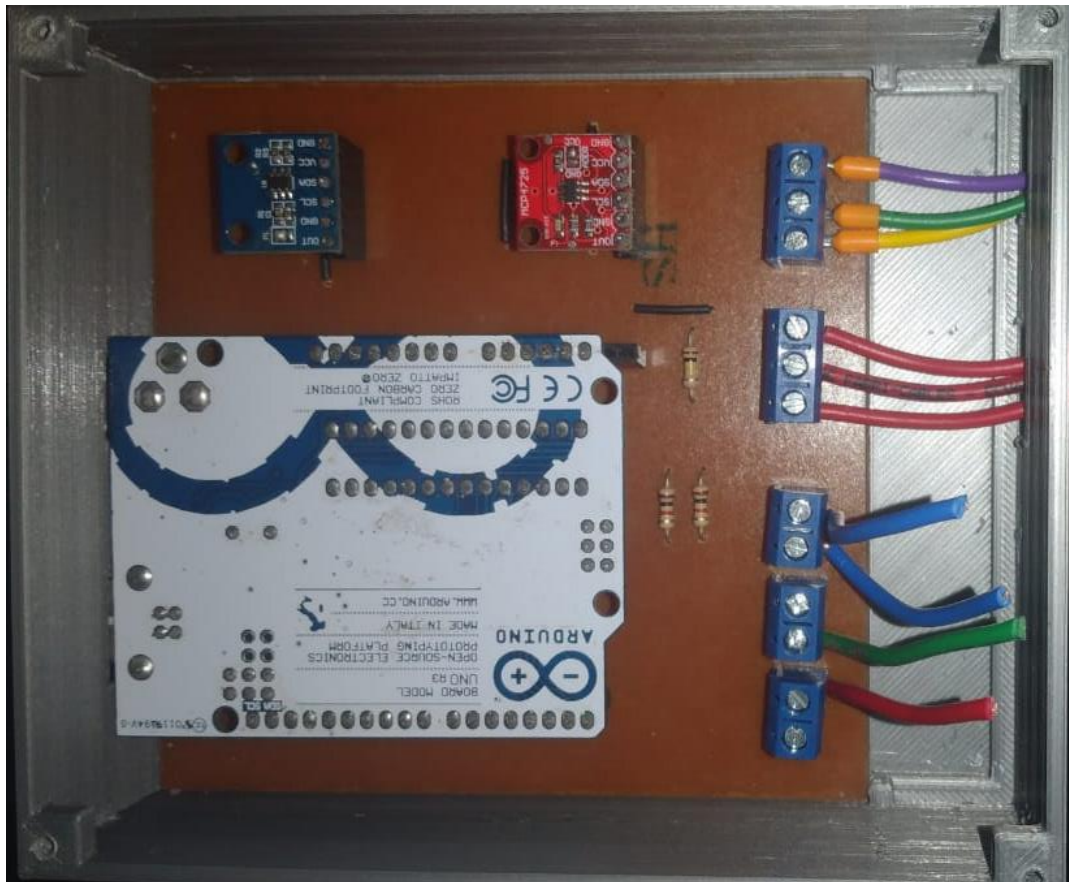
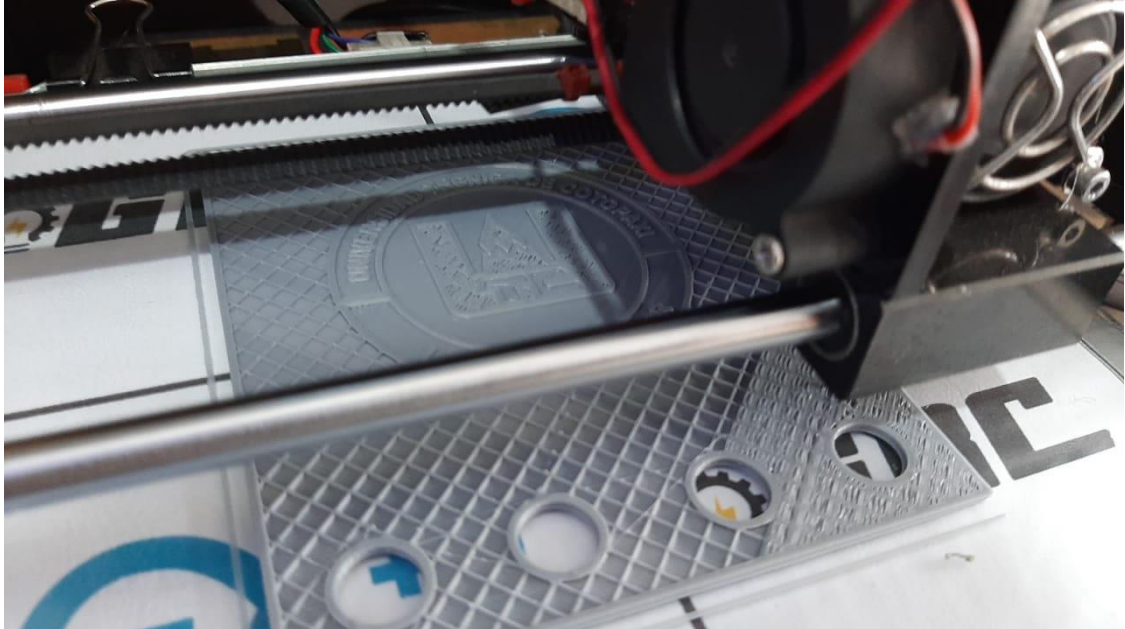












# UNIVERSIDAD TÉCNICA DE COTOPAXI

## INGENIERÍA ELECTROMECAÁNICA

### Manual de usuario

### Tarjeta de adquisición de datos



## 1. Introducción

El módulo para sistemas de control de nivel es un equipo electrónico que se incorpora con controladores externos, dentro del mismo cuenta de señales analógicas y digitales, además tiene componentes electrónicos.

## 2. Nota sobre las instrucciones

Lea este manual y siga todas las instrucciones mencionadas a continuación. De esta manera evite daños del equipo, personales y garantice un servicio y la larga vida útil del equipo.

Guarde este manual con cuidado para el uso del equipo.

En caso de entregarse el equipo a terceros, entregue también este manual.

Tenga en cuenta todas las indicaciones facilitadas en este manual. Contiene información básica importante para la instalación, el funcionamiento y el mantenimiento del equipo.

Por este motivo es imprescindible que tanto el técnico de instalación como los operarios y personal responsable lo lean antes de realizar trabajos de instalación, puesta en marcha y mantenimiento.

## 3. Seguridad



**¡Daño de componentes!**

**¡Emisor de ondas!**

Medidas preventivas:

- No manipular dentro de la tarjeta de adquisición.
- Asegúrese de que este bien conectado en los conectores hembra tipo banana para su adecuado funcionamiento.



- Al realizar la instalación eléctrica, respete todas las normativas vigentes.
- Los trabajos eléctricos o electrónicos deberán dejarse en manos de personal autorizado y calificado.
- Sobrepasar los valores límites (tanto máximos como mínimos) puede suponer un riesgo para los equipos, además de provocar averías en los aparatos.

#### 4. Descripción del equipo

##### Datos técnicos.

- MCP 4725
- Microcontrolador ATMEGA 328

##### Módulo de adquisición de datos

##### El proyecto es un módulo de adquisición tiene

- El proyecto es una tarjeta de adquisición de datos, conformada por un Microcontrolador ATMEGA 328, MCP 4725, consta de entradas y salidas

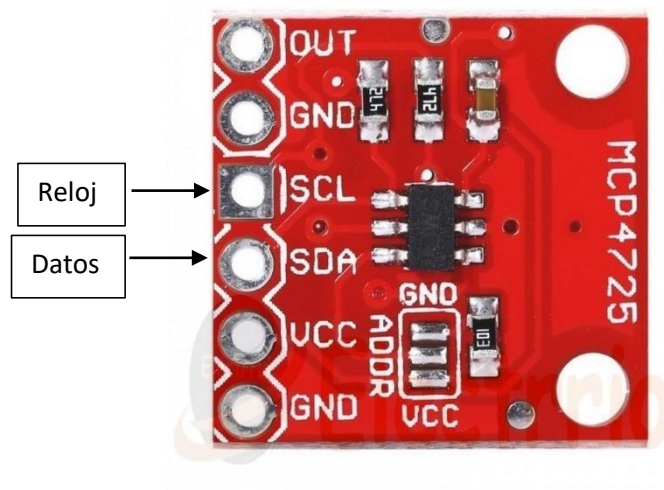
##### MCP 4725

El MCP tiene por objeto convertir la señal digital a analógica, este tiene dos direcciones porque es protocolo I2c por que tiene SCL y SDA, trabaja con 5 V.

Es un Dac de 12 bit de resolución.

Datasheet MCP 4725 se encuentra en la siguiente dirección:

<https://www.sparkfun.com/datasheets/BreakoutBoards/MCP4725.pdf>

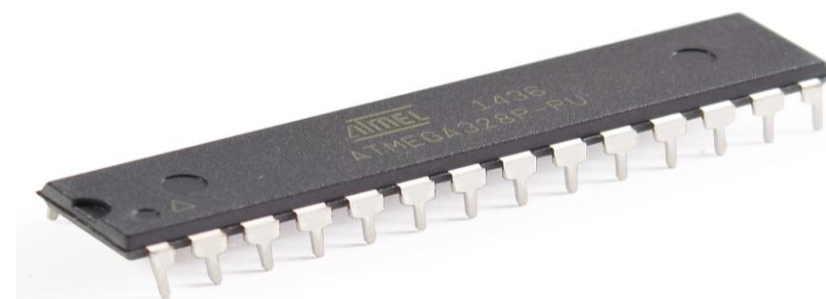


*Figura 4.MCP 4725.*

### **Microcontrolador ATMEGA 328**

El microcontrolador Atmega 328 fue utilizado por sus características.

Posee características como: 32 KB de memoria flash; con la capacidad de leer-mientras-escribe, 1 KB de memoria EEPROM, 2 KB de SRAM, 23 líneas de E/S de propósito general, 32 registros de proceso general, tres temporizadores flexibles/contadores con modo de comparación, interrupciones internas y externas, programador de modo USART, una interfaz serial orientada a byte de 2 cables, SPI e I2C, 6 canales 10-bit Conversor A/D, "watchdog timer" programable con oscilador interno, y cinco modos de ahorro de energía seleccionables por software. El dispositivo opera entre 1.8 y 5.5 voltios. Su arquitectura le permite ejecutar instrucciones en un solo ciclo de reloj, alcanzando una potencia de 1 MIPS.



*Figura 5. ATMEGA 328*

## **5. Interfaz con la tarjeta de adquisición de datos**

Se desarrolla un módulo real en donde el usuario podrá cargar la programación al PLC, una vez cargado la programación se conecta las salidas y entradas a la tarjeta de adquisición de datos para enviar y sacar las señales de 0 a 5V del software de simulación Unity 3D en la cual se lleva a cabo toda la ejecución del programa con los diferentes actuadores y equipos existentes en el proceso.

Para establecer el control mediante un PLC la tarjeta de adquisición de datos tiene integrado un circuito con acondicionadores de señal tanto analógicas como digitales las cuales son enviadas y recibidas por el PLC cumpliendo con la función de control, a continuación, se muestra un diagrama del acondicionamiento de señales.

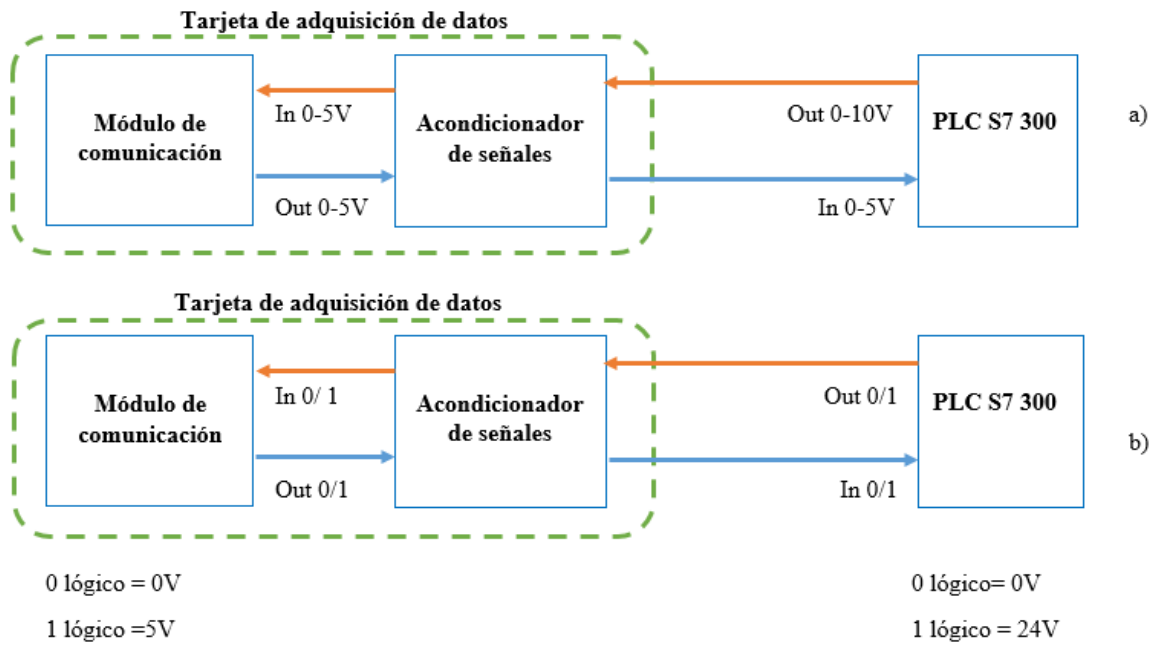
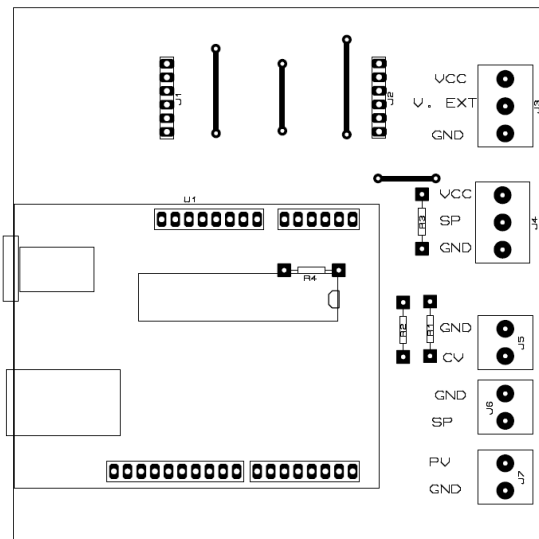


Figura 6. Acondicionamiento de señales analógicas.



b) Acondicionamiento de señales digitales.

Entradas y salidas analógicas de la tarjeta de adquisición de datos		
Variables	Denominación	Voltaje
PV	Process Value	0 – 5V
SP	Setpoint	0 – 5V
CV	Control Value	0 – 10V
B1	Alimentación y comunicación	5V

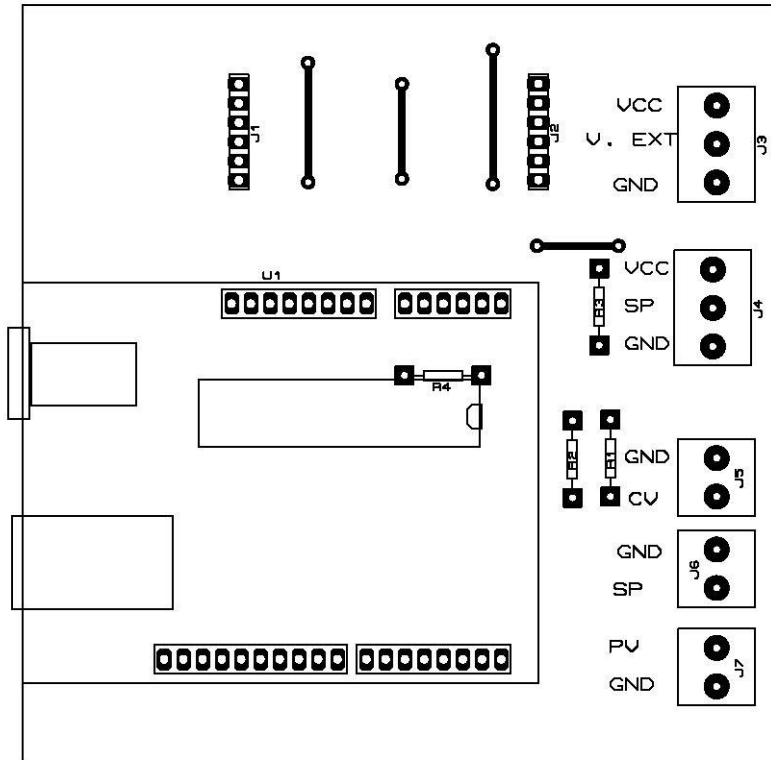


Figura 7. Placa de tarjeta de adquisición de datos.

Variables para conexión con PLC

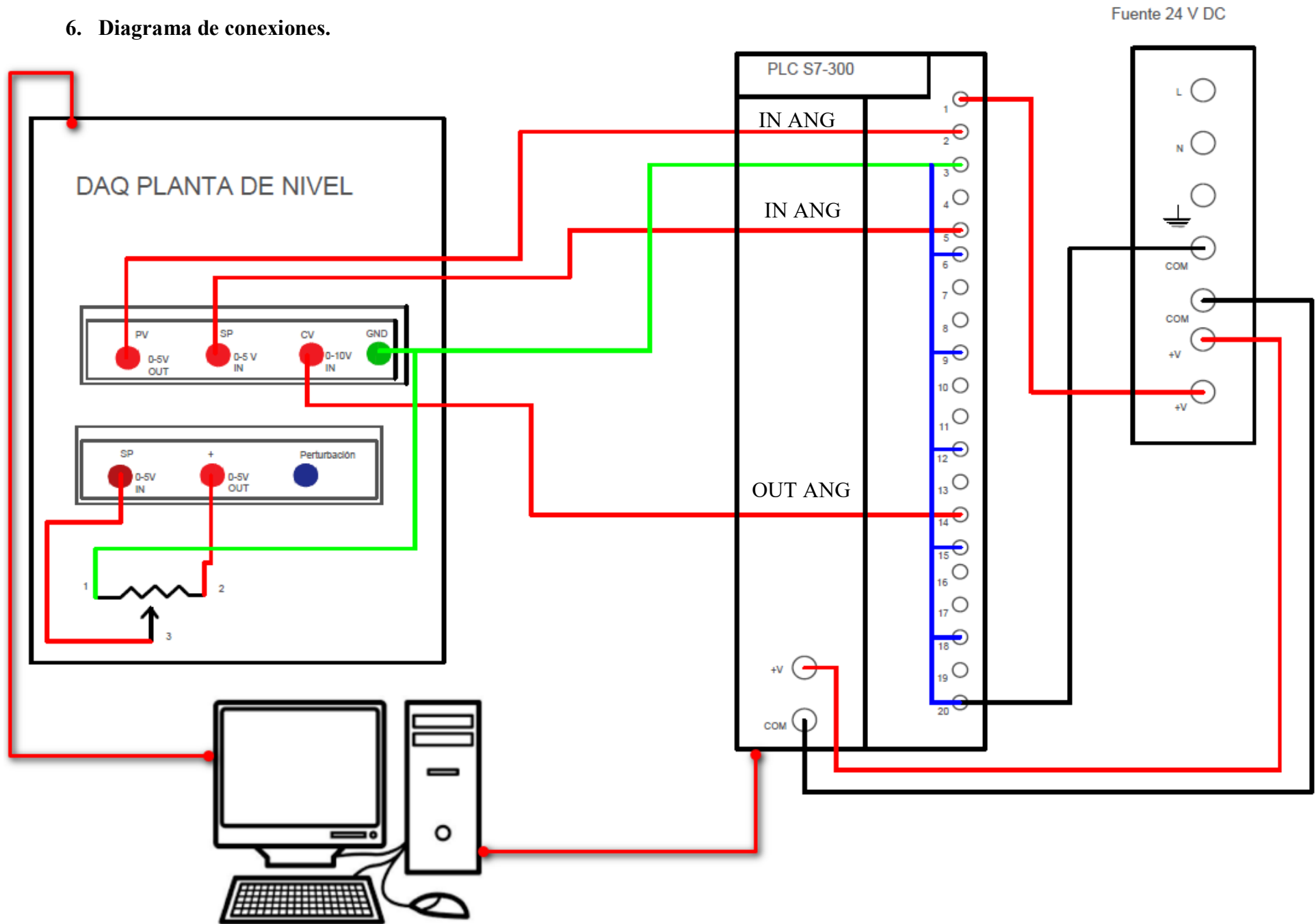
Módulo de entradas digitales.

<b>Señales Digitales enviadas</b>	Arduino Mega 1
<b>UNITY</b>	35473D
<b>Variables de entradas digitales</b>	<b>Denominación</b>
I5	Set point
I6	Process Valium

Módulo de entradas analógicas externas.

<i>Módulo de entradas analógicas externas.</i>	Tarjeta de adquisición de datos.
<b>Variables</b>	<b>Denominación</b>
Set point EXT	Bomba zona1
Válvula de descarga	Bomba zona 2

## 6. Diagrama de conexiones.



## **7. Funcionamiento del sistema**

Para el correcto funcionamiento del sistema se debe realizar los siguientes pasos:

1. Conectar su computador a una fuente de energía.
2. Realizar una copia del CD de la carpeta ArchivoNivel EJECUTABLE Y DAQ.
3. Conectar la DAQ PLANTA DE NIVEL mediante el cable USB.
4. Abrir la carpeta ArchivoNivelEJECUTABLE Y DAQ ingresar en la carpeta TarjetaPrincipalUNO doble clic en el archivo TarjetaPrincipalUno.
5. Una vez abierto el programa clic izquierdo en verificar el código para luego cargarlo.
6. Abrir el programa o ejecutable que está inmerso con el sistema del entorno virtual.
7. Conectar correctamente el interfaz con la tarjeta de adquisición de datos mediante conexión USB la cual ingresa al módulo de adquisición de datos.
8. Revisar el puerto de comunicación en el programa Arduino.
9. Dentro del programa seleccionar el puerto COM correctamente para obtener la comunicación serial con la tarjeta de adquisición de datos.
10. Dentro del programa o ejecutable correr la simulación para el desarrollo del sistema.
11. Realizar su control externo sea en PLC, Logo, Arduino.
12. Acondicionar sus señales de entrada y salidas de su controlador verificando el voltaje en el cual opera la DAQ.
13. Verificar conexiones y envió de señales.

## **8. Transporte y almacenamiento**

- Se debe transportar con mucho cuidado y de ser posible dentro de una caja para evitar el contacto directo con superficies, que puedan raspar o golpear al módulo, esto provocaría danos en el equipo.
- Para su almacenamiento evitar ambientes húmedos y el contacto con agua, así también evitar el apilamiento de otros objetos sobre el módulo, esto provocaría daños en el equipo.

**NOTA:** la programación del funcionamiento del sistema ya está precargada en el controlador por lo tanto no hace falta volver a cargar la programación en el mismo.

## 9. Operación del proceso

Se creó dos versiones del sistema: En la primera versión está en modo escritorio la cual cuenta con una programación para que el sistema sea controlado mediante un PLC, el accionamiento del sistema se realizó con la configuración del mouse y teclado para facilitar el uso del programa.



Figura 5. Instrucciones en el teclado.

### INSTRUCCIONES

- W Caminar hacia adelante
- S Caminar hacia atrás
- A Caminar hacia la izquierda
- D Caminar hacia la derecha
- Esc. pulsamos escape para fijar la pantalla
- Shift Seleccionar cursor mouse

### MOVIMIENTO DE LA CÁMARA

- ↑ Arriba
- ↓ Abajo
- Derecha
- ← Izquierda

## 10. RECOMENDACIONES

- Revisar los voltajes de entrada a la tarjeta de adquisición de datos.
- Revisar el correcto funcionamiento de los convertidores digitales analógicos revisando el Datashif a la salida de cada convertidor debe dar una variación de voltaje de 0 a 5v
- Revisar las conexiones del controlador usado.
- Si el control se realiza con un controlador PLC la entrada de señal analógica del CV debe ser un voltaje de entrada de 0-10v.

## 11. CONCLUSIONES

- La tarjeta de adquisición de datos se puede usar para realizar cualquier tipo de control tanto sea control PID, on-off, difuso, realizado en: Logo, PLC, Arduino.
- El entorno virtual como la tarjeta de adquisición de datos es de un uso fácil el cual lo hace didáctico.

Anexo 14	<b>UNIVERSIDAD TÉCNICA DE COTOPAXI</b>	
<b>PRÁCTICA DE LABORATORIO</b>		

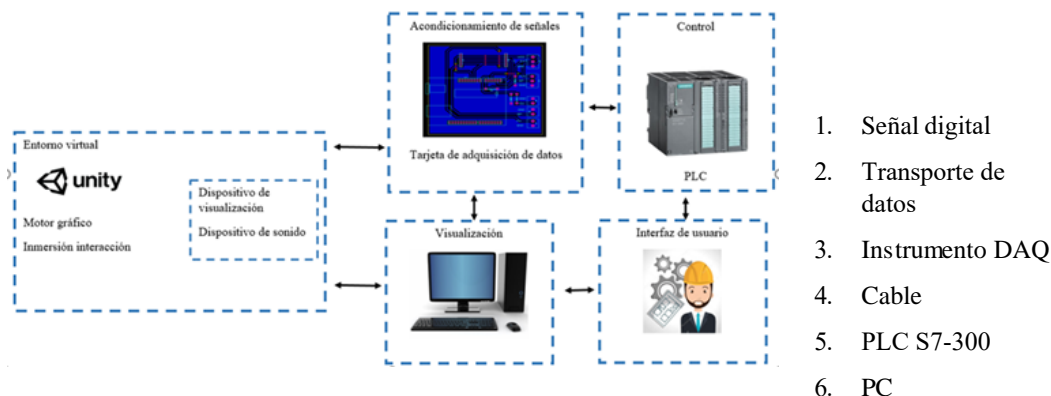
CARRERA	CÓDIGO DE LA ASIGNATURA	NOMBRE DE LA ASIGNATURA
INGENIERÍA ELECTROMECAÁNICA	ILM702M2	Sistemas de Control

PRÁCTICA N°	LABORATORIO:	Sistemas de Control	DURACIÓN (HORAS)
<b>01</b>	NOMBRE DE LA PRÁCTICA:	Sistemas PID de control de nivel	2

1	OBJETIVOS
	<ul style="list-style-type: none"> <li>• Realizar un sistema de control PID mediante software de elementos finitos.</li> <li>• Leer variables físicas mediante una tarjeta de adquisición de datos y acondicionamiento de las mismas.</li> <li>• Interactuar la interfaz de programación con la tarjeta de adquisición de datos.</li> </ul>

2	FUNDAMENTACIÓN TEÓRICA
	<p><b>2.1. INTRODUCCIÓN</b></p> <p>Un sistema DAQ está formado por tres tipos básicos de hardware, un bloque Terminal, un cable y un instrumento DAQ. Este apartado describe cada tipo de hardware, se centra en la función que desempeñan los componentes del instrumento DAQ.</p> <p>Funciones de Dispositivos DAQ</p> <p>Entradas analógicas miden señales analógicas Salidas analógicas generan señales analógicas Entradas/salidas digitales miden y generan señales digitales</p> <p>El procesamiento de datos de las DAQ se obtiene como se muestra en la figura 1</p>





**Figura 1.** Diagrama de adquisición de datos

Después de convertir el fenómeno físico en una señal medible con o sin acondicionarla, se debe adquirir esa señal. Para adquirir la señal es necesario un bloque Terminal, un cable, un dispositivo DAQ y un PC. Esta combinación de hardware puede transformar un PC estándar en un sistema de medición y automatización.

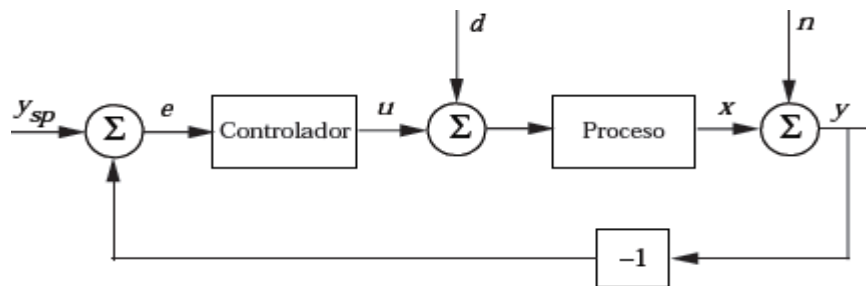
### Sistemas proporcionales integrales derivativos PID

El controlador PID es con mucho el algoritmo de control más común. La mayoría de los lazos de realimentación se controlan mediante este algoritmo u otro con pequeñas variaciones. Se implementa de muchas formas diferentes, como un controlador único o como parte de un paquete DDC (Control Digital Directo) o un sistema de control de procesos distribuido jerárquico. Muchos miles de ingenieros de instrumentación y control en todo el mundo están usando tales controladores en su trabajo diario. El algoritmo PID se puede contemplar desde muchas direcciones diferentes. Se puede ver como un dispositivo que puede operarse con unas pocas reglas heurísticas, pero que también puede estudiarse analíticamente.

La versión del algoritmo PID se puede describir como:

$$u(t) = K \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right)$$

Donde  $u$  es la señal de control y  $e$  es el error de control ( $e = y_{sp} - y$ ). La señal de control es así una suma de tres términos: el término-P (que es proporcional al error), el término-I (que es proporcional a la integral del error), y el término-D (que es proporcional a la derivada del error). Los parámetros del controlador son la ganancia proporcional  $K$ , el tiempo integral  $T_i$ , y el tiempo derivativo  $T_d$ .



**Figura 2.** Diagrama de bloques de un lazo de realimentación simple.

## 2.2. EQUIPO, INSTRUMENTOS Y MATERIALES NECESARIOS

Módulo DAQ planta control nivel

Fuente de alimentación de 24V

Cables jumper

Multímetro digital

PLC S7-300

Papel y lápiz

## 2.3. MEDIDAS DE SEGURIDAD

Realizar las conexiones del módulo con la tarjeta DAQ sin energización.

## 2.4. ACTIVIDADES A DESARROLLAR

En esta práctica se realizará un sistema de control PID mediante el control de señales digitales y analógicas partiendo de un entorno virtual, acondicionando en un módulo DAQ y un controlador S7-300 para realizar el control externo.

La tarjeta de adquisición de datos recibe señales digitales de nuestro entorno virtual para convertirlas en señales analógicas en un rango de 0 a 5 (V). Como también ingresa a la señal de CV del controlador en un voltaje de 0-10(V).

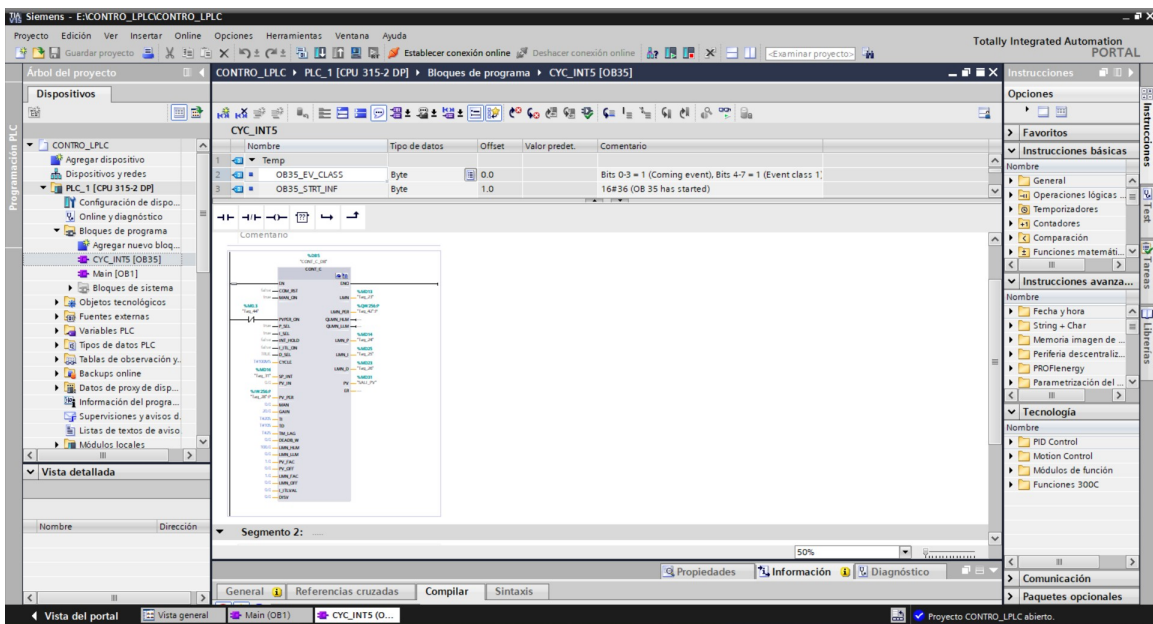
Dónde:

0(V)= 0 ltr

5(V)= 200 ltr

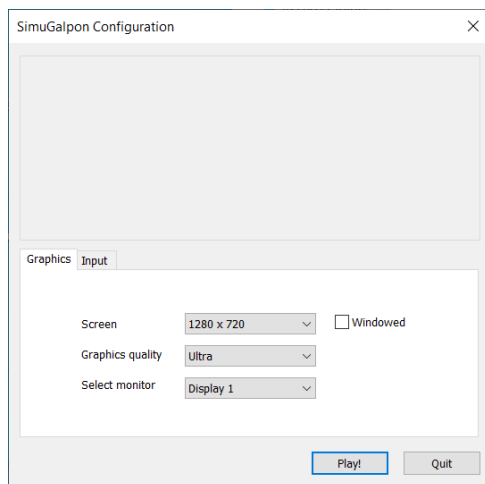
### Pasos para la realización del sistema de control PID de control de nivel

1. Conectar su computador a una fuente de energía
2. Realiza una copia del CD de la ArchivoNivel EJECUTABLE Y DAQ.
3. Conectar la DAQ PLANTA DE NIVEL mediante cable USB al computador.
4. Abrir la carpeta ArchivoNiveIEJECUTABLE Y DAQ ingresar en la carpeta TarjetaPrincipalUNO doble clic en el archivo TarjetaPrincipalUno.
5. Una vez abierto el programa clic izquierdo en verificar el código para luego cargarlo.
6. Realizar el código de programación de un control PID como se muestra en la figura 3.

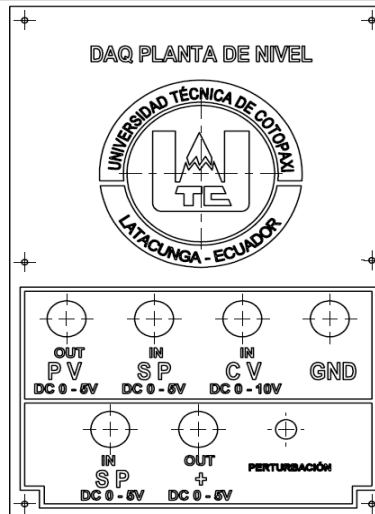


**Figura 3.** Código de programación TIA PORTAL PID COMPAC

7. Verificar la conexión del módulo DAQ PLANTA DE NIVEL para identificar nuestras señales SP, PV, CV seguir el diagrama de conexión del manual de equipo en la pág. 8. Identificando las señales de entrada y salida.
8. Buscar el ejecutable doble clic en el ejecutable sobre ellos en donde se desplegará una ventana como se muestra en la figura 4, en donde podrá configurar el tamaño de la ventana señalando Windowed.
9. Una vez dentro se despliega el menú de entrada donde tenemos Proceso, opciones, salir. Dando clic izquierdo en la pestaña opciones se despliega el menú para ingresar el puerto de comunicación de nuestra DAQ.
10. Escribir COM# dependiendo cual es el puerto que genera mi modulo.



**Figura 4.** Ventana de configuración de tamaño de ventana y menú de ingreso



**Figura 5.** Puertos de conexión de DAQ Planta de nivel

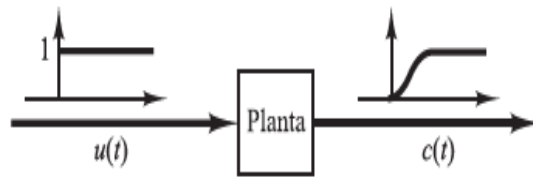
11. Procedo a conectar mis otras variables ya acondicionadas posterior mente en nuestra programación las cuales son SP, PV, CV.
12. Conecte el GND de la tarjeta DAQ a la GND recordar que el GND es común para todo nuestro sistema.
13. Para conectar nuestro SP externo pueden usar un potenciómetro o una fuente regulable de 5v máximo.
14. Revisar las conexiones tanto de nuestro controlador y fuente a usar (si usa una fuente externa).
15. Conecte la fuente de poder al módulo decontrol.
16. Verifique comandos de movimiento del player y cámara en el manual pag. 10.
17. Realice el control y tome los datos que se pide en los resultados.

## 2.5.METODOLOGÍA Y TÉCNICA EXPERIMENTAL

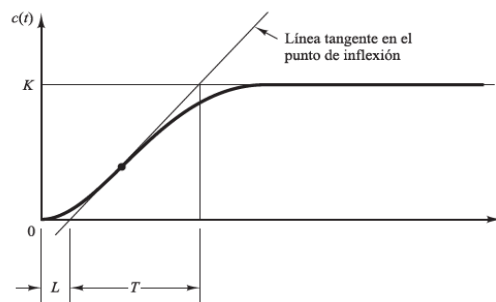
### Reglas de Ziegler-Nichols para sintonizar controladores PID.

Ziegler y Nichols propusieron reglas para determinar los valores de la ganancia proporcional  $K_p$ , del tiempo integral  $T_i$  y del tiempo derivativo  $T_d$ , basándose en las características de respuesta transitoria de una planta dada. Tal determinación de los parámetros de los controladores PID o sintonía de controladores PID la pueden realizar los ingenieros mediante experimentos sobre la planta. (Después de la propuesta inicial de Ziegler-Nichols han aparecido numerosas reglas de sintonía de controladores PID. Estas reglas están disponibles tanto en publicaciones técnicas como de los fabricantes de estos controladores.) Hay dos métodos denominados reglas de sintonía de Ziegler-Nichols: el primero y el segundo método. A continuación se hace una breve presentación de estos dos métodos. Primer método. En el primer método, la respuesta de la planta a una entrada escalón unitario se obtiene de manera experimental, tal como se muestra en la Figura 6 Si la planta no contiene integradores ni polos dominantes complejos conjugados, la curva de respuesta escalón unitario puede tener forma de S, como se observa en la Figura 7 Este método se puede aplicar si la respuesta muestra una curva con forma de S. Tales curvas de respuesta escalón se pueden generar experimentalmente o a partir de una simulación dinámica de la planta. La curva con forma de S se caracteriza por dos parámetros: el tiempo de retardo  $L$  y la constante de tiempo  $T$ . El tiempo de retardo y la constante de tiempo se determinan dibujando una

de esta tangente con el eje del tiempo y con la línea  $c(t)\%K$ , tal como se muestra en la figura 7



**Figura 6.** Respuesta a un escalón unitario de una planta.



**Figura 7.** Curva de respuesta en forma de S

**Tabla 1.** Regla de sintonía de Ziegler-Nichols basada en la respuesta escalón de la planta (primer método).

Tipo de controlador	$K_p$	$T_i$	$T_d$
P	$\frac{T}{L}$	$\infty$	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	$2L$	$0.5L$

Sintonía de controladores de sistemas PID por el segundo método Ziegler-Nichols

Este método no requiere retirar el controlador PID del lazo cerrado. En este caso solo hay que reducir al mínimo la acción derivativa y la acción integral del regulador PID. El ensayo en lazo cerrado consiste en aumentar poco a poco la ganancia proporcional hasta que el sistema oscile de forma mantenida ante cualquier perturbación. Esta oscilación debe ser lineal, sin saturaciones. En este momento hay que medir la ganancia proporcional, llamada ganancia crítica o  $K_c$ , y el periodo de oscilación  $T_c$  en segundos.

**Tabla 2.** Regla de sintonía de Ziegler-Nichols basada en la ganancia crítica  $K_c$  y periodo de oscilación  $T_c$  (segundo método).

Tipo de controlador	$K_p$	$T_i$	$T_d$
PID	$0.59 * K_c$	$1.18 * K_c / T_c$	$0.075 * K_c * T_c$

La constante  $K_p$  corresponde a la ganancia proporcional,  $T_i$  es la ganancia integral y  $T_d$  es la ganancia derivativa.

### 3.1. ANALISIS DE RESULTADOS

3.1.1. Obtener las curvas de sintonía del control PID.

3.1.2. Obtener valores de ganancia de las constantes de sintonía del control PID

**Tabla 3.** Constantes del PID

Configuración de ganancias	
Ganancia proporcional ( $K_c$ )	
Tiempo del Integral ( $T_i$ , min)	
Tiempo de la Derivada ( $T_d$ , min)	
Límites del PID	
Salida máxima	
Salida mínima	

3.1.3. Obtener formas de ondas reflejadas en el osciloscopio se visualizan en el entorno virtual.

3.1.4. Obtener voltajes de entrada y salidas del módulo usando las referencias marcadas en el módulo.

<b>4</b>	<b>CONCLUSIONES</b>
<b>5</b>	<b>RECOMENDACIONES</b>
<b>6</b>	<b>REFERENCIAS BIBLIOGRÁFICAS</b>
<p>[1] <b>Ogata, Katsuhiko.</b> Ingeniería de control moderna. Madrid : PEARSON EDUCACIÓN, S.A., 2010.</p> <p>[2] <b>Almaraz, Ruben Trejo.</b> Guía de LabVIEW Programación general y uso en Control. California : s.n., 1997.</p>	







PV			SP		
h(cm)	voltage	tiempo(s)	h(cm)	voltage	tiempo
0	0	0,000	0	0,000	0,000
1	0,025	0,210	1	0,025	0,210
2	0,050	0,420	2	0,050	0,420
3	0,075	0,630	3	0,075	0,630
4	0,099	0,840	4	0,100	0,840
5	0,124	1,050	5	0,125	1,050
6	0,149	1,259	6	0,150	1,259
7	0,174	1,469	7	0,175	1,469
8	0,199	1,679	8	0,199	1,679
9	0,224	1,889	9	0,224	1,889
10	0,249	2,099	10	0,249	2,099
11	0,273	2,309	11	0,274	2,309
12	0,298	2,519	12	0,299	2,519
13	0,323	2,729	13	0,324	2,729
14	0,348	2,939	14	0,349	2,939
15	0,373	3,149	15	0,374	3,149
16	0,398	3,358	16	0,399	3,358
17	0,423	3,568	17	0,424	3,568
18	0,447	3,778	18	0,449	3,778
19	0,472	3,988	19	0,474	3,988
20	0,497	4,198	20	0,499	4,198
21	0,522	4,408	21	0,524	4,408
22	0,547	4,618	22	0,548	4,618
23	0,572	4,828	23	0,573	4,828
24	0,597	5,038	24	0,598	5,038
25	0,622	5,248	25	0,623	5,248
26	0,646	5,457	26	0,648	5,457
27	0,671	5,667	27	0,673	5,667
28	0,696	5,877	28	0,698	5,877
29	0,721	6,087	29	0,723	6,087
30	0,746	6,297	30	0,748	6,297
31	0,771	6,507	31	0,773	6,507
32	0,796	6,717	32	0,798	6,717
33	0,820	6,927	33	0,823	6,927
34	0,845	7,137	34	0,848	7,137
35	0,870	7,347	35	0,873	7,347
36	0,895	7,556	36	0,897	7,556
37	0,920	7,766	37	0,922	7,766
38	0,945	7,976	38	0,947	7,976
39	0,970	8,186	39	0,972	8,186
40	0,994	8,396	40	0,997	8,396
41	1,019	8,606	41	1,022	8,606
42	1,044	8,816	42	1,047	8,816
43	1,069	9,026	43	1,072	9,026
44	1,094	9,236	44	1,097	9,236
45	1,119	9,446	45	1,122	9,446
46	1,144	9,655	46	1,147	9,655
47	1,168	9,865	47	1,172	9,865



48	1,193	10,075	48	1,197	10,075
49	1,218	10,285	49	1,222	10,285
50	1,243	10,495	50	1,247	10,495
51	1,268	10,705	51	1,271	10,705
52	1,293	10,915	52	1,296	10,915
53	1,318	11,125	53	1,321	11,125
54	1,342	11,335	54	1,346	11,335
55	1,367	11,545	55	1,371	11,545
56	1,392	11,754	56	1,396	11,754
57	1,417	11,964	57	1,421	11,964
58	1,442	12,174	58	1,446	12,174
59	1,467	12,384	59	1,471	12,384
60	1,492	12,594	60	1,496	12,594
61	1,516	12,804	61	1,521	12,804
62	1,541	13,014	62	1,546	13,014
63	1,566	13,224	63	1,571	13,224
64	1,591	13,434	64	1,596	13,434
65	1,616	13,644	65	1,620	13,644
66	1,641	13,853	66	1,645	13,853
67	1,666	14,063	67	1,670	14,063
68	1,690	14,273	68	1,695	14,273
69	1,715	14,483	69	1,720	14,483
70	1,740	14,693	70	1,745	14,693
71	1,765	14,903	71	1,770	14,903
72	1,790	15,113	72	1,795	15,113
73	1,815	15,323	73	1,820	15,323
74	1,840	15,533	74	1,845	15,533
75	1,865	15,743	75	1,870	15,743
76	1,889	15,952	76	1,895	15,952
77	1,914	16,162	77	1,920	16,162
78	1,939	16,372	78	1,945	16,372
79	1,964	16,582	79	1,969	16,582
80	1,989	16,792	80	1,994	16,792
81	2,014	17,002	81	2,019	17,002
82	2,039	17,212	82	2,044	17,212
83	2,063	17,422	83	2,069	17,422
84	2,088	17,632	84	2,094	17,632
85	2,113	17,842	85	2,119	17,842
86	2,138	18,051	86	2,144	18,051
87	2,163	18,261	87	2,169	18,261
88	2,188	18,471	88	2,194	18,471
89	2,213	18,681	89	2,219	18,681
90	2,237	18,891	90	2,244	18,891
91	2,262	19,101	91	2,269	19,101
92	2,287	19,311	92	2,294	19,311
93	2,312	19,521	93	2,318	19,521
94	2,337	19,731	94	2,343	19,731
95	2,362	19,941	95	2,368	19,941
96	2,387	20,150	96	2,393	20,150
97	2,411	20,360	97	2,418	20,360



98	2,436	20,570	98	2,443	20,570
99	2,461	20,780	99	2,468	20,780
100	2,486	20,990	100	2,493	20,990
101	2,511	21,200	101	2,518	21,200
102	2,536	21,410	102	2,543	21,410
103	2,561	21,620	103	2,568	21,620
104	2,585	21,830	104	2,593	21,830
105	2,610	22,040	105	2,618	22,040
106	2,635	22,249	106	2,643	22,249
107	2,660	22,459	107	2,668	22,459
108	2,685	22,669	108	2,692	22,669
109	2,710	22,879	109	2,717	22,879
110	2,735	23,089	110	2,742	23,089
111	2,759	23,299	111	2,767	23,299
112	2,784	23,509	112	2,792	23,509
113	2,809	23,719	113	2,817	23,719
114	2,834	23,929	114	2,842	23,929
115	2,859	24,139	115	2,867	24,139
116	2,884	24,348	116	2,892	24,348
117	2,909	24,558	117	2,917	24,558
118	2,933	24,768	118	2,942	24,768
119	2,958	24,978	119	2,967	24,978
120	2,983	25,188	120	2,992	25,188
121	3,008	25,398	121	3,017	25,398
122	3,033	25,608	122	3,041	25,608
123	3,058	25,818	123	3,066	25,818
124	3,083	26,028	124	3,091	26,028
125	3,108	26,238	125	3,116	26,238
126	3,132	26,447	126	3,141	26,447
127	3,157	26,657	127	3,166	26,657
128	3,182	26,867	128	3,191	26,867
129	3,207	27,077	129	3,216	27,077
130	3,232	27,287	130	3,241	27,287
131	3,257	27,497	131	3,266	27,497
132	3,282	27,707	132	3,291	27,707
133	3,306	27,917	133	3,316	27,917
134	3,331	28,127	134	3,341	28,127
135	3,356	28,337	135	3,366	28,337
136	3,381	28,546	136	3,390	28,546
137	3,406	28,756	137	3,415	28,756
138	3,431	28,966	138	3,440	28,966
139	3,456	29,176	139	3,465	29,176
140	3,480	29,386	140	3,490	29,386
141	3,505	29,596	141	3,515	29,596
142	3,530	29,806	142	3,540	29,806
143	3,555	30,016	143	3,565	30,016
144	3,580	30,226	144	3,590	30,226
145	3,605	30,436	145	3,615	30,436
146	3,630	30,645	146	3,640	30,645
147	3,654	30,855	147	3,665	30,855



148	3,679	31,065	148	3,690	31,065
149	3,704	31,275	149	3,715	31,275
150	3,729	31,485	150	3,740	31,485
151	3,754	31,695	151	3,764	31,695
152	3,779	31,905	152	3,789	31,905
153	3,804	32,115	153	3,814	32,115
154	3,828	32,325	154	3,839	32,325
155	3,853	32,535	155	3,864	32,535
156	3,878	32,744	156	3,889	32,744
157	3,903	32,954	157	3,914	32,954
158	3,928	33,164	158	3,939	33,164
159	3,953	33,374	159	3,964	33,374
160	3,978	33,584	160	3,989	33,584
161	4,002	33,794	161	4,014	33,794
162	4,027	34,004	162	4,039	34,004
163	4,052	34,214	163	4,064	34,214
164	4,077	34,424	164	4,089	34,424
165	4,102	34,634	165	4,113	34,634
166	4,127	34,843	166	4,138	34,843
167	4,152	35,053	167	4,163	35,053
168	4,176	35,263	168	4,188	35,263
169	4,201	35,473	169	4,213	35,473
170	4,226	35,683	170	4,238	35,683
171	4,251	35,893	171	4,263	35,893
172	4,276	36,103	172	4,288	36,103
173	4,301	36,313	173	4,313	36,313
174	4,326	36,523	174	4,338	36,523
175	4,351	36,733	175	4,363	36,733
176	4,375	36,942	176	4,388	36,942
177	4,400	37,152	177	4,413	37,152
178	4,425	37,362	178	4,438	37,362
179	4,450	37,572	179	4,462	37,572
180	4,475	37,782	180	4,487	37,782
181	4,500	37,992	181	4,512	37,992
182	4,525	38,202	182	4,537	38,202
183	4,549	38,412	183	4,562	38,412
184	4,574	38,622	184	4,587	38,622
185	4,599	38,832	185	4,612	38,832
186	4,624	39,041	186	4,637	39,041
187	4,649	39,251	187	4,662	39,251
188	4,674	39,461	188	4,687	39,461
189	4,699	39,671	189	4,712	39,671
190	4,723	39,881	190	4,737	39,881
191	4,748	40,091	191	4,762	40,091
192	4,773	40,301	192	4,787	40,301
193	4,798	40,511	193	4,811	40,511
194	4,823	40,721	194	4,836	40,721
195	4,848	40,931	195	4,861	40,931
196	4,873	41,140	196	4,886	41,140
197	4,897	41,350	197	4,911	41,350



198	4,922	41,560	198	4,936	41,560
199	4,947	41,770	199	4,961	41,770
200	4,972	41,98	200	4,986	41,98