

Aplicación HMI de software libre desarrollada en Python para monitorear un subsistema hidráulico del hospital pediátrico La Balear de La Habana

Free software HMI application developed on Python to monitoring a hydraulic subsystem at the pediatric hospital La Balear of Havana

Isidro Jorge Zertucha González^{1,*} Romar Alexei Borges Rivero

¹ Facultad de Ingeniería Automática y Biomédica. Universidad Tecnológica de La Habana “José Antonio Echeverría” (CUJAE). Calle 114 No 11901 entre 119 y 127, Marianao, La Habana, Cuba.

*Correspondencia: isidro.zg@automatica.cujae.edu.cu

Este documento posee una [licencia Creative Commons Reconocimiento/No Comercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/)



Resumen

El presente trabajo expone lo realizado como parte del proyecto de la disciplina de instrumentación para fomentar el uso de las tecnologías de software libre en las asignaturas que dentro de la carrera de ingeniería en automática le son afines. Para profundizar en el empleo de estas tecnologías se escoge como campo de acción un subsistema hidráulico del hospital pediátrico La Balear. Se realiza el estudio de la situación actual de su sistema de automatización. Se proponen los sensores y medios técnicos de automatización necesarios para conocer y controlar el nivel de agua en los depósitos asociados a dicho subsistema. Se emplea el software Python y un módulo de hardware Arduino para desarrollar y proponer una aplicación de monitoreo y control. La PC de monitoreo propuesta ejecuta la aplicación Python y MySQL como sistema de gestión de bases de datos relacional. Para la comunicación con los instrumentos del nivel de control se desarrolla una aplicación servidor OPC-UA también elaborada en Python. Finalmente, se realiza una valoración del efecto económico de la propuesta.

Palabras clave: Python, Kivy, MySQL, OPC, Arduino

Abstract

The present work exposes the realization as part of the Project of the instrumentation discipline to foment the use of free software technologies on signatures that inside the automation engineering career path are common. To deepen in the use of these technologies it is selected, as action field, a hydraulic sub-system of the pediatric hospital “La Balear”. It is realized the study of the actual situation of its automation system. Sensors and technical means of automation are proposed for knowing and controlling the water level at deposits associated to that subsystem. It is used the Python software and an Arduino hardware module to develop and propose a monitoring and control application. The proposed monitoring PC executes the Python application and MySQL as relational database gestor system. For communicating with the control level instruments, it is developed an OPC-UA server application, also created on Python. Finally, it is realized a valuation of the economic effect of the proposal.

Keywords: Python, Kivy, MySQL, OPC, Arduino

1. Introducción

Desde 2021 la disciplina de instrumentación del departamento de automática de la facultad de ingeniería en automática y biomédica de la Cujae ha venido desarrollando un proyecto denominado: Perfeccionamiento del proceso de la enseñanza y aprendizaje de la instrumentación en la Carrera de Ingeniería en Automática. El mismo se inserta en el programa denominado: Automática, Robótica e Inteligencia Artificial (ARIA) y tiene gran importancia para la adecuar las asignaturas de esta disciplina a las tendencias más actuales a nivel internacional.

Para ello se ha seleccionado el siguiente Objeto de Estudio: Los sistemas hidráulicos de bombeo de agua potable en las instalaciones hospitalarias.

Como Campo de Acción se ha seleccionado el siguiente: Subsistema hidráulico 3 del sistema hidráulico general del hospital pediátrico La Balear.

Situación Problema: El proceso de control de nivel a través del arranque parada de la bomba en el subsistema hidráulico 3, cuando se desea llenar los tanques, se realiza solo manualmente y de forma inapropiada, por la inexistencia de equipos de automatización tales como: Sensores de Nivel; un Accionamiento Eléctrico; un Controlador y un Sistema de Monitoreo adecuados.

Problema a Resolver: El proceso de control de nivel a través del arranque parada de la bomba en el subsistema hidráulico 3 es ineficiente, se incrementan las averías en el motor de la bomba, son numerosas las roturas, es necesario contar constantemente con un operador para manejar este proceso; por lo que el costo de las operaciones es elevado.

Hipótesis: Si el proceso de control de nivel a través del arranque parada de la bomba en el subsistema hidráulico 3, utilizada para el llenado de los tanques en el techo del policlínico del hospital, se perfecciona a través de la instalación de un sistema de automatización con el uso de tecnologías de software y hardware libre, se podrá solucionar los problemas actualmente existentes en el mismo.

Objetivo General: Diseñar un sistema de automatización con el uso de tecnologías de software y hardware libre, para dar solución a los problemas actuales en el proceso de control de nivel a través del arranque parada de la bomba en el subsistema hidráulico 3 de la Balear.

Objetivos Específicos:

- Analizar el estado actual de los sistemas hidráulicos para hospitales y del subsistema hidráulico 3 del hospital pediátrico La Balear.
- Analizar el estado actual de los instrumentos y medios técnicos de automatización del subsistema hidráulico 3 del hospital pediátrico La Balear.
- Proponer los instrumentos y medios técnicos de automatización necesarios para el control de nivel del subsistema hidráulico 3 del hospital pediátrico La Balear.
- Estudio de la factibilidad económica de la propuesta de automatización realizada.

2. Materiales y Métodos

Descripción del proceso

El proceso de estudio consta de una bomba de agua que debe ser encendida cuando la reserva de agua en los tanques elevados del hospital sea insuficiente, teniendo en cuenta que, para esto, la cisterna no puede estar vacía.

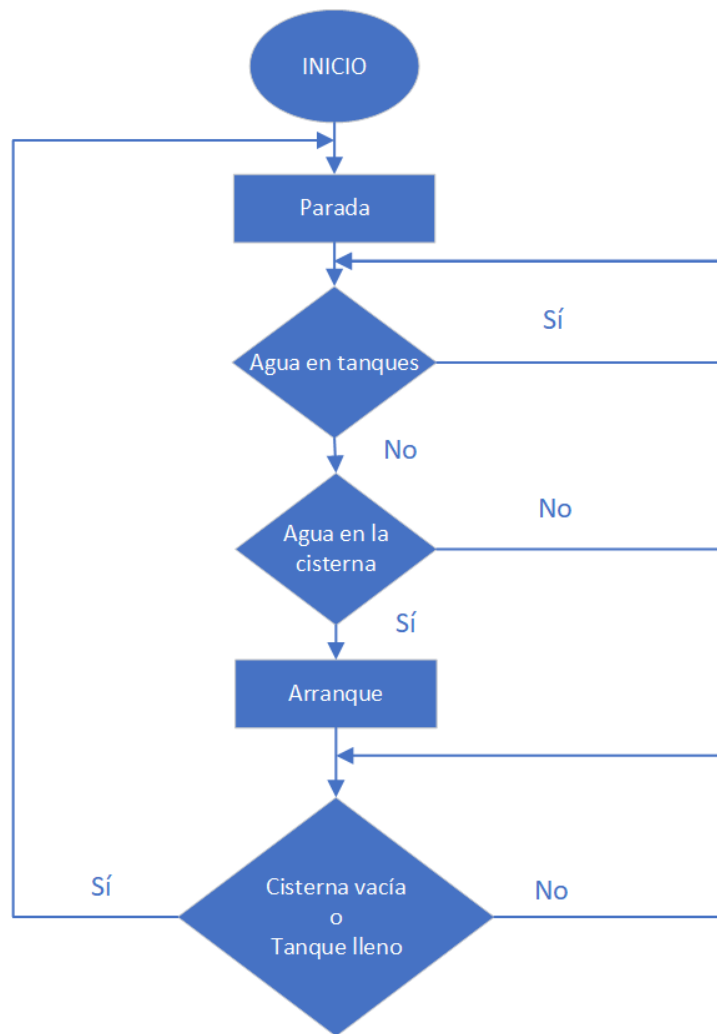


Fig.1 Diagrama de flujo del proceso

Propuestas de solución

Sensor de Nivel

Se necesita controlar el nivel de agua para que no se vacíe o exceda una altura dada. De ocurrir lo anterior se requeriría un dispositivo que cambie de estado y genere una señal de aviso al controlador de nivel. Existen varios tipos de sensor de nivel. Entre los más comunes se encuentra el sensor de nivel por presión diferencial. Al variar el nivel en el tanque, varía la presión que el agua ejerce sobre el instrumento. En el presente proyecto se propone el sensor de nivel de tipo presión hidrostática GLT501, que cuenta con un grado de protección IP68. Se ha elegido el este sensor debido a lo sencillo de instalar, a su disponibilidad actual en el mercado y su precio asequible. No requiere de un transductor externo, pues él mismo se encarga de otorgar una respuesta en magnitudes eléctricas.



Pressure range	0~1m ... 200mH2O		
Accuracy	0.25%F.S., 0.5%F.S.		
Power supply	12~30VDC, 5VDC		
Output	4~20mA、0~5VDC、0.5~4.5VDC、RS485,, customization		
Temp. range	Media Temp.	Operation Temp.	Storage Temp.
	-10~70℃	-10~70℃	-20~70℃
Ingress protection	Probe	IP68	
	Terminal box	IP67	
Mechanical connection	M42x1.5(M), Flange DN50 PN1.0, customization		
Anti-vibration	20g, (20~5000Hz)		
Shockproof	100g, 11m		
Probe material	PTFE, PVC, customization		
Service life	> 1x 108 Pressure cycling		

Fig.2 Propuesta de sensores de nivel para el subsistema hidráulico 3 de La Balear

Controlador

Arduino es una plataforma de hardware y software libre, su estructura se encuentra basada en un microcontrolador de la familia Atmel AVR. Además, Arduino tiene un entorno de programación que permite desarrollar los proyectos electrónicos de manera más sencilla y eficiente. Debido a la gran variedad que poseen dentro de los microcontroladores y el bajo costo se convierten en una gran opción para realizar grandes y múltiples diseños electrónicos [1].

Se decide usar M-Duino 21+ en el proyecto, el cual ofrece un microcontrolador ATmega2560 de 16 MHz de frecuencia reloj y soporta señales de hasta 24V. Cuenta con su propio software gratuito para programarlo. Se puede comunicar con una PC vía Ethernet mediante cableado, además es resistente a ruidos y campos electromagnéticos, vibraciones mecánicas, polvo y humedad.



Fig.3 M-Duino 21+

La alimentación del hospital es de 220V, para llevarlo a un estándar de trabajo de 24V se requiere de un dispositivo transformador. Esto se puede lograr implementando una fuente.



Fig.4 Fuente de alimentación conmutada impermeable tipo SMPS

Para el accionamiento del motor de la bomba del subsistema hidráulico 3 se propone el arranque en paralelo con el botón de marcha y en serie con el botón de parada del circuito de accionamiento eléctrico. Se sitúan las salidas del controlador para realizar el control en automático del nivel en los

tanques elevados de este proceso. Este tipo de control es una variante económica en comparación con el uso un variador o un arrancador suave y suficiente para atender la situación problema.

Características del equipamiento propuesto:

- Tipo: Arranque directo ON/OFF.
- Productos: 2 (1 Contactor, 1 Protección Magneto/térmica).
- Códigos: Contactor: GV2-P16, Protección Magneto/térmica: LC1-D25
- Fabricante: Schneider Electric

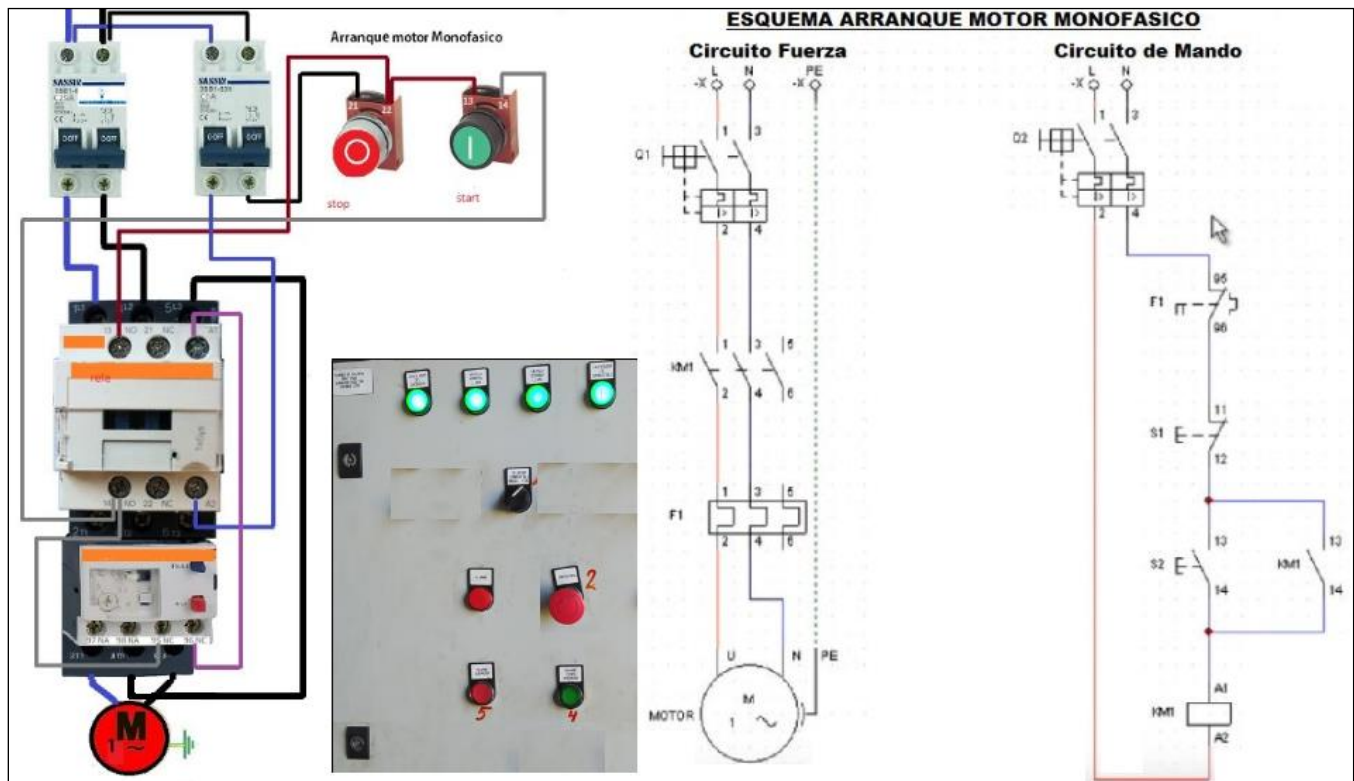


Fig.5 Propuesta de accionamiento eléctrico para el subsistema hidráulico 3 de La Balear

OPC-UA

Se le conoce como un método de conectividad que sirve para comunicar dispositivos, controladores y/o aplicaciones sin que se presenten los típicos problemas de conexión de los protocolos de comunicación de los propietarios. (KOMINEK, 2009). Este método de conectividad se consigue con el uso de dos componentes OPC llamados Cliente OPC y Servidor OPC. La comunicación entre estos dos componentes no implica que deje de usarse el protocolo de comunicación nativo de los fabricantes. (KOMINEK, 2009) [2].

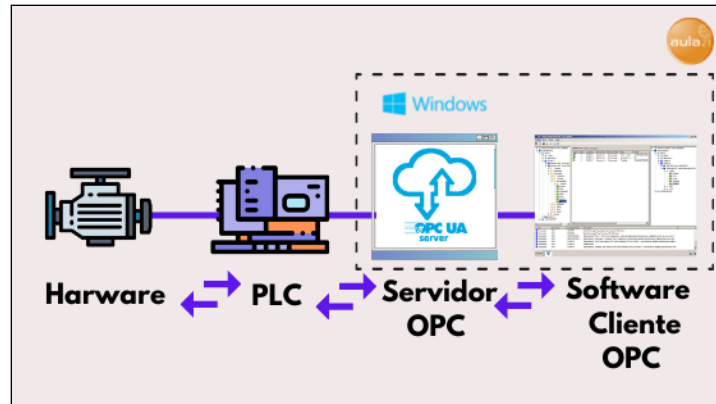


Fig.5 Esquema que ilustra el funcionamiento de OPC

Python

Según (van Rossum, 2009), Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas [3].

Xampp

XAMPP es una distribución de Apache que incluye varios softwares libres. El nombre es un acrónimo compuesto por las iniciales de los programas que lo constituyen: el servidor web Apache, los sistemas relacionales de administración de bases de datos MySQL y MariaDB, así como los lenguajes de programación Perl y PHP. La inicial X se usa para representar a los sistemas operativos Linux, Windows y Mac OS X.

MySQL es un sistema de gestión (o manejador) de bases de datos. Puede ser, desde una simple lista de artículos a las inmensas cantidades de información en una red corporativa. Al ser un software de código abierto, es accesible para cualquiera, para usarlo o modificarlo. Se puede descargar MySQL desde Internet y usarlo sin pagar nada, de esta manera cualquiera puede inclinarse a estudiar el código fuente y cambiarlo para adecuarlo a sus necesidades. MySQL es fácil de usar tanto para volúmenes de datos grandes como pequeños.

Entre sus características destaca:

- Velocidad y robustez.

- Multiproceso, es decir puede usar varias CPU si éstas están disponibles.
- Puede trabajar en distintas plataformas y sistemas operativos distintos.
- Sistema de contraseñas y privilegios muy flexible y seguro.
- Todas las palabras viajan encriptados en la red.
- Los clientes usan TCP o UNIX Socket para conectarse al servidor.
- El servidor soporta mensajes de error en distintos idiomas.
- Todos los comandos tienen help para las ayudas [4].

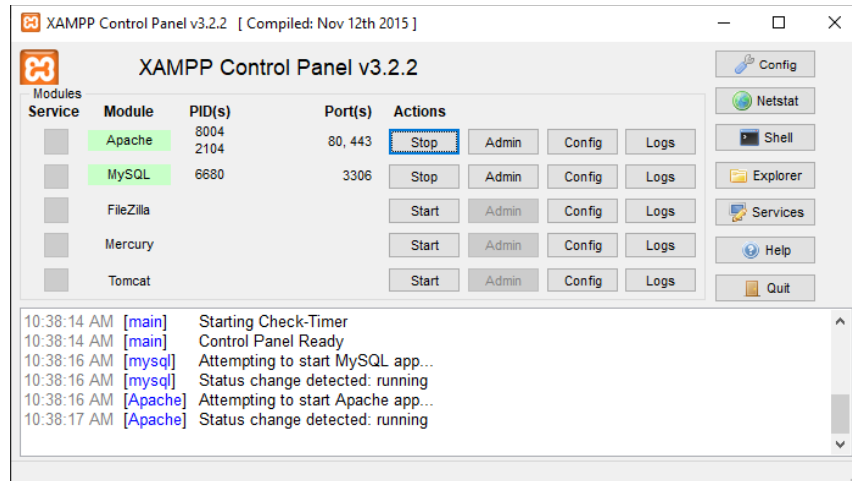


Fig.6 Paquete de software XAMPP

3. Resultados y Discusión

Evidencia del funcionamiento del diseño propuesto

Se realiza la programación del controlador mediante el software original de Arduino. En la figura siguiente se puede observar un fragmento del código resultante.

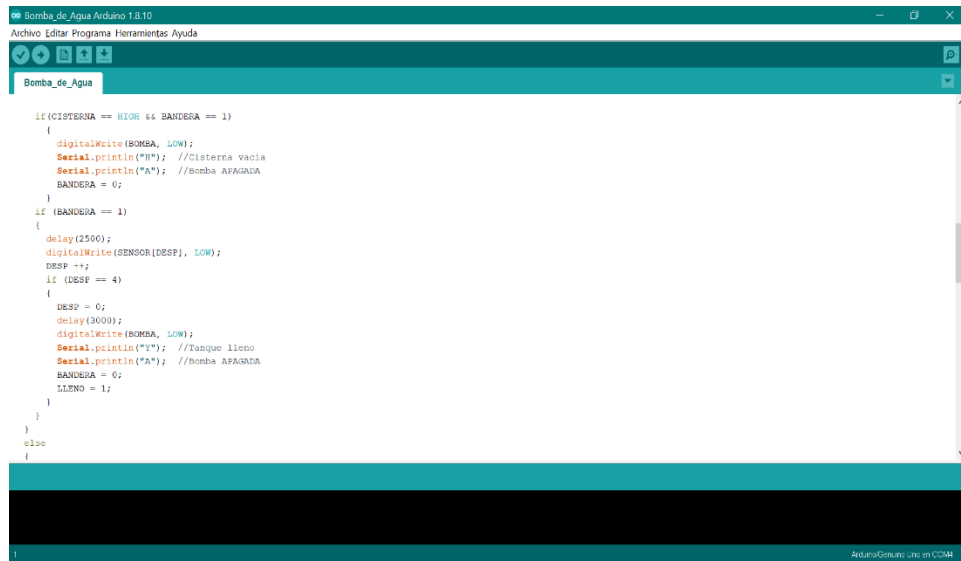


Fig.7 Código fuente del controlador Arduino empleado

Mediante el software XAMPP y su herramienta phpMyAdmin, se conformó la estructura de la base de datos, donde se almacenarán las credenciales de las cuentas administrativas, los registros históricos de alarmas y eventos.

Table	Action	Rows	Type	Collation	Size	Overhead
auth_group	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32 K	-
auth_group_permissions	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	48 K	-
auth_permission	Browse Structure Search Insert Empty Drop	36	InnoDB	latin1_swedish_ci	32 K	-
auth_user	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	32 K	-
auth_user_groups	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	48 K	-
auth_user_user_permissions	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	48 K	-
django_admin_log	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	48 K	-
django_content_type	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	32 K	-
django_migrations	Browse Structure Search Insert Empty Drop	22	InnoDB	latin1_swedish_ci	16 K	-
django_session	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	32 K	-
monitoreo_alarma	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 K	-
monitoreo_evento	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 K	-
12 tables	Sum	74	InnoDB	latin1_swedish_ci	400 K	8

Fig.8 Base de datos visualizada desde phpMyAdmin de Xampp

Empleando la librería OPC UA de Python se desarrolló un servidor OPC encargado intercambiar información vía serial con Arduino, almacenando las variables del proceso en tiempo real para su futuro análisis por otro software.

```

server.py - C:\Users\zael\Documents\PycharmProjects\VisualSCADA Hospital\opcserver.py (3.8.10)
File Edit Format Run Options Window Help
from opcua import Server
from random import randint
import time
from datetime import datetime
import serial
import mysql.connector

COM = "COM4"

arduino_is_online = True
try:
    arduino = serial.Serial(COM, 9600, timeout=0) # iniciar conexión a 9600 baudios
    time.sleep(2)
except:
    arduino_is_online = False
    print("Arduino offline")

server = Server()
server.set_endpoint("opc.tcp://127.0.0.1:4840")

addspace = server.register_namespace("OPCUA_SIMULATION_SERVER")

node = server.get_objects_node()

Param = node.add_object(addspace, "Parameters")

Tanque = Param.add_variable(addspace, "Estado del Tanque", 0)
Tanque.set_writable()
Cisterna = Param.add_variable(addspace, "Estado de la Cisterna", 0)
Cisterna.set_writable()
Bomba = Param.add_variable(addspace, "Estado de la Bomba", 0)
Bomba.set_writable()
EstadoMN = Param.add_variable(addspace, "Estado de Alarma/Normalidad", 0)
EstadoMN.set_writable()
SelectorMN = Param.add_variable(addspace, "Selector Manual/Automático", 0)
SelectorMN.set_writable()
BotEmergencia = Param.add_variable(addspace, "Estado del Botón de Emergencia", 0)
BotEmergencia.set_writable()
DispProtoc = Param.add_variable(addspace, "Disparo de Protección Térmica/Magnética", 0)
DispProtoc.set_writable()
BotReset = Param.add_variable(addspace, "Estado del Botón de Reset de Alarma", 0)

```

Fig.9 Aplicación en consola de servidor OPC desarrollada en Python

Las interfaces Hombre-Máquina de tipo pantallas de monitorización, son sistemas que reconocen el lenguaje de programación de la máquina y proporcionan una representación gráfica de la misma. Usando una pantalla de monitorización se puede obtener una representación gráfica de la máquina y se puede controlar su funcionamiento a través de los comandos de programación. Mediante programación en Python, se logró crear el software necesario para dicha labor. Esta aplicación se encargará de adquirir el estado de las variables del proceso alojadas en el servidor OPC, así como almacenar en base de datos los eventos y alarmas pertinentes.

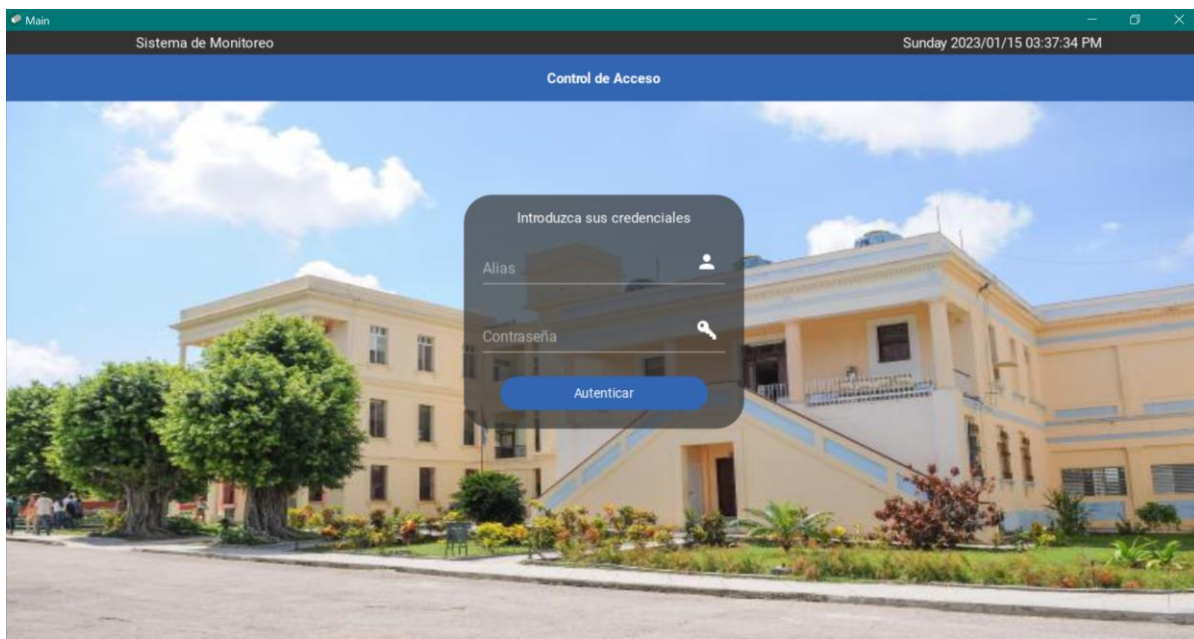


Fig.10 Pantalla de autenticación de HMI

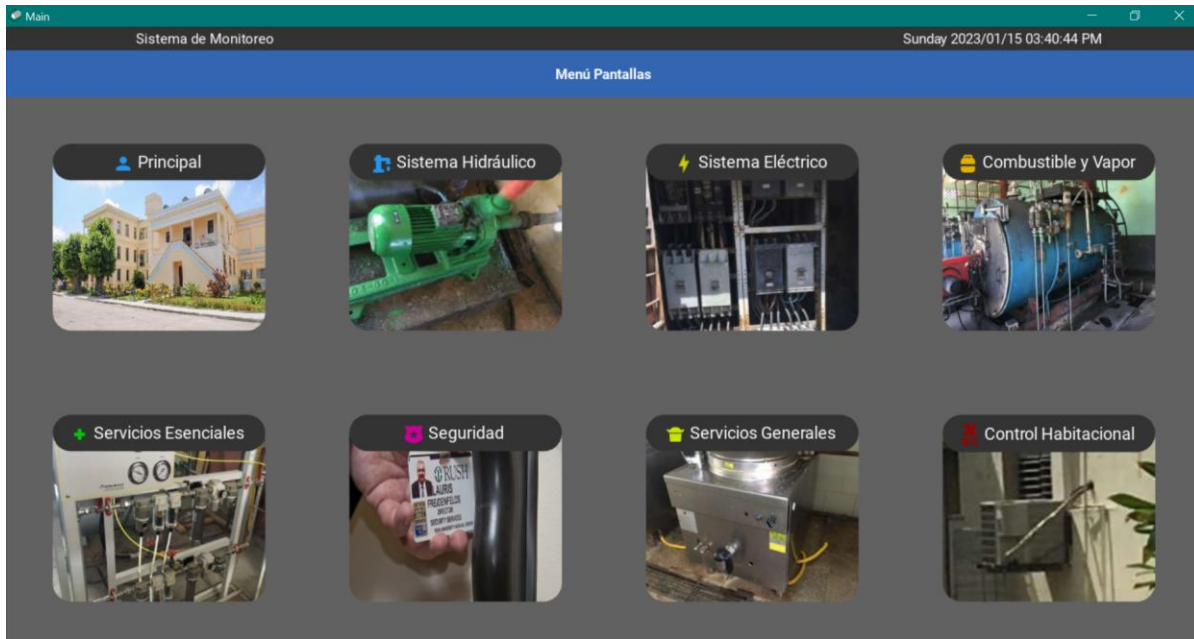


Fig.11 Menú principal

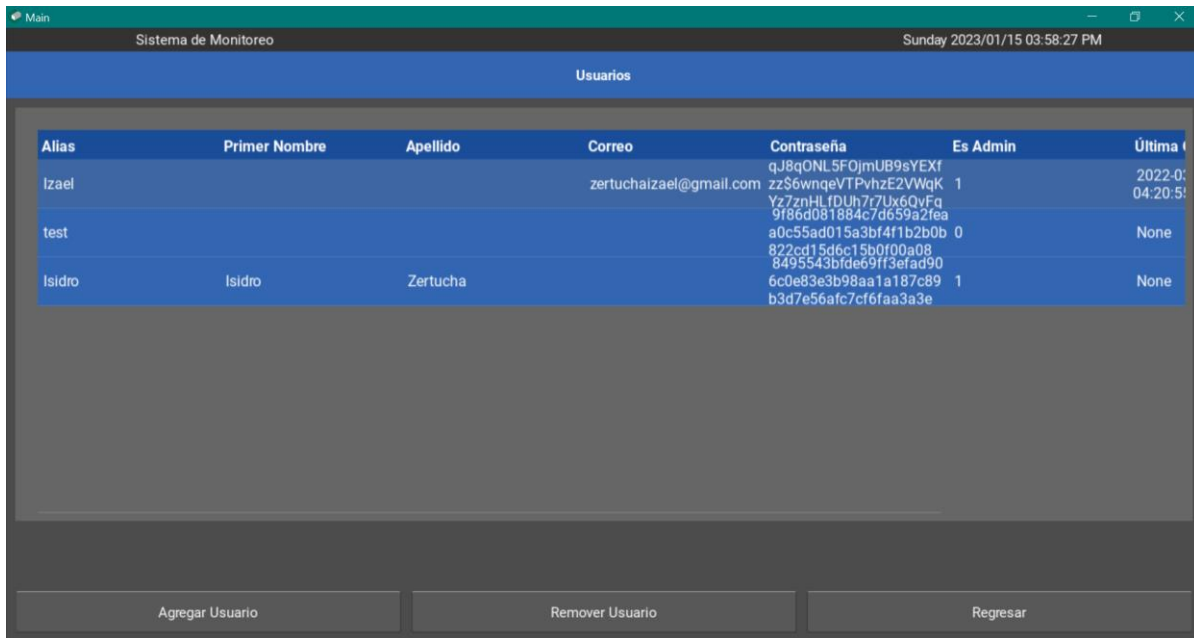


Fig.12 Pantalla de gestión de cuentas administrativas

Nombre	Fecha	Descripción	Lugar
Estado de alarma	2023-01-23 11:46:55.743458	Se ha detectado una alarma	Bomba de agua #3
Arranque	2023-01-23 11:46:55.727465	Se ha recibido una confirmación de marcha	Bomba de agua #3
Estado de alarma	2023-01-23 11:44:28.236401	Se ha detectado una alarma	Bomba de agua #3
Arranque	2023-01-23 11:44:28.225406	Se ha recibido una confirmación de marcha	Bomba de agua #3
Estado de alarma	2023-01-23 11:28:40.479366	Se ha detectado una alarma	Bomba de agua #3

Fig.13 Registro de eventos

ID	Nombre	Fecha Ocurrencia	Fecha Detectado	Fecha Resuelto	Imagen	Desc
7	Tanque vacío	2023-01-23 11:23:32.950124	2023-01-31 10:54:53.152624	None	src/alarm.png	No h
6	Cisterna vacía	2023-01-23 11:23:32.939130	2023-01-31 10:54:53.152624	None	src/alarm.png	No h cister bomt

Fig.14 Alarmas activas

Main Sistema de Monitoreo Wednesday 2023/02/08 02:31:04 PM

Sistema Hidráulico

Refrescar

ID	Nombre	Fecha Ocurrencia	Fecha Detectado	Fecha Resuelto	Imagen	Desc
6	Cisterna vacía	2023-01-23 11:23:32.939130	2023-01-31 10:54:53.152624	None	src/alarm.png	No h cister bomt
7	Tanque vacío	2023-01-23 11:23:32.950124	2023-01-31 10:54:53.152624	None	src/alarm.png	No h
3	Cisterna vacía	2023-01-19 13:24:02.487941	2023-01-19 13:51:28.905786	2023-01-19 13:53:19.722558	src/alarm.png	No h cister bomt
4	Cisterna vacía	2023-01-19 13:50:26.664344	2023-01-19 13:51:28.905786	2023-01-19 13:53:08.634994	src/alarm.png	No h cister bomt
5	Tanque vacío	2023-01-19 13:50:26.680335	2023-01-19 13:51:28.905786	2023-01-19 13:51:38.422448	src/alarm.png	No h
2	Cisterna vacía	2023-01-19 13:10:59.549600	2023-01-19 13:11:12.172988	2023-01-19 13:53:30.164918	src/alarm.png	No h cister bomt
		2020-01-19	2021-01-19			No h

Eventos Alarmas Activas Histórico de Alarmas Controlar Bomba Permisivos Histogramas Regresar

Fig.15 Histórico de alarmas

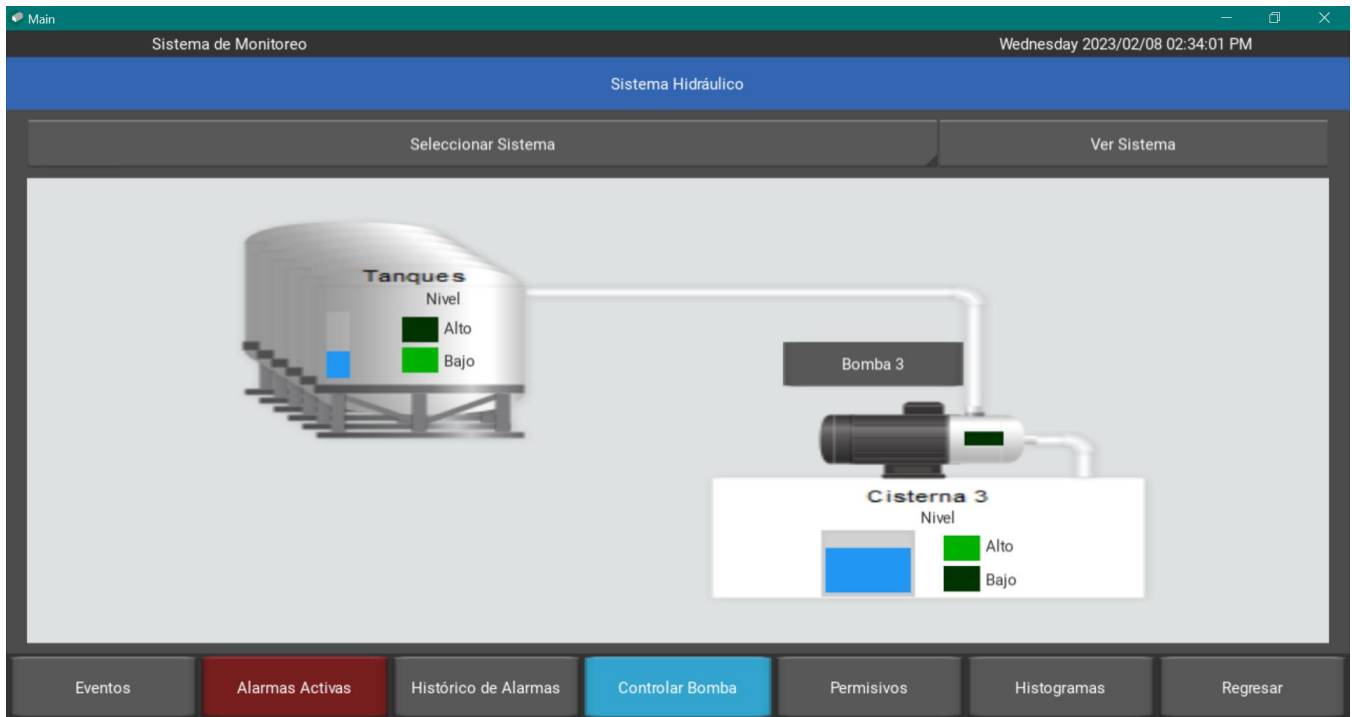


Fig.16 Control de bomba

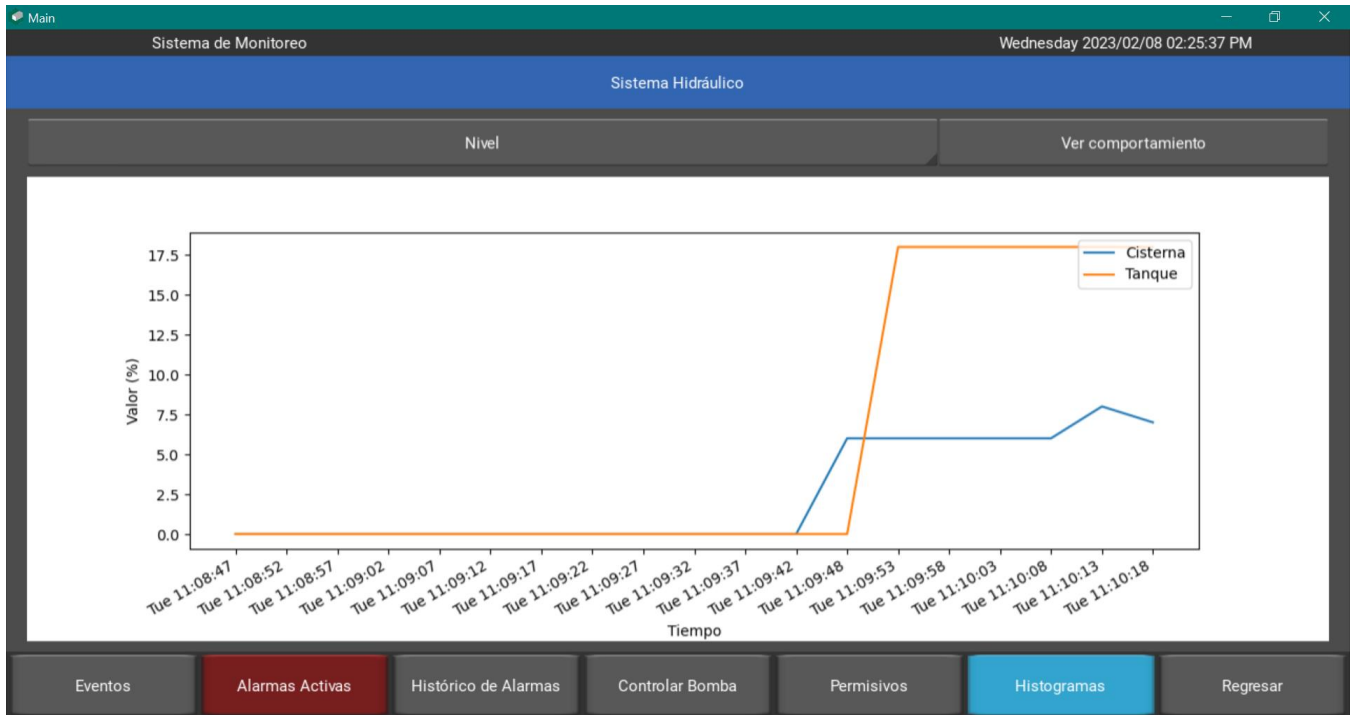


Fig.17 Histograma con el comportamiento de la variable nivel

```

main.py - C:\Users\lrael\Documents\PycharmProjects\VisualSCADA Hospital\main.py [3.9.10]
File Edit Format Run Options Window Help

    host = self.mysql_host,
    port= self.mysql_port,
    user= self.mysql_user,
    passwd= self.mysql_passwd,
    database= self.mysql_database

    )
    mycursor = mydb.cursor()
    sql = 'UPDATE `monitoreo_alarma` SET fecha_detectado=%s WHERE `fecha_detectado` IS NULL'
    values = [datetime.now()]
    mycursor.execute(sql, values)
    mydb.commit()
    mycursor.close()
    mydb.close()
except InterfaceError:
    print("DB ERROR Declare")
    return
# Refrescar cambios
self.refresh_now()

def change_screen(self, instance):
    # print(self.root.ids.scrn_mgr_main.current_screen.ids.scrn_mgr.current_screen)
    if instance.text == 'Alarmas' and instance.state == 'down':
        self.root.ids.scrn_mgr_main.current_screen.ids.scrn_mgr.current = 'scrn_alarm'
    elif instance.text == 'Editar Usuarios':
        self.root.ids.scrn_mgr_main.current_screen.ids.scrn_mgr.current = 'scrn_content'
    elif instance.text == 'Controlar Bomba':
        self.root.ids.scrn_mgr_main.current_screen.ids.scrn_mgr.current = 'scrn_pump'
    elif instance.text == 'Permisivos':
        self.root.ids.scrn_mgr_main.current_screen.ids.scrn_mgr.current = 'scrn_perm'
    elif instance.text == 'Eventos':
        self.root.ids.scrn_mgr_main.current_screen.ids.scrn_mgr.current = 'scrn_event'
    else:
        pass
    #self.root.ids.scrn_mgr_main.current_screen.ids.scrn_mgr.current = 'scrn_analysis'

def open_popup(self, instance):
    content = P()
    # content.add_widget(Label(text=self.seconds_string))
    content.ids.status_label.text = self.seconds_string
    popup = Popup(title="Control de Bomba", content=content, size_hint=(None,None), size=(400,200), auto_dismiss=True, separator_color=[0, .300, .250, 1])
    popup.open()
    
```

Fig.18 Fragmento de código fuente de la pantalla HMI

Viabilidad del diseño

En la Tabla 1 se evidencian los precios en USD para cada sistema. La ventaja de esta propuesta consiste en que al desarrollar el software desde cero y sustituir el uso de PLC industrial por hardware libre, no es necesario adquirir ni mantener una licencia.

Tabla 1. Precio de la propuesta por cada componente

Producto	Precio unitario de los elementos empleados en la solución	Cantidad	Fabricante
M-DUINO PLC Arduino Ethernet 21+	220	1	Industrial Shield
Panel de accionamiento (ver Tabla 2)	156.87	1	Tecnoind
Fuente 110V a 24V	46.27	1	Xypower
Sensor de nivel GLT501	30	2	GAMICOS
Total	483.14		

Tabla 2. Precio estimado del panel de accionamiento eléctrico

Código	Descripción	Unidad de medida	Cantidad	Precio unitario	Importe
4621-2-6-6640024CH	Panel de adosar a la pared de 400x300x200 (nxw5-4030/20)	U	1	48.12	48.12
4620-2-2-3550	Contactor NCI 2510 230V 50/60HZ 25A	U	1	8.5	8.5
4620-2-6-8112	Relé Térmico NR2-2517-25A	U	1	5.52	5.52
4621-2-1-A179871CH	Interruptor automático modular NB1-63H 3P 3D 25A curva C 10KA	U	1	11.39	11.39
4621-2-B-P574789CH	Pulsador verde-rojo iluminado (NP2-BW8465 230V LED)	U	2	5.32	10.64
4621-2-1-3574049CH	Indicador luminoso color rojo (NP2-BV64 220 LED)	U	1	2.78	2.78
4621-2-1-3574048CH	Green indicador (NP2-BV63 220 LED)	U	1	2.79	2.79
4621-2-2-0573982CH	Selector manual de 3 posiciones	U	1	2.27	2.27

4425-1-0-5549049CH	2NO (NP2-BD33) Bornera de conexión de 6 mm ² color Gris	U	10	1.17	11.7
4622-0-0-430X25	Canaleta de cableado con tapa para cuadros 30x25	U	1	5.96	5.96
4621-5-7-3778	Botón Emergencia NP2 Rojo BS64z	U	1	1.83	1.83
042E-R-K-PUNIVFC654	Perfil universal	U	1	25.61	25.61
123A-B-B-0TP40DA	Cable 12 AWG Monoconductor	m	100	0.1208	12.08
4621-2-0-4257022CH	Contacto auxiliar frontal 1NO+1NC (F4-11)	U	4	1.92	7.68
Total					156.87

4. Conclusiones

La propuesta presentada permite la implementación de un sistema de monitoreo automatizado para la el hospital pediátrico La Balear, con un registro histórico de eventos y alarmas respaldados en una base de datos. Este sistema fue creado con software y hardware libre, gracias a esto puede reutilizarse en otras instituciones o locales, abaratando costos al país y garantizando la soberanía tecnológica. Se concibió un enfoque sencillo en el diseño, facilitando la capacitación necesaria para que el personal técnico se pueda adiestrar en poco tiempo. Se ha empleado la plataforma de desarrollo Arduino para el controlador propuesto, el software Python 3.9.10, la base de datos MySQL y la herramienta de simulación Proteus. El uso de una segunda aplicación que corre al mismo tiempo en la PC, desarrollada en Python y con funciones de servidor a través del estándar de comunicación OPC-UA, facilita y normaliza las tareas de programación del código asociadas a la comunicación con el controlador Arduino.

Referencias

1. León, T., Luis J., *Evaluación de la eficiencia de los controladores Arduino Mega y Seimens Logo 230RC en procesos industriales*, 2017. Escuela Superior Politécnica de Chimborazo. Riobamba.
2. Jaramillo, M., Ernesto P., *Implementación de un módulo de simulación de procesos de control y automatización usando señales digitales y analógicas con interfaz de comunicación EasyPort para el Laboratorio de Control y Manipulación Automática*, 2019. Escuela Superior Politécnica de Chimborazo. Riobamba.
3. Córdova, P., José R., *Evaluación del rendimiento de la transferencia de imágenes entre los lenguajes de programación Java y Python*, 2018. Escuela Superior Politécnica de Chimborazo. Riobamba.

4. Guevara, A., Ximena, C., *Desarrollo de un sistema académico orientado a la web para la Unidad Educativa “Juan de Velasco” utilizando Symfony y MySQL*, 2016. Escuela Superior Politécnica de Chimborazo. Riobamba.