https://doi.org/10.37815/rte.v37nE1.1357
Original articles

Enhancing SCADA System Availability: A Case Study of Microgrid Universidad de Cuenca

Mejora de la Disponibilidad del Sistema SCADA: Caso de Estudio Microrred de la Universidad de Cuenca

Lourdes Gutierrez-Otavalo[1] https://orcid.org/0009-0009-6162-8081, Claudia Padilla-Guaman[1] https://orcid.org/0009-0000-7240-4547, Juan Diego Belesaca[1] https://orcid.org/0000-0001-8609-0358, Fabian Astudillo-Salinas[1] https://orcid.org/0000-0001-7644-0270, Edisson Villa-Ávila[1] https://orcid.org/0000-0002-2766-5913

[1]*Universidad de Cuenca, Departamento de Ingeniería Eléctrica, Electrónica y Telecomunicaciones*, Cuenca, Ecuador
lourdesv.gutierrez@ucuenca.edu.ec, claudia.padillag@ucuenca.edu.ec, juan.belesaca@ucuenca.edu.ec, fabian.astudillos@ucuenca.edu.ec, edisson.villa2809@ucuenca.edu.ec

Sent: 2025/06/29
Accepted: 2025/08/24
Published: 2025/10/15

Abstract

Ensuring the uninterrupted operation of Supervisory Control and Data Acquisition (SCADA) systems is critical for industrial environments that rely on continuous monitoring and control of equipment. However, in the Microgrid Laboratory at the University of Cuenca, routine maintenance procedures have revealed a significant vulnerability: the SCADA system temporarily loses communication and control capabilities when some of the devices are powered down, resulting in forced system downtime. This paper investigates practical strategies to enhance SCADA system availability by addressing this critical limitation. Three hypotheses were explored: first, an issue in network redundancy management; second, with disconnecting of a device from the fiber-optic ring network; and third, programming errors within the LabVIEW-developed SCADA application. Experimental results demonstrated that these approaches, particularly the software-level improvements in LabVIEW, successfully maintained system control and communication during maintenance. The proposed solutions offer a scalable and cost-effective pathway to increase the resilience of SCADA systems in microgrid environments.

Keywords: microgrid, SCADA, resilience, maintenance, OT network.

Summary: Introduction, Materials and Methods, Results and Discussion, Conclusions and Future Work.

How to cite: Gutierres-Otavalo, L., Padilla-Guaman, C., Belesaca, J. D., Astudillo-Salinas, F. & Villa-Ávila, E. (2025). Enhancing SCADA System Availability: A Case Study of Microgrid Universidad de Cuenca. *Revista Tecnológica - Espol, 37(E1), 25-37.* https://doi.org/10.37815/rte.v37nE1.1357

Resumen

Garantizar el funcionamiento ininterrumpido de los sistemas de supervisión, control y adquisición de datos (SCADA) es fundamental para los entornos industriales que dependen de la supervisión y el control continuos de los equipos. Sin embargo, en el Laboratorio de Microrredes de la Universidad de Cuenca, los procedimientos de mantenimiento rutinarios han revelado una vulnerabilidad significativa: el sistema SCADA pierde temporalmente sus capacidades de comunicación y control cuando se apaga alguno de los dispositivos, lo que provoca una interrupción forzosa del sistema. Este artículo investiga estrategias prácticas para mejorar la disponibilidad del sistema SCADA, abordando esta limitación crítica. Se exploraron tres hipótesis: en primer lugar, un problema en la gestión de la redundancia de la red; en segundo lugar, un problema en la desconexión de un dispositivo de la red de fibra óptica en anillo; y, en tercer lugar, errores de programación en la aplicación SCADA desarrollada con LabVIEW. Los resultados experimentales demostraron que estos enfoques, en particular las mejoras a nivel de software en LabVIEW mantuvieron con éxito el control y la comunicación del sistema durante el mantenimiento. Las soluciones propuestas ofrecen una vía escalable y rentable para aumentar la resiliencia de los sistemas SCADA en entornos de microrredes.

Palabras clave: microrred, SCADA, resiliencia, mantenimiento, red OT.

Introduction

Supervisory Control and Data Acquisition (SCADA) systems play a pivotal role in modern industrial automation, enabling the real-time monitoring and control of a wide array of physical processes. These systems depend on networks of Operational Technologies (OT), frequently established through wired or wireless infrastructures, to collect, process, and visualize critical operational data from field devices such as sensors, inverters, and programmable controllers as Programmable Logic Controller (PLC) y Programmable Automation Controller (PAC) (Chica Gallardo & Guamán Argudo, 2017; Daneels & Salter, 1999). The data acquired includes essential parameters such as power output, voltage, temperature, and frequency, which help optimize the performance and ensure system stability (Montesdeoca Chuva & Buñay Moncayo, 2021).

In recent years, the global shift toward sustainable energy has intensified the relevance of SCADA systems, particularly in smart grid and microgrid environments. The Microgrid Laboratory at the University of Cuenca was established as part of a national initiative aimed at supporting innovation and energy transition. This laboratory serves as a technological testbed for integrating renewable energy sources, storage systems, and electric vehicle infrastructure (Laboratorio de Microrred, 2021). Within this environment, a SCADA system implemented in Laboratory Virtual Instrument Engineering Workbench (LabVIEW) enables centralized management of a set of devices interconnected through a fiber-optic ring network (Montesdeoca Chuva & Buñay Moncayo, 2021).

Notwithstanding the advantages offered by SCADA systems, they are susceptible to interruptions during service operations. In the Microgrid Laboratory, power disconnection for routine maintenance results in the temporary loss of connectivity between SCADA and field devices. In turn, it triggers system-wide alerts and leads to forced shutdowns. This behavior not only compromises operational continuity but also limits the flexibility and resilience expected from intelligent energy systems (Ujvarosi, 2016). The existing infrastructure is deficient in its lack of robust fault-tolerant mechanisms capable of sustaining communication and control if individual components are taken offline. This poses a substantial challenge in the context of research laboratories and industrial facilities, where high availability and minimal downtime are paramount. Specifically, the current SCADA implementation exhibits a gap caused by GUI blocking and historical data loading during maintenance, which often leads to forced restarts. Therefore, the measurable objective of this study is to prevent such restarts and achieve a recovery time of less than 30 minutes.

The objective of this study is to address the challenge of enhancing the availability of SCADA systems during maintenance procedures. To this end, the study will evaluate and implement three complementary strategies. Initially, network redundancy protocols such as Spanning Tree Protocol (STP), Rapid STP, and Turbo Ring are configured on Weidmüller switches (Weidmüller, 2014) to mitigate single points of failure. Secondly, a physical network bypass is proposed through reconfiguration of the optical fiber layout, enhancing the system's fault tolerance. Thirdly, an analysis and adaptation of the LabVIEW-based SCADA architecture is conducted to overcome the limitations of its client-server communication model (Loayza, 2010; Zhou et al., 2009). Through experimental validation and technical adjustments, this work contributes to the development of more resilient SCADA systems for microgrid applications. It is anticipated that these outcomes will provide a foundation for future research and practical implementations in analogous laboratory and industrial contexts.

The remainder of this paper is organized as follows: The second section of the text outlines the materials and methodologies employed to implement the proposed strategies. In Section 3, the experimental results and system performance analysis are presented. Finally, Section 4 discusses the conclusions and suggests directions for future work.

Materials and Methods

This section presents the experimental setup and methodological approach used to investigate SCADA system interruptions during servicing activities in the Microgrid Laboratory at the University of Cuenca. The research combines a case study perspective with empirical observation and the design of controlled experiments to analyze the system's performance. The present study focuses on the SCADA system developed in LabVIEW, the OT network, and the communication infrastructure which is based on Weidmüller switches and a fiber-optic ring.

Experimental environment

The experiments were conducted in the Microgrid Laboratory at the University of Cuenca, a facility designed for research and testing of intelligent energy systems. The laboratory is equipped with a hybrid power generation infrastructure, comprising photovoltaic panels, a diesel generator, a liquefied petroleum gas (LPG) generator, a microturbine, and battery-based storage systems. These components are integrated into a microgrid configuration that supports both grid-connected and islanded modes of operation.

The interconnectedness of all equipment is facilitated by an OT network, which employs a fiber-optic ring topology to ensure high-speed data transmission and enhanced resilience. Programmable Logic Controllers (PLCs) are deployed across multiple subsystems, referred to as Advanced Power Integrated (API) station controllers. Each API is responsible for specific monitoring and control tasks. The communication infrastructure between the SCADA system and the field devices is established using the Modbus TCP/IP protocol. The utilization of Weidmüller switches, which are meticulously managed, plays a pivotal role in ensuring the seamless flow of data. These switches facilitate the continuity of data exchange, even during periods of physical reconfiguration or device-level failures.

A summary of the main components and characteristics of the microgrid environment is presented in Table 1. This environment provides a realistic and intricate platform assessing the behavior of the SCADA system under various maintenance scenarios, thereby simulating real-world disruptions and validating proposed solutions.

Table 1

Overview of the Microgrid Laboratory Infrastructure

COMPONENT	DESCRIPTION
Location	Microgrid Laboratory, University of Cuenca
Power sources	Photovoltaic panels, diesel generator, LPG generator, microturbine
Storage system	Battery-based energy storage units
Control devices	Programmable Logic Controllers (PLCs)
Communication protocols	Modbus TCP/IP, OPC UA
Network topology	Fiber-optic ring and ethernet to end devices
Network equipment	Managed Weidmüller switches, among others
SCADA platform	LabVIEW with Data Logging and Supervisory Control (DSC) module
Monitoring tools	NI MAX, Graylog, Scapy
Virtualization environment	VMware Workstation Pro (hosting Windows Server and Ubuntu Server instances)

SCADA System and Software Stack

The SCADA system utilized in the Microgrid Laboratory was developed in LabVIEW, employing the Data Logging and Supervisory Control (DSC) module to facilitate real-time monitoring, historical data visualization, alarm management, and automated control functions. The architecture employs a client-server model and is hosted on a virtualized Windows Server 2012 R2 instance.

The Modbus TCP/IP protocol facilitates the communication with field devices. The system gathers and logs real-time data, including power, temperature, and frequency, from multiple PLCs distributed across the microgrid. All acquired data are stored in the Citadel database, which is integrated into the SCADA system.

Additionally, NI Measurement & Automation Explorer (NI MAX) is employed for sensor configuration and real-time diagnostics. The management of all virtualized components is facilitated by VMware Workstation Pro, which serves as the host for both the LabVIEW SCADA and the log server environment. This process aligns with the methodically outlined setup in the laboratory's internal documentation.

Methodology

A mixed-methods approach was adopted to investigate the causes of SCADA system unavailability during equipment maintenance. This approach integrates three complementary strategies: the case study method, the empirical method, and the design of experiments (DoE). These strategies were outlined in the preliminary analysis and technical interactions with the Microgrid Laboratory staff.

The case study method explored the operational dynamics of the SCADA system in a real-world environment by focusing on specific incidents observed during serving activities. This method provided a detailed understanding of the system’s architecture, communication flows, and failure points by analyzing its behavior under normal and interrupted conditions. It was essential for formulating initial hypotheses, particularly in the absence of preexisting theoretical models regarding SCADA unavailability in fiber-optic ring networks.

The empirical method was used to directly observe and analyze system performance, with a focus on data flow continuity, communication stability, and device responsiveness. Through hands-on testing and observation, behaviors such as loss of connectivity, alert generation, and SCADA restarts were documented. These observations provided real-time evidence of system behavior, validating symptoms reported by operators, and narrowing down potential failure scenarios.

The design of experiments (DoE) methodology was employed to formally test three hypotheses derived from the case study. Each hypothesis was examined through targeted experiments:

- H1: Lack of proper network redundancy management using Weidmüller switches causes SCADA unavailability.
- H2: Disconnecting a device from the fiber-optic ring network causes system unavailability.
- H3: Programming errors within the LabVIEW-developed SCADA application are responsible for the system's failure during maintenance.

Each experiment was carefully structured to isolate relevant variables and capture measurable outputs. Network traffic was recorded using Scapy and Graylog, and system behavior was evaluated using log data, packet captures, and LabVIEW diagnostic outputs. These experiments enabled the data-driven validation or rejection of the proposed hypotheses, forming the basis for the technical solutions presented in subsequent sections.

Experimental design and procedures

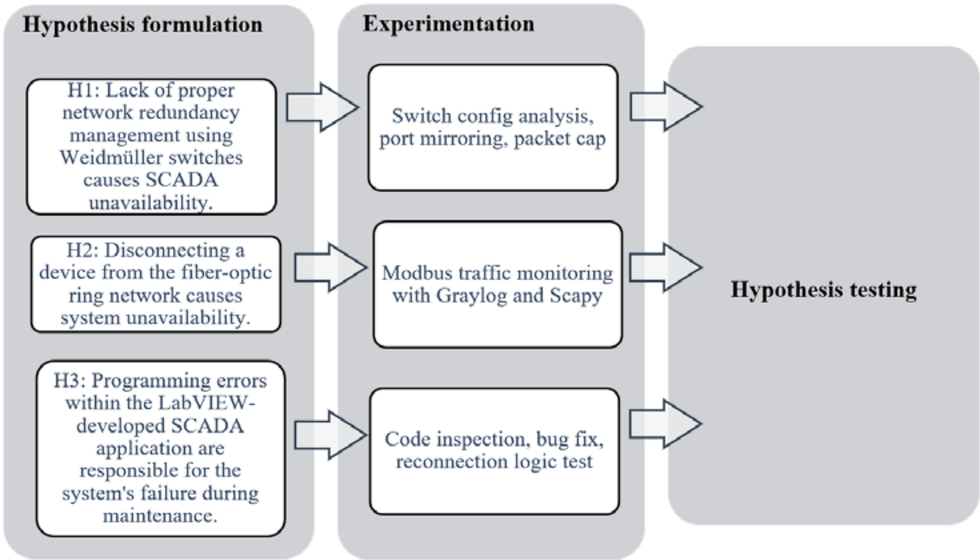
A series of three experiments were conducted to validate the hypotheses formulated within the methodological framework. Each experiment was meticulously designed to assess a particular aspect of the SCADA system's performance under maintenance-related disruptions. These experiments utilized tools and data collection techniques deemed suitable for real-time industrial environments.

The initial experiment aimed to determine whether inadequate redundancy management within the network contributed to the occurrence of SCADA unavailability. An examination of the Weidmüller switch configuration was conducted to verify the status of redundancy protocols, such as Turbo Ring, as per the manufacturer's documentation (Weidmüller, 2014). A port mirroring function was configured to capture and analyze traffic behavior during a simulated device disconnection. Network packets were captured using Wireshark and tcpdump, with a focus on communication continuity and protocol behavior. The analysis indicated that redundancy mechanisms were operational and efficacious, enabling communication to persist even in the event of a device's disconnection from the ring topology.

The second experiment evaluated the impact of disconnecting a single Programmable Logic Controller (PLC) on the behavior of the remaining devices. To this end, a virtual Ubuntu server was configured to collect and classify Modbus Query and Response messages using a custom Python script (Guachichullca Guamán, 2024) and the Scapy library. The results, which were visualized in real time through Graylog (Centralización y Análisis de Eventos de Seguridad Con Graylog, n.d.), demonstrated that while the disconnected PLC ceased sending data, the others continued transmitting without interruption, thereby refuting the second hypothesis.

Figure 1

Experimental validation flowchart for SCADA availability analysis



The third experiment focused on the identifying flaws in the SCADA software developed in LabVIEW. A thorough code inspection revealed issues in the handling of TCP connections and excessive startup delays due to unfiltered historical data loading. These issues were addressed by modifying the reconnection logic and eliminating superfluous initializations. Furthermore, unused trend monitoring blocks were eliminated, and database optimization was executed using SQL Server Management Studio. After the implementation of these adjustments, simulated serving procedures substantiated stable SCADA operation and sustained device communication.

To that end, Graylog (a centralized logging platform) as a means of monitoring system behavior and validating data flows was implemented. The platform is deployed on an Ubuntu 22.04 LTS server. The utilization of Scapy was employed for packet-level inspection of Modbus traffic between devices.

A summary of the experimental logic, including the hypotheses, procedures, and outcomes, is illustrated in Figure 1. Each experiment contributed to refining the understanding of the system's limitations and led to actionable improvements that enhanced SCADA availability in the Microgrid Laboratory environment.

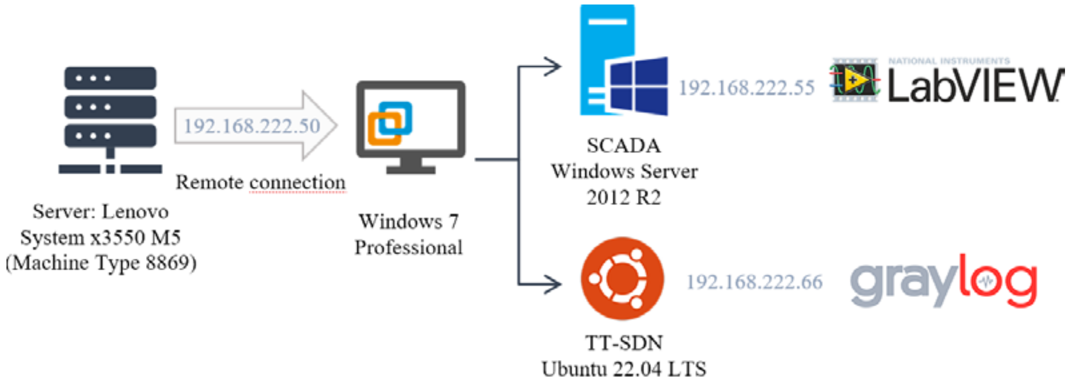
Results and Discussion

This section presents the results obtained through the experimental procedures described in the methodology. The primary objective of this study was to determine the underlying cause of the SCADA system's unavailability during maintenance operations in the Microgrid Laboratory. To this end, three hypotheses were formulated and evaluated through a series of targeted experiments.

The hypotheses addressed the potential causes of system failure, as described in the Methodology section. The testing of each hypothesis was conducted using the tools, protocols, and procedures outlined in Section 2. The experimental process entailed traffic monitoring, device disconnection, and software analysis. The ensuing subsections provide a comprehensive overview of the findings, incorporating both quantitative and qualitative data.

Figure 2 presents a diagram that summarizes the logical access topology of the SCADA system. This figure provides a comprehensive overview of the interconnection between the SCADA server, field devices, and network components.

Figure 2
Logical access topology of the SCADA system



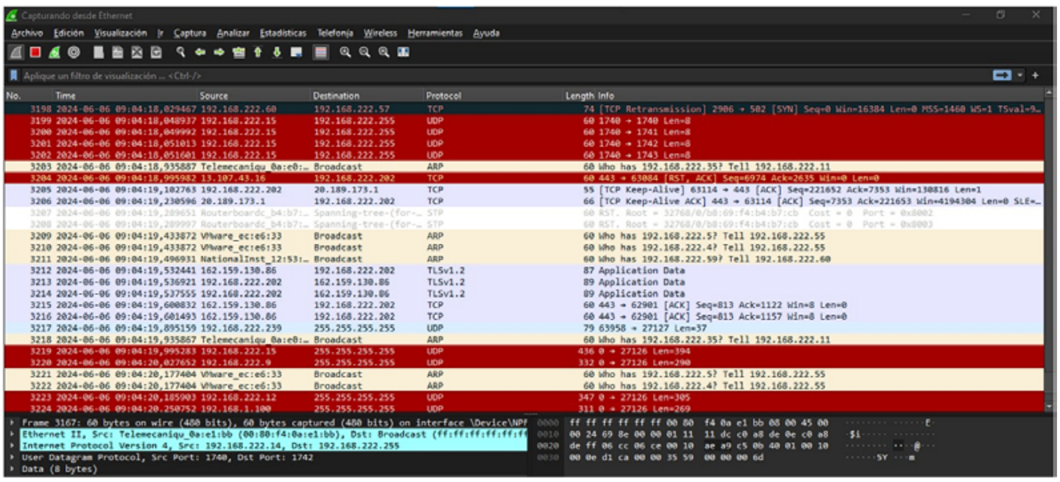
Hypotheses testing and experimental results

The first hypothesis H1 proposed that the unavailability of the SCADA system was due to a failure in managing network redundancy. More specifically, the failure involved the configuration of the fiber-optic ring that connects the OT devices via Weidmüller switches.

To validate this hypothesis, an experiment was conducted in which one of the switches in the fiber-optic ring was disconnected. Under the assumption of insufficient redundancy, the expected outcome was the loss of communication with devices located downstream of the disconnection point. However, packet captures obtained using Wireshark and tcpdump revealed continued data transmission between the SCADA system and the remaining field devices despite the disconnection.

These results indicate that the redundancy mechanisms were active and functioning properly. Ethernet connections between all peripheral switches and the central switch ensured uninterrupted communication by effectively bypassing the broken ring segment.

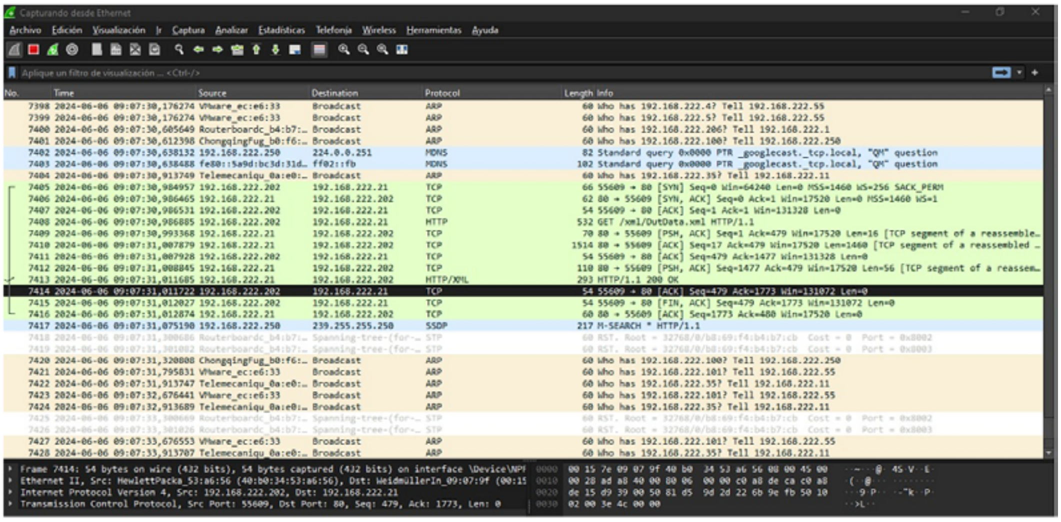
Figure 3
Packet capture before disconnection in Weidmüller switch



Figures 3 and 4 illustrate the traffic behavior before and after the disconnection. Figure 3 illustrates the packet flow prior to the disconnection occurred. Figure 3 confirms that data from the same devices continued to be received after the switch was disconnected, thus validating the redundancy. Consequently, Hypothesis H1 was rejected, and it was determined that the SCADA system's availability was not affected by the fiber-optic ring topology or redundancy configuration.

Figure 4

Packet capture after disconnection from Weidmüller switch



The second hypothesis H2 suggested that disconnecting a Programmable Logic Controller (PLC) from the fiber-optic ring network would interrupt the availability of the SCADA system. This assumption was based on the idea that the ring topology might propagate failure to the rest of the network.

The PLC in API 3 was selected for disconnection due to its ease of physical access and was chosen to evaluate this hypothesis. The test involved recording the number of Modbus TCP queries and responses per minute for each PLC using Graylog before and after disconnection. These values were obtained through the log server connected to the SCADA system.

Figure 5

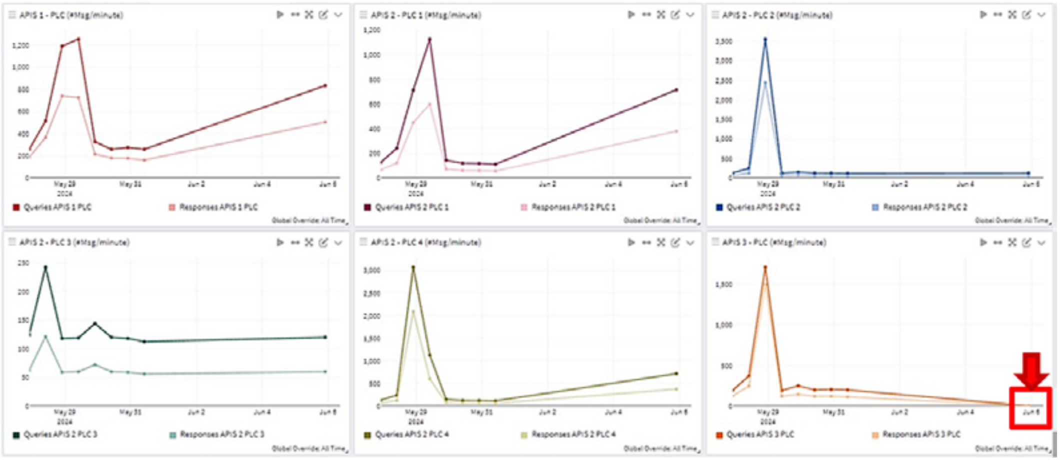
Message count before disconnection in APIs and PLCs

Figure 5 illustrates the number of Modbus messages before disconnection. All PLCs, including API 3, exhibit consistent data transmission and reception. After the disconnection of the API 3 PLC, the Modbus message count for that device decreases to zero. Concurrently, the communication patterns exhibited by the other PLCs remained consistent with their expected behaviors. Figure 6 further corroborates this finding by demonstrating the time-series behavior of query and response counts. Furthermore, Graylog furnished additional corroboration in the form of tabular logs during the disconnection process. Notably, communication with API 3 has been completely terminated, while the remaining devices remain unaffected.

These results clearly indicate that the loss of a single PLC does not compromise the network´s operation. Therefore, Hypothesis H2 is rejected as well, since the SCADA system maintains full availability and data integrity when a field device is temporarily disconnected.

Figure 6

Message count over time in APIs and PLCs



After rejecting the initial two hypotheses concerning the physical network infrastructure, the investigation focused on a third hypothesis H3. This hypothesis posited that the unavailability of the SCADA system was attributable to internal programming failures within the LabVIEW-based application. The validity of this hypothesis was ascertained through systematic testing of both the original and modified versions of the SCADA software.

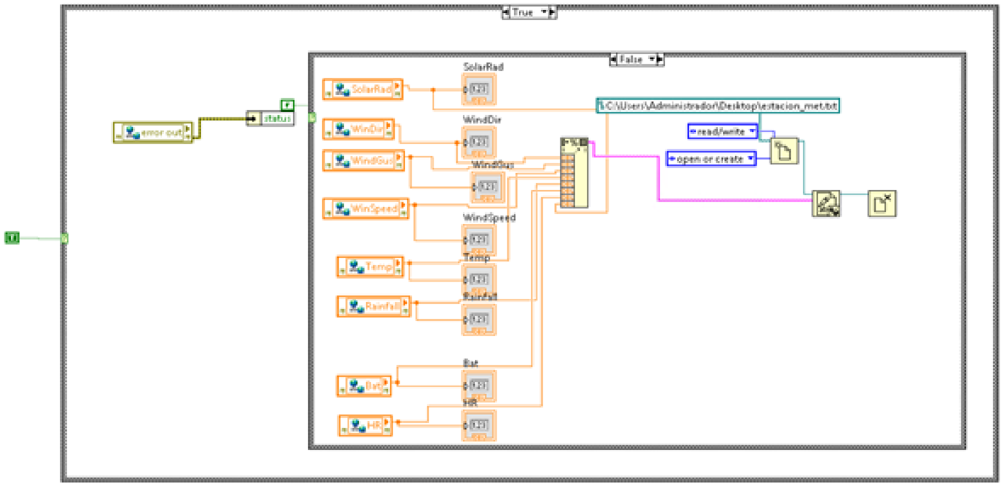
In the initial version, several issues were identified through a meticulous examination of the code and a series of interactions with system operators. These included excessive startup time, system crashes following temporary PLC disconnections, and the inability to automatically recover from Modbus communication errors. Upon disconnection of the PLC from API 3, the system exhibited a Modbus I/O error with code -1967353902, signifying a loss of communication with the slave device. The interface was locked, necessitating a manual restart, which had a substantial impact on system availability. Furthermore, the system's initialization process was found to require up to eight hours due to the loading of historical data from the database. This phenomenon was attributed to the incorporation of all historical trend traces since the system's deployment.

To remediate these issues, a series of enhancements were implemented. First, a reconnection protocol was programmed using a conditional case structure, allowing automatic recovery from TCP disconnections without requiring for user intervention. After, the graphical user interface was modified to eliminate unnecessary error prompts, thereby allowing operators to retain system control during serving. Finally, the process of loading historical data was enhanced by removing of the Historical Trend function and utilizing of SQL Server Management Studio (SSMS) to restrict data to a specific date range. Figure 7 shows that the program was implemented in LabVIEW to resolve the issues. After reconnection, the PLC resumes normal communication without requiring a full SCADA restart.

Finally, the system's startup time was reduced to just a few minutes, greatly improving operational readiness. Validation of these improvements confirms acceptance of Hypothesis H3 and establishes that the root cause of SCADA unavailability was related to software design flaws in the original LabVIEW implementation.

Figure 7

Final program that corrects the error in the TCP connection



Hypotheses validation

A summary of the validation process based on the experimental results obtained for each hypothesis is presented in Table 2. The table indicates whether each hypothesis was accepted or rejected and if it contributed directly to the final solution that improved the availability of the SCADA system.

Table 2

Hypotheses Validation Summary

HYPOTHESIS	VALIDATION RESULT	INCLUDE IN FINAL SOLUTION
H1: Lack of network redundancy management causes SCADA unavailability	Rejected	No
H2: PLC disconnection causes fiber ring failure	Rejected	No
H3: LabVIEW programming causes SCADA unavailability	Accepted	Yes

The results show that hypotheses H1 and H2, which attributed SCADA failures to network infrastructure issues, were rejected. The experiments demonstrated that the network's redundancy mechanisms and device independence were functioning correctly. In contrast, H3 was accepted because detailed software-level analysis confirmed that communication errors, system lockups, and performance delays were caused by limitations in the original LabVIEW implementation.

The software enhancements introduced in response to hypothesis H3 were successfully validated in a laboratory environment and are now part of the operational SCADA solution.

Complete system discussion

A comparative analysis was conducted between the original and updated versions of the SCADA system software to evaluate the effectiveness of the improvements made. The analysis focused on key performance indicators, including system availability during PLC disconnection, interface responsiveness, error handling, recovery behavior, and system startup time.

The original version of the system failed to handle temporary communication losses. When a PLC was disconnected, the SCADA system triggered a critical communication error followed by a Modbus I/O failure message. This error rendered the graphical user interface inoperative and required the system to be manually restarted. Even after reconnecting the PLC and restoring communication at the network level, the SCADA system remained unresponsive until a complete restart was performed. Additionally, the initialization time for the SCADA system in the original version was highly variable, ranging from several minutes to eight hours, depending on the volume of historical data loaded from the Citadel database. This delay reduced operational availability significantly and hindered routine maintenance procedures.

In contrast, the updated version of the SCADA software developed as part of this research demonstrated marked improvements in all evaluated aspects. Upon PLC disconnection, the system issued a non-intrusive visual alarm in SCADA without blocking operator interaction. The communication diagram (Figure 2) remained active, and other system components continued to function normally.

Once the PLC reconnected, Graylog confirmed the restoration of Modbus message flow, and the SCADA system automatically resumed normal operation without requiring a restart. Most notably, optimizing database access and removing unnecessary historical trend loading reduced the system's startup. The mean duration of system startup has been reduced from eight hours to 12 minutes, with a standard deviation of ±2 minutes, based on 20 runs. These findings are consolidated in Table 3, which summarizes the performance differences between the original and updated SCADA software versions across key indicators. Overall, the comparison confirms that the implemented software changes significantly increased the robustness and resilience of the SCADA system during both planned and unplanned serving events.

Table 3

Comparative analysis of the original vs. updated SCADA systems

INDICATOR	ORIGINAL VERSION	UPDATED VERSION
Startup time	8 hours (variable, depending on historical data loading)	12 ± 2 minutes (20 runs, optimized DB access and filtered historical data)
Behavior on PLC disconnect	Triggered critical error, Modbus I/O failure → forced shutdown	Issued non-intrusive alarm, communication diagram active, other components normal
UI blocking	GUI froze and required manual restart even after PLC reconnection	GUI remained responsive; automatic reconnection without restart
Recovery after PLC reconnection	Required complete restart	Automatic recovery confirmed via Graylog (no restart needed)
Data handling (historical)	Heavy Citadel database loading; caused variability in startup	Selective filtering; unnecessary historical data

Conclusions and Future Work

The present study evaluated the root causes of SCADA system unavailability during maintenance procedures in the Microgrid Laboratory at the University of Cuenca. A mixed method approach was employed, integrating case study analysis, empirical observation, and experimental validation, to test three hypotheses and ascertain the origin of system failures. The study's findings indicated that the physical infrastructure of the OT network, including redundancy protocols and device disconnections, did not contribute to the observed unavailability. Consequently, the initial two hypotheses were refuted.

Conversely, the third hypothesis was corroborated. The analysis revealed that the primary constraints originated from software-level issues within the LabVIEW-based SCADA system. These included improper handling of communication failures, long startup times due to unfiltered historical data loading, and the absence of an automatic reconnection mechanism following temporary PLC disconnections. To remediate these issues, a series of enhancements were implemented, encompassing optimized reconnection logic, user interface improvements , and database filtering. Consequently, the system demonstrated a significant decrease in initialization time, eliminated interface lockups, and maintained full functionality during servicing events. These outcomes demonstrate that the availability of SCADA systems can be significantly enhanced through the implementation of targeted software modifications without the necessity of altering the existing physical infrastructure.

Looking ahead, several directions for future work are identified. These include upgrading the microgrid's hardware infrastructure with faster and more robust components, such as high-performance RAM and solid-state drives, to further enhance system responsiveness. Implementing automated backup systems for the SCADA database would enhance data resilience and reduce system load. More detailed protocol analysis using external tools such as Modbus Poll could provide valuable insights into communication behavior and help refine implementation. It is also important to improve the traceability of maintenance procedures by recording and classifying interventions more systematically. Finally, the integration of emerging technologies such as artificial intelligence and the Internet of Things presents an opportunity to evolve the SCADA system into a more intelligent, predictive, and self-adaptive platform.

Acknowledgements and Declarations

The authors would like to express their gratitude to the Microgrid Laboratory at the Universidad de Cuenca and Department of Electrical, Electronic, and Telecommunications Engineering, which provided the laboratory and equipment for the development of this work. This work is part of the thesis: "ANÁLISIS DE LA FACTIBILIDAD DE MEJORAR LA DISPONIBILIDAD DEL SISTEMA SCADA DURANTE LOS PROCESOS DE MANTENIMIENTO DE LOS EQUIPOS DE LA MICRO-RED DE LA UNIVERSIDAD DE CUENCA".

The authors declare equal contribution and sharing of authorship roles for this publication.

The authors declared that, in the preparation of this article, AI tools were no used.

References

Centralización y análisis de eventos de seguridad con Graylog. (n.d.). Retrieved June 23, 2025, from https://openaccess.uoc.edu/items/1f2bc757-1b7e-43d3-8776-cc9a343964c1

Chica Gallardo, A. P., & Guamán Argudo, J. A. (2017). *Modelo de estado estacionario de la microrred del laboratorio de Balzay de la Universidad de Cuenca* [bachelorThesis]. https://dspace.ucuenca.edu.ec/handle/123456789/28606

Daneels, A., & Salter, W. (1999). What is SCADA? *Proceedings of the 7th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS '99)*. Trieste, Italia: International Conference on Accelerator and Large Experimental Physics Control Systems. https://cds.cern.ch/record/532624/files/mc1i01.pdf

Guachichullca Guamán, B. A. (2024). *Diseño e implementación de una Arquitectura de Ciberseguridad para la Micro-red de la Universidad de Cuenca* [Tesis de pregrado, Universidad de Cuenca]. Repositorio Institucional de la Universidad de Cuenca. https://dspace.ucuenca.edu.ec/handle/123456789/44828

Laboratorio De Microrred. (n.d.). Retrieved June 22, 2025, from https://www2.ucuenca.edu.ec/ingenieria/laboratorios/lab-microrred

Loayza, E. G. (2010). *Desarrollo de una guía práctica para la medición del tráfico de red IP y monitoreo de dispositivos en tiempo real mediante herramientas MRTG y PRTG*. Pontificia Universidad Católica del Ecuador. http://repositorio.puce.edu.ec/bitstream/handle/22000/3421/T-PUCE-3575.pdf?sequence=1&isAllowed=y

Montesdeoca Chuva, D. I., & Buñay Moncayo, D. E. (2021). *Creación de sistemas SCADA para el laboratorio de microred de la Universidad de Cuenca bajo el enfoque de desarrollo dirigido por modelos* [Tesis de pregrado, Universidad de Cuenca]. Repositorio Institucional de la Universidad de Cuenca. http://dspace.ucuenca.edu.ec/handle/123456789/37333

Ujvarosi, A. (2016). Evolution of SCADA systems. *Bulletin of the Transilvania University of Braşov. Series I: Engineering Sciences, 9*(58) (1), 63-68.

Weidmüller. (2014). *Industrial Ethernet managed switches: Manual for managed switches of series ValueLine and PremiumLine*. https://manualzz.com/doc/6766188/manual-managed-weidm%C3%BCller-switches

Zhou, J., Liu, D., Ma, X., & Ye, C. (2009). Application of industry ethernet and configuration software in heating network monitoring system. *2009 WRI World Congress on Computer Science and Information Engineering, CSIE 2009*, 131–134. https://doi.org/10.1109/CSIE.2009.439