

Sustainable Digital Signage with Free Software: An Applied Study with Raspberry Pi in Smart Campuses

Gabriel de Carvalho Barreto*, Vitor Marcelo Mignoni*, Pedro Vinicios Mazo*,
Heitor Scalco Neto* and Alisson Borges Zanetti*

*Instituto Federal Catarinense, Concórdia, Santa Catarina, Brazil

Email: gabrieloakbarreto@gmail.com

Abstract—This paper presents Next Signage, a sustainable digital signage system designed for academic environments, utilizing open-source software, Raspberry Pi microcomputers, and repurposed hardware. The project aims to reduce electronic waste and operational costs while enhancing communication within the IFC Campus Concórdia. We adopted a Model-View-Controller (MVC) architecture to create a scalable and low-resource system. The server-side, developed in PHP, manages content playlists, while the Python-based client, running on a Raspberry Pi, handles media playback. Performance analysis of existing open-source solutions like piSignage and Chromium Kiosk revealed gaps in efficiency and user interface design, justifying the need for a custom solution. Our system demonstrates that a combination of refurbished screens and low-power computing can create an effective and environmentally responsible communication platform, contributing to the goals of a modern smart campus.

Keywords—signage; smart campus; sustainability; Raspberry Pi; open-source software.

I. INTRODUCTION

Digital signage is the use of digital screens, such as TVs and monitors, to display information to the public in a quick and accessible way. These screens are connected to computer systems that allow for easy content management. This technology is more than just a tool for advertising; it can deliver specific content to targeted audiences and create customized visual experiences. This makes digital signage a socially and economically important form of communication. It allows for real-time updates, lowers costs related to paper and printing, and helps reduce visual pollution [1].

However, the production of the hardware needed for these systems has an environmental impact. This study proposes Next Signage, a solution that combines digital signage with sustainable, low-cost practices. Next utilizes Raspberry Pi microcomputers connected to repurposed digital screens to create a digital signage network at the IFC Campus Concórdia. This project aims to support the development of a smart campus by fostering a modern, sustainable environment and

improving the interaction between the academic community and the institution.

In a university setting, signage is a key tool for integrating students and staff into the daily life of the institution. When used effectively, it helps bridge information gaps, especially at the start of a new academic term. However, traditional methods like posters and flyers create negative environmental impacts, from production to disposal. A more efficient and sustainable option is the use of digital signage systems, which are already common in commercial settings [1]. These systems not only avoid the need for physical materials but can also be implemented in line with legislation promoting the reuse of electronic equipment [2].

II. BACKGROUND AND RELATED WORK

A. The Smart Campus Concept

The concept of a “smart campus” involves integrating Internet of Things (IoT) technologies into the university environment to optimize operations and enhance social interaction. A smart campus uses a network of connected devices to collect data and automate processes, leading to more efficient resource management and a better experience for students and staff. According to recent studies, technologies like IoT can be used to monitor and optimize building performance, reduce the campus’s carbon footprint, and improve safety [3]. Digital signage is a key component of a smart campus, as it provides a dynamic and efficient channel for information dissemination. The interaction between digital signage and educational institutions has a positive impact on the user experience by making communication more accessible and fluid [4].

B. Sustainability and E-Waste Reduction

Electronic waste (e-waste) is one of the most significant environmental challenges today. In many countries, the improper disposal of electronics is a widespread problem [5]. The UNEP (the United Nations Environment Programme) estimates 2.4 million tones of e-waste in Brazil every year [6]. Low recycling

rates lead to an increased demand for new hardware, which in turn drives up operational costs and worsens environmental damage. Refurbishing and reusing electronic equipment, such as monitors and computers, is a sustainable alternative that reduces waste and extends the lifespan of devices [7]. This approach aligns with a more conscious and responsible maintenance culture and is a core principle of this project. By repurposing discarded screens, our project directly addresses the e-waste problem and promotes a circular economy within the institution [8].

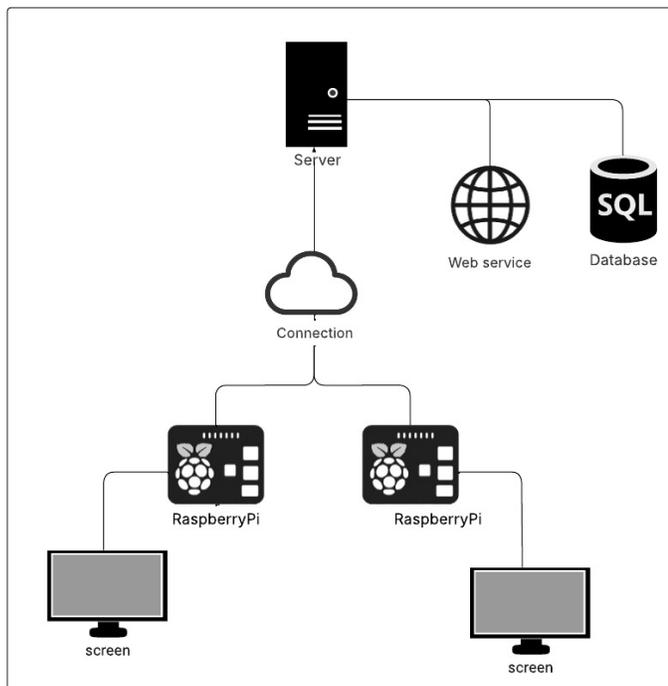


Fig. 1. Application Scheme. Source: Own elaboration (2025)

C. Open-Source Digital Signage Solutions

Several open-source digital signage platforms are available, offering a cost-effective alternative to commercial systems [9]. Solutions like Xibo [10], Screenly Open Source Edition (OSE), and PiSignage [11] are popular choices, particularly for projects using Raspberry Pi devices [12]. These systems provide flexibility and can be customized to meet specific needs. However, as our research shows, many free platforms suffer from performance issues or have outdated user interfaces compared to their paid counterparts. This performance gap, especially in resource consumption, was a key motivation for developing a new, optimized system tailored to our specific hardware and sustainability goals.

III. METHODOLOGICAL APPROACH

This study began with a review of academic articles on digital signage from repositories like CAPES and Korean Science, focusing on publications from the last six years that discussed relevant solutions and innovations [13]. From this review, we selected several systems that aligned with our project's goals for testing: piSignage, OptiSign [14], Xibo, StreetSign [15], and Chromium Kiosk [16]. We focused on the free, open-source versions of these systems to maintain our commitment to accessible and sustainable solutions.

The systems were tested using a client-server architecture. The server-side tests were conducted on an HP notebook with 4 GB of RAM and an Intel i3 processor, while the client-side tests used Raspberry Pi B3 devices. We evaluated each system based on functionality, resource usage (CPU and RAM), and ease of use. This initial testing phase helped us identify the strengths and weaknesses of existing solutions and refine the requirements for our custom-built system, Next Signage. The Figure 2 demonstrates communication between Raspberry Pi B3 and server (HP Notebook).

IV. DEVELOPMENT

A. System Architecture

To ensure a clear separation of concerns and facilitate future maintenance, we adopted the Model-View-Controller (MVC) design pattern for our system's architecture. MVC is a well-established pattern in software engineering that divides an application into three interconnected components: the Model (data and business logic), the View (user interface), and the Controller (handles user input and updates the Model and View) [17]. This structure makes the system more scalable and easier to manage.

The system operates on a client-server model designed for minimal hardware usage and high efficiency.

- **Server:** The server is responsible for all management tasks. It was developed using PHP and handles the core business logic, data processing, and communication with the database. Administrators use a web interface to create and manage playlists (queues of images or videos), set playback durations, and assign content to specific client screens.
- **Client:** The client acts as the media player. It was developed in Python, which is natively supported by the Raspberry Pi and provides excellent control over the hardware. The client's only requirement is an internet connection to periodically request updated playlists from the server. Once a new playlist is received, it is downloaded, and the old content is automatically deleted to save storage space. This process runs autonomously, ensuring that the display



Fig. 2. Setup Demonstration. Source: Own elaboration (2025)

continues to function even if the connection to the server is temporarily lost.

This architecture ensures that the client remains lightweight and efficient, as all the complex logic is handled by the server. The Figure 1 represents the overall scheme of the application.

The Figure 3 represents the Use Case Diagram of the application. This diagram shows how users and administrators interact with the system. A normal user can register and then log in to the system. After logging in, the user may update their personal information, select a player, and also manage playlists (create, read, update, and delete). The login process requires registration, and some actions extend from others, meaning they are optional or additional. On the other hand, the administrator (ADM) has access to special management functions that regular users cannot use.

V. RESULTS

In order to compare performance, we promoted tests directly in Raspberry Pi 3B hardware using existing digital signage systems. The tests consisted of playing MP4 HD resolution videos and showing them on an LCD display. We used Raspberry Pi OS utilitarian as top¹ and free² to monitor the average CPU and memory usage.

Our tests revealed significant performance differences, especially between free and paid options. Open-source applications,

¹A system monitor and process viewer for Unix-like systems, including CPU load (per core), RAM, and swap memory consumption.

²A command-line utility that provides a quick summary of the total amount of free and used physical memory (RAM) and swap memory in the system.

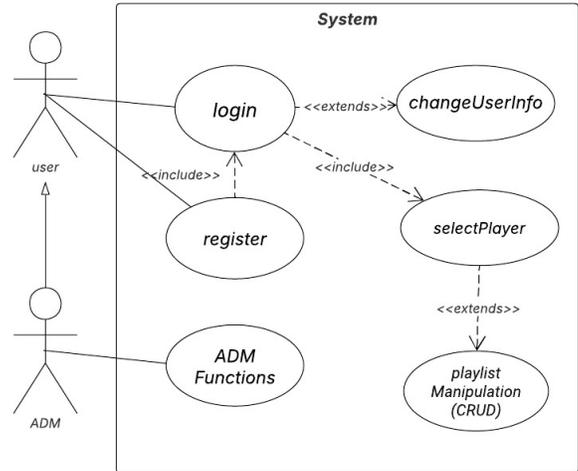


Fig. 3. Use Case Diagram. Source: Own elaboration (2025)

while cost-effective, tended to be less efficient in resource management. Their interfaces and features also often felt outdated compared to commercial solutions.

Table I summarizes the average performance results of the systems we analyzed on a Raspberry Pi 3B. The Consume Level in Table I is defined by the average CPU and RAM usage, where Very Low indicates sustained consumption $\leq 10\%$ of both resources; Low indicates $10\% < \text{consumption} \leq 15\%$; Average indicates $15\% < \text{consumption} \leq 20\%$; and Very High indicates $> 20\%$ for either resource.

TABLE I
COMPARATIVE PERFORMANCE

Model	License	Performance (RAM/CPU)	Consume Level
piSignage	Open Source	~15% / ~10%	Average
OptiSigns	Paid	~10% / ~6%	Low
Xibo	Paid	~10% / ~10%	Low
StreetSign	Open Source	~16% / ~20%	Very High
Chromium Kiosk	Open Source	~8% / ~8%	Very Low

Source: Own elaboration (2025)

The results show that systems like StreetSign were too resource-intensive for a low-power device like the Raspberry Pi. On the other hand, Chromium Kiosk was extremely efficient, demonstrating that a lightweight, browser-based approach could achieve very low resource consumption. The performance of paid systems like OptiSign and Xibo provided a benchmark for the user experience (UX) we wanted to achieve.

Based on these findings, we designed our custom client to be as lightweight as Chromium Kiosk while offering a centralized management interface inspired by the usability of Xibo. During long-term testing, our system running on a Raspberry Pi 3B consistently showed low CPU and RAM usage, similar to the “Very Low” consumption level, and did not exhibit any overheating issues. This confirmed that our architectural choices led to a high-performance, efficient, and reliable solution suitable for long-term deployment.

While Chromium Kiosk demonstrated excellent resource efficiency, its primary limitation, as noted in the Abstract, lies in its lack of a built-in centralized user interface (UI) for content management. It serves only as a dedicated media player, requiring external solutions for playlist and device control. Next Signage was designed to match the low-power consumption of Chromium Kiosk while integrating a robust, centralized, and user-friendly web-based management system, thereby filling the identified gap.

VI. CONCLUSION

This project successfully demonstrates that it is possible to build a sustainable and effective digital signage system for a smart campus using open-source software, low-cost hardware, and repurposed equipment. By focusing on resource efficiency and adopting the MVC architecture, we developed a solution that outperforms many existing open-source alternatives while remaining easy to manage.

The system contributes to the institution’s sustainability goals by giving a second life to discarded electronic screens and reducing the need for paper-based communication. It also serves as a practical example of how smart campus technologies can be implemented in a cost-effective and environmentally responsible manner.

For future work, we plan to expand the system’s capabilities to include support for interactive content and integration with other campus data systems, such as event calendars and emergency alerts. Further research could also explore the use of even more energy-efficient hardware, such as newer Raspberry Pi models, and conduct a formal user-satisfaction study to measure the system’s impact on the campus community.

REFERENCES

- [1] J. Schaeffler, *Digital Signage: Software, Networks, Advertising, and Displays*. Focal Press, 2008.
- [2] Brasil, Presidência da República, “Lei no 14.479, de 21 de dezembro de 2022: institui a política nacional de desfazimento e recondicionamento de equipamentos eletroeletrônicos e dispõe sobre o programa computadores para inclusão,” *Diário Oficial da União*, Dec. 2022, lei nº 14.479, de 21 de dezembro de 2022. [Online]. Available: https://www.planalto.gov.br/ccivil_03/_ato2019-2022/2022/lei/14479.htm
- [3] M. Al-Emran, V. Mezhuyev, and A. Kamaludin, “The impact of internet of things (iot) on higher education: A systematic review,” *Education and Information Technologies*, vol. 26, no. 3, pp. 2911–2932, 2021.
- [4] J.-Y. Cho, “A study of the effectiveness of digital signage: Importance of customized content,” *Journal of Digital Convergence*, vol. 17, no. 6, pp. 211–217, jun 2019.
- [5] International Telecommunication Union (ITU) and United Nations Institute for Training and Research (UNITAR), “Global e-waste monitor 2024: Electronic waste rising five times faster than recycling,” ITU and UNITAR, Geneva, Tech. Rep., 2024.
- [6] United Nations Institute for Training and Research (UNITAR), Sustainable Cycles (SCYCLE) Programme, United Nations Environment Programme (UNEP), and United Nations Industrial Development Organization (UNIDO), “Regional e-waste monitor for latin-america 2022,” UNITAR, Bonn, Tech. Rep., 2022.
- [7] Organisation for Economic Co-operation and Development (OECD), *Business Models for the Circular Economy: Opportunities and Challenges from a Policy Perspective*. Paris: OECD Publishing, 2019.
- [8] A. R. R. de la Cruz, “E-waste management: Challenges, opportunities, and the role of higher education institutions,” *Journal of Environmental Management*, vol. 298, 2021.
- [9] P. S. Hearty, “Open-source software and the corporate world,” *ACM SIGCAS Computers and Society*, vol. 38, no. 1, pp. 34–45, 2008.
- [10] Xibo Signage Ltd., “Xibo digital signage,” <https://xibosignage.org>, 2025, accessed: 2025-08-20.
- [11] piSignage, “pisignage digital signage,” <https://pisignage.com>, 2025, accessed: 2025-08-20.
- [12] A. P. D. L. da Silva, “The role of raspberry pi in the internet of things (iot): A review,” *International Journal of Computer Applications*, vol. 179, no. 3, pp. 8–12, 2018.
- [13] AhnSungHee, “A study on the development of public digital signage services,” *Journal of the Korea Convergence Society*, vol. 12, no. 12, pp. 185–196, 12 2021.
- [14] OptiSigns, Inc., “Optisigns digital signage software,” <https://www.optisigns.com>, 2025, accessed: 2025-08-20.
- [15] D. Name], “Streetsign digital signage software,” <https://github.com/dantheheckie/streetsign>, 2025, accessed: 2025-08-20.
- [16] The Chromium Authors, “The chromium projects,” <https://www.chromium.org/>, 2025, accessed: 2025-08-20.
- [17] S. R. Burris, “The model-view-controller (mvc) architecture as a framework for web application development,” *Journal of Computing Sciences in Colleges*, vol. 25, no. 5, pp. 118–124, 2010.