***American Worldviews: Toward a Sustainable Future (Workshop – March 18–21, 2025, Univer- sidad Tecnológica de El Salvador, San Salvador, El Salvador)}, Vol. 84, 2025***

**University of New Mexico**

Registro de Asistencia mediante una Aplicación Web (PWA)

# Alejandro León-Salazar 1, Natasha Mehler-Castro 1, Arianna Esteves-Valero 1, Nohelya Monar-Zambrano 1, Denisse Crespo-Arias 1

1 Facultad de Ciencias Matemáticas y Físicas, Universidad de Guayaquil, Guayaquil, Ecuador; [xavier.leonsal@ug.edu.ec](mailto:xavier.leonsal@ug.edu.ec), [natasha.mehlercas@ug.edu.ec](mailto:natasha.mehlercas@ug.edu.ec), [arianna.estevesval@ug.edu.ec](mailto:arianna.estevesval@ug.edu.ec), [nohelya.monarzam@ug.edu.ec,](mailto:nohelya.monarzam@ug.edu.ec,) [denisse.crespoari@ug.edu.ec](mailto:denisse.crespoari@ug.edu.ec)

**Resumen:** Este artículo presenta una aplicación web (PWA) enfocada en el registro y consulta de asistencias académicas, priorizando la experiencia de usuario, la accesibilidad y la facilidad de instalación en dispositivos personales. Desarrollada con tecnologías web estándar, la solución permite a los usuarios gestionar materias, controlar asistencias y consultar resúmenes, todo respaldado por un servidor PHP y una base de datos MySQL. Gracias a su diseño adaptable y su arquitectura instalable mediante manifest.json, la aplicación constituye una alternativa moderna y eficiente para el control de asistencia en entornos educativos y organizacionales.

**Palabras Clave:** registro de asistencia, aplicación web, PHP, MySQL, educación digital, instalación web

# Introducción

El control de asistencia es esencial en contextos educativos, pues ayuda en la organización, la evaluación y la gestión de las asistencias en cada materia. Muchas soluciones actuales ofrecen funciones avanzadas y una integración profunda, pero suelen estar acotadas a plataformas cerradas, exigir pagos por licencia, o requerir instalaciones complejas y mantenimiento especializado.

El sistema presentado en este trabajo responde a la necesidad de una herramienta sencilla y estandarizada, accesible a cualquier usuario con conocimientos básicos, sin sacrificar seguridad ni eficiencia. Desarrollada con PHP y MySQL como base tecnológica, la aplicación permite gestionar el ciclo completo de registro y consulta de asistencias desde cualquier navegador moderno. Su diseño, ligero y modular, busca reducir la curva de aprendizaje y facilitar su implementación en distintos entornos.

A diferencia de plataformas comerciales ampliamente conocidas como Jibble, Clockify o incluso módulos de asistencia incluidos en suites educativas más amplias, la propuesta se distingue por su enfoque minimalista: elimina funciones innecesarias, prioriza la simplicidad del flujo de uso y asegura que cualquier usuario pueda instalar y comenzar a utilizar la herramienta en minutos, sin dependencias externas ni configuraciones avanzadas.

# Materiales y Métodos

* 1. ***Descripción del Sistema***

El sistema consiste en una aplicación web, accesible desde cualquier dispositivo moderno (computadora, tablet o teléfono). Se apoya íntegramente en tecnologías estándar: HTML5, CSS3, JavaScript para la interfaz; PHP para la lógica de servidor; y MySQL como gestor de base de datos relacional.

El sistema permite registrar materias, definir horarios, marcar asistencias diarias y obtener reportes históricos. La arquitectura modular facilita la escalabilidad y el mantenimiento futuro del software.

Figura 1. Diagrama de componentes que describe la arquitectura del sistema.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Figura 1.** Diagrama de componentes del sistema.

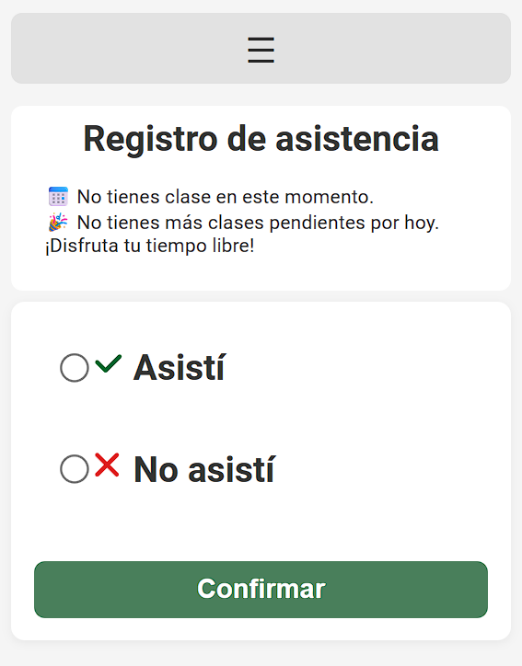
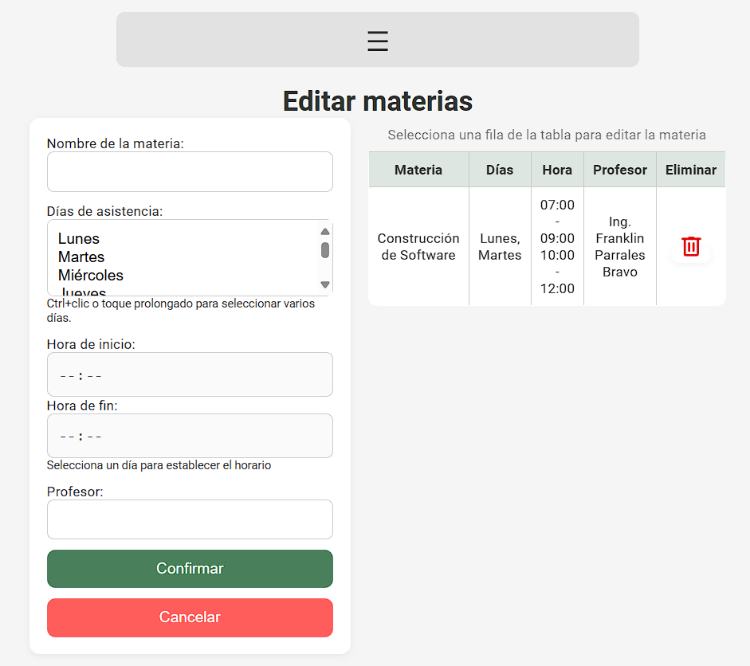
A continuación, se detallan los componentes principales del sistema, agrupados en función de su propósito:

1. Frontend:

* Proporciona formularios para la creación de cuentas, registro y edición de materias, y registro de asistencias. Incluye validaciones en tiempo real para mejorar la experiencia y prevenir errores comunes.
  + Manifest Web: El archivo manifest.json habilita la posibilidad de instalar la aplicación en el dispositivo del usuario, permitiendo su acceso directo y mejor integración con el sistema operativo.

1. Backend: Gestiona la lógica de negocio y la seguridad, incluyendo la autenticación de usuarios, la validación de sesiones y la protección de datos sensibles.
2. MySQL: Se emplea como sistema gestor de base de datos para almacenar toda la información relevante de usuarios, materias, horarios y asistencias. MySQL facilita consultas eficientes y garantiza la integridad en un entorno multiusuario.
3. Módulos Adicionales: Se han implementado mecanismos de tolerancia a fallos, manejo de errores, y políticas de mantenimiento preventivo y correctivo, documentadas en los planes respectivos.

En la Figura 2 y 3 se puede ver un ejemplo de la interfaz del sistema Para más detalles sobre su uso, puede consultar el manual de usuario disponible en su repositorio de GitHub [1].



**Figura 2.** Interfaz para registrar la asistencia. **Figura 3.** Interfaz para registrar materias.

**2.2 Arquitectura del Sistema**

El sistema adopta un modelo cliente-servidor clásico. El lado cliente consiste en una interfaz web responsiva construida con HTML5, CSS3 y JavaScript, optimizada para dispositivos móviles y de escritorio. El servidor utiliza PHP para la lógica de negocio y la gestión de sesiones, mientras que MySQL almacena los datos de usuarios, materias, horarios y asistencias. Las comunicaciones entre cliente y servidor emplean AJAX, permitiendo actualizaciones dinámicas sin recargar la página. Todas las transacciones sensibles requieren autenticación activa, reforzando la privacidad de los datos.

**2.3 Documentación Técnica**

Durante el desarrollo se elaboraron documentos de requisitos funcionales y no funcionales, alcance, tolerancia a fallos, políticas y plan de mantenimiento, plan de pruebas, diseño detallado de software y manual de usuario. Estas guías han permitido un desarrollo sistemático, minimizando riesgos y mejorando la calidad del producto final.

La tolerancia a fallos incluye validaciones en frontend y backend, manejo de sesiones expiradas y gestión de errores de conexión o de base de datos, garantizando la robustez del sistema en ambientes reales, como recomienda [Sommerville, 2011](https://www.pearson.com/en-us/subject-catalog/p/software-engineering/P200000007433/9780137035151).

**3. Resultados**

La solución fue implementada y desplegada en un entorno de pruebas con usuarios reales.  
Se validó la correcta gestión de usuarios, materias y asistencias mediante pruebas funcionales y de usabilidad, siguiendo la metodología propuesta por Nielsen (1994) para la evaluación de interfaces de usuario [Nielsen, 1994].

El sistema fue capaz de registrar y consultar asistencias de manera eficiente, con tiempos de respuesta inferiores a 500 ms en operaciones CRUD principales (crear, leer, actualizar, borrar).  
Las pruebas de tolerancia a fallos demostraron que el sistema maneja adecuadamente la pérdida de conexión, errores de datos y sesiones expiradas, mostrando mensajes claros al usuario y evitando la pérdida de información.

Los usuarios destacaron la facilidad de uso del sistema, la rapidez para registrar asistencias y la claridad de los reportes generados. La curva de aprendizaje fue mínima, en contraste con soluciones comerciales más complejas (véase [Fernández-García et al., 2019](https://www.sciencedirect.com/science/article/pii/S0360131519302490)).

**4. Discusión**

Comparado con plataformas comerciales como Jibble, Clockify o sistemas institucionales de gestión académica, la aplicación propuesta se distingue por su simplicidad, libre acceso y facilidad de instalación y uso.

Sistemas comerciales suelen incluir funciones avanzadas para grandes empresas, pero resultan excesivos para contextos educativos simples y pueden requerir licencias o integraciones complejas ([Rashid et al., 2016](https://ieeexplore.ieee.org/document/7733302)).

El sistema presentado cubre las necesidades esenciales de registro de asistencia, evitando sobrecarga funcional y priorizando la experiencia del usuario final, como sugiere la literatura sobre diseño centrado en el usuario ([Norman, 2013](https://mitpress.mit.edu/9780262525671/the-design-of-everyday-things/)).

Entre las limitaciones identificadas están la falta de integración con otras plataformas educativas y la ausencia de reportes avanzados para administración institucional, aspectos que pueden ser abordados en trabajos futuros.

**5. Conclusiones**

El sistema de registro de asistencias presentado demuestra que es posible ofrecer una herramienta digital sencilla, eficiente y accesible para el control académico, evitando la complejidad y sobrecarga de soluciones comerciales existentes.

Su arquitectura modular y el uso de tecnologías estándar facilitan la adaptación y el mantenimiento. La experiencia del usuario y la seguridad de la información son pilares del diseño, cubriendo las necesidades básicas del entorno educativo digital.

Como trabajo futuro, se plantea la integración con plataformas de gestión académica, el desarrollo de módulos de análisis avanzado de datos de asistencia e implementación con autenticación con Google.

# Referencias

1. Xavier Alejandro, L. Código del Sistema de Registro de Asistencia. Disponible en: https://github.com/Alejandro-Leon24/Proyecto-web.git
2. Fernández-García, Á. J., García-Peñalvo, F. J., & Therón, R. (2019). Usability evaluation of learning analytics dashboards: A systematic literature review. Computers & Education, 141, 103647. <https://doi.org/10.1016/j.compedu.2019.103647>
3. Norman, D. A. (2013). The Design of Everyday Things: Revised and Expanded Edition. MIT Press.
4. Nielsen, J. (1994). Usability Engineering. Academic Press.
5. Rashid, A., Rahman, A., & Musa, R. M. (2016). Comparison of Attendance Monitoring Systems: A Review. 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), 2016, 439–444. <https://ieeexplore.ieee.org/document/7733302>
6. Sommerville, I. (2011). Software Engineering (9th ed.). Pearson.