

Sistema de Gestión y Administración de Lockers (SGAL)

Prototipo funcional

Categoría: Desafío STEAM

Eje temático: Tecnologías de la Información Aplicadas a la Informática

Estudiante: Josué Alexander Segura Brizuela

Docente tutor: Greddy Hernández Pérez

Centro educativo: CTP José María Zeledón Brenes

CORVEC: Unidos por la Excelencia

Correo de la persona estudiante: **segurabjosue@gmail.com**

Resumen

SGAL es un prototipo web desarrollado para centralizar y facilitar la administración de lockers estudiantiles. El sistema integra autenticación, gestión de estudiantes y lockers, asignaciones con validación de disponibilidad, comprobantes, pagos, morosidad, gastos y reportes administrativos y financieros. Su desarrollo se realizó de forma progresiva mediante análisis de la necesidad, diseño de base de datos e interfaces, programación en PHP, integración con MySQL, pruebas funcionales y corrección de errores. Como aplicación práctica, Josué cargó personalmente 159 registros de estudiantes; el tablero mostró 300 lockers, 159 ocupados y 141 disponibles, equivalentes a 53 % de ocupación y 47 % de disponibilidad. Los resultados evidencian un prototipo funcional y útil para apoyar el control administrativo. Antes de una implementación institucional completa se recomienda fortalecer seguridad, respaldos, mantenimiento, costos, escalabilidad y desempeño.

1. Introducción

La administración de lockers estudiantiles requiere mantener información organizada sobre estudiantes, espacios disponibles, asignaciones, comprobantes, pagos y consultas administrativas. SGAL surge como una propuesta tecnológica para centralizar estos procesos mediante una aplicación web conectada a una base de datos, con el propósito de reducir la dependencia de registros dispersos y facilitar el seguimiento del servicio.

1.1 Antecedentes, problema y necesidad

Durante la etapa inicial se identificó la necesidad de digitalizar el registro, asignación y control de lockers y de disponer de consultas y reportes administrativos. El proceso requiere conocer qué estudiantes utilizan el servicio, cuáles lockers están disponibles u ocupados y mantener información actualizada de las asignaciones. La pregunta que orienta el proyecto es: ¿Cómo desarrollar un prototipo web que permita centralizar y facilitar la gestión de estudiantes, lockers, asignaciones y demás información administrativa relacionada con este servicio en el centro educativo?

1.2 Justificación y relevancia

SGAL aborda una necesidad concreta del entorno educativo mediante la aplicación de programación, bases de datos y desarrollo web. La solución integra en una misma plataforma el registro de estudiantes, la administración de lockers, la verificación de disponibilidad, las asignaciones, la generación de comprobantes y diferentes consultas y reportes. Su relevancia práctica se fortaleció cuando Josué realizó personalmente la carga de 159 registros de estudiantes solicitados para el proceso de lockers, utilizando el prototipo en una actividad directamente relacionada con su propósito.

1.3 Criterios técnicos, viabilidad, alcance y limitaciones

La solución se desarrolló con PHP y MySQL, apoyada por HTML, CSS, Bootstrap, phpMyAdmin y Visual Studio Code. Se incorporaron autenticación, validaciones de disponibilidad y operaciones de registro, consulta, modificación y eliminación. El prototipo es técnicamente viable en el alcance probado y utiliza tecnologías de amplio acceso; sin embargo, una implementación

institucional definitiva requiere valorar alojamiento, respaldo, mantenimiento, seguridad, niveles de permisos, escalabilidad, soporte y costos reales.

El alcance actual comprende gestión de estudiantes, administración y asignación de lockers, comprobantes, pagos, obligaciones pendientes, gastos y reportes. Entre las limitaciones se encuentran la dependencia de Internet para la versión alojada, la necesidad de mantener los datos actualizados, fortalecer la seguridad, validar el desempeño con mayor concurrencia y realizar una prueba piloto prolongada. SGAL se presenta como prototipo funcional y no como implementación oficial institucional.

2. Marco teórico

SGAL se sustenta en el concepto de sistema de información, entendido como la integración de personas, procesos y tecnologías para administrar información y apoyar actividades organizacionales. Wallace (2023) destaca esta relación, que en SGAL se concreta al centralizar estudiantes, lockers, asignaciones y procesos administrativos.

La automatización de procesos permite reducir tareas manuales repetitivas y mejorar la disponibilidad de información. En el prototipo se aplica mediante módulos de registro, asignación, validación, comprobantes, pagos y reportes. A su vez, las bases de datos permiten estructurar y relacionar información para su consulta y mantenimiento; SGAL utiliza MySQL para almacenar los datos necesarios para sus módulos (Kroenke et al., 2023; Oracle, 2026).

Una aplicación web combina una interfaz accesible desde navegador con lógica de procesamiento y almacenamiento. PHP se utiliza para procesar las operaciones del sistema y comunicarse con MySQL (PHP Documentation Group, 2026). Bootstrap apoya la construcción de una interfaz consistente y adaptable.

La autenticación constituye una medida inicial de control de acceso, mientras que las validaciones contribuyen a la integridad de los datos al comprobar disponibilidad y evitar asignaciones duplicadas. Para una eventual implementación institucional se requieren controles adicionales de seguridad. La ISO/IEC 27001:2022 se utiliza únicamente como referencia conceptual sobre la protección de la información; no se afirma que SGAL esté certificado ni que demuestre conformidad con dicha norma.

Las pruebas de software y la mejora continua permiten detectar problemas, validar cambios y retroalimentar el desarrollo. Erthal et al. (2023) describen la experimentación continua como una práctica útil para aprender de los resultados y ajustar el software. En SGAL este principio se aplicó mediante pruebas progresivas, correcciones y nuevas verificaciones.

3. Objetivos

3.1 Objetivo general

Desarrollar un prototipo web para la gestión y administración de lockers estudiantiles, mediante PHP, MySQL y tecnologías web, que permita centralizar el registro de estudiantes, la administración y asignación de lockers y la consulta de información para apoyar el control administrativo dentro del centro educativo.

3.2 Objetivos específicos

1. Diseñar una base de datos en MySQL que permita almacenar y relacionar la información necesaria para estudiantes, lockers y asignaciones.
2. Implementar una aplicación web mediante PHP y MySQL para registrar, consultar, modificar y administrar información de estudiantes y lockers.
3. Desarrollar un módulo de asignación que verifique disponibilidad, evite asignaciones duplicadas y contribuya a mantener la integridad de los datos.
4. Implementar autenticación y validación de credenciales para restringir el acceso al prototipo.
5. Generar comprobantes y reportes administrativos que faciliten el seguimiento de asignaciones y el estado de los lockers.
6. Evaluar el funcionamiento de los módulos mediante pruebas, identificar errores, realizar correcciones y comprobar la integración del sistema.

4. Metodología

El proyecto se desarrolló de manera progresiva y cronológica. Las actividades se reconstruyeron a partir de las bitácoras y de las evidencias del prototipo, respondiendo a qué se hizo, quién lo realizó, cuándo se desarrolló y qué producto se obtuvo. El proceso incluyó análisis, diseño, programación, integración, pruebas, correcciones, aplicación práctica y retroalimentación.

Fase	Actividad principal	Producto / resultado
I. Investigación y análisis	Identificación de la necesidad, requerimientos y módulos iniciales.	Requerimientos del prototipo
II. Diseño	Diseño de base de datos, relaciones e interfaces.	Estructura inicial
III. Integración	Conexión PHP-MySQL y pruebas CRUD.	Comunicación aplicación-base de datos
IV. Desarrollo	Autenticación, estudiantes, lockers, asignaciones, comprobantes y reportes.	Módulos funcionales
V. Ampliación administrativa	Pagos, morosidad, gastos y reportes financieros.	Mayor alcance administrativo
VI. Pruebas y corrección	Pruebas de formularios, consultas, reportes e interfaz.	Correcciones y mayor estabilidad
VII. Aplicación práctica	Carga personal de 159 estudiantes y revisión del tablero.	Evidencia de utilización
VIII. Retroalimentación	Revisión de observaciones y fortalecimiento de la documentación.	Mejora del informe y defensa

4.1 Cronograma y seguimiento

Periodo / etapa	Actividad	Estado
Inicio	Problema y requerimientos	Cumplida
Diseño	Base de datos e interfaces	Cumplida
Desarrollo inicial	Conexión PHP-MySQL y CRUD	Cumplida
Desarrollo progresivo	Autenticación, estudiantes, lockers y asignaciones	Cumplida
Desarrollo avanzado	Pagos, morosidad, gastos y reportes	Cumplida
Pruebas	Corrección de formularios, consultas, reportes e interfaz	Cumplida
7 agosto 2026	Carga personal de 159 estudiantes	Cumplida
Posterior a Expo	Revisión y fortalecimiento del informe	Cumplida

4.2 Materiales y recursos

Se utilizaron computadora, conexión a Internet, navegador web, PHP, MySQL, phpMyAdmin, HTML, CSS, Bootstrap, Visual Studio Code y servicio de alojamiento web para desarrollo y pruebas. Estos recursos permitieron programar, administrar la base de datos, publicar el prototipo y verificar su funcionamiento.

4.3 Diseño, arquitectura y desarrollo

La arquitectura general sigue el flujo: persona usuaria → navegador web → aplicación SGAL en PHP → base de datos MySQL. Para la asignación de lockers, el proceso comprende inicio de sesión, selección del estudiante, consulta del locker, verificación de disponibilidad, asignación, almacenamiento y generación de comprobante.

El desarrollo integró base de datos, conexión PHP-MySQL, autenticación, gestión CRUD de estudiantes, administración de lockers, validaciones de asignación, comprobantes, reportes y funciones financieras. La innovación no se plantea como la invención de los sistemas de lockers, sino como la aplicación creativa e integrada de conocimientos de programación y bases de datos a una necesidad concreta del colegio.

5. Discusión de resultados y conclusiones

5.1 Resultados obtenidos

1. Base de datos MySQL funcional y conectada con la aplicación.
2. Autenticación y validación de credenciales.
3. Gestión de estudiantes mediante operaciones CRUD.
4. Administración de lockers y control de disponibilidad.
5. Asignación con validaciones para evitar duplicidades.
6. Generación de comprobantes y reportes administrativos.
7. Funciones de pagos, obligaciones pendientes, gastos y reportes financieros.
8. Corrección de errores y mejoras de interfaz a partir de pruebas.
9. Aplicación práctica mediante la carga personal de 159 registros de estudiantes.

5.2 Pruebas y mejora continua

Etapas	Situación	Acción	Resultado
Base de datos	Necesidad de organizar información	Diseño de tablas y relaciones	Estructura funcional
Conexión	Comunicar aplicación y datos	Conexión PHP-MySQL y pruebas CRUD	Comunicación funcional
Seguridad	Restringir acceso	Inicio de sesión y validación	Control de acceso inicial
Asignaciones	Riesgo de duplicidad	Validación de disponibilidad	Mayor integridad
Reportes	Necesidad de consulta administrativa	Desarrollo y revisión de reportes	Visualización del estado
Pruebas	Errores en formularios, consultas y reportes	Correcciones y nuevas verificaciones	Mayor estabilidad
Aplicación práctica	Observar comportamiento con más datos	Carga de 159 estudiantes	Evidencia de uso práctico

5.3 Análisis cuantitativo

En la prueba práctica, el tablero mostró 300 lockers totales, 159 ocupados y 141 disponibles. La cantidad disponible se comprueba mediante $300 - 159 = 141$. La consistencia del total se verifica con $159 + 141 = 300$. El porcentaje de ocupación es $(159 \div 300) \times 100 = 53 \%$, mientras que la disponibilidad corresponde a $(141 \div 300) \times 100 = 47 \%$. Estos resultados coinciden con los indicadores mostrados por el prototipo.

Indicador	Cálculo	Resultado
Lockers disponibles	$300 - 159$	141
Comprobación del total	$159 + 141$	300
Ocupación	$(159 \div 300) \times 100$	53 %
Disponibilidad	$(141 \div 300) \times 100$	47 %

5.4 Interpretación, viabilidad e impacto

Los resultados muestran la evolución desde una necesidad administrativa hasta una solución web integrada. La utilización con

159 registros constituye evidencia de aplicación práctica y permite comprobar el comportamiento del tablero con una cantidad considerable de información. Los módulos financieros y de reportes amplían el alcance más allá de la asignación básica de lockers.

La viabilidad técnica se evidencia en la integración funcional de PHP, MySQL y los módulos publicados. La viabilidad operativa se apoya en la prueba con datos del proceso de lockers. En el plano económico, las tecnologías utilizadas son de amplio acceso, pero una implementación institucional requiere estimar alojamiento, dominio, mantenimiento, respaldo, seguridad y soporte.

El impacto social potencial se relaciona con una administración más organizada y una consulta más ágil de la información. El impacto económico potencial se vincula con el control de pagos, gastos e ingresos. El impacto ambiental potencial se asocia con la disminución del uso de registros impresos cuando la información se gestiona digitalmente. Estos efectos se presentan como potenciales y deberán medirse en una implementación institucional prolongada.

5.5 Conclusiones

1. SGAL permitió construir un prototipo web funcional orientado a una necesidad real de administración de lockers del centro educativo.
2. La integración de PHP y MySQL permitió centralizar estudiantes, lockers, asignaciones y otros datos administrativos.
3. Las validaciones de disponibilidad contribuyen a reducir el riesgo de asignaciones duplicadas y a mantener coherencia en los registros.
4. Las pruebas progresivas permitieron identificar errores, aplicar correcciones y mejorar el funcionamiento del prototipo.
5. La carga personal de 159 estudiantes y los indicadores de 300 lockers, 159 ocupados y 141 disponibles constituyen evidencia cuantitativa de uso práctico.
6. Los comprobantes, pagos y reportes ampliaron la utilidad administrativa del prototipo.
7. Antes de una implementación institucional completa deben fortalecerse seguridad, respaldos, escalabilidad, mantenimiento y análisis de costos.

6. Recomendaciones

1. Realizar una prueba piloto prolongada con las personas responsables del proceso de lockers.
2. Fortalecer los mecanismos de seguridad y definir niveles de permisos según los tipos de usuario.
3. Implementar procedimientos periódicos y, cuando sea posible, automáticos de respaldo y recuperación.
4. Mantener pruebas funcionales cada vez que se incorporen nuevas funcionalidades.
5. Evaluar la experiencia de las personas usuarias responsables de administrar lockers.
6. Determinar los costos reales de alojamiento, dominio, mantenimiento y soporte para una implementación institucional.
7. Generar y conservar el diagrama entidad-relación a partir de la estructura real de la base de datos como parte de la documentación técnica.

7. Referencias

- Erthal, V. M., de Souza, B. P., dos Santos, P. S. M., & Travassos, G. H. (2023). Characterization of continuous experimentation in software engineering: Expressions, models, and strategies. *Science of Computer Programming*, 229, 102961. <https://doi.org/10.1016/j.scico.2023.102961>
- International Organization for Standardization. (2022). ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection - Information security management systems - Requirements.
- Kroenke, D. M., Auer, D. J., Vandenberg, S. L., & Yoder, R. C. (2023). *Database concepts* (10th ed.). Pearson.
- Oracle. (2026). *MySQL 8.4 Reference Manual*. MySQL Documentation.
- PHP Documentation Group. (2026). *PHP Manual*.
- Wallace, P. (2023). *Introduction to information systems* (5th ed.). Pearson.

8. Apéndice: evidencias del prototipo

Las siguientes figuras documentan interfaces y funciones desarrolladas en SGAL. Se presentan como evidencia visual complementaria del proceso descrito en el informe.



Figura A1. Tablero principal de SGAL con 159 estudiantes, 300 lockers, 141 disponibles y 159 ocupados.



Figura A2. Pantalla de inicio de sesión del prototipo.



Figura A3. Menú principal y módulos integrados en SGAL.

The image shows a screenshot of the phpMyAdmin interface displaying the database structure. The table 'tablas' contains the following data:

Tabla	Acción	Fila	Tipo	Coligamiento	Tamaño	Reserva a
gastos	Examinar Estructura Buscar Insertar Vacío Eliminar	12	InnoDB	utf8mb4_general_ci	16.0 KB	-
lockers	Examinar Estructura Buscar Insertar Vacío Eliminar	120	InnoDB	utf8mb4_general_ci	32.0 KB	-
pagos	Examinar Estructura Buscar Insertar Vacío Eliminar	350	InnoDB	utf8mb4_general_ci	48.0 KB	-
recibos	Examinar Estructura Buscar Insertar Vacío Eliminar	350	InnoDB	utf8mb4_general_ci	48.0 KB	-
usuarios	Examinar Estructura Buscar Insertar Vacío Eliminar	8	InnoDB	utf8mb4_general_ci	16.0 KB	-
estudiantes	Examinar Estructura Buscar Insertar Vacío Eliminar	250	InnoDB	utf8mb4_general_ci	48.0 KB	-
6 tablas	Número de filas	1,090	InnoDB	utf8mb4_general_ci	200.0 KB	0 B

Figura A4. Estructura de datos observada en phpMyAdmin.

Registro de Gastos

Total Gastado: ₡0

Historial de Gastos

Fecha	Descripción	Monto
-------	-------------	-------

Volver

Figura A5. Módulo de registro de gastos.

Estado Financiero

Ingresos
₡0

Gastos
₡0

Saldo
₡0

Volver

Figura A6. Módulo de estado financiero.

Reporte Mensual

Total recaudado en Febrero: ₡0

Estudiante	Mes	Monto	Tipo	Fecha
------------	-----	-------	------	-------

Volver

Figura A7. Reporte mensual.

8.1 Declaración de uso de inteligencia artificial

Durante la preparación y revisión de la documentación se utilizó una herramienta de inteligencia artificial como apoyo complementario para organizar información, revisar redacción y estructura, y contrastar el documento con los lineamientos suministrados. Los datos del prototipo, actividades, resultados, evidencias y decisiones del proyecto corresponden al trabajo realizado y revisado por la persona estudiante y su tutor. La persona estudiante mantiene la responsabilidad sobre el contenido presentado y debe poder explicar y defender el proyecto.