

### Diseño de un Sistema de Gestión para una Empresa de Máquinas Recreativas mediante un Software Automatizado Design of a Management System for a Recreational Machine Company through Automated Software

**Joshúa Castillo-Merejildo¹, Jean Castro-Murillo¹, Joel Gabino-Villao¹, Edú Sabando-Barberán¹, y Euro Quiroz-Gómez¹**  
  
joshua.castillomer@ug.edu.ec, jean.castromur@ug.edu.ec, joel.gabinovil@ug.edu.ec, edu.sabandobar@ug.edu.ec, euro.quirozgom@ug.edu.ec

¹Carrera de Software, Facultad de Ciencias Matemáticas y Físicas, Universidad de Guayaquil, Guayaquil, Ecuador

**Resumen**

El presente proyecto se enfoca en el desarrollo de un sistema de gestión de información para una empresa de máquinas recreativas. Se busca solucionar la dispersión de información, los retrasos operativos y la falta de trazabilidad mediante un sistema automatizado que permita registrar procesos como montaje, distribución, recaudación y mantenimiento de las máquinas. El desarrollo sigue los lineamientos del estándar IEEE 1016-2009, aplicando principios de modularidad, escalabilidad y seguridad. Este enfoque permite optimizar las operaciones y mejorar la toma de decisiones estratégicas de la empresa.

**Palabras Clave:** Gestión de sistemas, máquinas recreativas, automatización, trazabilidad, arquitectura modular.

**Palabras clave:** ITS; tiempo real; arquitectura de transporte; transporte.

**Abstract**

This project focuses on Designing an information management system for a recreational machine company. It aims to solve information dispersion, operational delays, and lack of traceability through an automated system capable of recording assembly, distribution, collection, and maintenance processes. The development follows the IEEE 1016-2009 standard, applying principles of modularity, scalability, and security. This approach optimizes operations and improves strategic decision-making for the company.

**Keywords:** Systems management, recreational machines, automation, traceability, modular architecture.

## 1. Introducción

En el mercado actual, soluciones como GESTIMAQ (Micromaq, 2025) y PROMARE (Procode, 2025) dominan la gestión de máquinas recreativas, pero se centran en telemetría y facturación, dejando brechas en trazabilidad integral. A diferencia de estos sistemas, nuestro proyecto aborda todas las etapas del ciclo de vida (montaje, distribución, recaudación, mantenimiento y baja), con roles de usuario especializados (Logística, Técnicos, Contabilidad). Esto optimiza flujos de trabajo en empresas con departamentos segmentados, aunque carece temporalmente de telemetría en tiempo real, lo cual es una ventaja clave de GESTIMAQ. La justificación de este diseño radica en su escalabilidad y bajo costo, ideales para PYMES que priorizan control interno sobre funcionalidades avanzadas de monitorización remota.

En el entorno empresarial contemporáneo, la competitividad en el sector de servicios técnicos demanda una optimización constante de los procesos internos, especialmente en industrias como la de las máquinas recreativas, donde convergen tareas de montaje, distribución, mantenimiento y recaudación. En este contexto, el desarrollo de soluciones automatizadas resulta crucial para superar las limitaciones propias de la gestión manual de la información.

Desde una perspectiva de ingeniería de software, Pressman (2005) define el diseño de sistemas como el proceso de estructurar arquitecturas, componentes e interfaces que permitan satisfacer necesidades funcionales y no funcionales, mediante soluciones técnicas mantenibles y escalables. Esta concepción implica no solo la implementación de tecnologías, sino la configuración de módulos interrelacionados que favorezcan la evolución y comprensión del sistema en entornos complejos.

Por su parte, Sommerville (2011) advierte que los sistemas no automatizados incrementan el riesgo de errores operativos, afectan la productividad y entorpecen la toma de decisiones basada en datos confiables. Bajo esa premisa, la automatización no solo representa una estrategia de eficiencia, sino también un factor crítico para la calidad organizacional en procesos sensibles como el control financiero o logístico. Así, un sistema de gestión informatizado debe contemplar atributos clave como la confiabilidad, la eficiencia y la capacidad de adaptación a contextos cambiantes.

Además, el aseguramiento de la trazabilidad de los datos a lo largo del ciclo de vida de las operaciones se configura como un eje fundamental en la arquitectura del sistema. Según lo planteado por Parrales-Bravo et al. (2024), la trazabilidad bien implementada permite detectar con precisión requisitos ambiguos, corregir inconsistencias en tiempo real y garantizar una base de datos coherente, especialmente cuando se manejan múltiples actores y flujos de información simultáneos.

Por lo que en el artículo científico presentado a continuación se muestra el diseño e implementación de un sistema de gestión de información que centralice y automatice los procesos de montaje, distribución, recaudación y mantenimiento de máquinas recreativas, aplicando criterios de calidad del software, modularidad estructural y seguridad informática.

**2. Materiales y métodos**

**2.1 Estimación de esfuerzo utilizando el modelo COCOMO I básico**

Siguiendo la metodología de agregación de criterios propuesta por Cañizares Galarza et al. (2021) para tratar la incertidumbre en la toma de decisiones, a continuación se detallan los pasos y resultados de la estimación de esfuerzo, duración y tamaño de equipo mediante el modelo básico de COCOMO I, aplicable a un proyecto web que emplea tecnologías como HTML, CSS, JavaScript, PHP, MySQL y JSX.

**2.1.1 Cálculo de puntos de función**

| Categoría | Ítems identificados | Complejidad | Peso unitario | Total PF |
| --- | --- | --- | --- | --- |
| Entradas de usuario | 10 (Login, registrar usuario, registrar recaudación, registrar máquina, registrar actividades logísticas, registrar reporte, actualizar perfil, recuperar usuario/contraseña, chat de usuarios) | Media | 4 | 10 × 4 = 40 |
| Salidas de usuario | 5 (Informe de recaudación, reporte de estado de máquinas, notificaciones a usuarios, confirmaciones de eliminación, mensajes de errores/mensajes de éxito de formularios) | Media | 4 | 5 × 4 = 20 |
| Consultas de usuario | 8 (consultar usuarios, consultar recaudaciones, consultar máquina, consultar proveedores, consultar perfil, buscar por fecha reportes, consultar actividad logística/técnicos) | Media | 4 | 8 × 4 = 32 |
| Archivos lógicos internos (ILF) | 6 (Usuarios, máquinas, recaudaciones, proveedores, técnicos, reportes y comentarios) | Media | 7 | 6 × 7 = 42 |
| Interfaces externas (EIF) | 3 (API, página estática, dashboard) | Media | 5 | 3 × 5 = 15 |
| Total de PF |  |  |  | **149 PF** |

**2.1.2 Conversión a Líneas de Código (LOC)**

F**actor de conversión**: 1 PF = 58 LOC

**LOC estimados**: 149 PF × 58 LOC/PF = 8 642 LOC ≈ 8.642 KLOC

**2.1.3 Cálculo del esfuerzo (EAF)**

Para reflejar la influencia de factores de costo (EAF), se emplea la agregación multiplicativa de los 14 atributos tradicionales de COCOMO I, cuyos valores se listan a continuación:

1. **Software**
   * Fiabilidad: 0.88
   * Tamaño de BD: 1.08
   * Complejidad: 1.15
2. **Hardware**
   * Restric. tiempo de ejecución: 1.30
   * Restric. memoria virtual: 1.00
   * Tiempo de respuesta: 1.15
3. **Personal**
   * Capacidad de análisis: 1.00
   * Experiencia en la app: 1.29
   * Calidad de programadores: 0.86
   * Experiencia en VM: 1.21
   * Experiencia en lenguaje: 1.07
4. **Proyecto**
   * Técnicas actualizadas de programación: 0.91
   * Uso de herramientas de SW: 0.91
   * Restricción de tiempo de desarrollo: 1.10

**EAF** = 0.88 × 1.08 × 1.15 × 1.30 × 1.00 × 1.15 × 1.00 × 1.29 × 0.86 × 1.21 × 1.07 × 0.91 × 0.91 × 1.10 = **2.137**

**2.1.4 Modelo básico COCOMO I (Submodelo semidesacoplado)**

Parámetros para un proyecto de tamaño medio con equipo experimentado en entorno conocido:

* a = 3.0
* b = 1.12
* c = 2.5
* d = 0.35

1. **Esfuerzo no ajustado**

E0​=a×(SIZE)^b=3.0×(8.642)^1.12=3.0×10.23=30.69 persona-mes

1. **Esfuerzo ajustado**

E=E0​×EAF=30.69×2.137= 65.59 persona-mes

1. **Duración**

D=c×E^d=2.5×(65.59)^0.35=2.5×3.11=7.78 meses

1. **Personal promedio**

**P=E/D​= 65,59 / 7,78= 8.43 personas**

1. **Productividad**

LOC/E= 8642/65,59 = 132 LOC / persona-mes

El primer paso consistió en identificar los elementos funcionales del sistema y asignarles pesos según su complejidad. Esto arrojó un total de 149 Puntos de Función (PF), los cuales fueron convertidos a líneas de código (LOC) mediante un factor estándar de 58 LOC por PF, resultando en una estimación de 8.642 KLOC.

Posteriormente, se calcularon los factores de ajuste del esfuerzo (EAF), considerando 14 atributos relacionados con el software, el hardware, el personal y el proyecto. La multiplicación de estos valores generó un EAF de 2.137, lo que refleja un entorno con ciertas restricciones técnicas pero también con fortalezas como la experiencia del equipo.

Aplicando los parámetros típicos del modelo básico de COCOMO I para un proyecto de tamaño medio (a = 3.0, b = 1.12, c = 2.5, d = 0.35), se obtuvo un esfuerzo base de 30.69 persona-mes, que al ser ajustado con el EAF, resultó en un esfuerzo total estimado de 65.59 persona-mes.

A partir de este esfuerzo ajustado, se estimó una duración del proyecto de 7.78 meses, y un equipo promedio de 8.43 personas. Finalmente, la productividad esperada se calculó en 132 líneas de código por persona-mes, lo cual es coherente con proyectos desarrollados con tecnologías web modernas y equipos con experiencia moderada.

**2.2 Automatización y calidad del software**

El uso de tecnologías automatizadas permite reducir significativamente los errores humanos derivados de la gestión manual. Según Sommerville (2011), la automatización de tareas repetitivas no solo mejora la eficiencia operativa, sino que proporciona datos confiables que sustentan decisiones gerenciales estratégicas. En el caso de las empresas de entretenimiento electrónico, la automatización permite, por ejemplo, programar mantenimientos preventivos, generar informes financieros en tiempo real y coordinar rutas de distribución con menor margen de error.

Asimismo, Pressman (2005) resalta que los productos de software deben ser diseñados considerando la mantenibilidad, la reutilización y la modularidad, ya que estas propiedades garantizan que los sistemas puedan evolucionar sin comprometer su estabilidad o aumentar su complejidad técnica.

**2.3 Arquitectura en capas**

El sistema propuesto adopta una arquitectura distribuida en tres capas: presentación, lógica de negocio y acceso a datos. Este enfoque se alinea con las recomendaciones de diseño estructurado planteadas por Pressman (2005), lo que permite una separación clara de responsabilidades y promueve la reutilización de componentes. Una arquitectura de este tipo facilita el mantenimiento evolutivo del sistema, ya que cada capa puede modificarse o actualizarse de forma independiente sin afectar la integridad funcional global.

De acuerdo con Ajiga et al. (2024), la adopción de una arquitectura en capas también es fundamental para garantizar la escalabilidad del software. Este enfoque permite agregar nuevos módulos o integrar tecnologías emergentes, como inteligencia artificial o analítica predictiva, sin la necesidad de rediseñar el sistema de manera extensiva. Este tipo de arquitectura también facilita la incorporación de nuevas funcionalidades o servicios según se desarrollen nuevas necesidades empresariales, lo cual es una característica clave de los sistemas diseñados para apoyar el crecimiento de negocios dinámicos.

**2.4 Seguridad y control de acceso**

La gestión de la seguridad en entornos de software que manejan información crítica, como la de las máquinas recreativas, requiere mecanismos robustos de autenticación y políticas claras de control de accesos. Menéndez (2023) señala que un sistema informático seguro debe incluir la validación de identidades, la restricción de privilegios según el rol del usuario y la trazabilidad de las actividades a través de bitácoras de auditoría.

Ajiga et al. (2024) destacan que la implementación de controles de acceso en sistemas empresariales debe ser un proceso continuo y adaptable a las nuevas amenazas. Para ello, proponen el uso de marcos de trabajo escalables que no solo protejan los datos, sino que también permitan una gestión eficiente de las políticas de seguridad, asegurando que el sistema pueda crecer sin comprometer la seguridad a medida que cambian las necesidades del negocio.

**3. Diagnóstico y justificación del rediseño del sistema**

El análisis del contexto inicial reveló un conjunto de deficiencias estructurales vinculadas al uso de procedimientos manuales y registros físicos, lo que generó retrasos en la operación, errores en la gestión documental y limitaciones en la visibilidad de datos. Este tipo de problemática, según lo plantea Sommerville (2011), es característico de organizaciones que aún no han adoptado un enfoque sistemático en la ingeniería de software, lo cual limita su capacidad de adaptación, respuesta y control.

Entre las principales fallas detectadas destacan los retrasos operativos derivados de una escasa sincronización entre áreas, consecuencia de la desactualización de datos compartidos. Estos cuellos de botella, como afirma Pressman (2005), suelen originarse por la inexistencia de sistemas de información integrados que respalden las actividades críticas del negocio. Asimismo, se observaron errores de registro que comprometieron la confiabilidad de los reportes financieros y operativos, situación que Menéndez (2023) relaciona directamente con vulnerabilidades en los procesos de captura, almacenamiento y verificación de datos. A esto se suma la falta de visibilidad, manifestada en la imposibilidad de generar reportes centralizados y actualizados para la toma de decisiones estratégicas.

Frente a este diagnóstico, se diseñó un sistema con capacidad para centralizar información operativa y administrativa, alineándose con las recomendaciones de Pressman (2005), quien resalta la importancia de sistemas que reduzcan redundancias y garanticen una única fuente de verdad. El objetivo fue optimizar recursos humanos eliminando tareas repetitivas, así como mejorar la precisión en la contabilización de ingresos y en el control de inventarios, lo que responde a los principios de calidad del software descritos por Sommerville (2011). También se buscó automatizar la generación de reportes, dotando a la organización de herramientas más robustas para el análisis y la toma de decisiones.

**4. Metodología de investigación**

La investigación se sustentó en un enfoque descriptivo-aplicativo, apoyado en el ciclo de vida tradicional del desarrollo de software, lo que permitió seguir una secuencia estructurada de actividades, conforme a los principios de ingeniería sistemática propuestos por Pressman (2005). El proceso se desarrolló en seis fases detalladas a continuación:

1. **Levantamiento de información**: Se identificaron los actores clave (técnicos, contables y gerentes) para comprender los procesos actuales. Esta etapa inicial coincide con lo propuesto por Sommerville (2011), quien destaca la necesidad de involucrar a los stakeholders para lograr especificaciones más precisas y relevantes.
2. **Análisis de requerimientos**: Se priorizaron más de 20 requerimientos funcionales y no funcionales. Según Pressman (2005), este análisis garantiza que el software responda a las verdaderas necesidades del negocio y permite establecer criterios de validación desde el inicio del proyecto.
3. **Diseño del sistema**: Utilizando el estándar IEEE 1016-2009, se elaboraron diagramas de casos de uso, clases y despliegue. Esta documentación formal promueve la trazabilidad del sistema y su mantenibilidad, tal como lo argumenta Sommerville (2011).
4. **Desarrollo del prototipo**: Se implementó una versión funcional utilizando tecnologías como HTML, CSS, JS, PHP y MySQL. Este prototipo permitió validar la lógica de negocio y la integración de componentes, en concordancia con las prácticas de desarrollo incremental sugeridas por Pressman (2005).
5. **Pruebas de aceptación**: Los usuarios finales evaluaron el sistema considerando tiempo de respuesta, integridad de datos y eficiencia percibida. Este enfoque se relaciona con el modelo de satisfacción evaluado por Parrales-Bravo et al. (2025), quienes, a través de la Escala Likert Neutrosófica, demostraron que características como facilidad de uso, velocidad y confiabilidad son determinantes para lograr una alta satisfacción del usuario final.
6. **Despliegue y capacitación**: Finalmente, se procedió con la instalación del sistema y la formación de usuarios clave. Este paso fue clave para garantizar una adopción efectiva y sostenible del sistema, de acuerdo con los principios de implementación gradual y acompañamiento definidos por Sommerville (2011).

Este rediseño responde de manera integral a las necesidades operativas detectadas, incorporando buenas prácticas de ingeniería de software, criterios de seguridad informática y herramientas metodológicas que permiten obtener una retroalimentación continua por parte de los usuarios, tal como lo recomienda Parrales-Bravo et al. (2025), para asegurar mejoras iterativas y sostenidas en el tiempo.

**Diseño del sistema**

**2.1 Arquitectura modular**

El sistema se ha estructurado bajo una arquitectura modular que permite la escalabilidad, mantenibilidad y separación de responsabilidades entre sus componentes. Está organizado en tres capas principales:

* **Capa de presentación:** Interfaz web responsiva desarrollada con **ReactJS**, que interactúa con el backend mediante peticiones HTTP. El enrutamiento principal se gestiona desde App.jsx, mientras que cada módulo posee vistas específicas para sus funcionalidades.
* **Capa de lógica de negocio:** Implementada mediante servicios RESTful en **PHP**, donde cada módulo cuenta con sus propios controladores (Controller.php) que comunican la presentación con la lógica central definida en los servicios (Service.php).
* **Capa de acceso a datos:** Utiliza una base de datos relacional MYSQL, optimizada mediante el uso de tablas especializadas para mejorar el rendimiento y la integridad de la información.

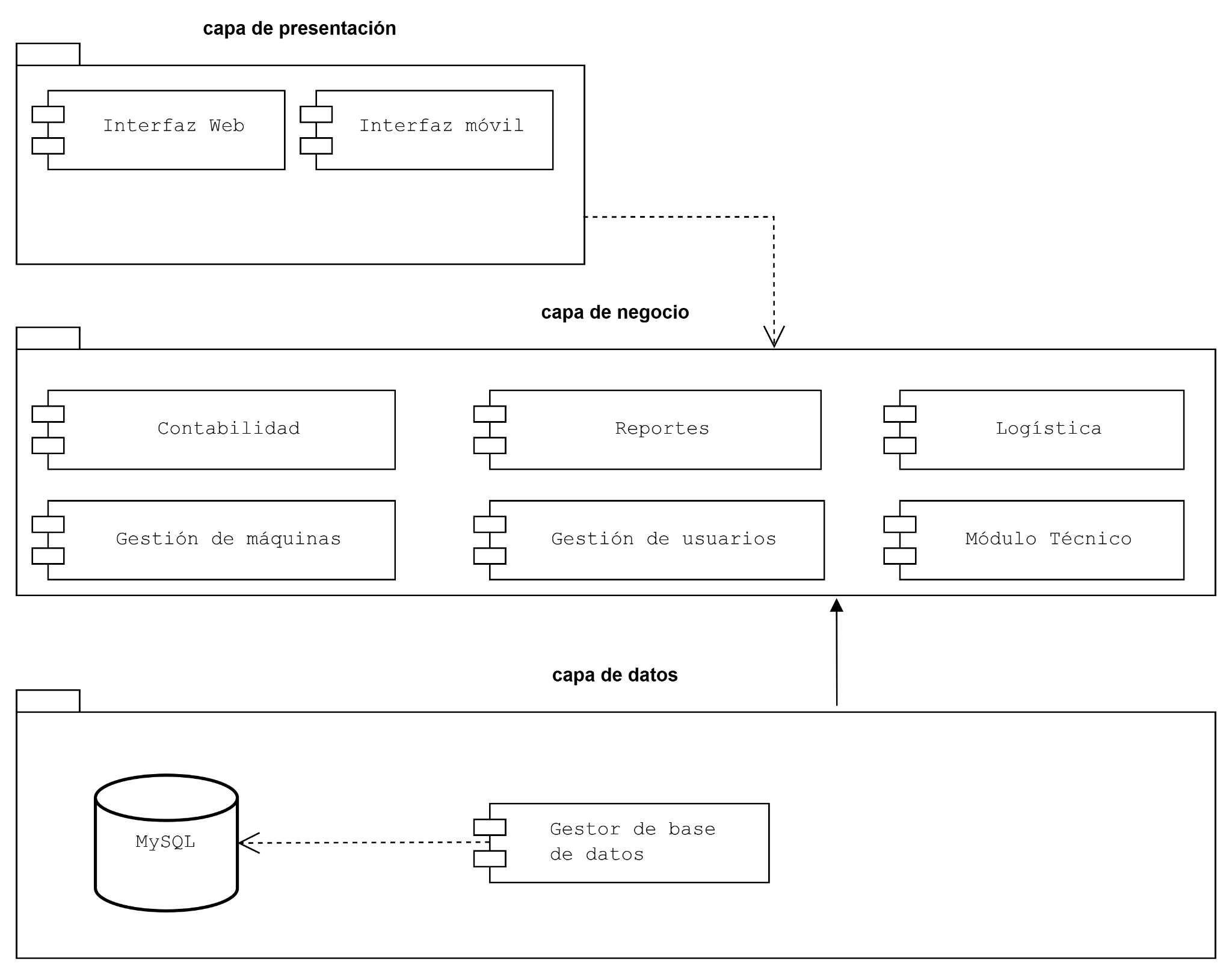
**2.2 Componentes principales del sistema**

Los módulos del sistema se organizan funcionalmente, permitiendo una operación distribuida pero integrada:

* **Módulo administrador:** Gestiona cuentas de usuario, roles y permisos. Incluye funciones de creación, edición, eliminación y consulta de usuarios.
* **Módulo contabilidad:** Administra el registro y control de recaudaciones económicas del sistema. Permite registrar, editar y visualizar ingresos.
* **Módulo logística:** Supervisa la distribución y seguimiento de las máquinas recreativas en distintas ubicaciones. Gestiona el inventario geolocalizado.
* **Módulo máquina recreativa:** Lleva el control del inventario, el estado físico y los mantenimientos de las máquinas operativas.
* **Módulo reporte:** Centraliza la gestión de reportes, comentarios entre usuarios y notificaciones del sistema en tiempo real.
* **Módulo técnico:** Separa tareas técnicas como ensamblaje, mantenimiento y verificación de máquinas, asignando roles diferenciados a cada tipo de técnico.
* **Módulo seguridad y acceso:** Controla el inicio de sesión, autenticación y autorización por roles. Incluye protección de rutas privadas dentro del sistema.
* **Módulo perfil y recuperación:** Permite la actualización del perfil del usuario, así como la recuperación de credenciales mediante correo electrónico.

Figura 1

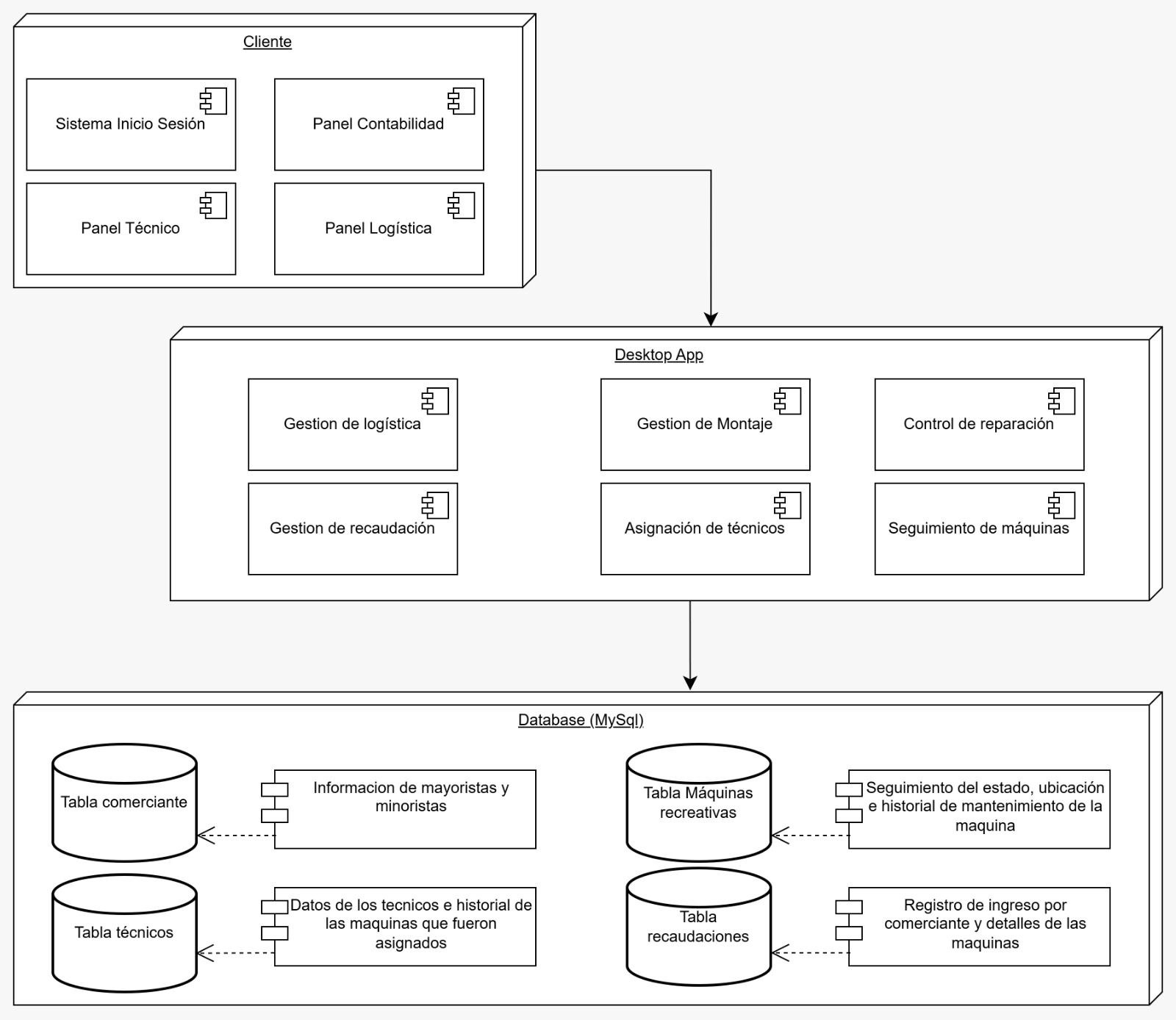
*Diagrama de componentes del sistema*



**Nota:** Este diagrama muestra la arquitectura en capas del sistema, organizada en tres niveles: presentación, negocio y datos.

Figura 2

*Vista de Despliegue - Diagrama del Ambiente Físico*



**Nota:** Este diagrama representa cómo se distribuyen físicamente los componentes del sistema en el entorno real.  
Muestra la interacción entre los módulos del cliente (interfaz de usuario) y la aplicación de escritorio, los cuales acceden a distintas tablas en un servidor de base de datos (MySQL).  
Incluye funcionalidades específicas como gestión de logística, montaje, recaudación, y el uso de tablas que almacenan información crítica sobre comerciantes, máquinas recreativas y técnicos.

## 5. Resultados y discusión

Los resultados evidencian un impacto positivo tras la implementación del sistema de información en la empresa de máquinas recreativas. La arquitectura desarrollada permitió automatizar funciones clave como el registro de montaje, la gestión de componentes y proveedores, la asignación de técnicos, la emisión de reportes financieros y la generación automática de notificaciones de mantenimiento.

De acuerdo con Castro Bermúdez, Solarte Martínez y Muñoz Guerrero (2019), la planificación y gestión de la calidad del software debe centrarse en la correcta definición de procesos, roles y métricas, promoviendo la estandarización y reduciendo la incidencia de errores humanos. La solución propuesta sigue este principio, mostrando mejoras en la trazabilidad, en la eficiencia operativa y en la flexibilidad del sistema para adaptarse a necesidades futuras.

### 5.1. Impacto en la coordinación y adopción del sistema

La implementación de una arquitectura modular favoreció la integración de funciones y la coordinación entre áreas operativas. Según Castro Bermúdez et al. (2019), la calidad del software no debe entenderse únicamente desde lo técnico, sino también desde su capacidad para responder a los objetivos del negocio mediante una correcta gestión de procesos, comunicación efectiva y cumplimiento de estándares.

Los beneficios observados incluyen:

* Centralización de datos operativos, mejorando la accesibilidad y fiabilidad de la información.
* Disminución de tiempos de respuesta en mantenimiento y distribución.
* Generación de reportes precisos y oportunos para la toma de decisiones.
* Optimización del desempeño del personal técnico y administrativo.

Estas mejoras se alinean con lo planteado por los autores al resaltar que un sistema de aseguramiento de la calidad eficaz debe incorporar tanto elementos de verificación como de validación continua durante el ciclo de vida del software.

### 5.2. Comparación con Soluciones Existentes en el Mercado

Este sistema supera a alternativas como GESTIMAQ y PROMARE en trazabilidad de procesos, gracias a su enfoque en el ciclo de vida completo. Sin embargo, estudios previos (Castro Bermúdez et al., 2019) destacan que la telemetría, como la ofrecida por GESTIMAQ-LTE, es crítica para empresas con máquinas en ubicaciones remotas. Nuestra solución prioriza la adaptabilidad a PYMES mediante tecnologías abiertas (PHP, MySQL), evitando costos de licencia. Futuras iteraciones cerrarán esta brecha integrando IoT, tal como recomienda Ajiga et al. (2024) para frameworks escalables.

## 6. Conclusiones

El desarrollo de un sistema de información adecuado no solo resuelve problemas inmediatos, sino que también fortalece la competitividad y sostenibilidad de la empresa en el mercado de máquinas recreativas.

La implementación de bitácoras automáticas permitió reconstruir el historial completo de cada máquina, facilitando la identificación de fallas recurrentes y el análisis de causas raíz.

### 6.1. Acciones y mejoras propuestas

Desde una mirada hacia el futuro del producto, se contempla que ciertos requisitos funcionales y no funcionales podrían postergarse para ser implementados en versiones más avanzadas del sistema. En cuanto a las funcionalidades, se prevé que el registro de mantenimientos, la generación de informes correspondientes, la gestión del proceso de distribución de máquinas recreativas y el registro de recaudaciones mensuales sean incorporados en etapas futuras, conforme evolucione la necesidad operativa del sistema.

Por otro lado, también se han identificado varios requisitos no funcionales cuya implementación se priorizará progresivamente. Entre ellos, se encuentran aspectos cruciales como la seguridad de la información y de los datos, la protección ante posibles ataques, y los mecanismos de respaldo y recuperación de información. Además, se considera fundamental asegurar un buen rendimiento y tiempo de respuesta, aplicar balanceo de carga, garantizar la escalabilidad vertical para la gestión de recaudaciones y facilitar el despliegue eficiente del sistema.

Como parte de las acciones y mejoras propuestas que acompañarán el crecimiento del producto, se plantea el desarrollo de un módulo móvil que permita registrar información en tiempo real desde dispositivos Android, lo que incrementa la agilidad operativa en campo. También se proyecta la incorporación de análisis predictivo mediante técnicas de aprendizaje automático para anticipar posibles fallas y mejorar la gestión preventiva.

Si bien el sistema desarrollado resuelve necesidades críticas de trazabilidad y modularidad, áreas subestimadas por GESTIMAQ y PROMARE, su principal limitación es la ausencia de telemetría en tiempo real. Trabajos futuros integrarán sensores IoT y una app móvil, ya mencionada, para recaudadores, acercándose a las capacidades de soluciones comerciales pero manteniendo la ventaja competitiva.

## Referencias bibliográficas

Ajiga, D., Okeleke, P. A., Folorunsho, S. O., & Ezeigweneme, C. (2024). Methodologies for developing scalable software frameworks that support growing business needs. *International Journal of Management, Entrepreneurship and Research, 6*, 2661–2683.

Castro Bermúdez, Y. V., Solarte Martínez, G. R., & Muñoz Guerrero, L. E. (2019). *Planificación, Gestión y Control de la Calidad del Software*. Scientia Et Technica, 24(4), 611–617

Cañizares Galarza, F. P., Becerra Arévalo, N. P., Jalón Arias, E. J., & Acosta Espinoza, J. L. (2021). Prioritization of

software requirements using neutrosophic TOPSIS. *Neutrosophic Sets and Systems, 44*(1), 23–30.

Gonzales Arechavala, Y.; De Cuadra, F. Calidad del software II. <https://www.researchgate.net/publication/265964918_Calidad_del_software_el_camino_al_exito>

Micromaq. (s.f.). Gestimaq: Software para gestión de máquinas recreativas. Recuperado el 6 de julio de 2025, de <https://www.micromaq.es/recreativo.html>

Menéndez, S. (2023). *Auditoria de seguridad informática, curso práctico*. Alfaomega.

Procode. (s.f.). Software de gestión para máquinas recreativas. Recuperado el 6 de julio de 2025, de <https://procode.es/home/sectores/recreativo/>

Parrales-Bravo, F., Gómez-Rodríguez, V., Chiquito-Vera, L., Rendón-Quijije, I., Caicedo-Quiroz, R., Tolozano-Benites, E., & Cevallos-Torres, L. (2024). DEAR: Detecting ambiguous requirements as a way to develop skills in requirement specifications. *Electronics, 13*(15), 3079. <https://doi.org/10.3390/electronics13153079>

Parrales-Bravo, F., Agurto-Pincay, J., Tolozano-Benites, R., Arteaga-Yaguar, E., Vasquez-Cevallos, L., Cevallos-Torres, L., Rumbaut-Rangel, D., & Gómez-Rodríguez, V. (2025). Neutrosophic Likert Scale to assess the user satisfaction with retail software developed by software engineering students. Neutrosophic Sets and Systems, 84, 312–325.

Pressman, R. S. (2005). *Ingeniería de software: Un enfoque práctico* (7.ª ed.). McGraw-Hill.

Ramírez, V. M. & Simbaña, J. S. (2019). Diseño de un sistema de gestión de mantenimiento...  
 <https://dspace.ups.edu.ec/handle/123456789/17759>

Sánchez-Rodríguez, A. P. (2010). La gestión de los activos físicos en la función mantenimiento.  
 <http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59442010000200008>

Sommerville, I. (2011). *Ingeniería de software* (9.ª ed.). Pearson Educación.