**20 resúmenes**

**Nombre:**

**Johan Calderón Perdomo**

**Instructor:**

**Jesús Ariel González**

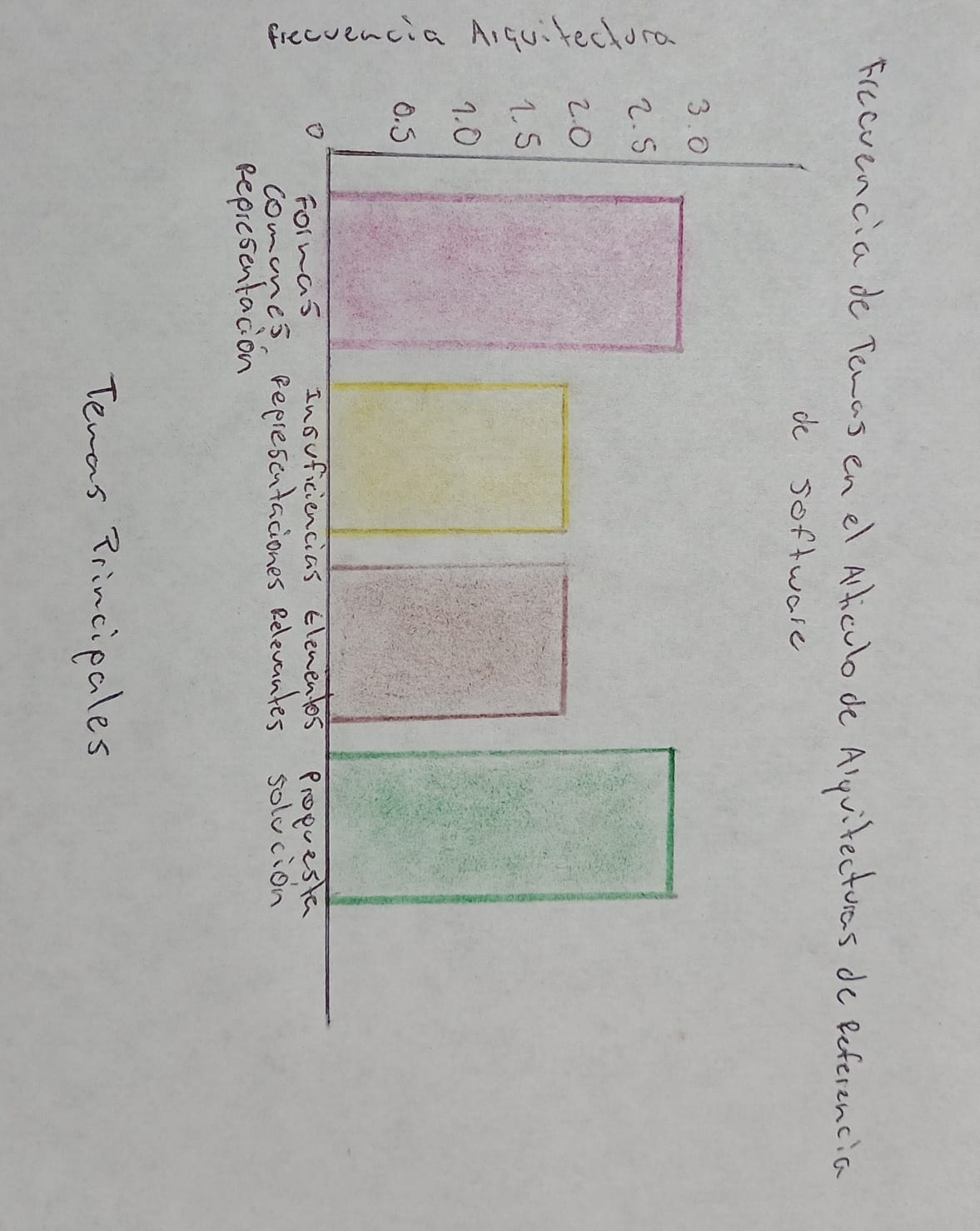
**Servicio Nacional de Aprendizaje SENA Sede Industria**

**Neiva – Huila**

**14-09-2024**

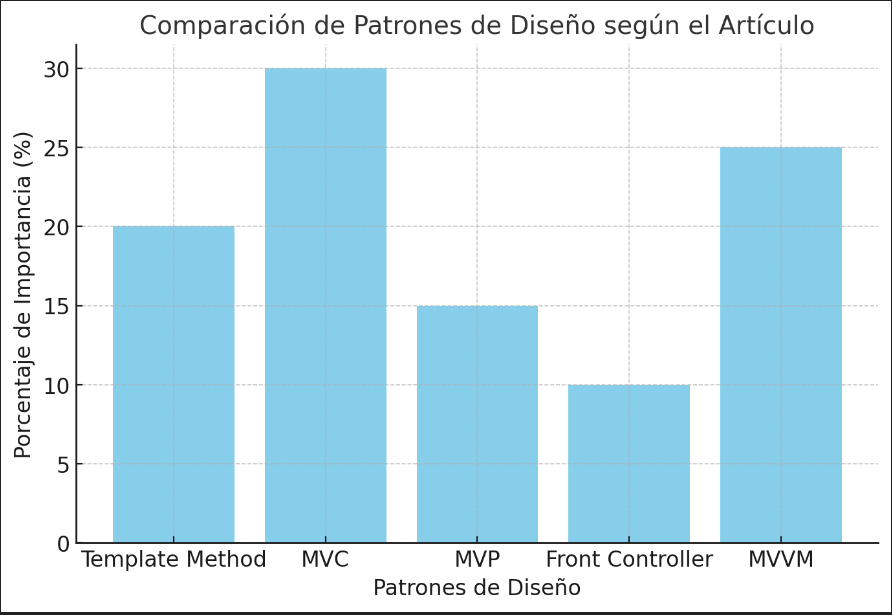
**Revisión de elementos conceptuales para la representación de las arquitecturas de referencias de software**

La Arquitectura de Software organiza un sistema a nivel alto, definiendo elementos y sus relaciones. Facilita el desarrollo al reutilizar componentes mediante arquitecturas de referencia. Ha evolucionado a través de enfoques estructurales, patrones y escenarios, guiando el diseño y la comprensión de sistemas complejos.



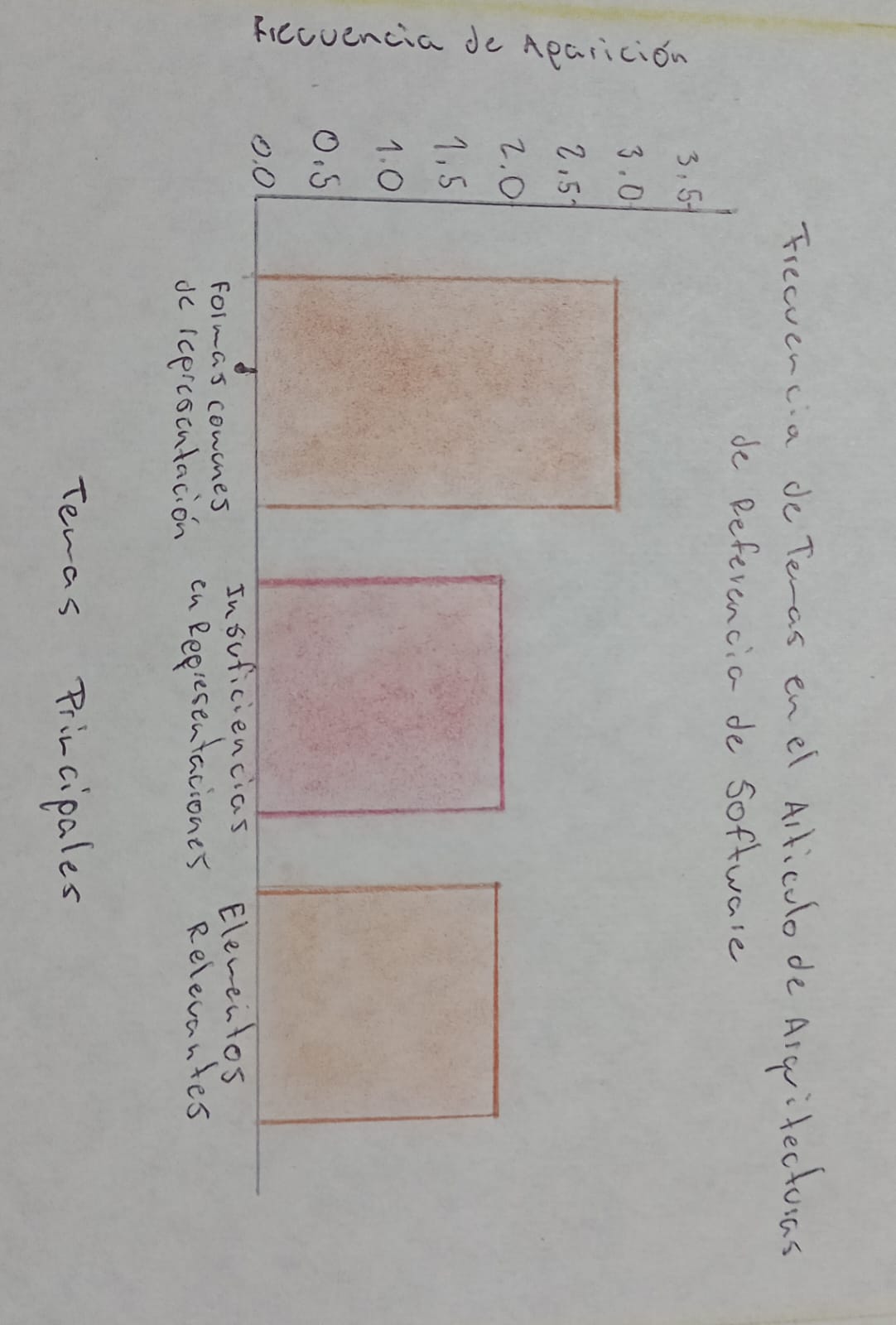
**Análisis comparativo de Patrones de Diseño de Software**

Los patrones de diseño evitan duplicación de código y mejoran la calidad del software. El artículo compara varios patrones, concluyendo que no hay uno superior, ya que cada uno tiene un propósito específico, facilitando el mantenimiento y el modularidad del sistema.



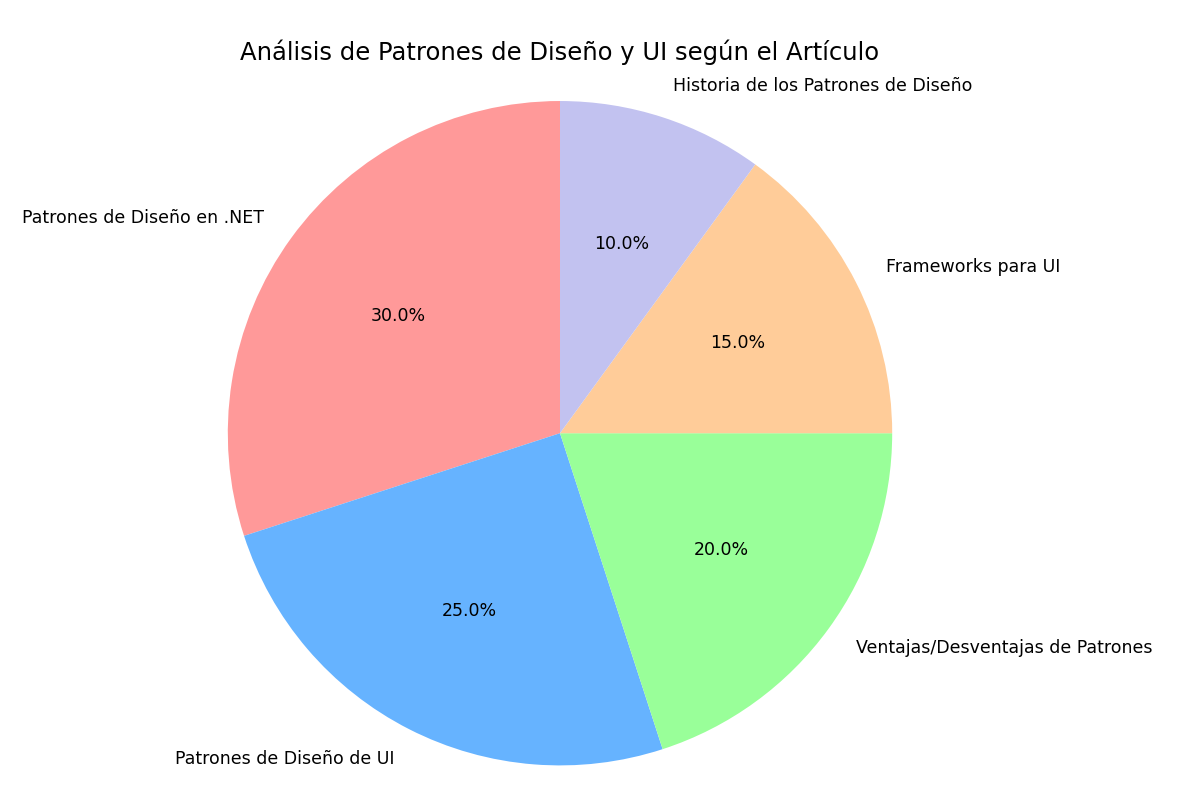
**Análisis de secuencia discretas para la detección de Patrones de Diseño de Software**

Los Agentes de Interfaz ofrecen asistencia personalizada en software. Este trabajo propone usar Modelos de Márkov para detectar patrones de diseño en herramientas CASE, mejorando la precisión y tiempos de respuesta, y generando recomendaciones basadas en las acciones del usuario.



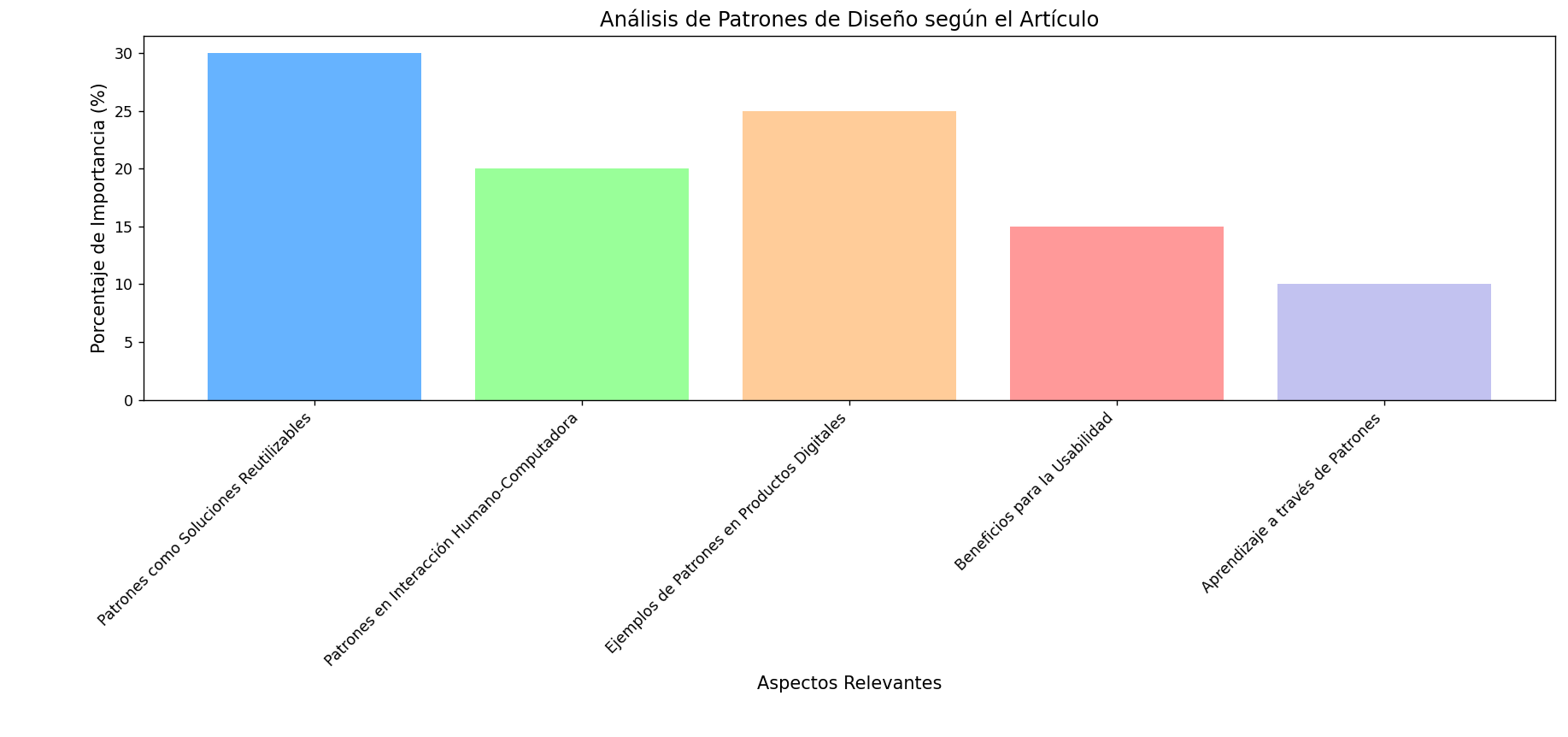
**Patrones de Diseño**

Los patrones de diseño, popularizados por el libro Design Patterns en 1994, proporcionan soluciones reutilizables para crear software escalable y flexible. Este documento aborda dos tipos de patrones: los de diseño para la implementación en código, explicando seis patrones usados en .NET, y los de diseño de interfaz de usuario, enfocados en mejorar la experiencia del usuario en sitios web, con ejemplos y frameworks recomendados para facilitar el desarrollo.



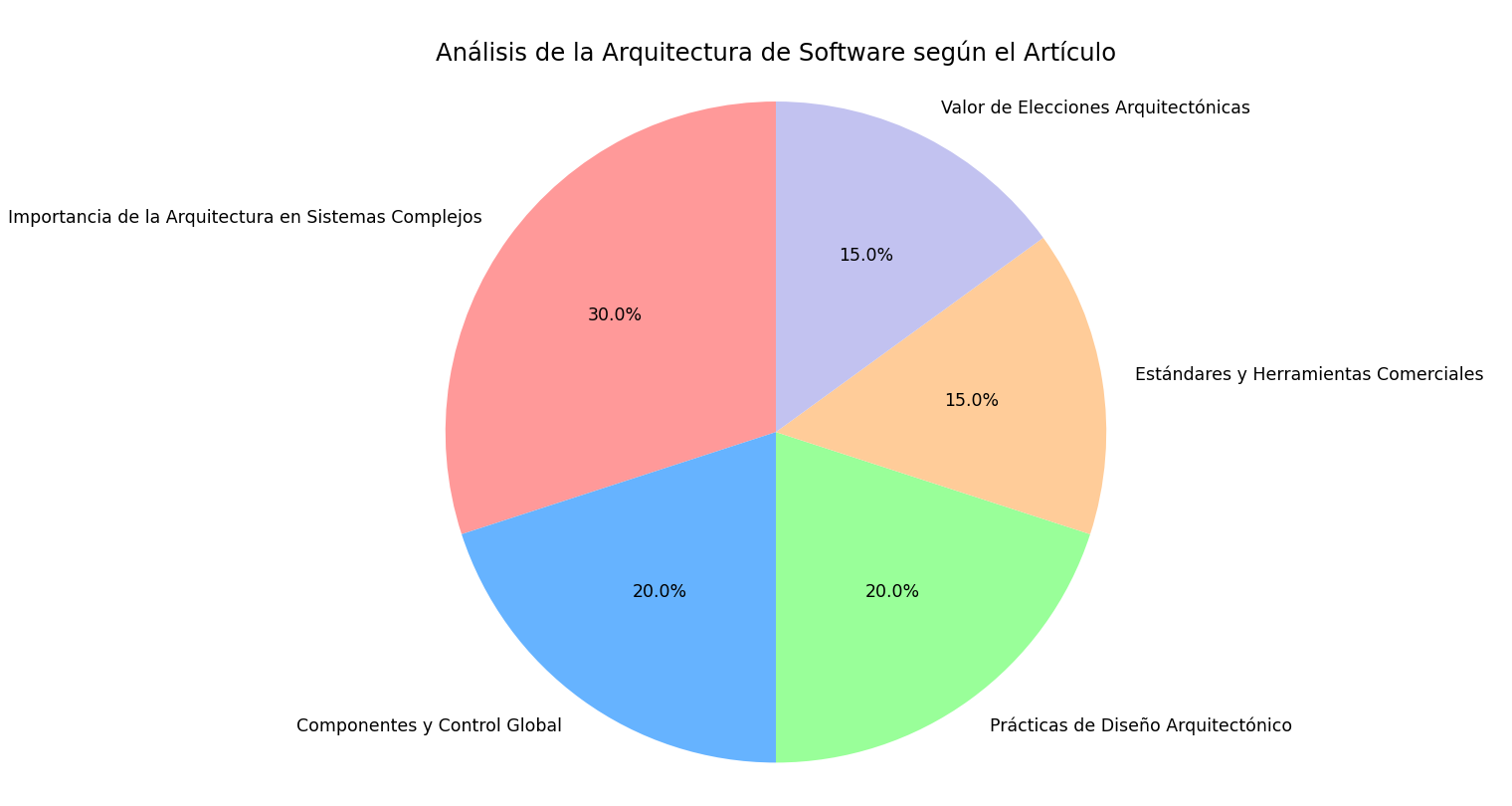
**Patrones de Diseño: Mejores Experiencias y Productos Fácilmente Escalables**

Los patrones de diseño en experiencias digitales facilitan el uso para los usuarios, quienes ya están familiarizados con ciertos estándares. Siguiendo estos patrones, se mejora la usabilidad y se evita una curva de aprendizaje empinada, lo que ayuda a que los usuarios se concentren en el contenido del sitio en lugar de en cómo usarlo. Esto también aumenta la eficiencia del equipo de desarrollo.



**Software Architecture**

El diseño arquitectónico en software es crucial para manejar la creciente complejidad de los sistemas. Una buena arquitectura garantiza requisitos clave como rendimiento, escalabilidad y confiabilidad. En la última década, ha ganado importancia con la creación de libros, herramientas, cursos y estándares formales que ayudan a reutilizar y mejorar los diseños arquitectónicos.

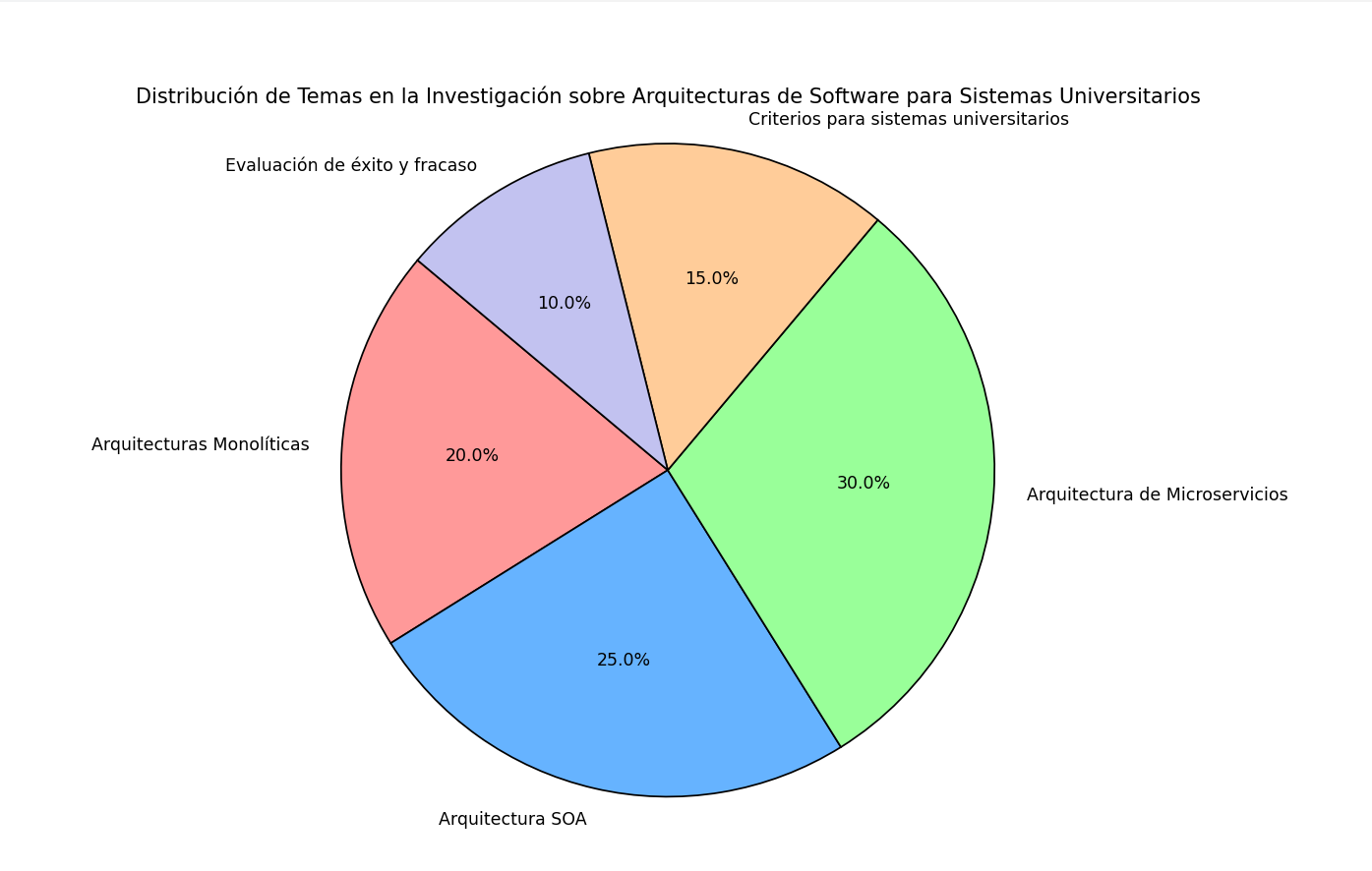


**A Systematic Review on Software Architectures for IoT Systems and Future Direction to the Adoption of Microservices Architecture**

Este estudio revisa la investigación sobre la adopción de la Arquitectura de Microservicios (MSA) en sistemas de Internet de las Cosas (IoT). Analiza 140 artículos para identificar patrones arquitectónicos, desafíos y soluciones en IoT. Destaca trece patrones de MSA, clasifica arquitecturas en nueve categorías, y aborda veintitrés desafíos principales. Los hallazgos ofrecen recomendaciones para mejorar la arquitectura y el desarrollo de software de IoT, beneficiando a la industria y a la comunidad académica.

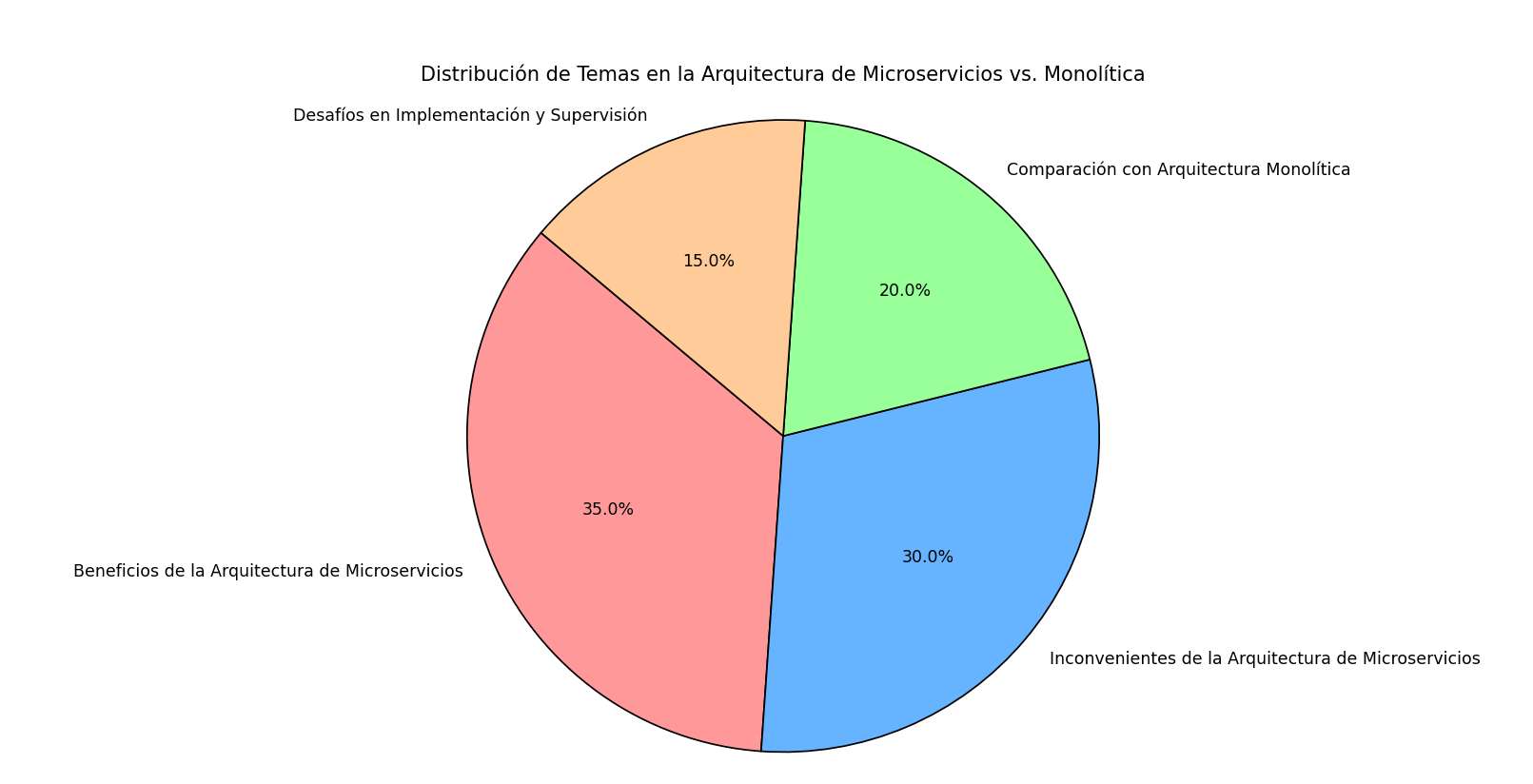
**Software Architectures - Present and Visions**

El artículo analiza diferentes tipos de arquitecturas de sistemas de software, como monolíticas, SOA y microservicios, y evalúa cuál es más adecuada para sistemas universitarios. Examina desafíos específicos, factores de éxito y causas de fracaso, y ofrece una recomendación objetiva para la implementación en el ámbito académico.



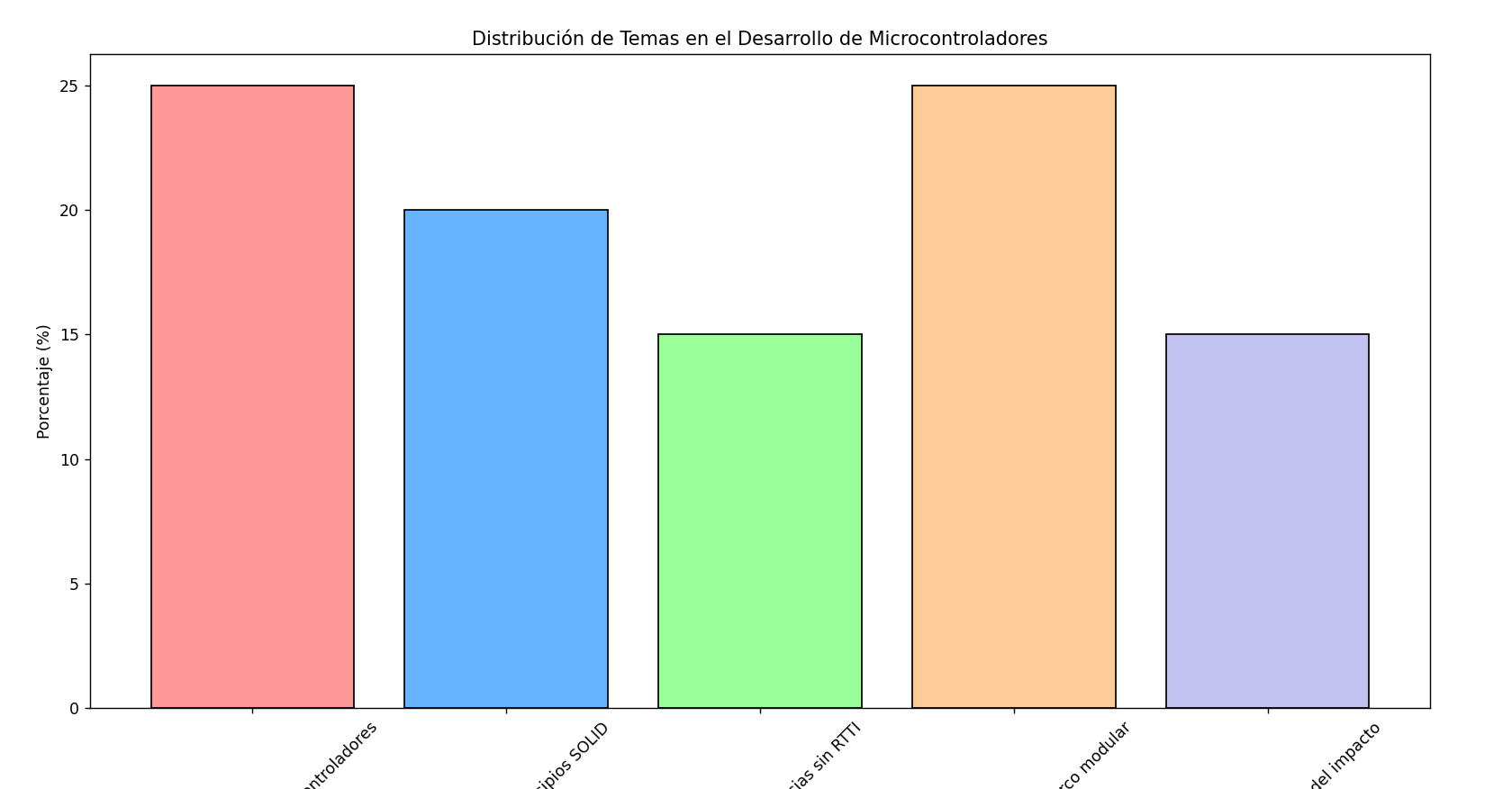
**A scalable architecture for automated monitoring of microservices**

El artículo propone una arquitectura para monitorear microservicios, registrando métricas clave de rendimiento y utilizando algoritmos avanzados para clasificar eventos y automatizar decisiones. Emplea Prometheus para métricas a corto plazo, OpenTSDB para almacenamiento a largo plazo y RabbitMQ para la transmisión de mensajes.



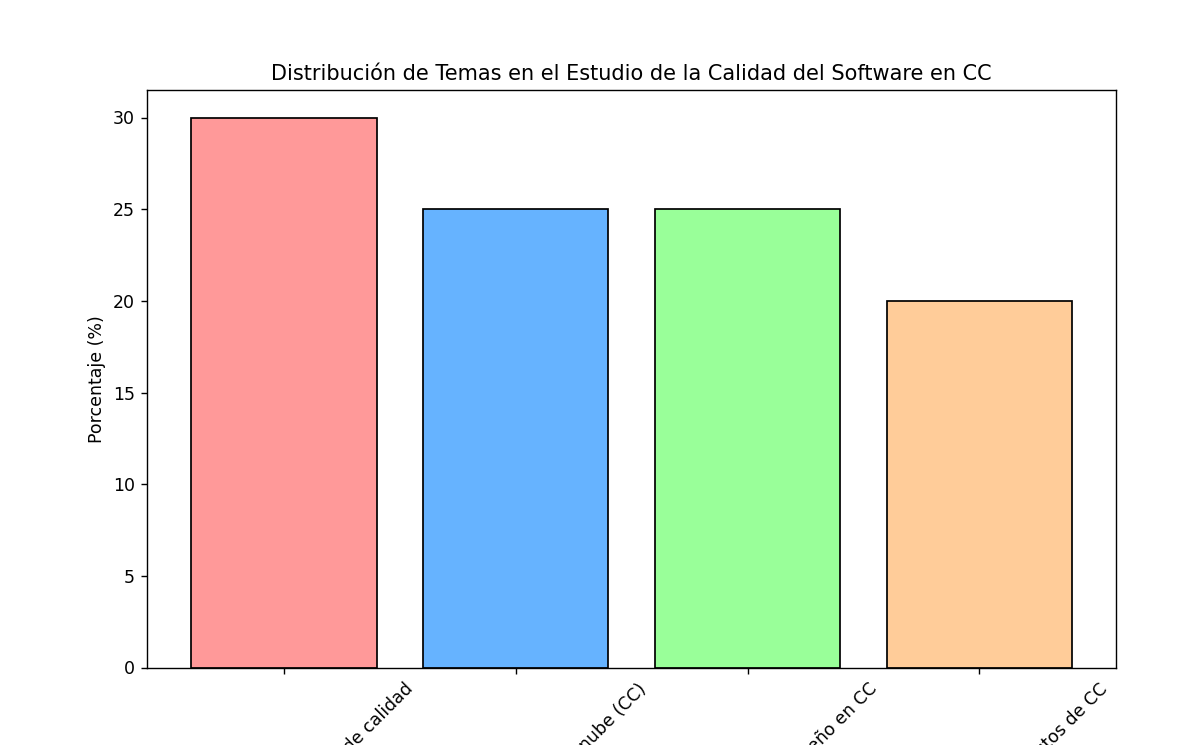
**Implementation of a Universal Framework Using Design Patterns for Application Development on Microcontrollers**

El artículo presenta un marco modular para microcontroladores que aplica principios modernos y patrones de diseño para mejorar la calidad del software en sistemas integrados e IoT. Este enfoque permite la comunicación independiente entre módulos, integración flexible de hardware y adaptabilidad a diferentes entornos de desarrollo, mejorando la sostenibilidad y escalabilidad a largo plazo.



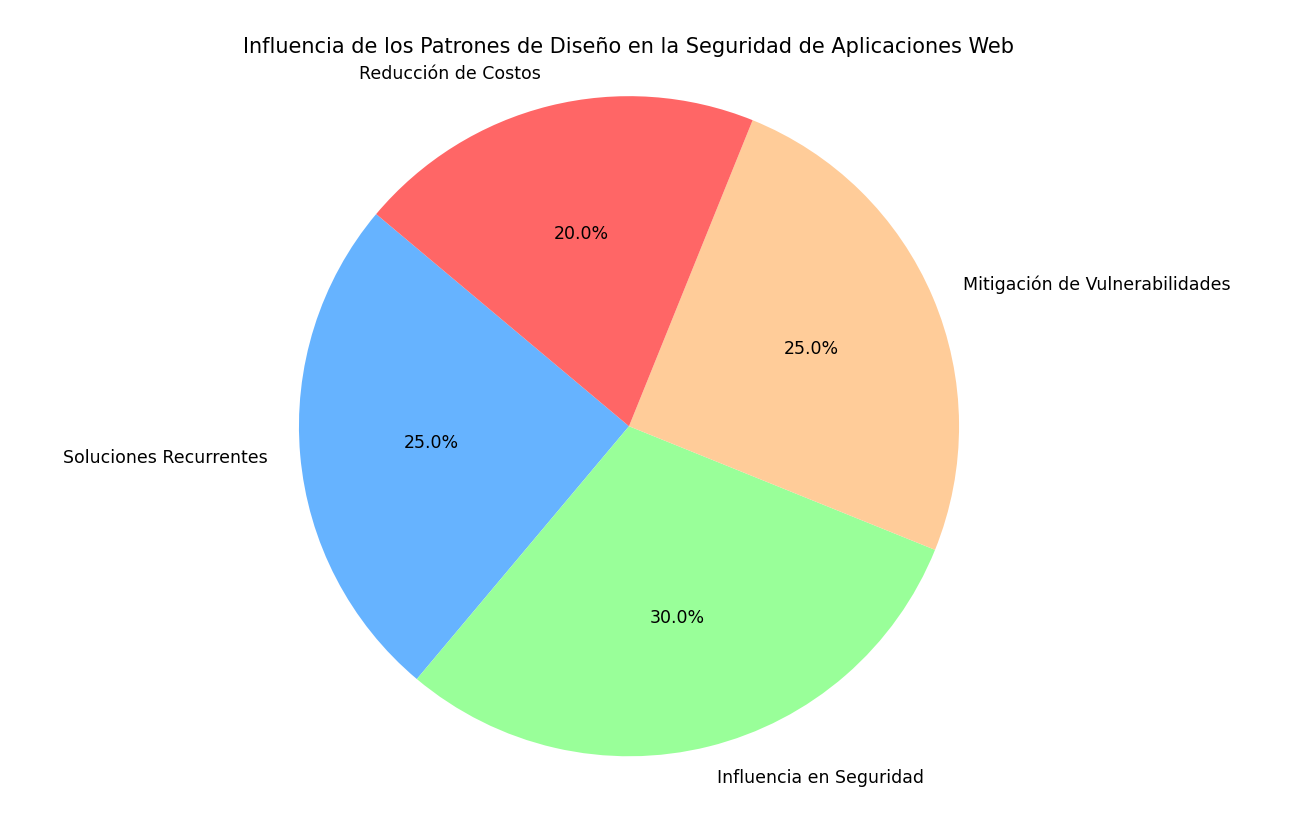
**Modelado y Verificación de Patrones de Diseño de Arquitectura de Software para Entornos de Computación en la Nube**

El trabajo presenta una herramienta gráfica para diseñar arquitecturas de software en aplicaciones web, basada en patrones de diseño. Se enfoca en cómo las funcionalidades del software se relacionan con los atributos de calidad y utiliza el paradigma de computación en la nube para ofrecer acceso a recursos configurables de manera eficiente.



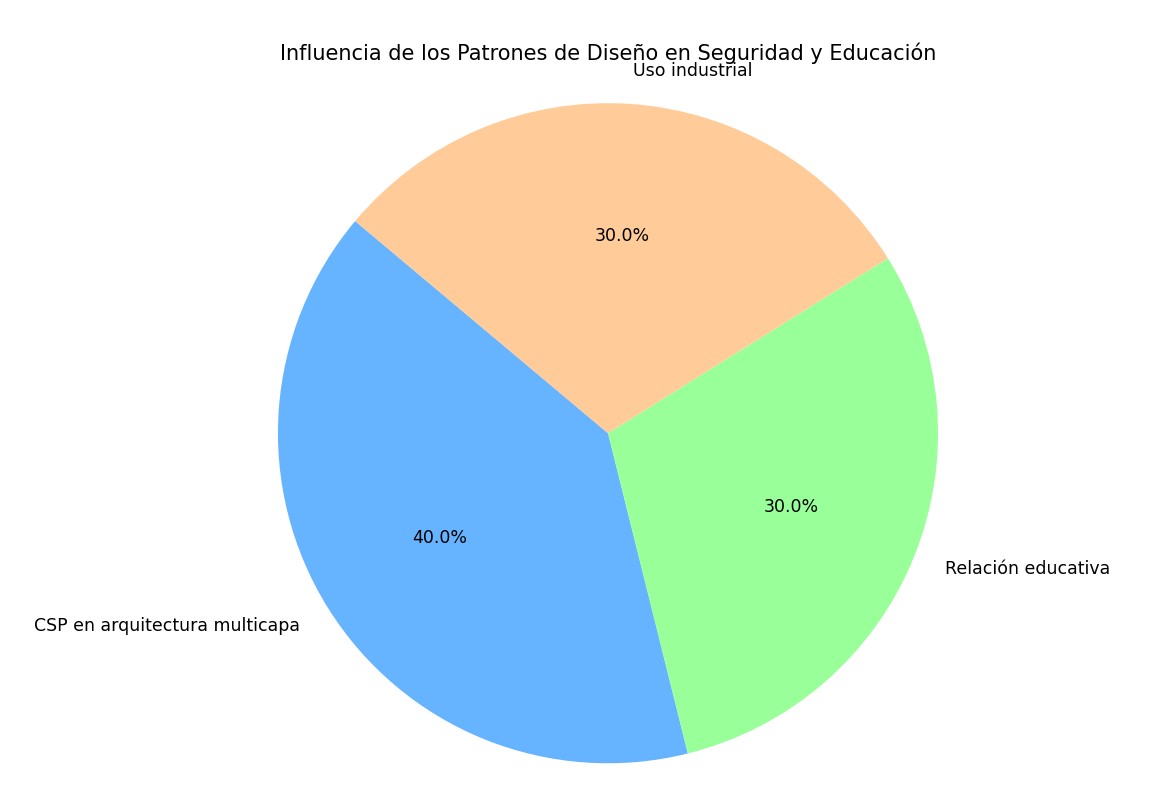
**Incidencia de los Patrones de Diseño de Software en la Seguridad de Aplicaciones Web**

Los patrones de diseño ayudan a estructurar y organizar el código de manera eficiente, y pueden mejorar la seguridad de aplicaciones web al abordar problemas como validación de datos y autenticación. La investigación busca determinar cómo estos patrones ayudan a mitigar vulnerabilidades, reduciendo costos y mejorando la mantenibilidad del software.



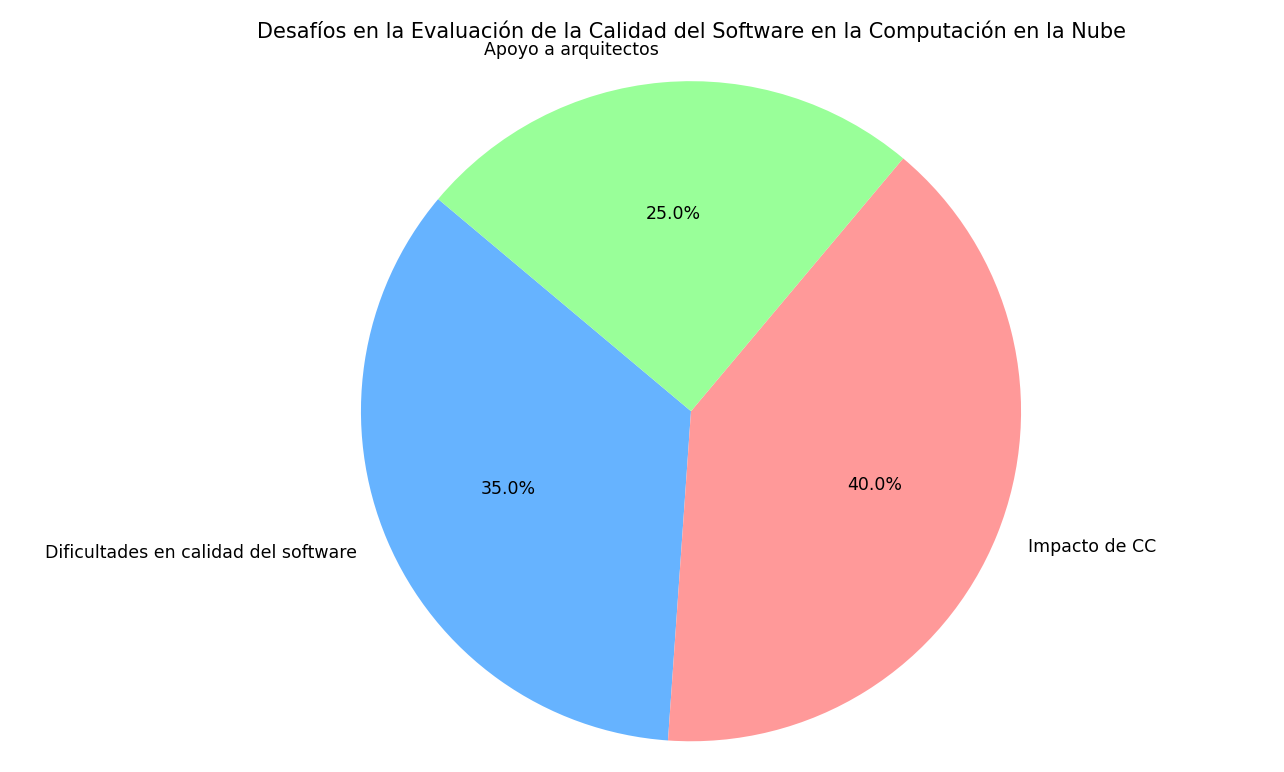
**Integración de patrones de seguridad y patrones de diseño J2EE**

El trabajo analiza dos catálogos populares de patrones de diseño de software de Sun Microsystems (Oracle), define sus relaciones para un uso coherente y presenta casos prácticos de aplicaciones que integran patrones de ambos catálogos.



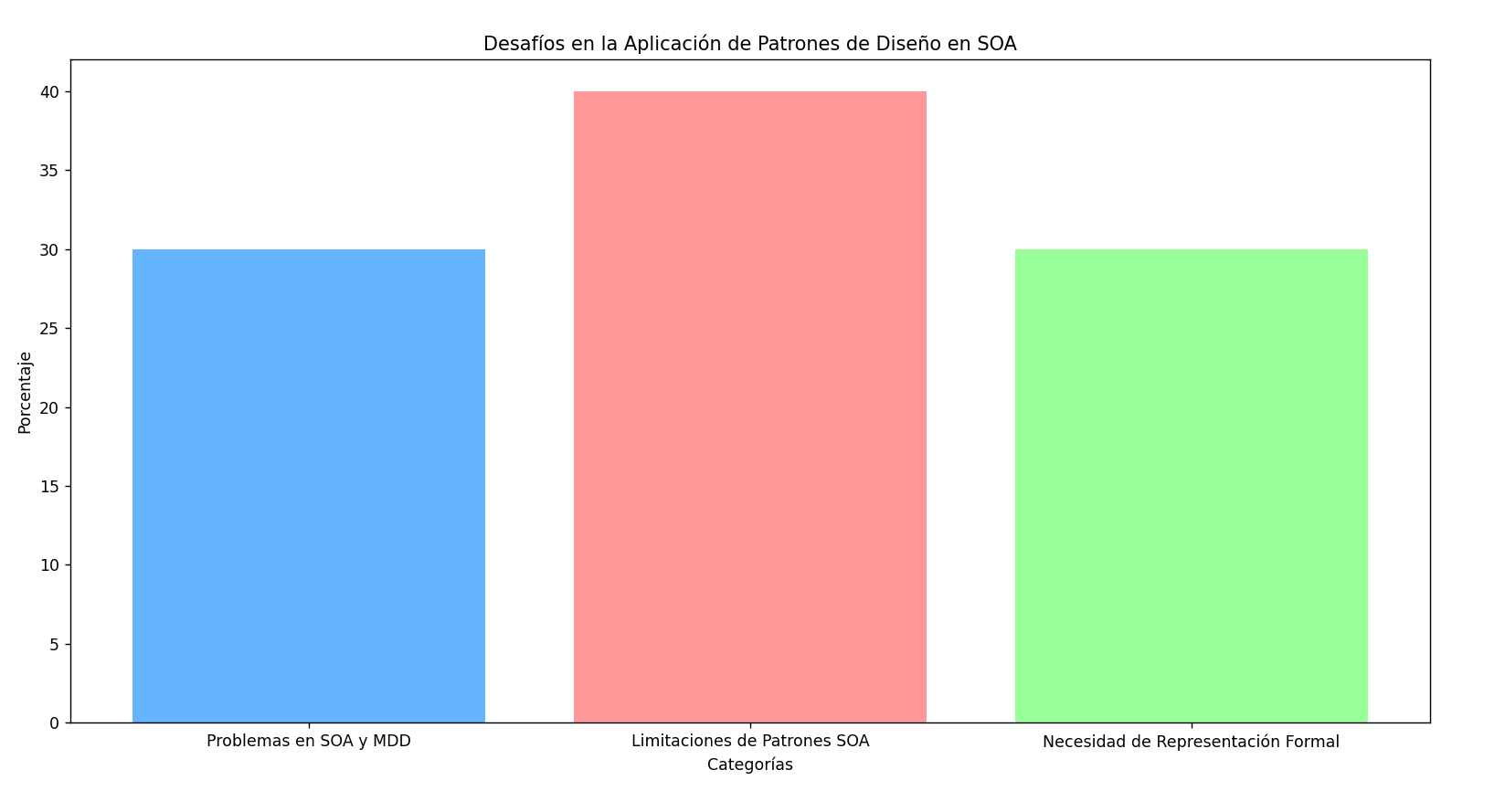
**Modelado y Verificación de Patrones de Diseño de Arquitectura de Software para Entornos de Computación en la Nube**

El trabajo presenta una herramienta gráfica para diseñar arquitecturas de software en aplicaciones web, utilizando un modelo de soporte. Destaca que las funcionalidades del software se separan de los atributos de calidad y que la computación en la nube facilita el acceso a recursos configurables de manera eficiente.



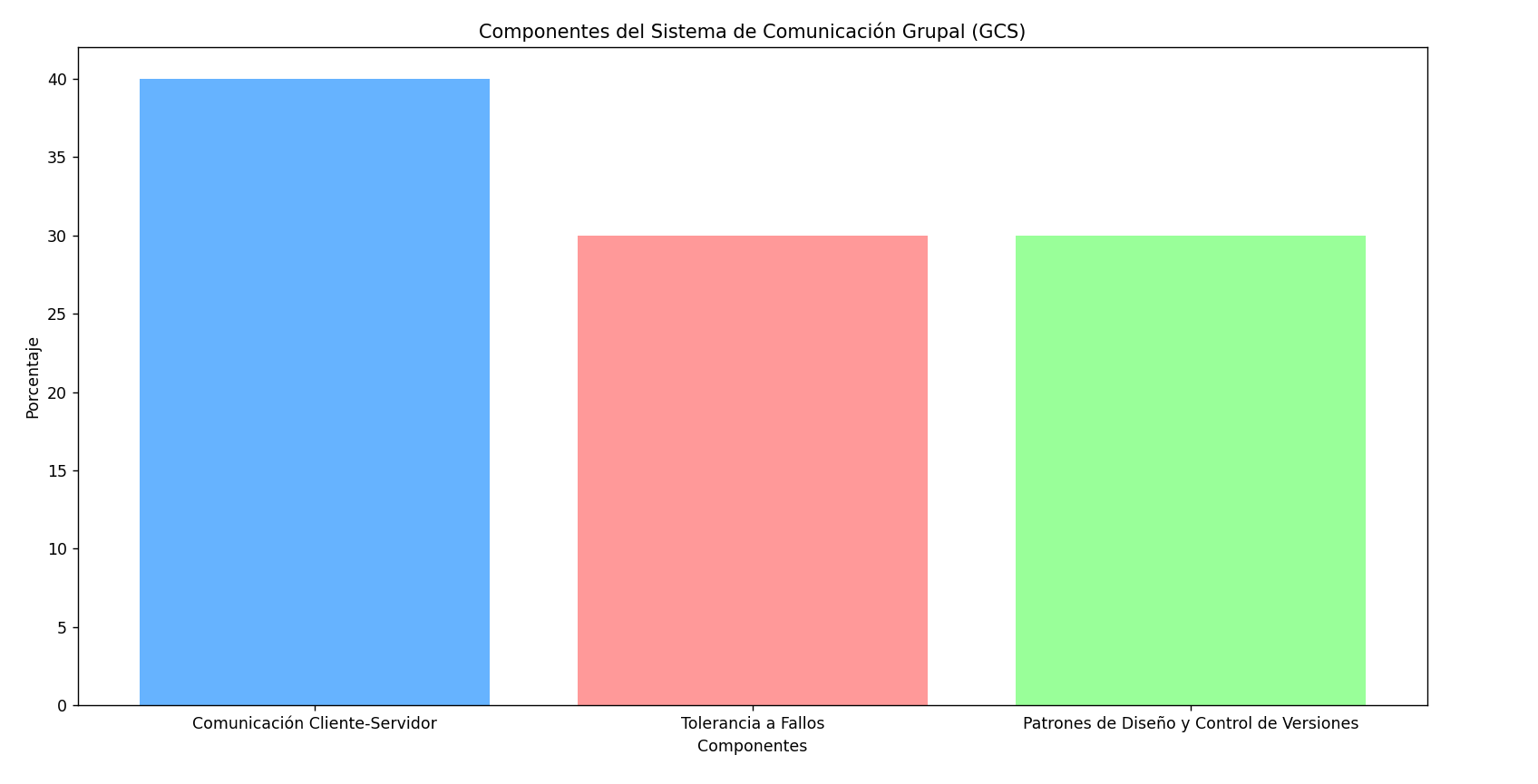
**Application of Design Patterns in Service Oriented Architecture**

El artículo investiga cómo los patrones de diseño informales de Arquitectura Orientada a Servicios (SOA) pueden resolver problemas en el desarrollo de software basado en modelos. Utiliza análisis orientado a objetos y teorías de categorías para formalizar estos patrones, mejorando su aplicación en el modelado de sistemas.



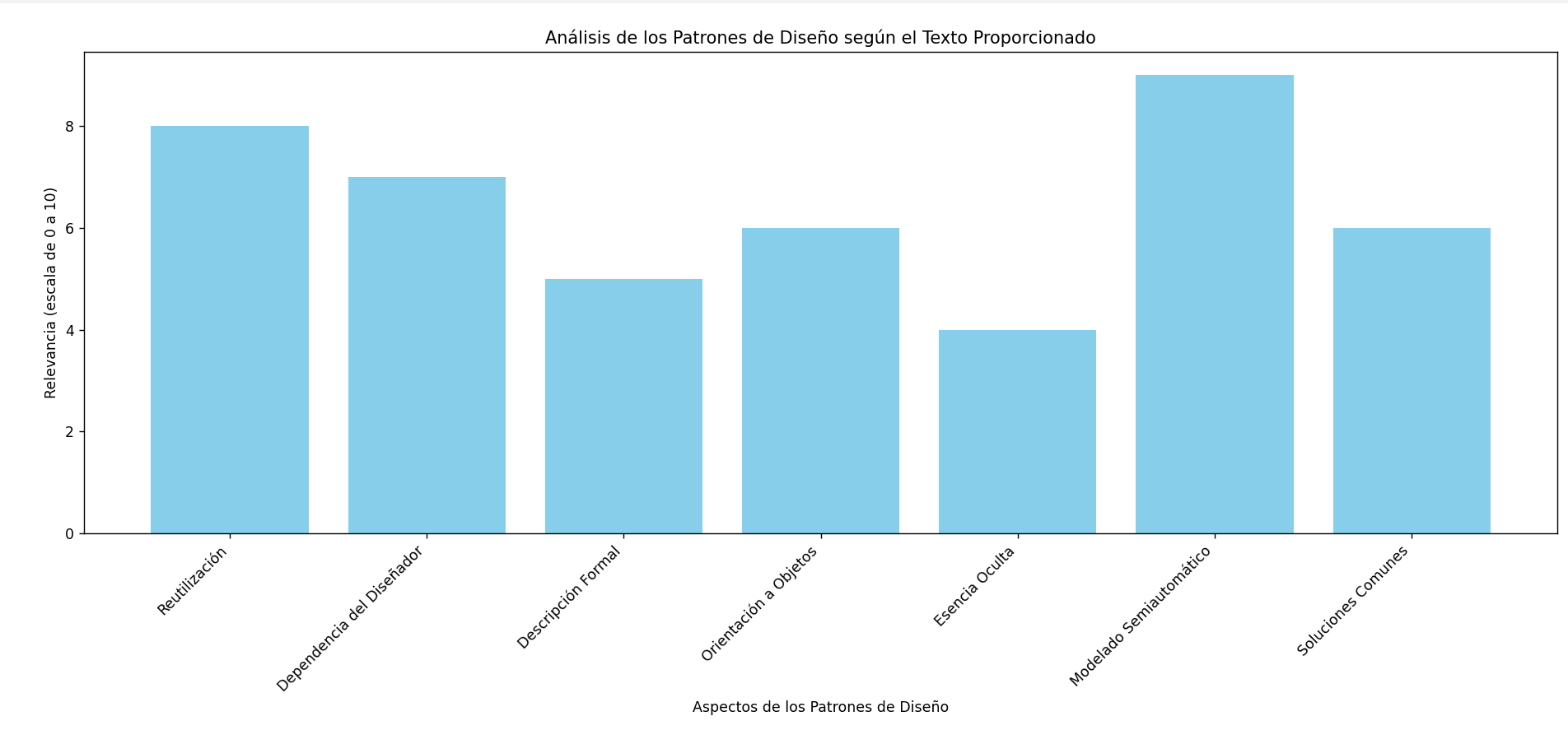
**Design and Implementation of a Java-Based Client-Server Application**

El informe describe el desarrollo del Sistema de Comunicación Grupal (GCS) en Java, que utiliza sockets y protocolos de comunicación para gestionar grupos a través de una interfaz de línea de comandos. Destaca la tolerancia a errores, patrones de diseño y uso de control de versiones, proporcionando una visión completa de su arquitectura e implementación.



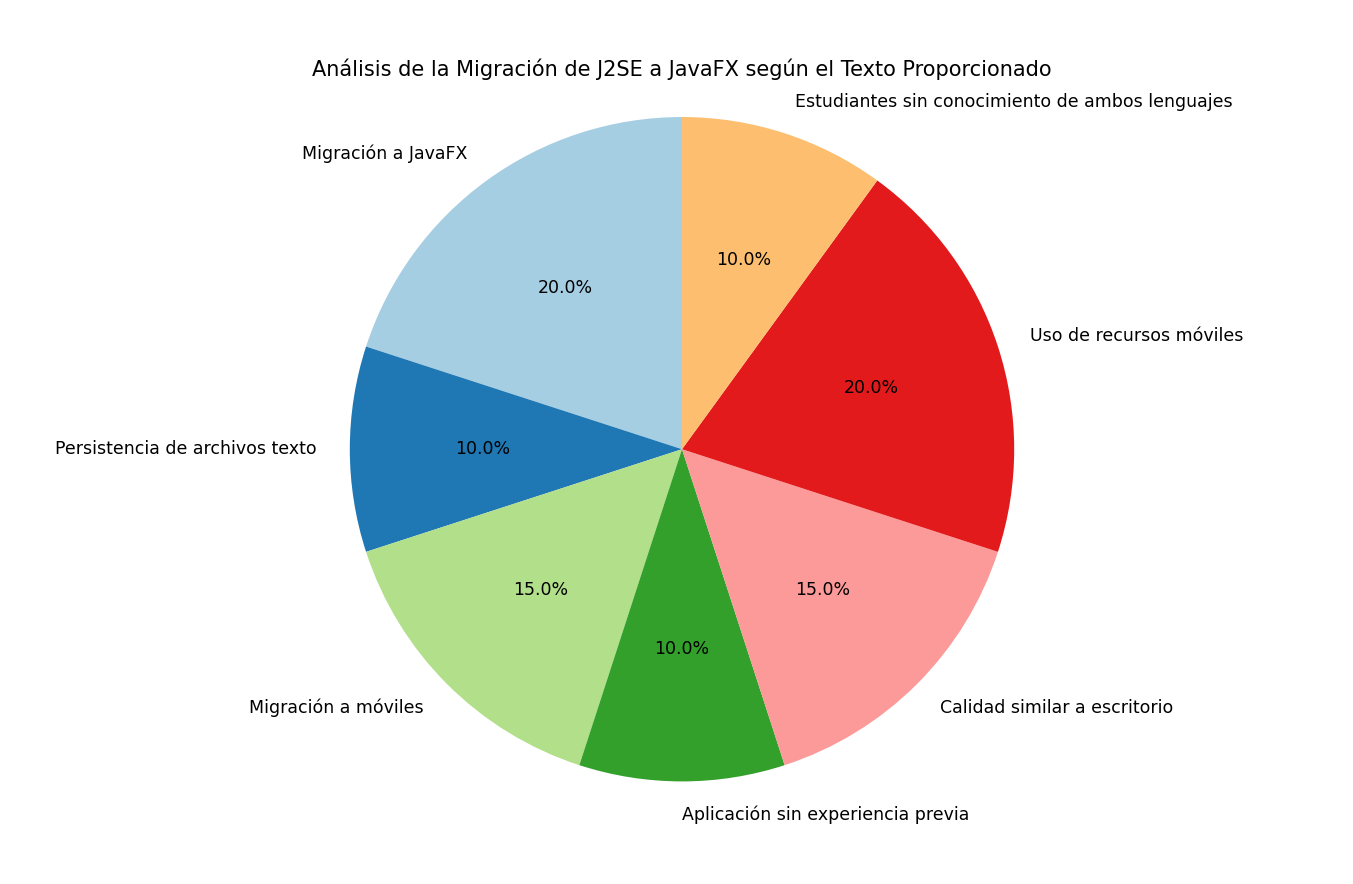
**Modeling Design Patterns for Semi-Automatic Reuse in System Design**

Los patrones de diseño ofrecen soluciones reutilizables para problemas comunes en el desarrollo de software, pero su eficacia depende del diseñador. A menudo, se describen de manera informal o demasiado orientada a objetos. El artículo propone un enfoque para modelar estos patrones y utilizarlos semiautomáticamente para mejorar el diseño del software, mostrando que este método ayuda a identificar problemas clave en aplicaciones prácticas.



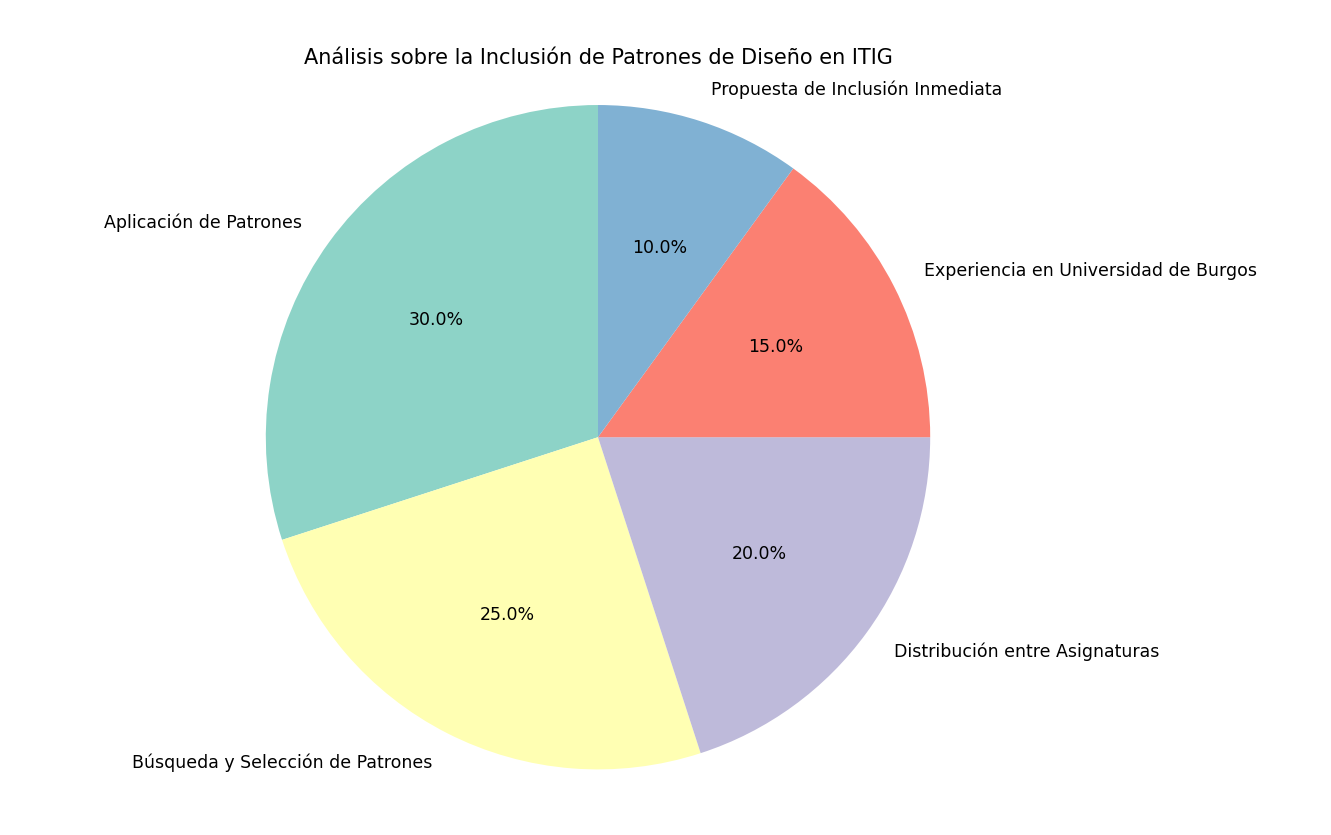
**Convertir aplicaciones de escritorio en aplicaciones móviles con Java**

El artículo presenta un proceso para convertir aplicaciones J2SE a JavaFX, enfocándose en aplicaciones locales que gestionan archivos de texto y se desean migrar a móviles. Proporciona pautas para crear aplicaciones JavaFX desde cero, manteniendo la calidad y funcionalidad de las versiones de escritorio, y se aplicó en un curso avanzado en UNIMINUTO, logrando una conversión exitosa sin necesidad de experiencia previa en JavaFX.



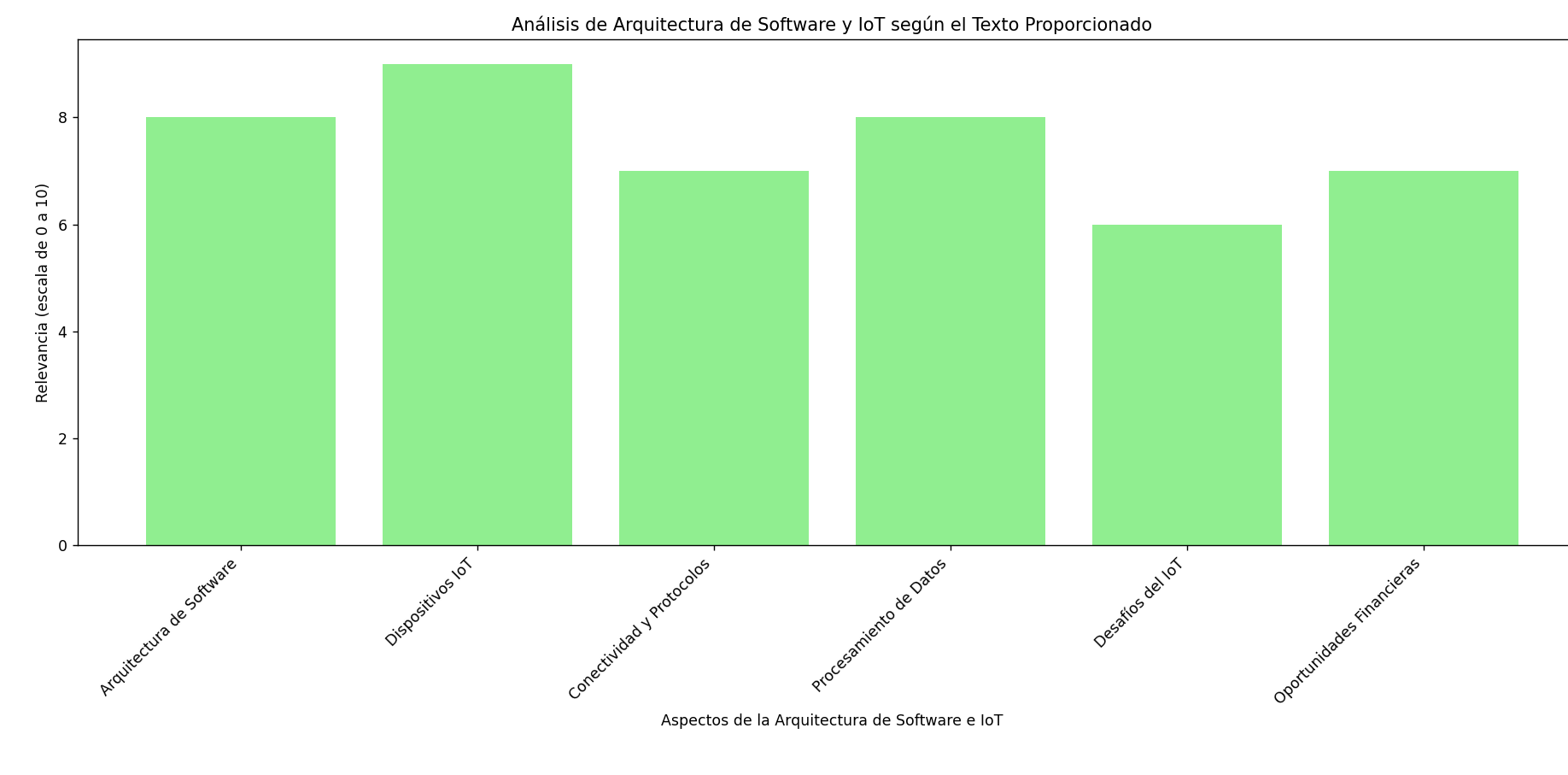
**Inclusión de patrones de diseño en un plan de estudios de Ingeniería Técnica en Informática de Gestión**

El trabajo propone incluir patrones de diseño en el currículo de Ingeniería Técnica en Informática de Gestión (ITIG), diferenciando entre la aplicación y la selección de patrones. Se basa en la experiencia previa de la Universidad de Burgos y sugiere una integración adecuada de estos conceptos en las asignaturas del plan de estudios.



**A Systematic Review on Software Architectures for IoT Systems and Future Direction to the Adoption of Microservices Architecture**

Este estudio revisa la investigación sobre la adopción de la Arquitectura de Microservicios (MSA) en sistemas de Internet de las Cosas (IoT). Analiza 140 artículos para identificar patrones arquitectónicos, desafíos y soluciones en IoT. Destaca trece patrones de MSA, clasifica arquitecturas en nueve categorías, y aborda veintitrés desafíos principales. Los hallazgos ofrecen recomendaciones para mejorar la arquitectura y el desarrollo de software de IoT, beneficiando a la industria y a la comunidad académica.



**Referencias Bibliográficas**

Palmero, M. A. S., Martínez, N. S., & Grass, O. Y. R. (2019). Revisión de elementos conceptuales para la representación de las arquitecturas de referencias de software. *Revista cubana de ciencias informáticas*, *13*(1), 143–157. http://scielo.sld.cu/scielo.php?pid=S2227-18992019000100143&script=sci\_arttext

(S/f). Unirioja.es. Recuperado el 14 de septiembre de 2024, de https://dialnet.unirioja.es/servlet/articulo?codigo=9042927

Logroño, J. F. S., Berdún, L., Armentano, M., & Amandi, A. (s/f). *Análisis de secuencias discretas para la detección de Patrones de Diseño de Software*. Edu.ar. Recuperado el 14 de septiembre de 2024, de https://sedici.unlp.edu.ar/bitstream/handle/10915/152663/Documento\_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y

(S/f-b). Academia.edu. Recuperado el 14 de septiembre de 2024, de https://www.academia.edu/download/57338244/Patrones.pdf

Español, M. (2022, febrero 15). *Patrones de Diseño: Mejores experiencias y productos fácilmente escalables*. Modyo. https://es.modyo.com/blog/patrones-de-diseno-mejores-experiencias-y-productos-facilmente-escalables

Garlan, D. (s/f). *Software Architecture*. Cmu.edu. Recuperado el 14 de septiembre de 2024, de https://kilthub.cmu.edu/articles/journal\_contribution/Software\_Architecture/6609593/files/12101711.pdf

(S/f-c). Proquest.com. Recuperado el 14 de septiembre de 2024, de https://www.proquest.com/docview/2933155446/E860FCE7DDB74622PQ/23?accountid=31491&sourcetype=Scholarly%20Journals

(S/f-d). Proquest.com. Recuperado el 14 de septiembre de 2024, de https://www.proquest.com/docview/1758012314/C6BB69EB2E54D0EPQ/1?accountid=31491&sourcetype=Scholarly%20Journals

(S/f). Proquest.com. Recuperado el 14 de septiembre de 2024, de https://www.proquest.com/docview/2933155446/E860FCE7DDB74622PQ/23?accountid=31491&sourcetype=Scholarly%20Journals

(S/f). Proquest.com. Recuperado el 14 de septiembre de 2024, de https://www.proquest.com/docview/2172010739/38B8A9878E894A8APQ/1?accountid=31491&sourcetype=Scholarly%20Journals

(S/f-b). Proquest.com. Recuperado el 14 de septiembre de 2024, de https://www.proquest.com/docview/3059721054/339A7C7B154647BAPQ/4?accountid=31491&sourcetype=Scholarly%20Journals

(S/f-c). Proquest.com. Recuperado el 14 de septiembre de 2024, de https://www.proquest.com/docview/2343015882/90C25C73A9E343F0PQ/6?accountid=31491&sourcetype=Scholarly%20Journals

Mesías-Valencia, J. J. (2024). Incidencia de los patrones de diseño de software en la seguridad de aplicaciones web. *MQRInvestigar*, *8*(1), 236–259. https://doi.org/10.56048/mqr20225.8.1.2024.236-259

(S/f-d). Ucm.es. Recuperado el 14 de septiembre de 2024, de https://docta.ucm.es/entities/publication/14b254d6-2bc6-481b-abd8-460469bf2dd0

(S/f-e). Proquest.com. Recuperado el 14 de septiembre de 2024, de https://www.proquest.com/docview/2343015882/82B908C636404259PQ/4?accountid=31491&sourcetype=Scholarly%20Journals

(S/f-f). Proquest.com. Recuperado el 14 de septiembre de 2024, de https://www.proquest.com/docview/2162719666/579077592A2F4496PQ/102?accountid=31491

(S/f-g). Proquest.com. Recuperado el 14 de septiembre de 2024, de https://www.proquest.com/docview/3040141605/579077592A2F4496PQ/111?accountid=31491&sourcetype=Working%20Papers

(S/f-h). Proquest.com. Recuperado el 14 de septiembre de 2024, de https://www.proquest.com/docview/2954642945/abstract/D88747EC066945C7PQ/2?accountid=31491&sourcetype=Scholarly%20Journals

(S/f-i). Proquest.com. Recuperado el 14 de septiembre de 2024, de https://www.proquest.com/docview/2018732382/abstract/A0AF1FD47215498BPQ/8?accountid=31491&sourcetype=Scholarly%20Journals

(S/f-j). Rua.ua.es. Recuperado el 14 de septiembre de 2024, de https://rua.ua.es/dspace/handle/10045/127856