

**1. Desarrollo y certificación de un Sistema Web basado en Microservicios:** Este documento describe en detalle el proceso de desarrollo de un sistema web utilizando una arquitectura basada en microservicios, destacando cómo esta arquitectura facilita la escalabilidad, modularidad y flexibilidad del sistema. Además, aborda el proceso de certificación del sistema, asegurando que cumple con los estándares de calidad y funcionalidad requeridos. Se analiza la implementación de microservicios desde la fase de diseño hasta su despliegue, resaltando las ventajas en comparación con las arquitecturas monolíticas tradicionales.

**3. Estudio de Métricas y Patrones de Seguridad en Microservicios:** El estudio se enfoca en la seguridad dentro de arquitecturas de microservicios, examinando las métricas y patrones de seguridad más relevantes que se deben considerar al diseñar y desarrollar estos sistemas. Se destacan las estrategias para mitigar riesgos, tales como la segmentación de servicios y la protección de endpoints, así como la implementación de prácticas de seguridad desde el inicio del desarrollo para asegurar la integridad y confidencialidad de los datos en entornos distribuidos. También se analizan casos de estudio y mejores prácticas para la gestión segura de microservicios.

**4. Desarrollo de aplicaciones basadas en microservicios: tendencias y desafíos de investigación:** Este artículo ofrece una revisión exhaustiva de las tendencias actuales y los desafíos en el desarrollo de aplicaciones basadas en microservicios. Se identifican las áreas de investigación más relevantes, como la definición de la granularidad adecuada de los servicios, la modularización, la integración con la interfaz de usuario y la gestión de atributos de calidad como la escalabilidad y la calidad del servicio. El documento también resalta los desafíos específicos en las fases de desarrollo, pruebas y despliegue, incluyendo la orquestación de servicios, la seguridad y el monitoreo continuo, así como la necesidad de más investigación en la automatización de estas áreas.

**5. Marco de Trabajo para Seleccionar un Patrón Arquitectónico en el Desarrollo de Software:** Este documento presenta un framework diseñado para seleccionar el patrón arquitectónico más adecuado en el desarrollo de software. El enfoque principal es resolver problemas de desacoplamiento en la lógica empresarial, la interfaz de usuario, la navegación y la arquitectura de la información, todo lo cual afecta la calidad del producto final. El marco propuesto se basa en la caracterización de patrones de arquitectura de software, soportado por una revisión de literatura y consultas a expertos. Se identifican patrones con un alto impacto en el sector productivo y se proporciona una guía para seleccionar el patrón adecuado según el tipo de proyecto de desarrollo y sus necesidades específicas.

**6. Arquitectura Basada en Microservicios y DevOps para una Ingeniería de Software Continua:** Este documento propone una arquitectura basada en microservicios y DevOps para lograr una ingeniería de software continua. Los microservicios permiten desarrollar aplicaciones a partir de servicios independientes, escalables y colaborativos, mientras que DevOps facilita la entrega e integración continua en un entorno ágil y colaborativo. El documento incluye un caso de estudio

donde se aplica esta arquitectura en un proyecto académico, resaltando la implementación de tecnologías de vanguardia y la cultura colaborativa impulsada por DevOps. El enfoque está en la automatización y escalabilidad, cruciales para enfrentar los desafíos actuales en la industria del software

#### **7. Metodología para la Migración de Aplicaciones Monolíticas a Sistemas Basados en**

**Microservicios:** Este documento detalla una metodología para migrar aplicaciones monolíticas a arquitecturas basadas en microservicios. La propuesta aborda las limitaciones de los sistemas monolíticos, como la dificultad para escalar y mantener, y describe un proceso de migración estructurado para transformar una aplicación existente en un conjunto de microservicios. La metodología incluye la identificación de componentes, la creación de servicios independientes y la implementación de prácticas de DevOps para asegurar la integración y entrega continua, facilitando la adaptación a infraestructuras modernas como la nube.

#### **8.Estrategia de integración de diferentes plataformas web dinámicas mediante un framework**

**del lado del cliente:** Este documento propone una estrategia para integrar diversas plataformas web dinámicas en una sola aplicación utilizando un framework del lado del cliente, específicamente AngularJS. Se realiza un análisis detallado de diferentes tecnologías frontend y backend para seleccionar las herramientas con mejor rendimiento. El estudio incluye la implementación de un prototipo basado en microservicios con arquitectura RESTful para validar la eficiencia y comportamiento de la estrategia propuesta

#### **9.Análisis de rendimiento entre una arquitectura monolítica y una arquitectura de**

**microservicios basada en contenedores:** Este estudio compara el rendimiento de arquitecturas monolíticas y de microservicios, específicamente en entornos de contenedores. A través de pruebas de rendimiento y análisis, se demuestra cómo la arquitectura de microservicios puede superar a la monolítica en términos de escalabilidad y gestión de recursos, aunque también se identifican desafíos asociados a la complejidad y administración de microservicios

#### **10.Desarrollo ágil del nuevo sistema institucional basado en una arquitectura orientada a**

**microservicios:** Este documento aborda el desarrollo de un sistema institucional en una universidad utilizando una arquitectura de microservicios, destacando la agilidad obtenida en comparación con una arquitectura monolítica. Se implementa la metodología de puntos de casos de uso para medir el rendimiento y se concluye que la nueva arquitectura facilita un desarrollo más rápido y flexible, validado a través de su implementación en módulos clave del sistema

#### **11.Definición de una arquitectura para la transformación de software centralizado a software**

**basado en microservicios en el ámbito web :** Este documento trata sobre la definición de una arquitectura para transformar software monolítico, que es centralizado y difícil de escalar, en software basado en microservicios, más flexible y escalable. La propuesta se centra en aplicaciones web y aborda los desafíos de la transformación, tales como la identificación de servicios, la

comunicación entre ellos, y la implementación de estrategias para migrar de manera gradual sin interrumpir el funcionamiento del sistema. Aporta una guía práctica para organizaciones que buscan modernizar sus sistemas, mejorando su capacidad de adaptación y mantenimiento.

#### **12. Implementación de microservicios para la evaluación de programas basado en casos de prueba:**

Este documento describe la implementación de un sistema de evaluación de programas informáticos utilizando una arquitectura de microservicios. El objetivo principal es mejorar el rendimiento y escalabilidad del sistema actual de la Carrera de Informática de la Universidad Mayor de San Andrés, que se basa en un sistema monolítico. El nuevo sistema permite la evaluación automática y en tiempo real de programas enviados por estudiantes, manejando casos de prueba específicos y proporcionando resultados instantáneos. Se destaca el uso de metodologías ágiles como Scrum para el desarrollo del sistema y se analizan las ventajas y desventajas de la arquitectura de microservicios. Este documento es útil para entender cómo implementar y evaluar un sistema de microservicios en un entorno académico.

#### **13- Arquitectura de microservicios para mejorar la calidad de software en una entidad bancaria de lima:**

Este documento aborda la implementación de una arquitectura de microservicios en una entidad bancaria en Lima con el objetivo de mejorar la calidad del software. La investigación destaca cómo los microservicios permiten la escalabilidad y flexibilidad en el desarrollo de software, lo que resulta en una mayor eficiencia y un menor tiempo de respuesta en comparación con las arquitecturas monolíticas tradicionales. Se examinan los desafíos que enfrentan las organizaciones al adoptar microservicios, como la gestión de la complejidad y la seguridad, y se presentan soluciones prácticas para superarlos. Este trabajo es valioso para cualquier organización que busque mejorar la calidad de su software mediante la adopción de arquitecturas modernas y más eficientes.

**14-UNACIENCIA: Un acercamiento a los microservicios:** Este artículo ofrece una visión general de la arquitectura de microservicios, destacando su emergencia como una tendencia clave en el desarrollo de aplicaciones web. Los autores explican cómo los microservicios permiten descomponer una aplicación en servicios pequeños y autónomos que pueden ser desplegados, evolucionados y escalados de manera independiente, lo que mejora la flexibilidad y la escalabilidad del sistema. Además, se comparan los microservicios con otras arquitecturas, como SOA y las arquitecturas monolíticas, señalando las ventajas y desafíos de cada enfoque. Este documento es útil para desarrolladores y arquitectos de software interesados en entender los beneficios y retos de adoptar microservicios en sus proyectos

#### **15-Comparación del rendimiento de las arquitecturas monolíticas y microservicios en los sistemas web:**

Esta tesis compara el rendimiento de las arquitecturas monolíticas y de microservicios en sistemas web, analizando cómo cada una maneja aspectos clave como el consumo de memoria, uso de CPU y tiempo de respuesta bajo condiciones de estrés. Los resultados muestran que la arquitectura de microservicios supera a la monolítica en eficiencia y

fiabilidad, mejorando en un 10.63% y 27.87% respectivamente. La investigación sugiere que los microservicios ofrecen una solución superior en términos de escalabilidad y manejo de grandes volúmenes de tráfico, aunque también recomienda evaluar los costos y la viabilidad de su implementación antes de proceder. Este estudio es esencial para quienes buscan justificar o evaluar la transición a microservicios en sistemas web

**16. Aplicación web para el servicio de trámites académicos de la UNACH usando una arquitectura basada en microservicios:** Este documento presenta el desarrollo de una aplicación web diseñada para facilitar los trámites académicos en la Universidad Autónoma de Chiapas (UNACH). Utiliza una arquitectura basada en microservicios, lo que permite modularidad, escalabilidad y una mejor gestión de los diferentes servicios que componen el sistema académico. El enfoque en microservicios facilita la integración y el mantenimiento de nuevas funcionalidades sin afectar la operatividad general. Este trabajo es especialmente relevante para quienes buscan optimizar la gestión de servicios académicos mediante el uso de tecnologías modernas y arquitecturas escalables.

**17. Herramienta para el modelado y generación de código de Arquitecturas de Software basadas en Microservicios:** Este documento describe el desarrollo de una herramienta que permite el modelado y la generación automática de código para arquitecturas de software basadas en microservicios. La herramienta está diseñada para simplificar y agilizar el proceso de desarrollo de software mediante la automatización de tareas repetitivas y la estandarización de patrones de diseño. Es útil para desarrolladores de software y arquitectos que buscan mejorar la eficiencia en la creación de sistemas complejos y distribuidos.

**18. Arquitectura de software para el desarrollo de aplicaciones web orientada a micro-servicios en TecNM campus Escárcega:**

Este trabajo se enfoca en la creación de una arquitectura de software para el desarrollo de aplicaciones web en el Instituto Tecnológico de Escárcega, orientada al uso de microservicios. El objetivo es mejorar la productividad y reducir los tiempos de desarrollo para los docentes y estudiantes, permitiendo la reutilización de componentes y facilitando la transición del desarrollo monolítico a microservicios. Se utilizaron tecnologías como Angular y Spring Boot para construir una arquitectura modular y escalable que pueda ser aplicada en proyectos institucionales. El estudio también incluye una evaluación del impacto en la productividad y usabilidad, mostrando una aceptación positiva entre los usuarios.

**19. Reduciendo la brecha de seguridad del IoT con una arquitectura de Microservicios basada en TLS y OAuth2:** Este artículo aborda las brechas de seguridad en el Internet de las Cosas (IoT), especialmente en entornos de hogares inteligentes. Se enfoca en la integración de tecnologías como TLS y OAuth2 en una arquitectura de microservicios para mejorar la seguridad. La arquitectura propuesta busca resolver problemas de autenticación y confidencialidad en redes IoT, proporcionando flexibilidad y robustez, validada por implementaciones de referencia. Además, se

presenta una estructura de seguridad jerárquica para adaptarse a dispositivos con diferentes capacidades computacionales

**20. Diseño e implementación de un microservicio con Spring:** Este trabajo explora el desarrollo de aplicaciones web mediante microservicios usando tecnologías modernas como Spring Boot, Spring Data y MongoDB. Se implementa un microservicio que gestiona listas de reproducción de Spotify, filtrando las canciones escuchadas por el usuario. El proyecto analiza arquitecturas, patrones de diseño y frameworks, proporcionando una implementación eficiente de microservicios. También destaca las ventajas de usar Spring para simplificar el desarrollo y optimizar la funcionalidad de la aplicación

**21.Comparación del rendimiento de las arquitecturas monolíticas y microservicios en los sistemas web:** Este documento es una tesis que busca comparar el rendimiento de las arquitecturas monolíticas y de microservicios en sistemas web. A través de un estudio cuantitativo y preexperimental, se evaluaron 30 procesos similares en sistemas web para determinar cómo cada arquitectura afecta la eficiencia y la fiabilidad del sistema. Los resultados mostraron que la arquitectura de microservicios mejora la eficiencia en un 10.63% y la fiabilidad en un 27.87% en comparación con la arquitectura monolítica. Se concluye que los microservicios son más efectivos para sistemas web modernos y se recomienda investigar más sobre los costos y la viabilidad de implementar esta arquitectura. Nos puede ser útil para comprender cómo las arquitecturas de software influyen en el rendimiento de los sistemas web y para evaluar la transición de sistemas monolíticos a microservicios, especialmente en proyectos que requieren alta escalabilidad y fiabilidad

**22.Arquitecturas de microservicios para aplicaciones desplegadas en contenedores:** Este trabajo de fin de grado se centra en la definición y despliegue de arquitecturas de microservicios utilizando contenedores, con un enfoque en la nube, específicamente en Microsoft Azure. El documento explora las buenas prácticas y patrones arquitectónicos para el desarrollo de aplicaciones escalables y de alta disponibilidad. Se utilizan tecnologías como .NET Core para el desarrollo y Docker como motor de contenedores, con Kubernetes como orquestador. El proyecto aborda cómo gestionar la escalabilidad, la replicación rápida y la implementación continua en entornos cloud, destacando la importancia de los contenedores y orquestadores para maximizar el potencial de los microservicios. Sirve mucho al momento de diseñar, implementar y escalar aplicaciones modernas basadas en microservicios, aprovechando tecnologías cloud y contenedores. Es particularmente útil si estás considerando un despliegue en la nube y quieres entender cómo orquestadores como Kubernetes pueden facilitar la gestión de aplicaciones distribuidas.

**23. Estrategia para la implementación de modelos de aprendizaje automático (machine learning) usando arquitecturas basadas en microservicios en el contexto de la seguridad en ciudades inteligentes:** Este documento propone una estrategia para implementar modelos de aprendizaje automático (ML) con arquitecturas de microservicios en sistemas de seguridad de ciudades inteligentes. Dado el crecimiento de la dependencia tecnológica, surgen desafíos como la

protección ante ciberataques y la gestión de grandes volúmenes de datos. La solución implica el uso de microservicios para mejorar la escalabilidad y resiliencia de estos modelos. Se desarrollan dos microservicios: uno para el preprocesamiento de datos y otro para la predicción, utilizando herramientas como Kafka para la comunicación asíncrona y un balanceador de carga para optimizar el tráfico de solicitudes. Se validan estos enfoques con bases de datos de tráfico de red. brinda una visión clara sobre cómo implementar sistemas escalables y versátiles para manejar modelos de ML en ciudades inteligentes, con un enfoque en la seguridad y el manejo eficiente de datos en tiempo real.

#### **24.Desarrollo de un módulo contable para fortalecer la gestión financiera utilizando una arquitectura de software basada en microservicios en la empresa Inveservice FGL S.A.S.**

Este estudio presenta el desarrollo de un módulo contable basado en microservicios para optimizar la gestión financiera en Inveservice FGL S.A.S. La empresa enfrentaba ineficiencias debido a procesos manuales y un sistema contable monolítico. Se propone una solución automatizada, flexible y escalable que permita mejorar la precisión en los registros financieros y agilizar la generación de informes. Se utilizó la metodología Scrum para el desarrollo del software y se evaluó su impacto con el modelo de éxito de DeLone y McLean, obteniendo resultados positivos en cuanto a calidad del sistema, satisfacción del usuario y eficiencia. proporciona un enfoque práctico para automatizar y mejorar la eficiencia en la gestión financiera mediante microservicios, con una validación detallada sobre cómo esta arquitectura impacta positivamente en los procesos contables de la empresa.

#### **25. Análisis de patrones de resiliencia en una arquitectura basada en microservicios**

Este documento analiza patrones de resiliencia utilizados en arquitecturas de microservicios, enfocándose en cómo estos patrones ayudan a mantener el funcionamiento del sistema ante fallos. Se estudia un caso práctico en la empresa PedidosYa, que procesa 4 millones de órdenes semanales en América Latina, aplicando los patrones Timeout, Retry, Circuit Breaker y Bulkhead. Estos patrones mejoran la tolerancia a fallos al gestionar de forma eficiente errores de red, picos de tráfico y problemas de priorización. También se prueba la combinación de estos patrones para ofrecer una mayor robustez y evitar fallos en cascada

#### **26.Sistema de gestión de cotizaciones de servicios para empresas de telecomunicaciones**

**mediante microservicios:** Este trabajo aborda la problemática de gestionar cotizaciones en empresas de telecomunicaciones. A través de la implementación de un sistema basado en microservicios, se optimiza la gestión de recursos compartidos como torres de telefonía, infraestructura de fibra óptica, y servicios de roaming entre empresas. Se utiliza Kafka para la mensajería y tecnologías como Angular, Java, y Docker, lo que permite escalabilidad, alta disponibilidad, y tiempos de respuesta mejorados. El sistema se centra en facilitar la colaboración entre empresas a través de una interfaz ágil y flexible, mejorando notablemente el rendimiento y la experiencia del usuario.

#### **27.Metodología de conversión de aplicaciones monolíticas a microservicios desplegable en la**

**nube para pequeñas empresas:** Este documento presenta una metodología para la conversión de aplicaciones monolíticas a microservicios, enfocada en pequeñas empresas que buscan migrar sus

sistemas hacia soluciones más escalables y eficientes. La metodología abarca las etapas de análisis, diseño y despliegue, utilizando tecnologías cloud como Azure o AWS. Se enfoca en reducir la complejidad del monolito, dividir las funciones en microservicios autónomos y garantizar que la transición se realice sin afectar la operación. Se destacan las mejoras en el mantenimiento, la flexibilidad y el rendimiento del sistema tras la conversión.

**28.Desarrollo de software basado en microservicios: un caso de estudio para evaluar sus ventajas e inconvenientes:** Este estudio de caso explora el desarrollo de software utilizando una arquitectura de microservicios, evaluando tanto sus beneficios como sus limitaciones. Entre las ventajas destaca la escalabilidad, facilidad de mantenimiento, y capacidad de desarrollar y desplegar componentes de manera independiente. Sin embargo, se señalan desafíos en la gestión de dependencias, la complejidad de la implementación y la mayor carga administrativa. El trabajo se basa en un proyecto real, analizando las métricas de rendimiento antes y después de la migración, y concluye que la adopción de microservicios es especialmente adecuada para aplicaciones de gran escala.

**29. Arquitectura de microservicios para optimizar el acceso a datos del SiT Log Lab:** Este documento describe cómo implementar una arquitectura de microservicios en el Laboratorio Nacional en Sistemas de Transporte y Logística (SiT-LOG Lab). La motivación principal es mejorar la interoperabilidad y escalabilidad de los sistemas monolíticos actuales, facilitando el acceso y la comunicación entre proyectos. Se detallan los beneficios de los microservicios, como su capacidad para permitir el desarrollo y mantenimiento independiente de componentes. También se describen los frameworks y técnicas de migración utilizadas para este propósito, como Spring Boot y patrones de migración como "Strangler Fig".

**30. Método de automatización del despliegue continuo en la nube para la implementación de microservicios:** Este documento propone un método para automatizar el despliegue continuo de microservicios en la nube usando prácticas DevOps. Se describe cómo implementar un pipeline de despliegue continuo utilizando herramientas como Bitbucket y AWS EC2 para garantizar pruebas automáticas y despliegues sin intervención humana. El enfoque se centra en mejorar la agilidad del desarrollo y reducir errores mediante la integración continua y pruebas automatizadas. Se identifican desafíos en la adopción de microservicios, como la gestión de transacciones distribuidas y la complejidad de los sistemas distribuidos

**31. Adaptación de un patrón de software en seguridad a la arquitectura de un Microservicio:** Este documento aborda cómo adaptar patrones de software de seguridad dentro de arquitecturas de microservicios. Se enfoca en integrar mecanismos de seguridad como autenticación, autorización y protección de datos en un entorno de microservicios, donde la naturaleza distribuida presenta nuevos desafíos. Explica cómo aplicar estas técnicas en cada componente de la arquitectura sin afectar el rendimiento ni la escalabilidad del sistema.

**32. Despliegue de microservicios mediante técnicas de virtualización ligeras basadas en contenedores:** Este trabajo analiza el despliegue de aplicaciones y funciones de red virtualizadas (VNF) utilizando técnicas de contenedores ligeros. El enfoque principal es la arquitectura de microservicios y la virtualización basada en contenedores como Docker y Kubernetes. Proporciona casos de uso y escenarios prácticos donde se aplican estas tecnologías, enfocándose en la escalabilidad, agilidad y eficiencia en el uso de recursos.

**33. Monolitos vs. Microservicios en Arquitectura de Software:** Perspectivas para un Desarrollo Eficiente Este estudio compara las arquitecturas monolíticas y de microservicios, analizando sus ventajas, desventajas y los escenarios en los que cada una es más eficiente. Se examinan casos de empresas como Amazon y eBay que han migrado de una arquitectura a otra, explorando las dificultades y beneficios en términos de escalabilidad, mantenimiento y desarrollo.

**34. ¿Son los microservicios la mejor opción? Una evaluación de su eficacia y eficiencia frente a los monolitos:** Este documento evalúa la eficacia y eficiencia de los microservicios frente a los monolitos, considerando aspectos como la escalabilidad, modularidad y la complejidad. Aunque los microservicios ofrecen ventajas en términos de agilidad y escalabilidad, también presentan desafíos significativos, como la coordinación entre servicios y la sobrecarga operativa.

**35. Arquitectura de Software basada en Microservicios para desarrollo de aplicaciones web:** Este artículo examina cómo las aplicaciones web de la Asamblea Nacional del Ecuador (ANE) utilizan actualmente una arquitectura monolítica, lo que causa problemas de mantenimiento, escalabilidad y entregas. El estudio propone una transición hacia una arquitectura basada en microservicios para superar estos desafíos. La investigación se centra en identificar tecnologías y metodologías necesarias para este cambio, explorando el diseño de software modular, autonomía de servicios, y tolerancia a fallos. Se busca mejorar la agilidad, escalabilidad y flexibilidad del desarrollo de aplicaciones web.

**36. Pruebas de Software para Microservicios** Este artículo trata los retos y soluciones en la realización de pruebas de software para aplicaciones basadas en microservicios. Dado que estas aplicaciones están compuestas por múltiples servicios, es esencial realizar pruebas automatizadas y eficientes. Se abordan las pruebas unitarias, de integración, de componentes y de extremo a extremo, destacando el papel crucial de las pruebas de rendimiento y seguridad. Además, se evalúan herramientas para realizar pruebas end-to-end, siguiendo estándares internacionales como IEEE 14102-2010, con el fin de garantizar la calidad del sistema.

**37. Migración de aplicaciones monolíticas basadas en microservicios** Este trabajo de fin de grado se enfoca en analizar aplicaciones monolíticas para su migración a arquitecturas basadas en microservicios. Examina diferentes técnicas, algoritmos y metodologías que facilitan la transición, como el uso del KDM (Knowledge Discovery Metamodel) para el análisis y mapeo de sistemas. Además, se discuten las ventajas de la arquitectura de microservicios, como la flexibilidad, escalabilidad y mejor tolerancia a fallos, en comparación con la arquitectura monolítica.



### **38. Diseño de una arquitectura basada en contenedores para la integración y el despliegue**

**continuo CI-CD:** Este documento detalla el diseño de una arquitectura basada en contenedores, específicamente orientada hacia los procesos de integración continua (CI) y despliegue continuo (CD). Propone un enfoque que optimiza el ciclo de desarrollo y despliegue de software utilizando tecnologías de contenedores como Docker y Kubernetes. Aporta valor en la automatización del ciclo de vida del software, reduciendo tiempos de entrega y mejorando la eficiencia operativa de los equipos de desarrollo y operaciones.

**39. Autorización distribuida para aplicaciones basadas en microservicios:** Este trabajo investiga estrategias de autorización para aplicaciones distribuidas basadas en microservicios. Identifica que el uso de tokens firmados digitalmente es el mecanismo más adecuado para gestionar el control de acceso en estas arquitecturas. Aporta un análisis profundo de diferentes enfoques de autorización, como sesiones replicadas y centralizadas, comparando su compatibilidad con las características de microservicios, como la autonomía y la gestión descentralizada de datos.

### **40. Recomendación de prácticas de desarrollo de software en una arquitectura de microservicios que sean ambientalmente sostenibles. Caso de uso hipotético: sistema de compra de tiquetes en línea:**

Este documento se centra en la sostenibilidad ambiental en el desarrollo de software utilizando una arquitectura de microservicios. El autor propone una serie de prácticas y recomendaciones para minimizar el impacto ambiental durante el ciclo de desarrollo, como la optimización del uso de recursos y la eficiencia energética. Aporta una visión novedosa sobre cómo integrar la sostenibilidad en el desarrollo de software a gran escala, utilizando como ejemplo un sistema de compra de tiquetes en línea.