**Resúmenes de Artículos**

**Aprendiz: Carlos Andres Pantoja Jaramillo**

**Instructor: Jesus Ariel**

**Sena: Centro De La Industria La Empresa Y Los Servicios**

**Análisis Y Desarrollo De Software**

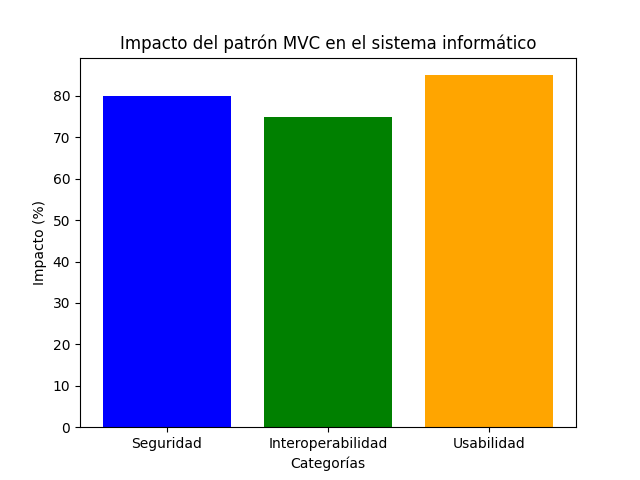
**2024**

**Perfil**

Técnico de multimedia con conocimientos de Adobe Photoshop, Adobe Illustrator, Filmora y Audicity. Soy aprendiz del Sena, cursando el tecnólogo de análisis y desarrollo de softwares, con conocimientos de Spring Boot, HTML, JavaScript y CSS.

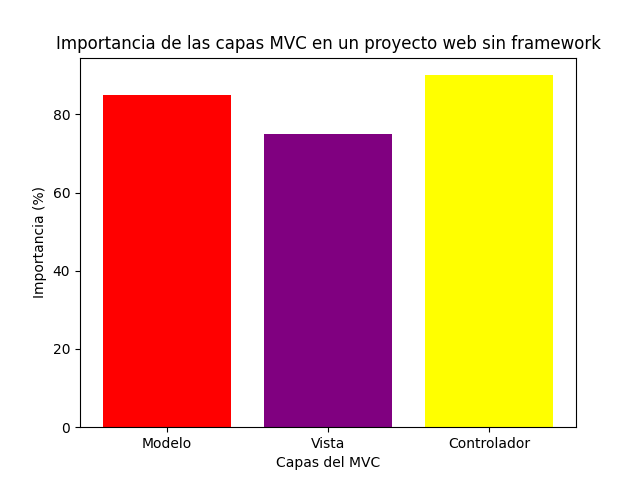
****

**Artículo 1: Impacto del patrón modelo vista controlador (MVC) en la seguridad, interoperabilidad y usabilidad de un sistema informático durante su ciclo debido.**

****

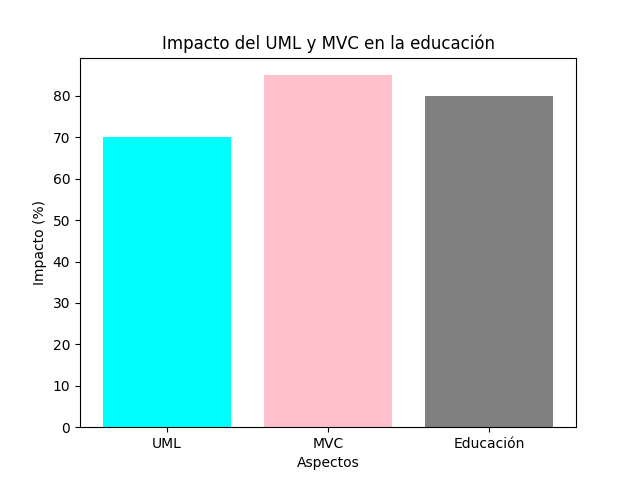
El artículo de Corazza (año) aborda la "programación abierta", una filosofía de diseño de software que enfatiza la flexibilidad y la modularidad en la construcción de sistemas. Se enfoca en los lenguajes de programación que apoyan esta filosofía, destacando cómo permiten la creación de aplicaciones más adaptables y extensibles. Menciona que el MVC promueve la separación de responsabilidades y la modularidad en el desarrollo de software.

**Artículo 2: Desarrollo de Aplicaciones Web con Java aplicando el patrón de diseño MVC Sin Utilizar un Framework.**

****

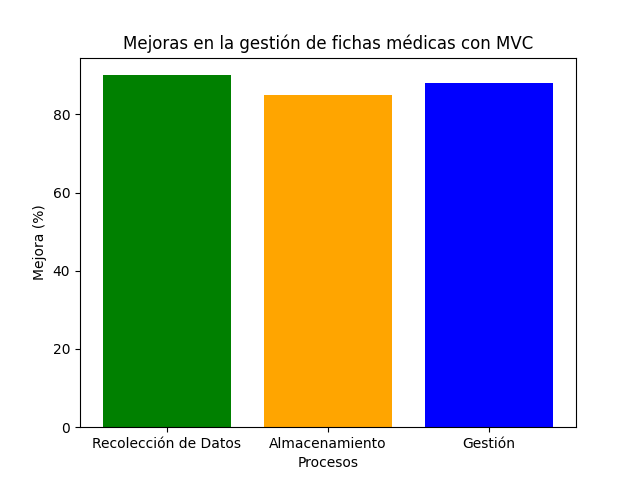
El artículo se centra en el patrón Modelo - Vista – Controlador (MUC), describiendo su estructura y funcionamiento. Se menciona cómo se implementó un proyecto web utilizando Eclipse, donde las clases específicas como, listarLibrosCuentos y listarOtrosLibros. El enfoque está en cómo se gestionan las interacciones a través de las capas MVC, destacando la importancia de la separación de responsabilidades en el desarrollo de aplicaciones web, lo que facilita en mantenimiento y la escalabilidad de software.

**Artículo 3: MODELO-VISTA-CONTROLADOR. LENGUAJE UML.**

****

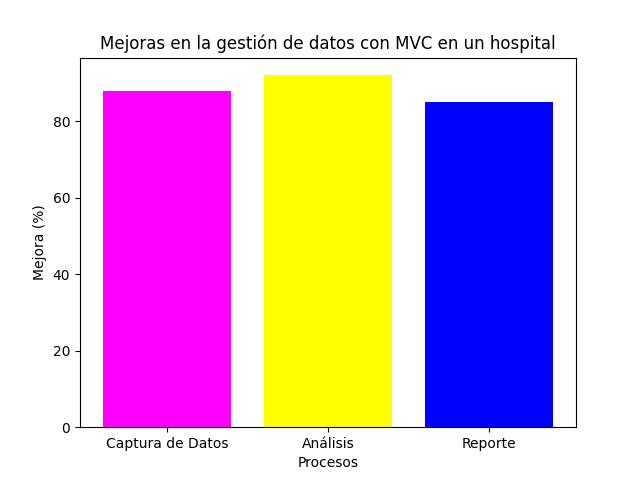
El artículo se divide en dos partes principales. La primera consiste en un estudio epistemológico Sobre el Modelo - Vista - Controlador y el lenguaje UML analizando sus antecedentes y las tendencias futuras en patrones de programación. La segunda parte elabora una unidad didacta que contextualiza los contenidos discutidos y su aplicación en un entorno educativo, específicamente en el contexto del ciclo superior del desarrollo de aplicaciones web.

**Artículo 4: SISTEMA WEB BASADO EN LA ARQUITECTURA MODELO VISTA CONTROLADOR (MVC) PARA LA GESTIÓN DE FICHAS MÉDICAS DE DOCENTES Y ESTUDIANTES DE LA UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ DE LA CIUDAD DE AMBATO.**

****

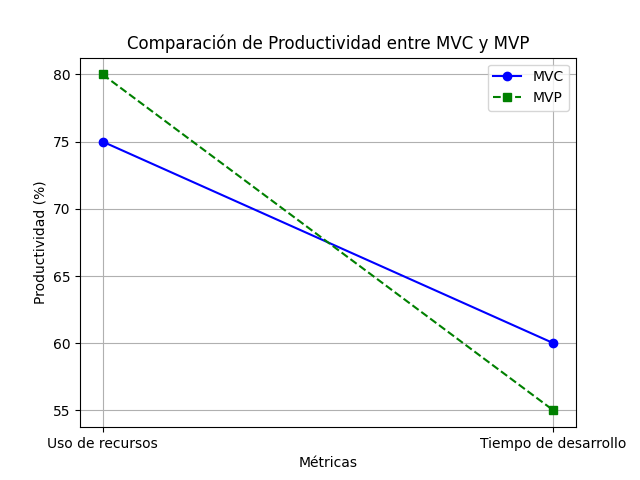
Es un proyecto, consiste en el desarrollo de un sistema web para la gestión de fichas médicas de docentes y estudiantes de la unidad educativa Gonzales Suárez, utilizando la arquitectura Modelo Vista Controlador (MVC). El objetivo es automatizar y mejorar los procesos de recolección, almacenamiento y gestión de la información médica de la institución. La arquitectura MVC se implementó utilizando Framework Laravel. El Laravel facilita la creación de un sistema web robusto y eficiente para la gestión de fichas médicas, al tiempo permite una integración con bases de datos y otras tecnologías.

**Artículo 5: Patrón MVC, un componente para la implementación de una Estrategia Informática para mejorar gestión de datos en el área de estadística: Caso de Estudio Hospital Maternidad Babahoyo.**

****

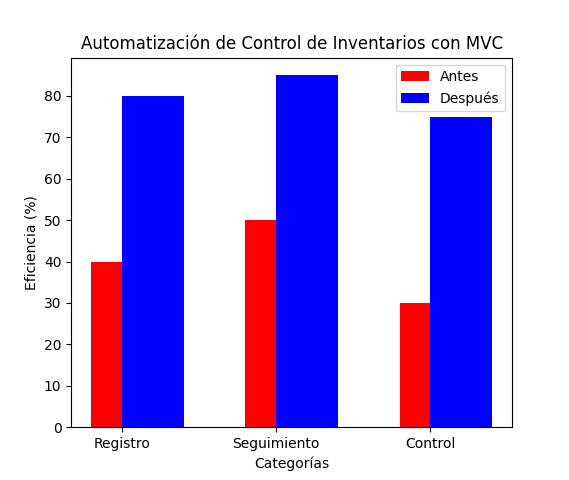
El artículo explora la implementación de patrón Modelo - Vista - Controlador como parte de una estrategia informática para optimizar la gestión de datos estadística del Hospital Maternidad Babahoyo. A través del desarrollo del sistema SIGEC se mejoraron los procesos de captura, análisis y reporte de datos, lo cual es crucial para la planificación y tomo de decisiones dentro del hospital. El artículo destaca las ventajas del patrón MVC Como la modularidad y facilidad de mantenimiento, al separar la lógica de negocio, la interfaz de usuario y el control de eventos en sistemas de información médica.

**Artículo 6: “ANÁLISIS COMPARATIVO DE LA PRODUCTIVIDAD ENTRE LOS PATRONES DE DISEÑO MODELO VISTA CONTROLADOR (MVC) Y MODELO VISTA PRESENTADOR (MVP) APLICADO AL DESARROLLO DEL SISTEMA NÓMINA DE EMPLEADOS Y ROL DE PAGOS DE LA “DISTRIBUIDORA SORIA C.A”**

****

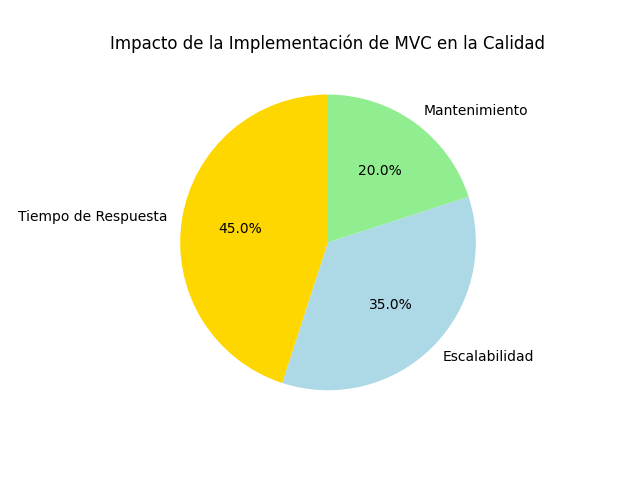
Centra a comparar la productividad entre dos patrones de diseño de software, MVC y MVP, utilizados en el desarrollo de un sistema de nómina y rol de pagos para la Distribuidora Soria C.A. A través de este análisis, se evalúan métricas clave como el uso de recursos del sistema y el tiempo de desarrollo, concluyendo cuál de los dos patrones ofrece mejores resultados en términos de eficiencia. En cuanto a MVC, el estudio resalta su capacidad para gestionar grandes cantidades de datos y su flexibilidad en la separación de la lógica de negocio y la interfaz de usuario.

**Artículo 7: SISTEMA WEB APLICANDO ARQUITECTURA MODELO VISTA CONTROLADOR (MVC) PARA EL CONTROL DE INVENTARIO Y PRODUCCIÓN EN LA PASTEURIZADORA J.S DEL CANTÓN SALCEDO.**

****

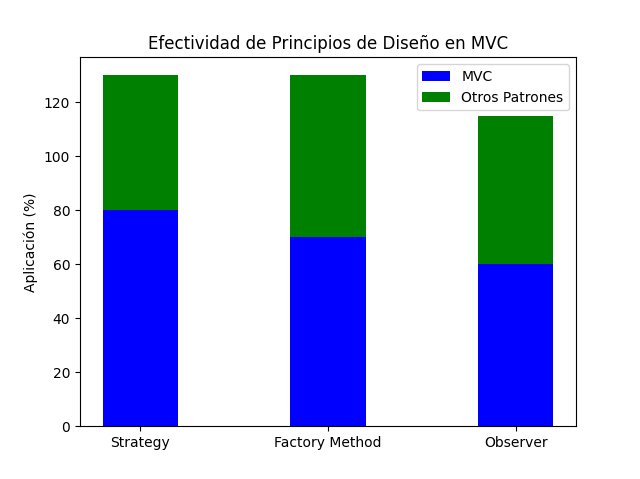
Desarrollo de un sistema web para la gestión de inventarios y producción en la pasteurizadora J.S, utilizando la arquitectura (MVC). Esta implementación buco resolver problemas de ineficiencia y errores que surgían en el manejo manual de los datos, logrando una automatización efectiva que facilita el registro, seguimiento y control de materias primas, productos terminados y ventas.

**Artículo 8: Impacto de implementaciones web del patrón MVC en los requisitos de calidad percibidos**

****

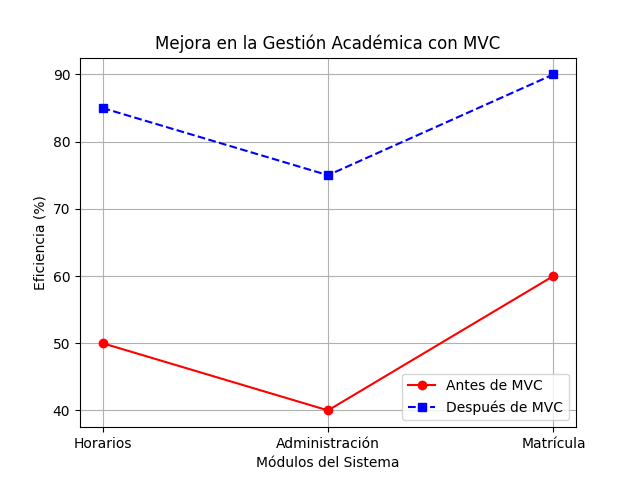
El artículo estudió dos variantes de la implementación del patrón Modelo - Vista - Controlador (MVC) en aplicaciones web y su impacto en los atributos de calidad percibidos, Como el tiempo de respuesta y la escalabilidad. El autor compara la implementación. Clásica de MVC, que utiliza un enfoque de llamada y respuesta, con una alternativa que emplea tuberías y filtros implementados a través de cortinas Python. La investigación concluye que, aunque prometedora, la implementación, con tubería, requiere más estudios para su aplicación práctica.

**Artículo 9: Principios y patrones de diseño de software en torno al patrón compuesto Modelo Vista Controlador para una arquitectura de aplicaciones interactivas**

****

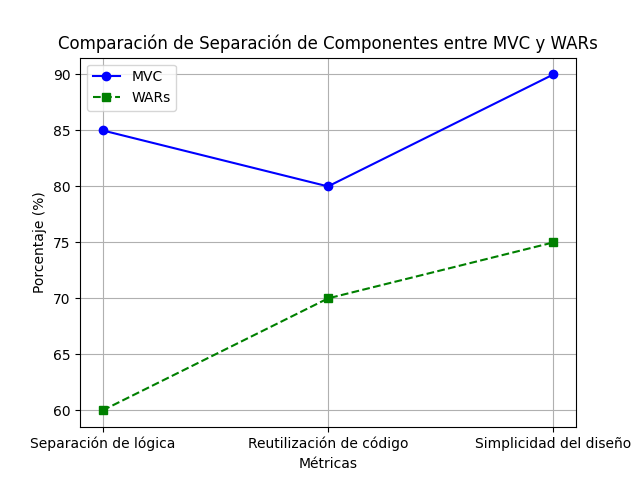
El artículo explora el patrón arquitectónico (MVC) puede ser implementado de manera efectiva en el desarrollo de aplicaciones interactivas. se centra en la importancia de entender y aplicar correctamente los principios y patrones de diseño que subyacen en MVC para lograr sistemas que Sean extensibles, mantenibles y escalables. El autor también analiza cómo estos patrones. Como el Strategy, Factory Method, y Observer, se combinan en el contexto MVC para abordar problemas comunes en la implementación de interfaces. de usuario y la lógica de negocio.

**Artículo 10: Sistema de Gestión Académica a través del desarrollo de Modelo-Vista-Controlador**

****

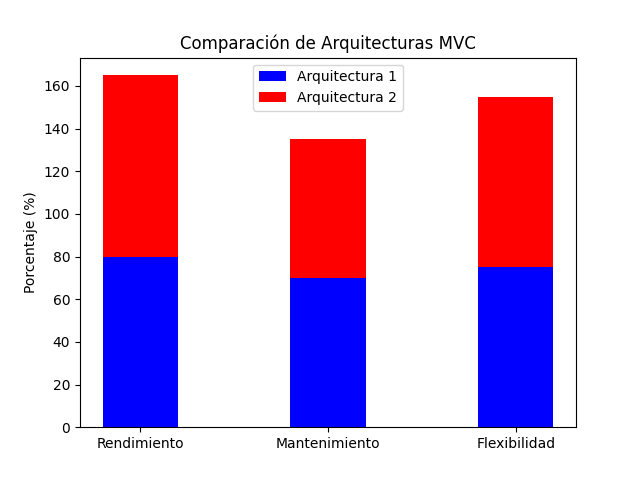
El artículo representa el desarrollo de un sistema de gestión académica (SIGAF) para la facultad de contaduría y administración, utilizando la arquitectura (MVC) implementada con framework Laravel. El sistema busca mejorar la gestión de horarios y administración académica mediante PHP, JavaScript, y Css3. Se detalla que estructura los módulos del sistema, separando la lógica de negocio, la presentación, y la persistencia de datos. Además, uso de metodologías ágiles como SERUM para el desarrollo de proyecto.

**Artículo 11: "MODELO VISTA CONTROLADOR (MVC) Y WEB ARCHIVES (WARS)"**

****

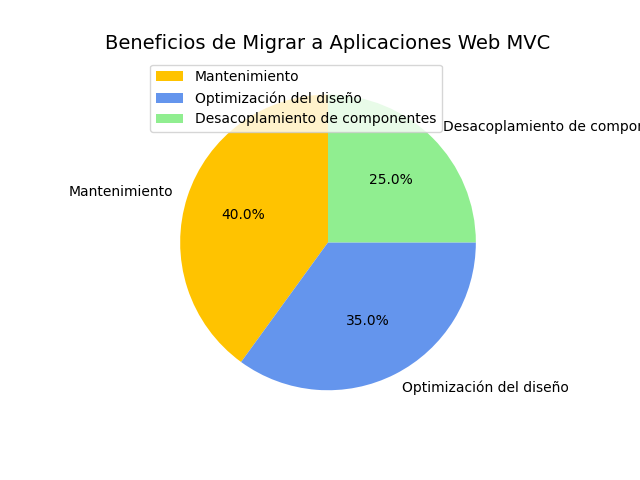
El artículo trata sobre el modelo vista controlador (MVC). se explica el concepto y sus ventajas, como la clara separación de componente, la simplificación del diseño y reutilización de código. Además, se menciona concepto de web-archives (Wars), que agrupar aplicaciones web en un formato comprimido que debe ser desconocido para su lectura, y se detalla su estructura, que incluye archivos HTML, JSP

**Artículo 12: Comparativa de arquitecturas MVC**

****

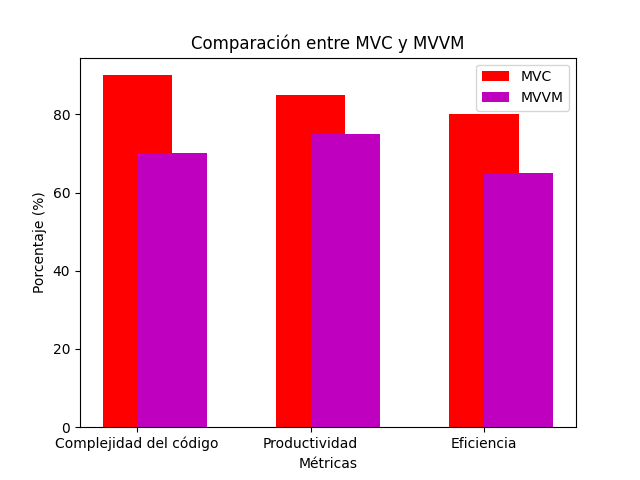
El artículo trata sobre la comparación de diferentes arquitecturas basadas en el patrón MVC, analizando tanto sus aspectos teóricos como prácticas o se discuten las métricas de rendimiento y se evalúan las implicaciones de realizar cambios en estas arquitecturas, como la introducción de evento a la descomposición de clases. Se enfatiza la importancia de entender cómo afecta el desarrollo y mantenimiento del software.

**Artículo 13: Procedimiento para la migración de aplicaciones transaccionales de escritorio a aplicaciones web MVC.**

****

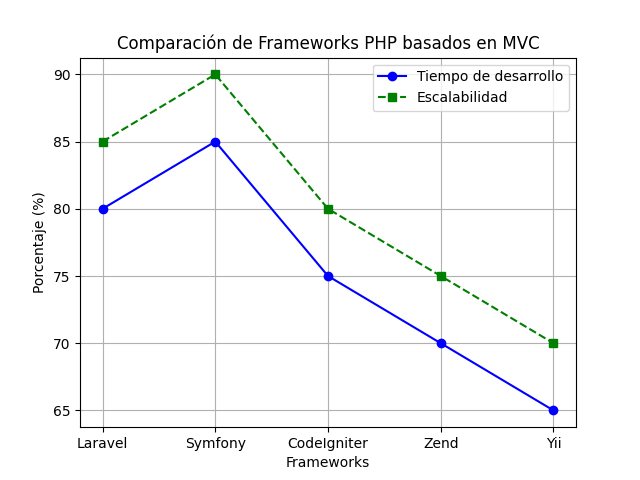
El artículo trata sobre la migración de aplicaciones de escritorio a aplicaciones web, centrándose en la implementación del patrón de diseño MVC. se discuten los principios teóricos detrás de MVC, que permite desacoplar los componentes del software para facilitar su mantenimiento y optimizando el diseño e implementación en su entorno contemporáneo.

**Artículo 14: Estudio comparativo de los patrones para interfaces de usuario MVVM y MVC aplicado el desarrollo del sitio de gestión de ventas para Vidrialum**

****

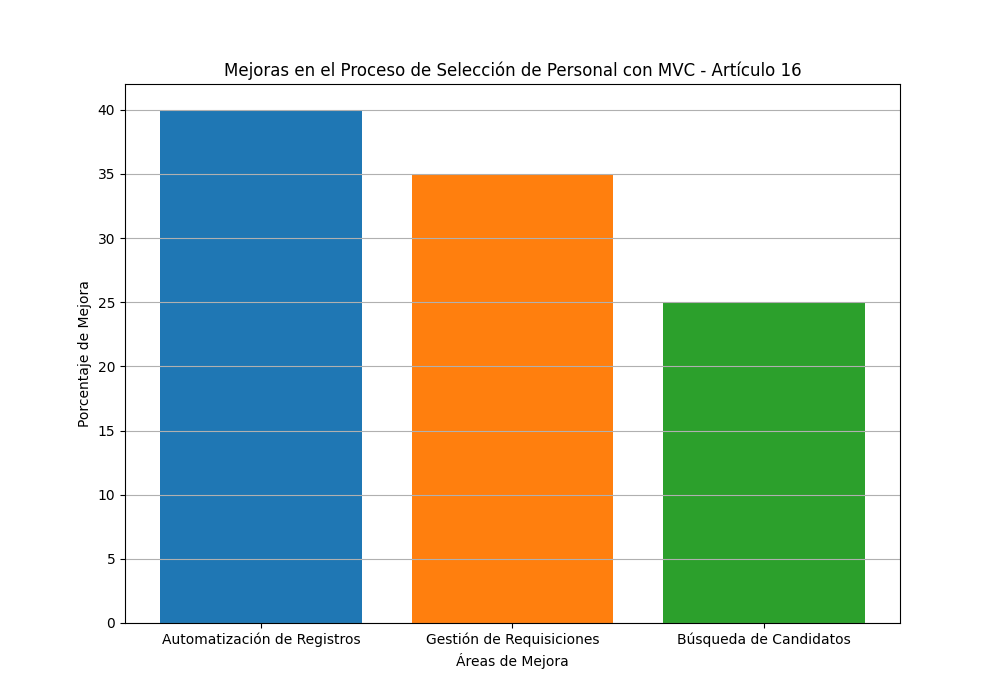
El artículo aborda la comparación entre los patrones de diseño MVC y MVVM (Modelo - Vista - Vista Modelo), evaluando su rendimiento y productividad en el desarrollo de aplicaciones. Se discuten aspectos como la complejidad del código, la disponibilidad de información y la calificación de cada patrón en función de su eficacia, se concluye que el MVC presenta mejores condiciones en varios indicadores, lo que hace más adecuado Para ciertos entornos de desarrollo.

**Artículo 15: "Frameworks PHP basados en la arquitectura Modelo-Vista-Controlador para desarrollo de aplicaciones web"**

****

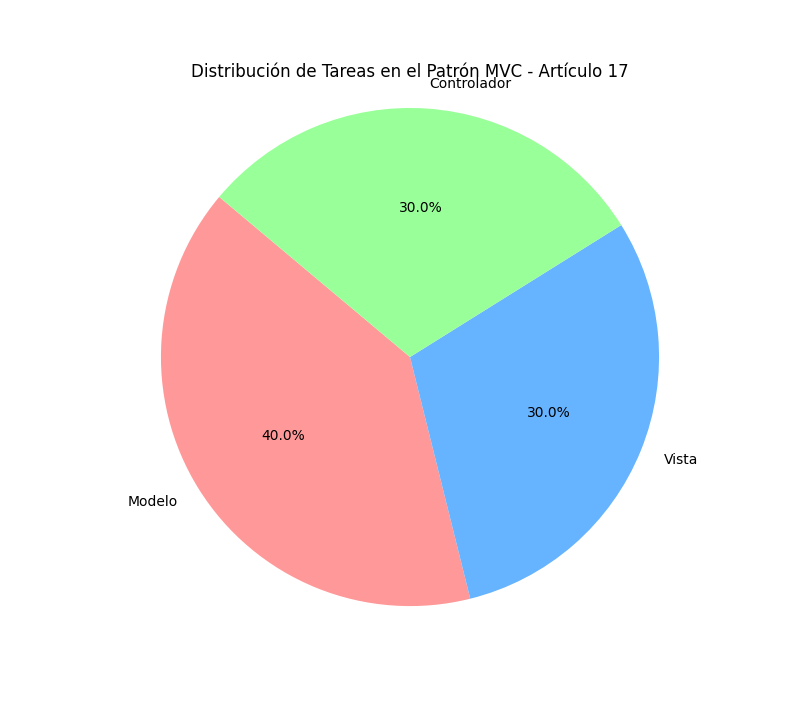
El artículo revisa y comparte varios framewortss PHP basadas en el patrón de diseño (MVC) incluyendo Laravel, Symfony, codelgniter, Zend, cakePHP y Yii. Estas frameworks Son herramientas poderosas para el desarrollo de aplicaciones web dinámicas, seguras y escalables, permitiendo a los desarrolladores evitar empezar desde cero al ofrecer. bibliotecas integradas y herramienta que optimizan el tiempo de desarrollo.

**Artículo 16: Implementación del patrón de MVC para el proceso de selección de personal**

****

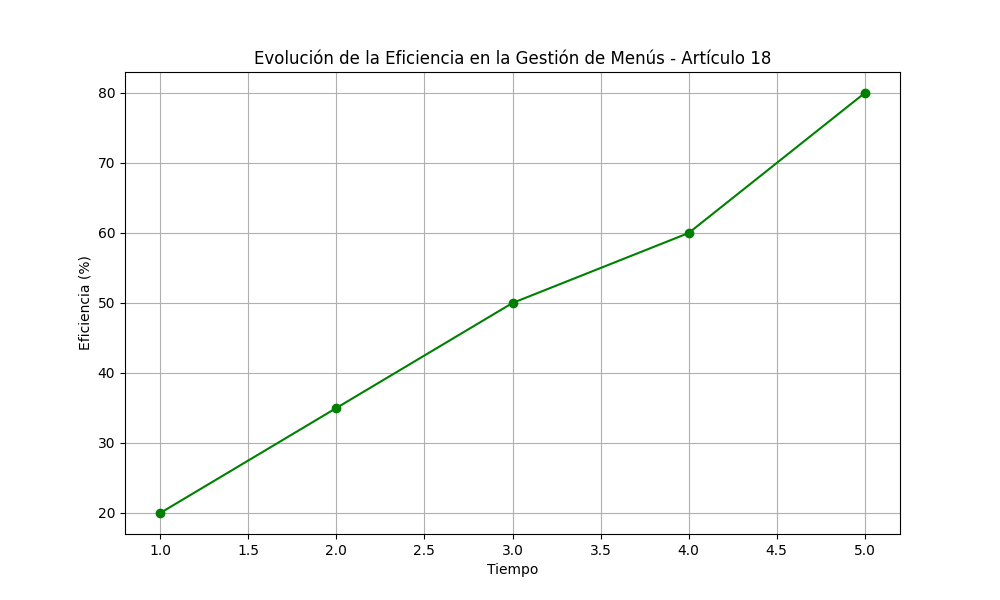
El trabajo presenta el desarrollo de un prototipo de software denominado "SisRePat "para mejorar el proceso de selección de personal en la empresa Persom S. A, utilizando el patrón de diseño (MVC). El proyecto aborda problemas relacionados con la falta de sistematización en los procesos. de reclutamiento y selección, lo que generaba ineficiencias del patrón MVC y pérdidas de clientes. Mediante la implementación del patrón MVC, se logró desarrollar una aplicación web que permite automatizar el registro de hojas de vida, la gestión de requisiciones de personal y la búsqueda de candidatos.

**Artículo 17: Patrón Modelo-Vista-Controlador.**

****

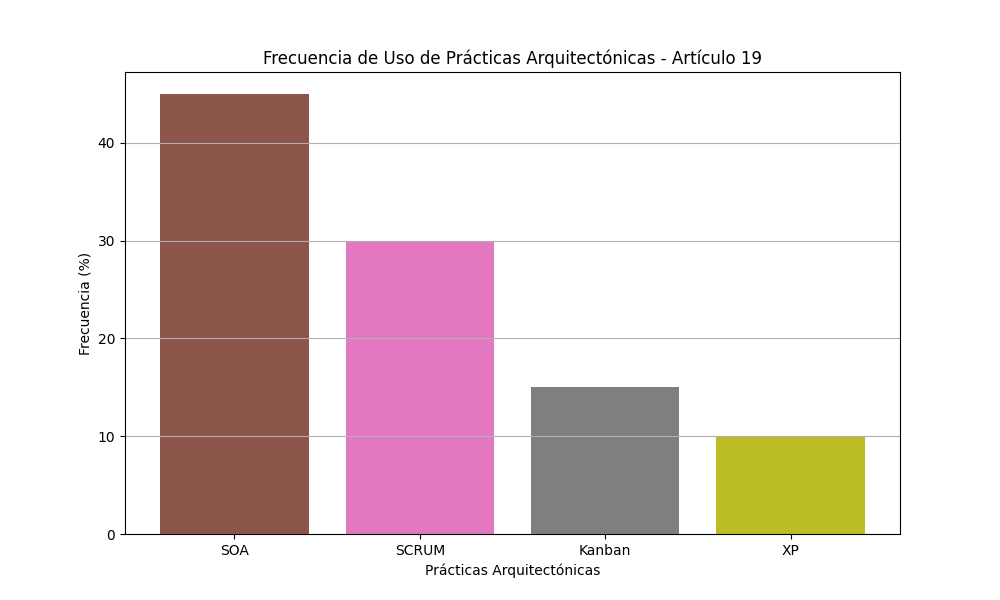
El artículo presenta el patrón (MVC) como una estrategia que busca optimizar el desarrollo de aplicaciones mediante la separación de sus componentes en tres partes: Modelo Vista Controlador. Esto permite que cada parte sea desarrollada y mantenida de forma independiente, facilitando la actualización del software y reduciendo el esfuerzo de programación. También se menciona la importancia de utilizar framework basados en MVC para mejorar la organización del trabajo y la especialización de los desarrolladores.

**Artículo 18: "Administrador de Menús Para Sistemas de Información Orientados a la Web"**

****

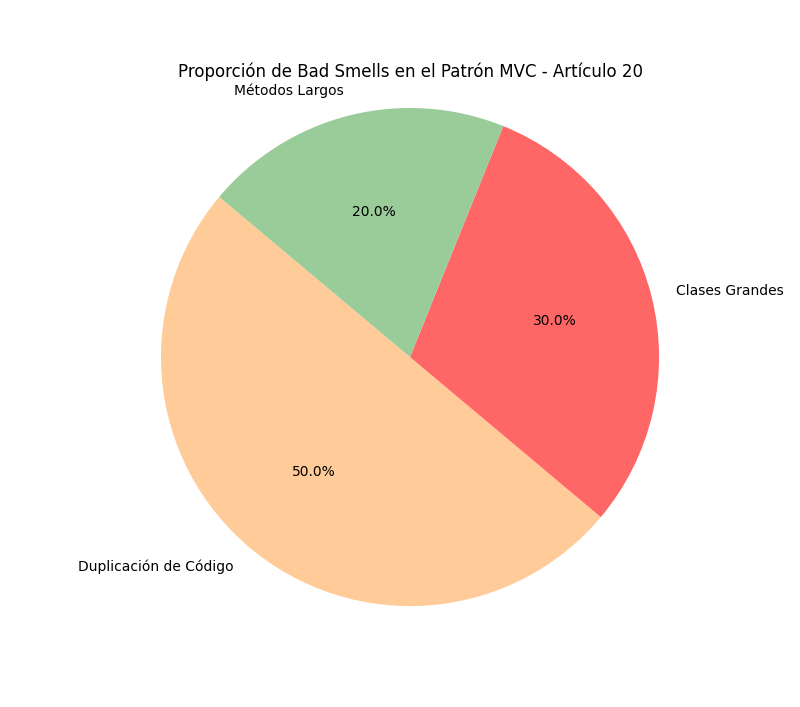
El artículo aborda la necesidad de un sistema. eficiente para gestionar los menús de aplicaciones web, proponiendo un administrador de menús jerárquico y multi idioma basado en el patrón de diseño MVC. Se discute cómo este patrón permite separar las diferentes capas de una aplicación, facilitando el desarrollo y mantenimiento de sistemas de información al reducir el acoplamiento entre el código de presentación y la lógica de negocio. Esto resulta en una mejor experiencia para el usuario y un sistema más robusto.

**Artículo 19: Mapeo Sistemático Sobre las Arquitecturas de Software en el Desarrollo Ágil**

****

El artículo examina el uso de arquitecturas de software, resaltando la omisión frecuente de estas actividades debido a la priorización de resultados inmediatos. A través de un mapeo sistemático, se identifican patrones y prácticas, destacando la combinación de arquitecturas SOA con metodologías ágiles como SCRUM. Se concluye que una buena comunicación y una arquitectura sólida son esenciales para el éxito en proyecto ágiles.

**Artículo 20: "Caracterización y Detección Automática de Bad Smells MVC"**

****

El artículo se centra en los "bad smells", que son indicadores de problema en el diseño y desarrollo de software que pueden llevar a una acumulación de deuda técnica. Aunque existe investigación sobre la caracterización y detección de estos problemas, pocos estudios se enfocan en los bad smells arquitectónicos, especialmente en el contexto del patrón MUC. Este trabajo busca llenar este vacío mediante la caracterización detallada de estos bad smells y la propuesta de un método para su detección automática a través de análisis estático.

**Bibliografia.**

 Corazza, G. (n.d.). Impacto del patrón modelo vista controlador (MVC) en la seguridad, interoperabilidad y usabilidad de un sistema informático durante su ciclo debido. Recuperado de <https://revistas.ug.edu.ec/index.php/easi/article/view/821>

**Artículo 1:** González, A., & Martínez, C. (n.d.). Desarrollo de Aplicaciones Web con Java aplicando el patrón de diseño MVC Sin Utilizar un Framework. Recuperado de <https://tecnocultura.org/index.php/Tecnocultura/article/view/260>

**Artículo 2:** Rodríguez, J. (2020). MODELO-VISTA-CONTROLADOR. LENGUAJE UML. Universidad de Jaén. Recuperado de <https://crea.ujaen.es/handle/10953.1/11437>

**Artículo 3:** Pérez, L., & López, M. (2021). SISTEMA WEB BASADO EN LA ARQUITECTURA MODELO VISTA CONTROLADOR (MVC) PARA LA GESTIÓN DE FICHAS MÉDICAS DE DOCENTES Y ESTUDIANTES DE LA UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ DE LA CIUDAD DE AMBATO. Recuperado de <https://repositorio.uta.edu.ec/handle/123456789/41236>

**Artículo 5:** Castro, F. (2021). Patrón MVC, un componente para la implementación de una Estrategia Informática para mejorar gestión de datos en el área de estadística: Caso de Estudio Hospital Maternidad Babahoyo. Recuperado de <https://www.redalyc.org/pdf/5646/564677242010.pdf>

**Artículo 6:** Salazar, J., & Miranda, R. (2020). ANÁLISIS COMPARATIVO DE LA PRODUCTIVIDAD ENTRE LOS PATRONES DE DISEÑO MODELO VISTA CONTROLADOR (MVC) Y MODELO VISTA PRESENTADOR (MVP) APLICADO AL DESARROLLO DEL SISTEMA NÓMINA DE EMPLEADOS Y ROL DE PAGOS DE LA “DISTRIBUIDORA SORIA C.A”. Recuperado de <http://dspace.espoch.edu.ec/handle/123456789/3583>

**Artículo 7:** Fernández, A., & Rodríguez, P. (2022). SISTEMA WEB APLICANDO ARQUITECTURA MODELO VISTA CONTROLADOR (MVC) PARA EL CONTROL DE INVENTARIO Y PRODUCCIÓN EN LA PASTEURIZADORA J.S DEL CANTÓN SALCEDO. Recuperado de <https://repositorio.uta.edu.ec:8443/handle/123456789/36643>

**Artículo 8:** Martínez, C. (2021). Impacto de implementaciones web del patrón MVC en los requisitos de calidad percibidos. Recuperado de <https://laccei.org/LACCEI2010-Peru/Copyright%20and%20Payment%20missing/ED054_Vazquez.pdf>

**Artículo 9:** Gómez, A. (2020). Principios y patrones de diseño de software en torno al patrón compuesto Modelo Vista Controlador para una arquitectura de aplicaciones interactivas. Recuperado de <https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15>

**Artículo 10:** López, M., & Ramírez, J. (2021). Sistema de Gestión Académica a través del desarrollo de Modelo-Vista-Controlador. Recuperado de <https://www.proquest.com/openview/b7f8919dbb75fa3e02bd941a78a6e890/1?pq-origsite=gscholar&cbl=1006393>

**Artículo 11:** Vallejo, E., & Sola, J. (2019). Modelo vista controlador (MVC) y web archives (WARS). Universidad Autónoma del Estado de México. <http://ri.uaemex.mx/bitstream/handle/20.500.11799/108712/secme-35486_1.pdf?sequence=1>

**Artículo 12:** Guzmán López, B. (2018). Comparativa de arquitecturas MVC (Tesis de maestría). Universidad Politécnica de Madrid. <https://oa.upm.es/54237/1/TESIS_MASTER_BORJA_GUZMAN_LOPEZ.pdf>

**Artículo 13:** Angarita, M. (2017). Procedimiento para la migración de aplicaciones transaccionales de escritorio a aplicaciones web MVC (Trabajo de grado). Universidad de Pamplona. <http://repositoriodspace.unipamplona.edu.co/jspui/bitstream/20.500.12744/5437/1/Angarita_2017_TG.pdf>

**Artículo 14:** Zambrano, J. (2019). Estudio comparativo de los patrones para interfaces de usuario MVVM y MVC aplicado al desarrollo del sitio de gestión de ventas para Vidrialum (Tesis de maestría). Escuela Superior Politécnica de Chimborazo. <http://dspace.espoch.edu.ec/bitstream/123456789/3559/1/18T00567.pdf>

**Artículo 15:** Jiménez, J. A., & Mendoza, C. (2020). Frameworks PHP basados en la arquitectura Modelo-Vista-Controlador para desarrollo de aplicaciones web. Revista Científica de Tecnologías de la Información. <http://scielo.senescyt.gob.ec/pdf/rctu/v10n1/1390-7697-rctu-10-01-00070.pdf>

**Artículo 16:** Guzmán, G., & Tovar, N. (2021). Implementación del patrón de MVC para el proceso de selección de personal (Trabajo de grado). Universidad Libre. <https://repository.unilibre.edu.co/bitstream/handle/10901/8881/Trabajo%20de%20Grado%20Gelen%20Guzman%20y%20Natalia%20Tovar.pdf?sequence=1&isAllowed=y>

**Artículo 17:** Pérez, R. (2022). Patrón Modelo-Vista-Controlador. Revista Telemática. <https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15/10>

**Artículo 18:** Tancara, J. (2009). Administrador de menús para sistemas de información orientados a la web. Ingeniería y Sistemas. <https://ingsisucblp.wordpress.com/wp-content/uploads/2011/01/jtancara-papers_jta_2_2009.pdf>

**Artículo 19:** González, L., & Martínez, R. (2020). Mapeo sistemático sobre las arquitecturas de software en el desarrollo ágil. Revista de Tecnología y Gestión. <https://www.proquest.com/openview/1b52d4c1d201b1226f1436f2c2b26424/1?pq-origsite=gscholar&cbl=2026366&diss=y>

**Artículo 20:** Vázquez, S. (2018). Caracterización y detección automática de bad smells MVC (Tesis de doctorado). ResearchGate. <https://www.researchgate.net/profile/Sodel-Vazquez-Reyes/publication/326152679_Caracterizacion_y_Deteccion_Automatica_de_Bad_Smells_MVC/links/5cfaba11a6fdccd1308a6199/Caracterizacion-y-Deteccion-Automatica-de-Bad-Smells-MVC.pdf>