

Resúmenes de Artículos

Carlos Andres Pantoja Jaramillo

Jesus Ariel González Boni

Sena: Centro De La Industria La
Empresa Y Los Servicios

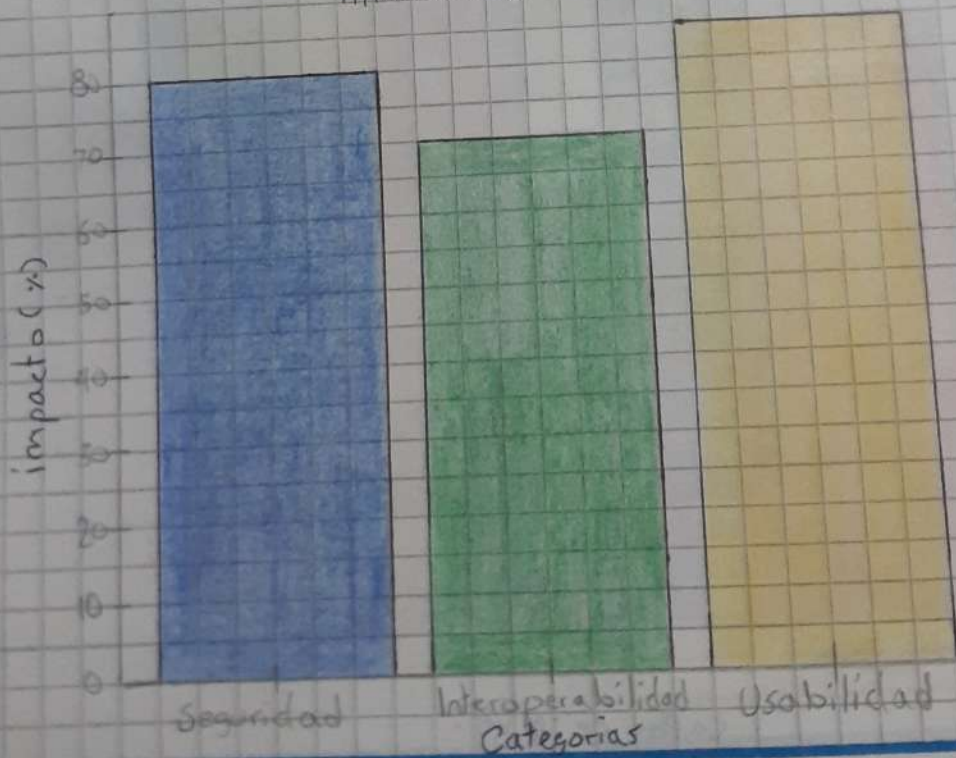
Análisis Y Desarrollo De Software

2024

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

El artículo de Giovanni Corazza aborda la "programación abierta", una filosofía de diseño de software que enfatiza la flexibilidad y la modularidad en la construcción de sistema. 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 responsabilidad y la modularidad en el desarrollo de Software.

Impacto del patrón MVC en el sistema informático



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 (MVC), 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 el mantenimiento y la escalabilidad de software.

Importancia de las capas MVC en un Proyecto web sin framework



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 didáctica que contextualiza los contenidos discutidos y su aplicación en un entorno educativo, específicamente en el contexto del ciclo superior del desarrollo de aplicaciones web.

Impacto del UML y MVC en la Educación



Artículo 4: Sistema web basado en la arquitectura Mode 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 González 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 implementa 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 otra tecnologías.



Mejoras en la
gestión de fichas
médicas con MVC

Artículo 5: Patrón MVC, un componente para la implementación de una estrategia informática para mejorar gestión de dato en el área de estadística: Caso de estudio Hospital Maternidad Babahoyo.

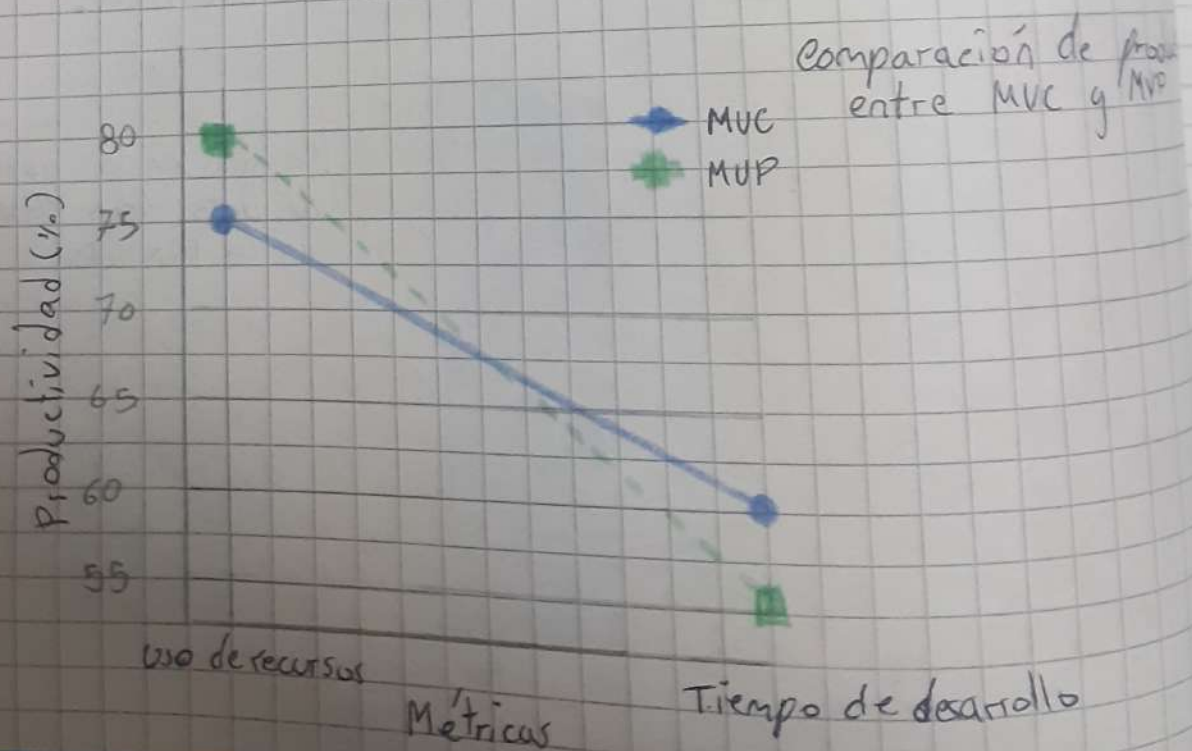
El artículo explora la implementación de patrón Modelo-Vista-Controlador como parte de una estrategia informática para optimizar la gestión de datos en el área de estadística del Hospital Maternidad Babahoyo. A través del desarrollo del sistema SIGEC se mejoraron los procesos de captura, análisis y reporte de datos, lo cual es crucial para la planificación y toma 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:



Mejoras en la gestión de datos con MVC en un hospital

Artículo 6°. Análisis comparativo de la productividad de los patrones de diseño MVC y modelo vista (MVP) aplicados al desarrollo del sistema Nómina de empleados y del sistema de la distribuidora Soria C.A.

Centra a comparar la productividad entre los patrones de diseño de software, MVC y MVP, utilizados en el desarrollo de un sistema de nómina y el rol de parámetro de la distribuidora Soria C.A. A través de este análisis se evaluarán métricas clave como el uso de recursos, el tiempo de desarrollo, concluyendo cuál de los dos ofrece mejores resultados en términos de eficiencia. En comparación con MVC la 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 (MVC) para el control del inventario y producción en la pasteurizadora J.S del cantón salcedo.

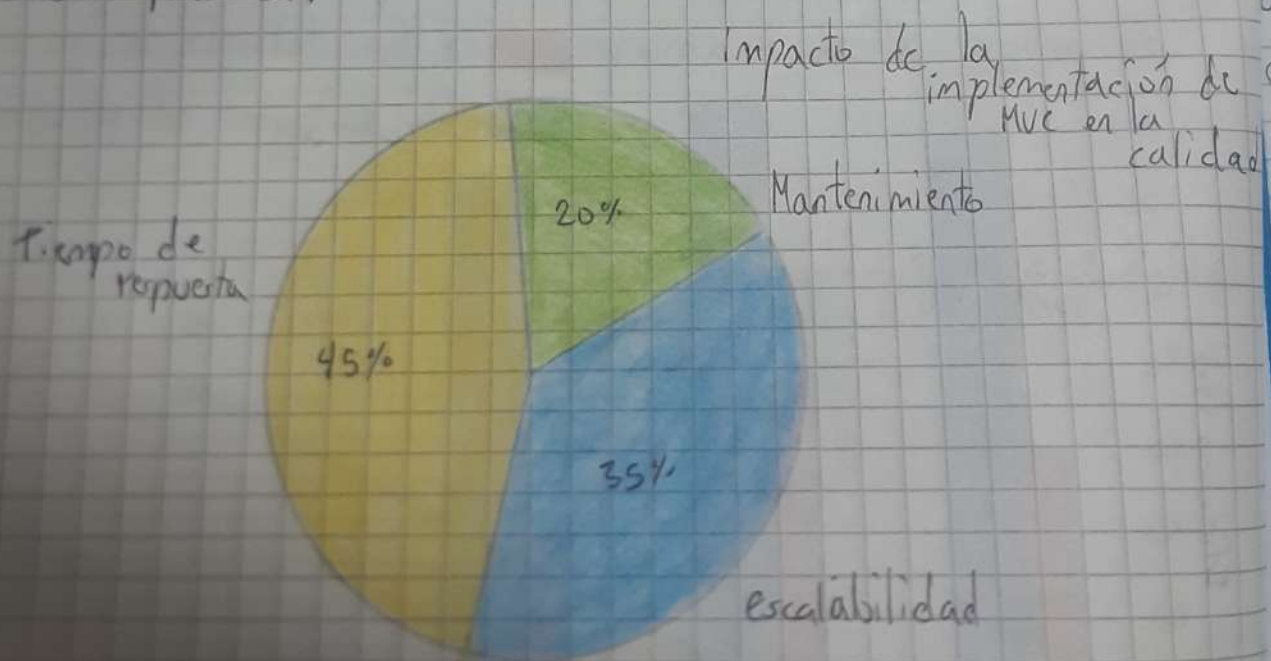
Desarrollo de un sistema web para la gestión de inventarios y producción en la pasteurizadora J.S, utilizando la arquitectura (MVC). Esta implementación busca resolver problemas de ineficiencia y errores que surgían en el manejo manual de los datos, logrando una automatización efectiva que facilita el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

Automatización de control de inventarios con MVC



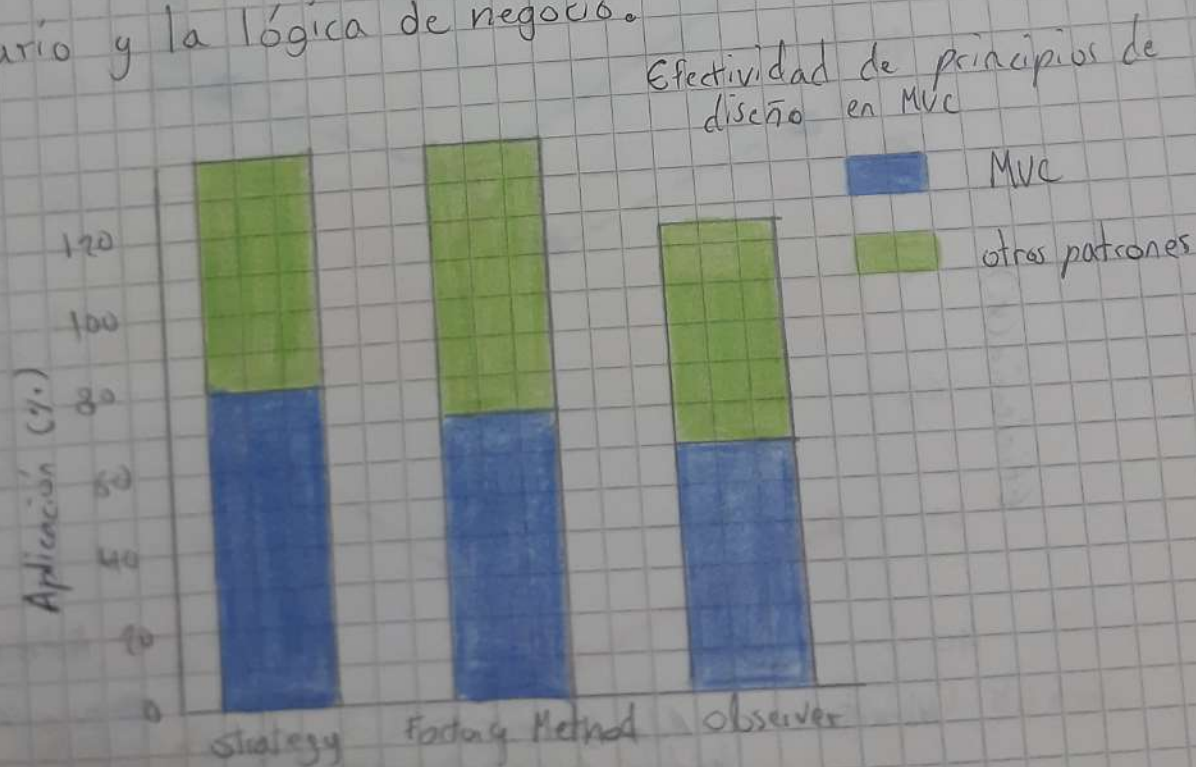
Artículo 8: Impacto de implementaciones web del patrón MVC en los requisitos de calidad percibidos.

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, implementadas a través de `coroutines` en 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.

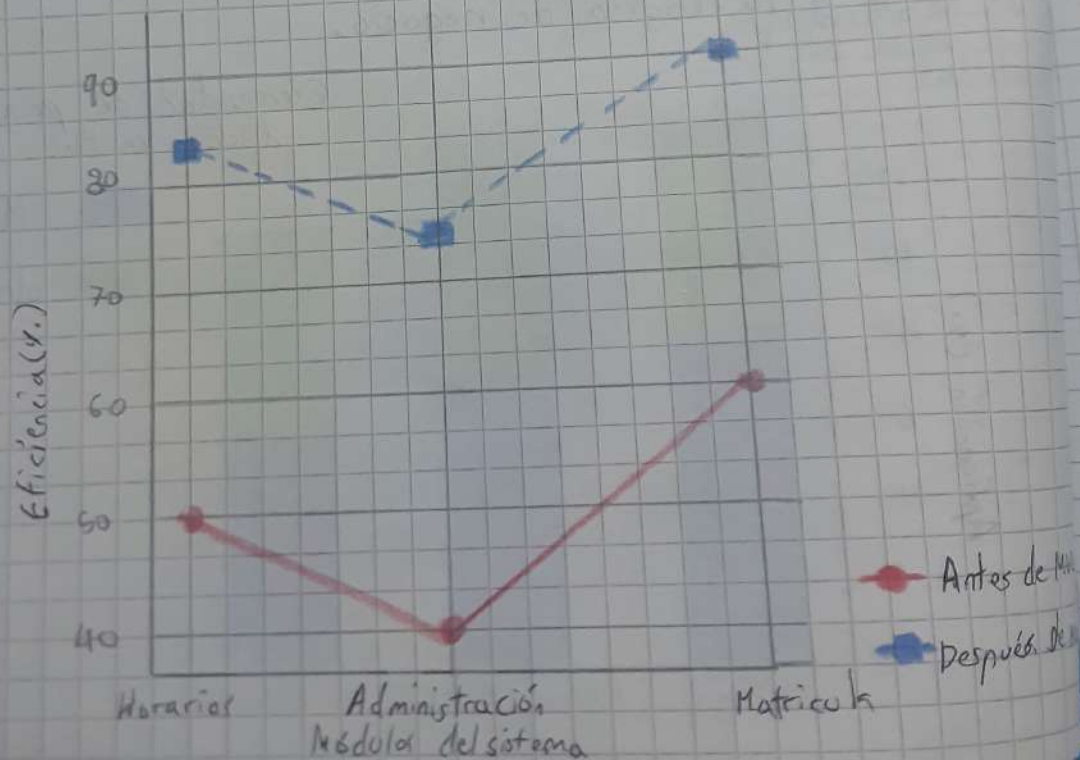
El artículo explora el patrón arquitectónico (MVC) puede ser implementado de manera efectiva en el desarrollo de aplicaciones interactivas. Se centra en la importancia de entender y aplicar correctamente los principios y patrones de diseño que subyacen en MVC para lograr sistemas que sean extensibles, 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 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 SCRUM para el desarrollo de proyecto.

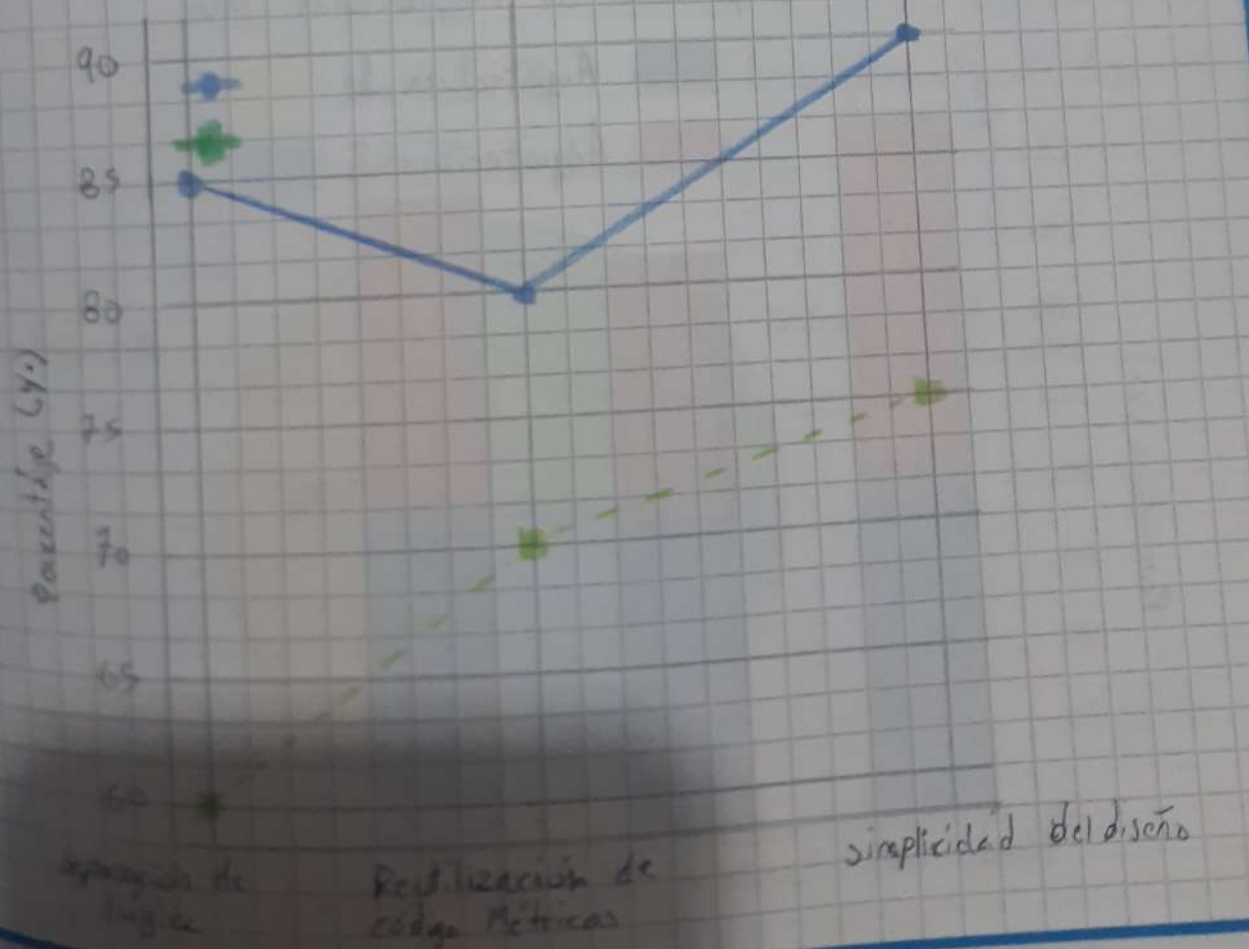
Mejora en la Gestión Académica con MVC



Artículo 11: Modelo Vista Controlador (MVC) y web Archives (WARs).

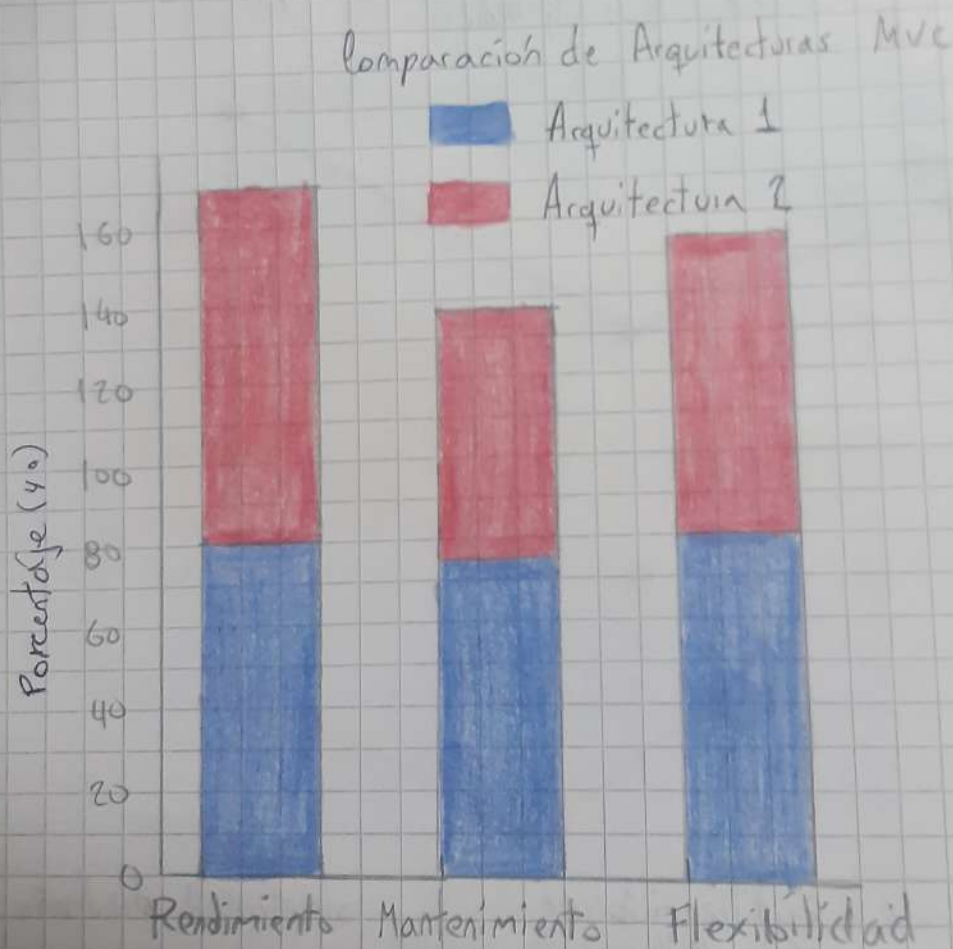
El artículo trata sobre el modelo vista controlador (MVC). Se explica el concepto y sus ventajas, como la clara separación de componente, la simplificación del diseño y reutilización de código. Además, se menciona concepto de web-archives (Wars), que agrupan aplicaciones web en un formato comprimido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

Comparación de separación de componentes entre MVC y WARs



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 prácticos o se discuten las métricas de rendimiento y se evalúan las implicaciones de realizar cambios en estas arquitecturas, como la introducción de evento o 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.

Beneficios de Migrar a Aplicaciones web MVC



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

El artículo aborda la comparación entre los patrones de diseño MVC y MVVM (Modelo - Vista - VistaModelo), 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.

Comparación entre MVC y MVVM



ara
del

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

es

ando

iones.

la

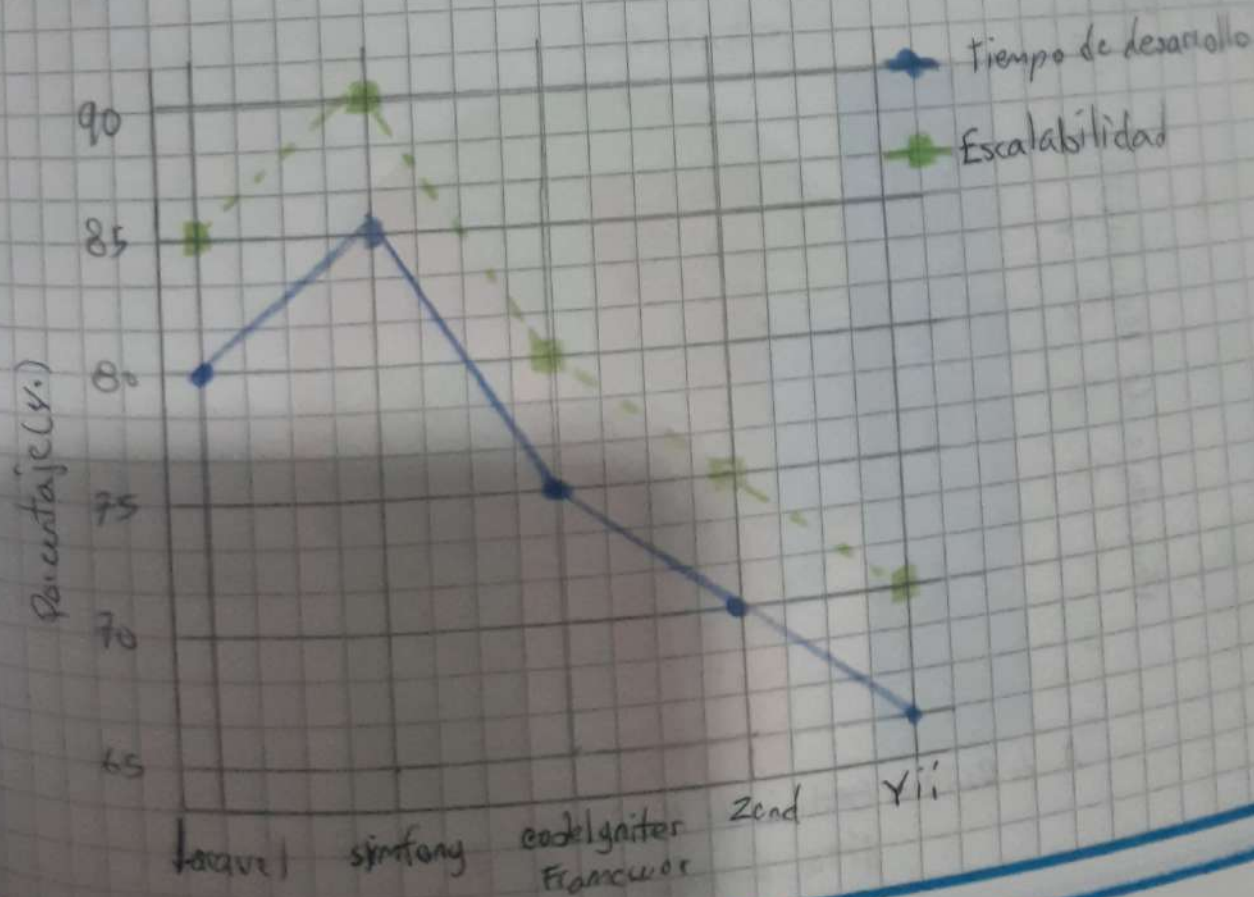
a

ic

ce

El artículo revisa y comparte varios frameworks PHP basados en el patrón de diseño (MVC) incluyendo Laravel, Symfony, CodeIgniter, Zend, CakePHP y Yii. Estos 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.

Comparación de framework PHP basados
en MVC



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.

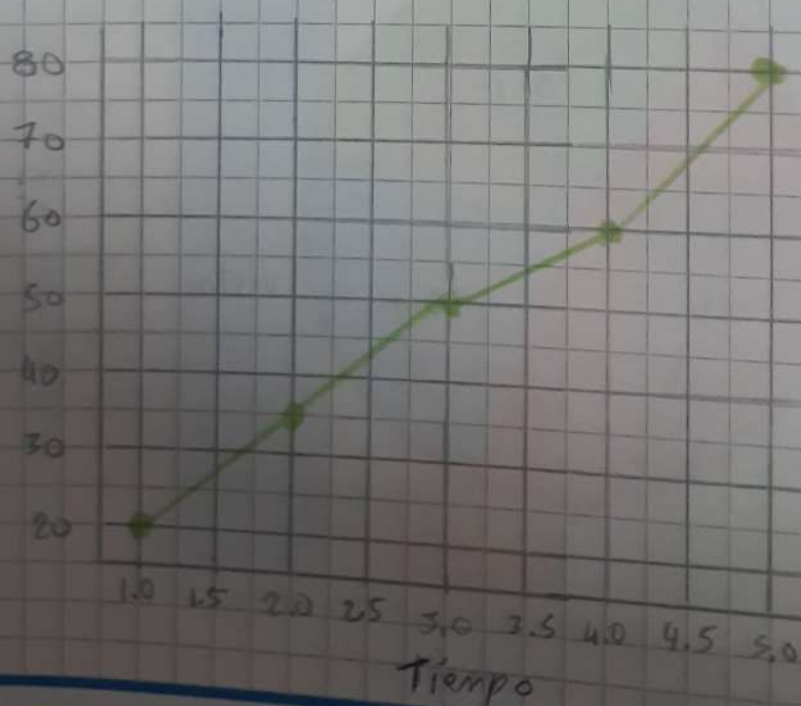
El artículo presenta el patrón (MVC) como una estrategia que busca optimizar el desarrollo de aplicaciones mediante la separación de sus componentes en tres partes: Modelo, vista y controlador. Esto permite que cada parte sea desarrollada y mantenida de forma independiente, facilitando la actualización 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.

Distribución de tareas en el patrón MVC



Artículo 18: Administrador de menús para sistemas de información orientados a la web.

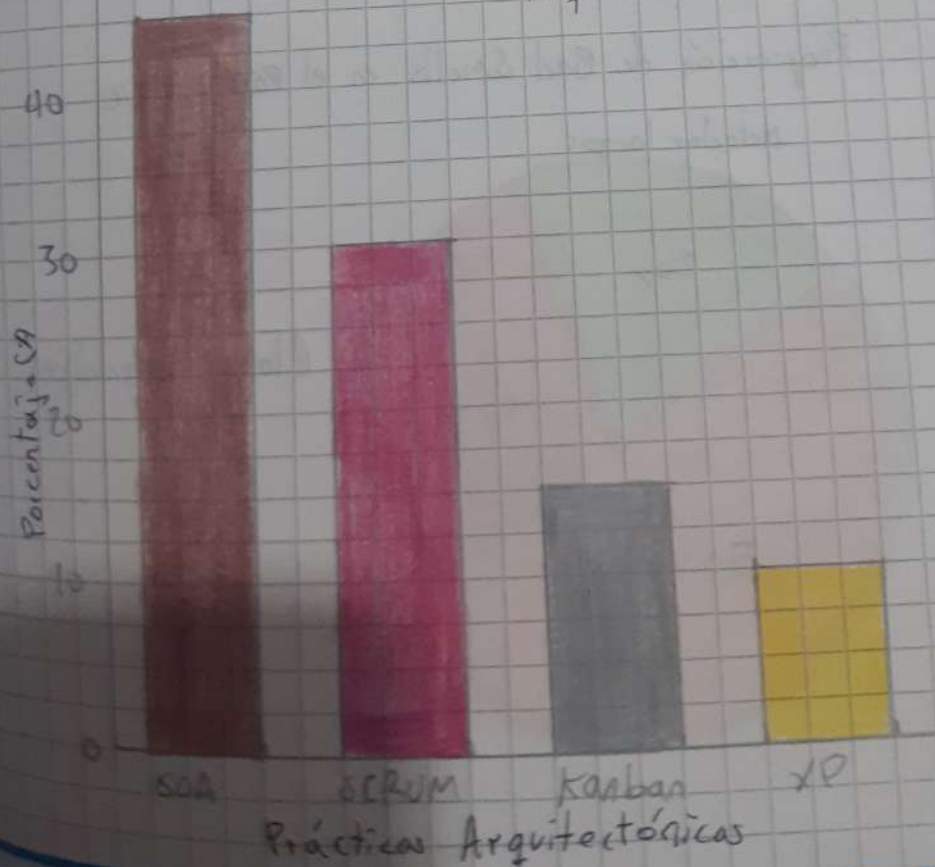
El artículo aborda la necesidad de un sistema eficiente para gestionar los menús de aplicaciones web, proponiendo un administrador de menús jerárquico y multi-idioma basado en el patrón de diseño MVC. Se discute cómo este patrón permite separar las diferentes capas de una aplicación, facilitando el desarrollo 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 identificaron 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.

Frecuencia de uso de Prácticas Arquitectónicas



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 MVC. 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.

Proporción de Bad Smells en el Patrón MVC

