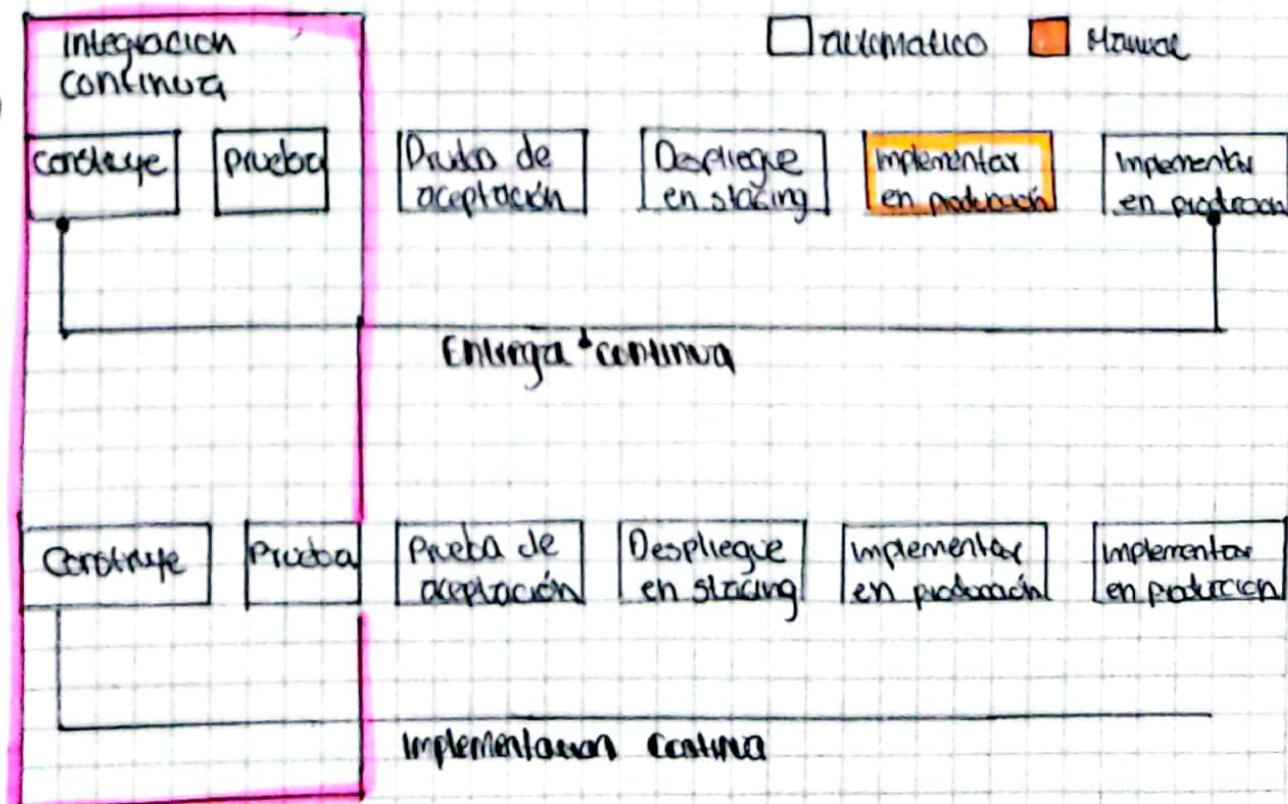


Modelado y Verificación de patrones de diseño de arquitectura de software para entornos de la computación en la nube

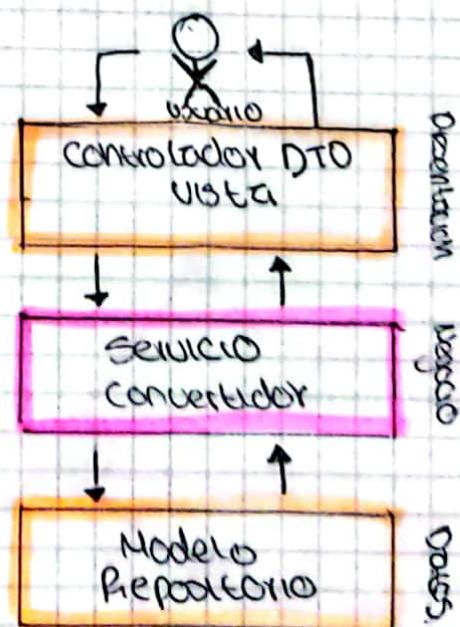
el documento aborda los desafíos de garantizar la calidad en la producción de software, especialmente en entornos de informática en la nube (computación en la nube). Los desafíos incluyen la falta de modelos de arquitectura maduros, documentación y un lenguaje deficiente para comunicar exigencias de calidad. Para tratar el problema se propone un instrumento basado en un matamodo de patrones que facilita el modelado y verifica de manera automática su UDDI, mejorando la eficiencia y precisión del diseño. El propósito del ambiente integrado es guiar a los arquitectos mediante validaciones y sugerencias inteligentes para potenciar la calidad de las soluciones en la computación en la Nube.



Blas, M. J., & Garnet, S. (2019). Modelado y verificación de patrones de diseño de Arquitectura de software para entornos de computación en la nube. *Avist - revista ibérica de sistemas e tecnologías de información*, 35, 1-17.

Desarrollo de una herramienta para el aprendizaje de patrones de diseño software

En este artículo se destaca y detalla un proyecto enfocado en el diseño y desarrollo de un instrumento para simplificar el aprendizaje del diseño en programación, teniendo en cuenta su relevancia en la creación de software. Por las limitaciones del covid-19, el proyecto se llevó a cabo de forma remota, el estudiante adquirió experiencias en las tecnologías Spring y mejoró las habilidades de desarrollo web en Frontend y Backend. Aunque no se lograron alcanzar todas las características esperadas el proyecto da como resultado un prototipo funcional que evolucionará en futuras versiones.

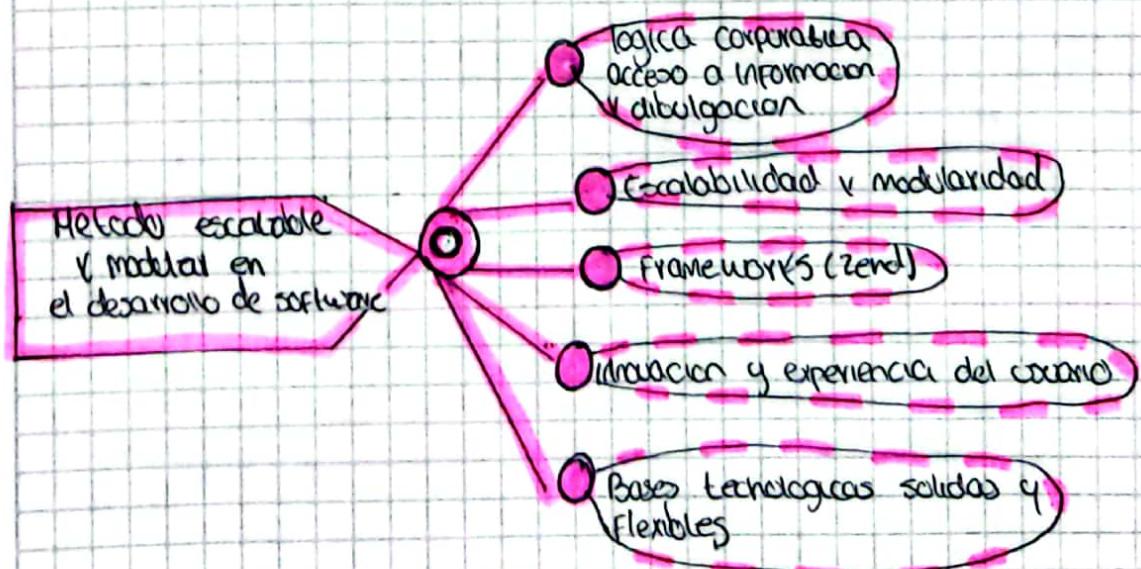


- Zulian, E. R. (2024). Implementación de un framework para el desarrollo de aplicaciones web utilizando patrones de diseño y arquitectura MVC/REST. Ub. edu. cx.
<http://repositorio.edu.edu.cu:8080/xml/handle/123456789/640>

Norma

Implementación de un framework para el desarrollo de aplicaciones web utilizando patrones de diseño y arquitectura MVC/REST

El artículo se centra en la importancia de utilizar un método escalable y modular en la elaboración de software, al distinguir la lógica corporativa, el acceso a la información y la difusión. Esta configuración no solo facilita la gestión de cambios y ampliaciones, sino que además proporciona flexibilidad y sostenibilidad, el uso de frameworks verificados y flexibles, como zend, resalta la manera en que las herramientas ya existentes puedan ser mejoradas para satisfacer las necesidades de la empresa, este método posibilita que las empresas se concentren más en la innovación y la experiencia del usuario, asegurando la protección de una base tecnológica sólida y flexible.

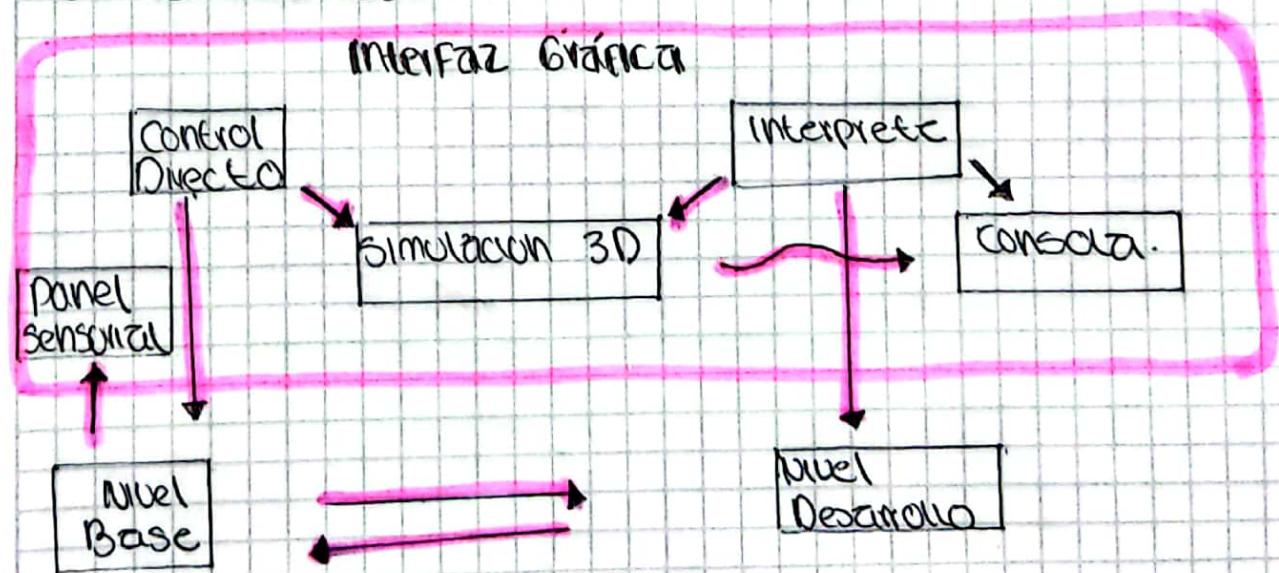


) Tesina, L. (s/f). Universidad de Belgrano. Ed. ux. Recuperado el 12 de noviembre de 2024, de https://repositorio.ub.edu.ar/bitstream/handle/123456789/646/354_Tesina-Zulian.pdf?sequence=1&allow=1

Desarrollo de una arquitectura de software para el robot móvil Lázaro

el texto aborda diferentes tipos de arquitecturas de software aplicadas a robots, centrándose en modelos deliberativo, reactivo e híbrido.

Las arquitecturas híbridas fusionan ambos enfoques con el objetivo de optimizar la capacidad de respuesta de los robots ante situaciones inesperadas y entornos complicados. Un ejemplo es la arquitectura del robot móvil Lázaro, que combina control jerárquico y reactivo, empleando C# para la gestión de actuadores y sensores, lo que proporciona flexibilidad y escalabilidad. La creación de arquitecturas de software para robots resalta la relevancia de la flexibilidad, escalabilidad, y adaptabilidad para cumplir con las necesidades de los usuarios.

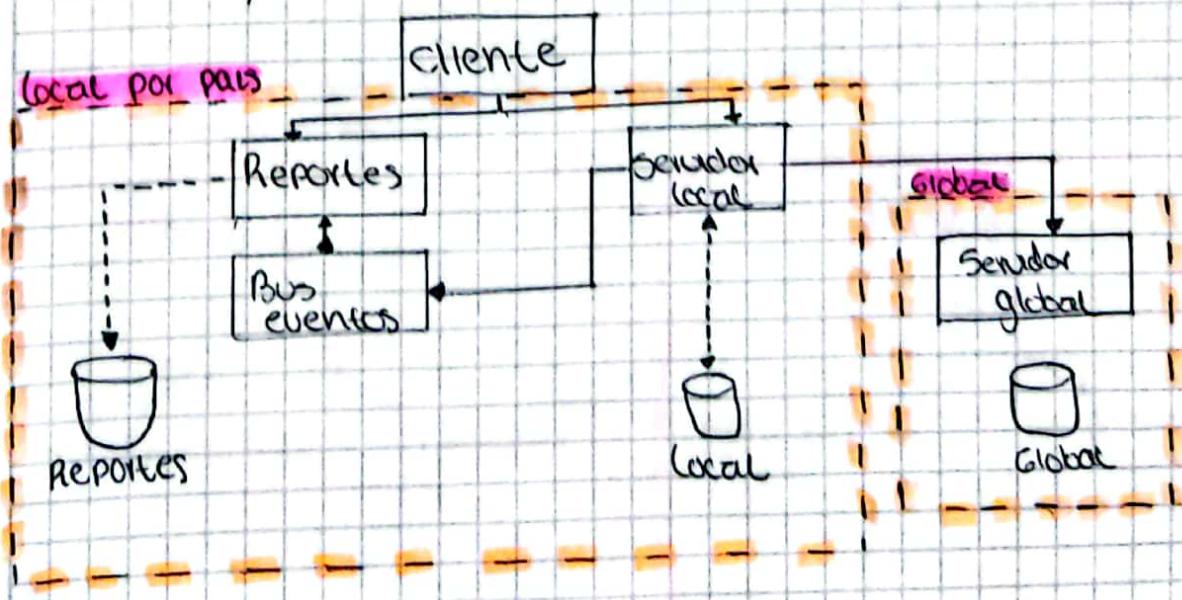


) (S/F). Scielo.cl. Recuperado el 12 de noviembre de 2024,
de https://www.scielo.cl/scielo.php?pid=50718-330520180003003768&script=sci_context

Arquitectura de software para el desarrollo de herramienta tecnológica de costo, presupuesto y programación de obra.

Se destaca la importancia de las tecnologías de la información y las comunicaciones (TIC) como pilares de la economía, la educación y la sociedad en general, que afecta de manera directa a la educación superior al requerir nuevas herramientas y métodos de enseñanza que se adapten a el avance tecnológico. En particular en el campo de la Ingeniería Civil enfrenta desafíos debido a el aumento en la construcción de proyectos.

Para abordar estas necesidades, se propone desarrollar un software educativo en la Universidad de Francisco de Paula Santander enfocada en la administración de costo, presupuesto y planificación de trabajos contribuyendo a la formación profesional.



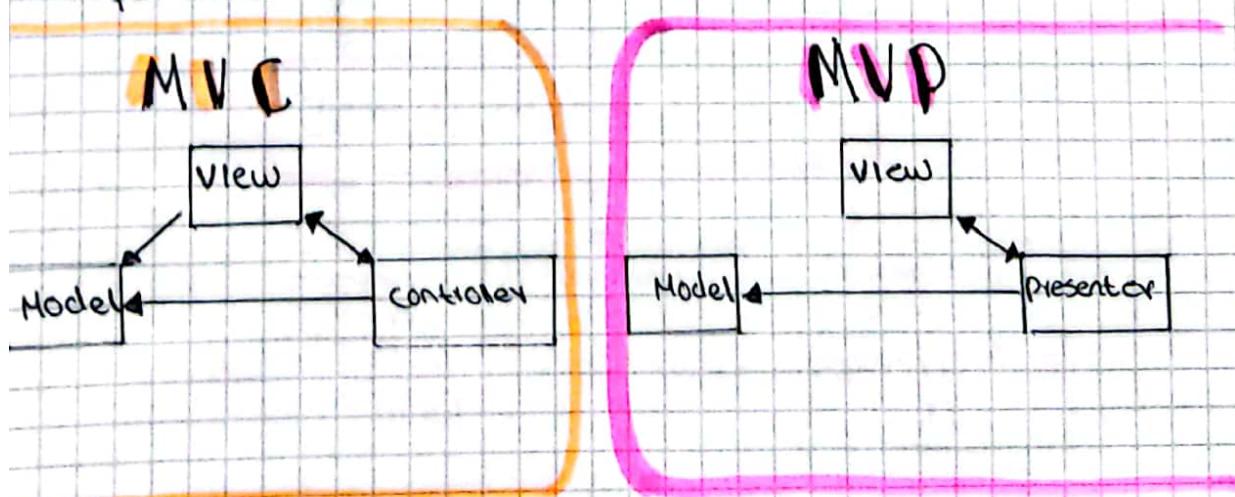
Cáñadas-Gutiérrez, C., Alfonso, J., Baileños-Monsalve, José, E., & Molina Salazar. (n.d.). Cómo citar el artículo número completo. Más información del artículo, página de la revista en redalyc.org. Sistema de Información Científica Redalyc. Red de Revistas Científicas de América Latina y el Caribe, España y Portugal. Proyecto TIC Académico sin fines de lucro, desarrollado bajo la Norma

Lenguajes de patrones de arquitectura de software: Una aproximación al estado de arte

La evolución de la arquitectura de software ha ayudado mucho a los desarrolladores a enfrentar los problemas complicados de los sistemas de información. Con el avance de la tecnología, es importante seguir mejorando estas herramientas y crear procesos más sólidos que permitan a los diseñadores de software construir arquitecturas que no solo solucionen problemas actuales, sino que también se adapten y crezcan para enfrentar los desafíos del futuro.

El objetivo general de este artículo es dar una visión general sobre el estado actual de la arquitectura de software, especialmente en el uso de lenguajes de patrones.

Los lenguajes de patrones son presentados como herramientas esenciales para los diseñadores e implementadores de software, ya que ayudan a mejorar la flexibilidad y la aplicabilidad de las soluciones arquitectónicas.

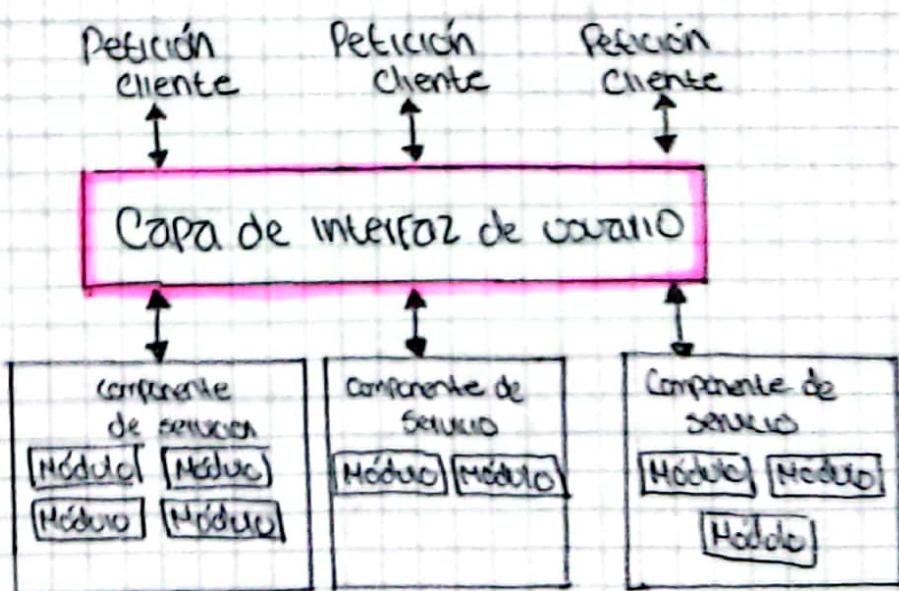


(o/f-b1 - Recuperado el 12 noviembre de 2024, de <http://file:///C:/Users/Angel/Downloads/Digital-ArquitecturaDeSoftware-Esquemas/Servicios-4786605.pdf>)

Arquitectura de software basada en microservicios para el desarrollo

Hoy en día muchos compañías, tanto el sector público como el sector privado, crean programas para hacer más eficientes sus procesos internos, usando arquitecturas tradicionales y monolíticas, no obstante el uso de arquitecturas monolíticas ha traído consigo problemas con el mantenimiento complicado, la falta de escalabilidad y dificultades al agregar nuevas funciones.

Actualmente se está investigando para encontrar una nueva arquitectura de software que se ajuste mejor a las tecnologías actuales y que ayuden a solucionar estos inconvenientes. La idea de combinar a una arquitectura más moderna es una respuesta sensible a estos retos.



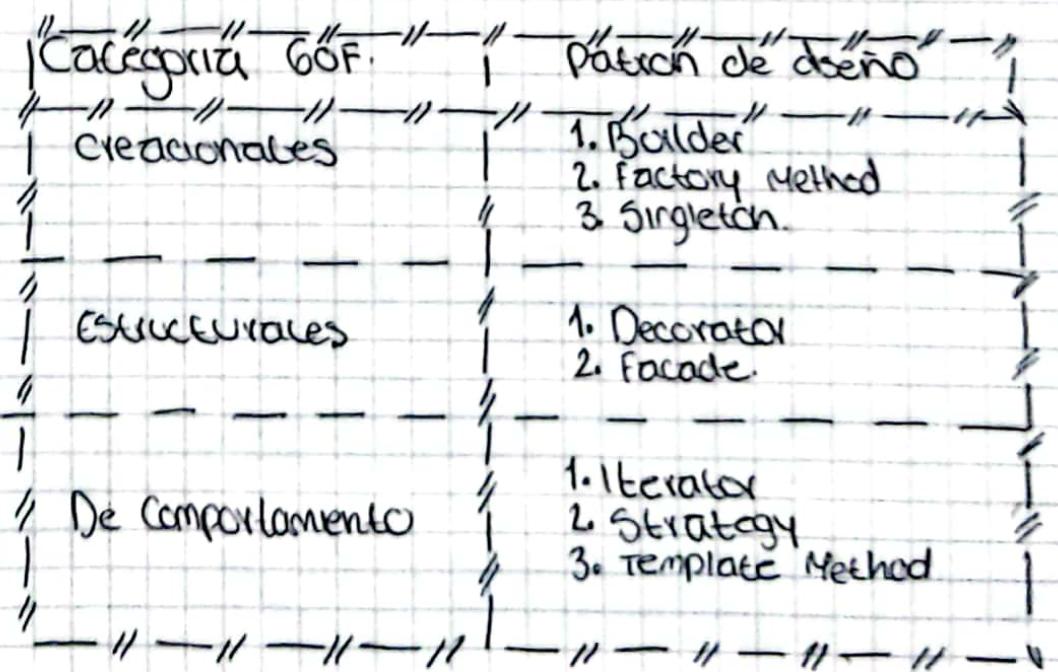
López, D., & Moya, E. (SIP). Arquitectura de software basada en microservicios para desarrollos de aplicaciones web. SA.13.30. Recuperado el 12 de noviembre de 2024, de <https://130.59.13.30/bitstream/10786/1279/1/193%20%20arquitectura%20de%20%20software%20%20basada%20en%20%20microservicios.pdf>

Patrones de diseño GOF en el contexto de procesos de desarrollo de aplicaciones orientadas a la web

Este artículo aborda la investigación sobre los patrones de diseño GOF (Gang of Four) en el contexto de la ingeniería del software, especialmente en la industria de software de Colombia.

El grupo de investigación en ingeniería de software (GRIS) identificó los patrones más utilizados en el desarrollo de software en Colombia, destacando que su implementación mejora la calidad y facilita el mantenimiento.

Se advierte sobre el uso cuidadoso de patrones, ya que no todos los problemas los requieren, evitando complicaciones en el diseño.

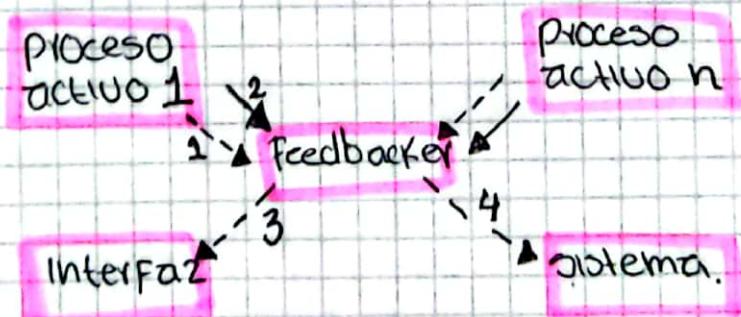


Gómez, C. A., Suárez, J. M., & Gutiérrez, L. E. (2013). Patrones de diseño GOF (The Gang of Four) en el contexto de procesos de desarrollo de aplicaciones orientadas a la web. *CIT Información Tecnológica*, 24(3), 103-114. <https://doi.org/10.4067/S0718-070420130030012>

Patrones de usabilidad: Mejora la usabilidad del software desde el momento arquitectónico.

El artículo se centra en la importancia de seleccionar y aplicar patrones arquitectónicos desde el principio para optimizar la usabilidad de un sistema, permitiendo que la arquitectura se diseñe para cumplir con requisitos específicos. Aunque aún no hay resultados empíricos disponibles, se espera que su enfoque mejore la usabilidad de software final.

El proyecto STATES propone mejorar la usabilidad desde las primeras etapas del desarrollo de software mediante la integración de evaluaciones arquitectónicas y el uso de patrones específicos.



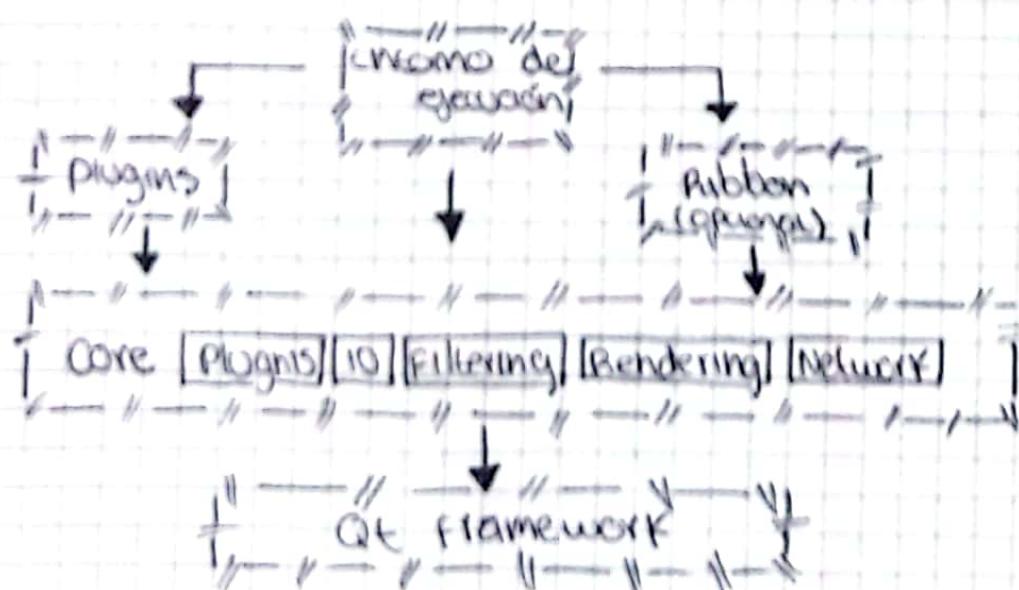
(S/F-C). Researchgate.net. Recuperado el 12 de noviembre de 2021, de https://www.researchgate.net/profile/ana-moreno-56/publication/221505496_Patrones_de_usabilidad_mejora_de_la_usabilidad_del_software_desde_el_momento_arquitectonico/;

Arquitectura de software para el sistema de visualización médica *vismedic*

El diseño de la arquitectura del software es esencial para satisfacer las necesidades de las organizaciones.

La reutilización de conceptos de otros proyectos, junto con una red de flujo de datos dinámica, mejora la eficiencia del sistema y fomenta la innovación.

La integración de tecnologías existentes y configuraciones dinámicas no solo resuelven problemas técnicos, sino que también mejora la experiencia del usuario al permitir soluciones personalizadas sin conocimientos técnicos profundos.



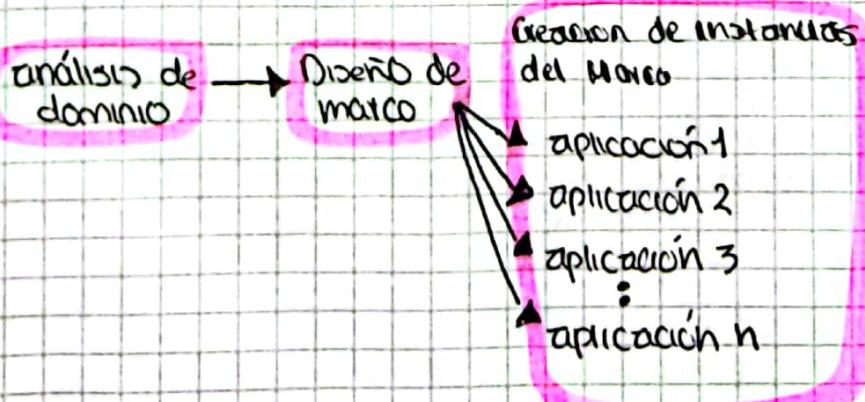
(5/F-d). s1d.co Recuperado el 12 de noviembre de 2024 de <http://s1d.co/pdf/ym/ym11/ym0616.pdf>

Arquitectura de software reutilizable basada en patrones de diseño y patrones de interacción, para el desarrollo rápido de aplicaciones web

El anterior artículo muestra cómo se ha creado un marco reutilizable que permite hacer aplicaciones web de manera rápida y eficiente, especialmente para manejar contenidos. Se basa en patrones de diseño, como el Patrón MVC (Modelo vista controlador) y otros patrones para la interfaz de usuario. El objetivo de este marco es proporcionar una solución común a problemas que suelen aparecer en el desarrollo de software, lo que ayuda a que las aplicaciones sean más fáciles de escribir, mantener y usar.



Diseño tradicional orientado a objetos



Proceso de desarrollo del marco

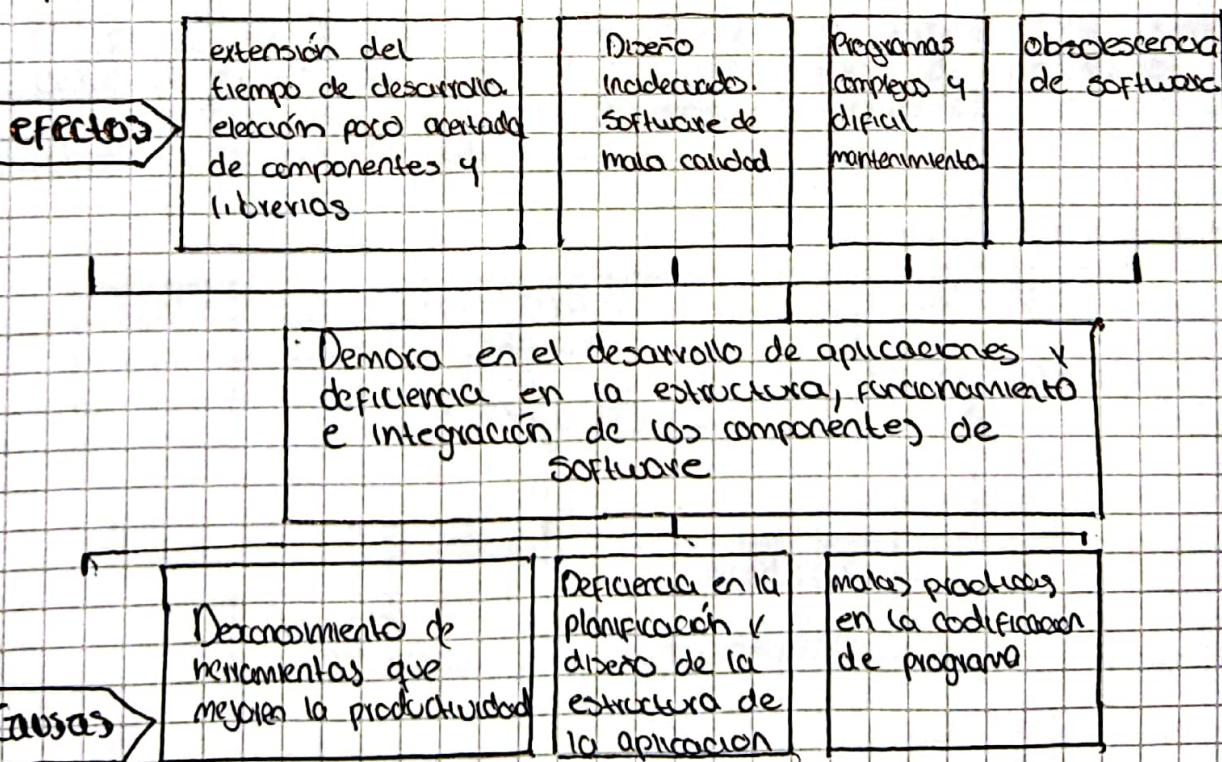
Santiago, D., Esteban, D., Ocaña, F., Exactas, C., Tecnología, I., Rossi, M., & De Gómez, P. (2009). UNIVERSIDAD NACIONAL PROFESIONAL Gómez. <http://170.210.231.10:8080/jspui/bitstream/123456789/191/1/MONICO%20%20Marte%20C3%A0n%20Miguel.pdf>

Revisión sistemática sobre generadores de código fuente y patrones arquitectónicos.

Este artículo trata sobre los generadores de código que son herramientas que ayudan a automatizar gran parte de la programación.

Estas herramientas crean código usando plantillas y reglas ya establecidas. Esto hace que el desarrollo de software sea más rápido, disminuye los errores y mejora la uniformidad del código.

Al seguir patrones arquitectónicos, estos generadores de código permiten a los programadores enfocarse en la lógica del negocio y en la arquitectura, en lugar de perder tiempo en tareas de codificación que se repiten.

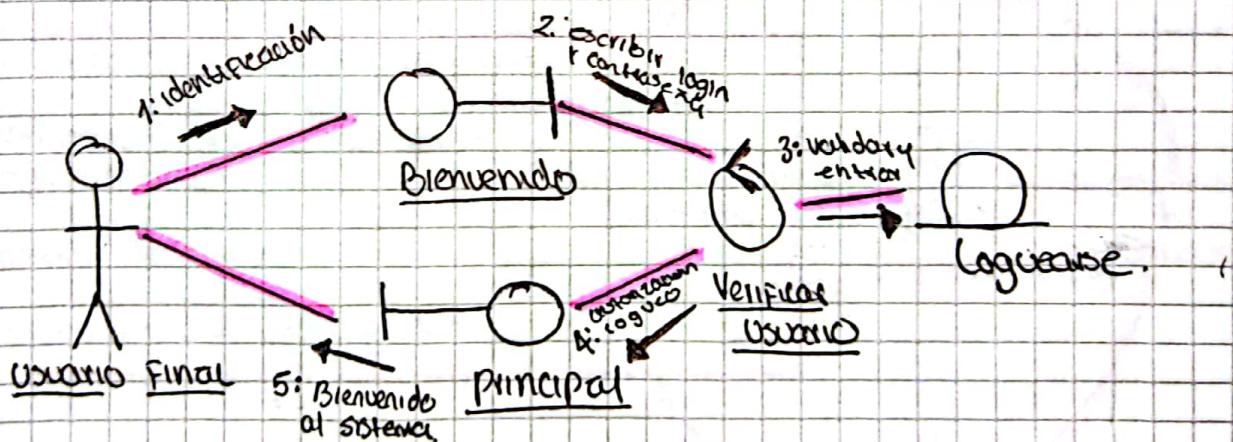


Revisión sistemática sobre generadores de código fuente y patrones de arquitectura - ProQuest. (2020). Proquest.com. <https://www.proquest.com/openview/03dd8f10be1930ffgfb034646c5530d/1?pqorigsite=gscholar&ddid=193008>

Herramientas para reuso de código javascript orientado a patrones de interacción

Hoy en día, hay más páginas web disponibles en internet. Este artículo nos habla sobre la importancia de usar patrones de interacción, ya que estos ayudan a los diseñadores a crear de manera más efectiva.

Al reimaginar soluciones efectivas a problemas que surgen los desarrolladores de sitios web y aplicaciones están siempre buscando nuevas formas de diseñar páginas que ofrecen soluciones efectivas a problemas que siguen apareciendo, utilizando métodos que ya han demostrado ser útiles.



De interacción (n.d.). HERRAMIENTA PARA REUSO DE CODIGO JAVASCRIPT ORIENTADO A PATRONES. Retrieved November 12, 2024, from <https://www.redalyc.org/pdf/4127/41271429330438009.pdf>

Introducción a la arquitectura de software

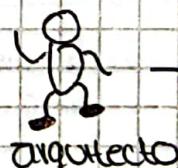
Este texto habla sobre por qué es importante ver la arquitectura de software como una etapa diferente y más general en el ciclo de la vida del software. También menciona que es crucial tener en cuenta tanto los requisitos funcionales como los no funcionales al definir la arquitectura.

Es esencial entender la diferencia entre arquitectura y diseño, ya que la arquitectura se ocupa del sistema de manera más general, mientras que el diseño se enfoca en los detalles de la implementación. Al centrarse en estilos arquitectónicos y conectores complejos, se puede formalizar esta idea y ayudar a crear sistemas que sean fuertes y adaptables.

Influencias sobre el arquitecto

Intereses
organización
estado de la profesión
experiencia del arquitecto.

→ requerimientos
→ Funcionales
→ no funcionales
(calidades)



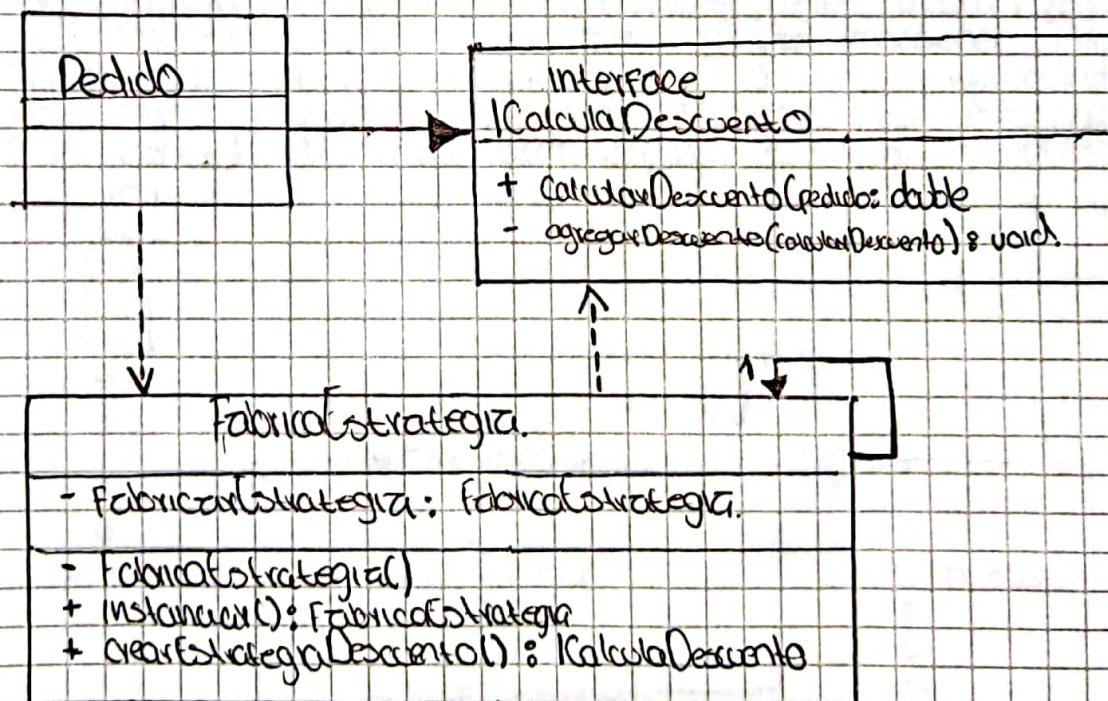
arquitecto

→ arquitectura
↓
Sistema

Apliación de patrones de diseño para garantizar alta flexibilidad en el software

Para conseguir esta flexibilidad, se sugiere utilizar patrones de diseño como estrategias, como el patrón composite y el patrón de fábrica.

Estos patrones ayudan a crear aplicaciones que se adaptan de forma eficiente a nuevos requisitos sin necesitar mucho tiempo o recursos. En pocas palabras al aplicar estos principios de diseño, se puede desarrollar software muy flexibles, especialmente en sistemas que tienen muchas reglas de negocio que cambian con el tiempo. Esto permite que las empresas se mantengan competitivas en un entorno que está en constante cambio.

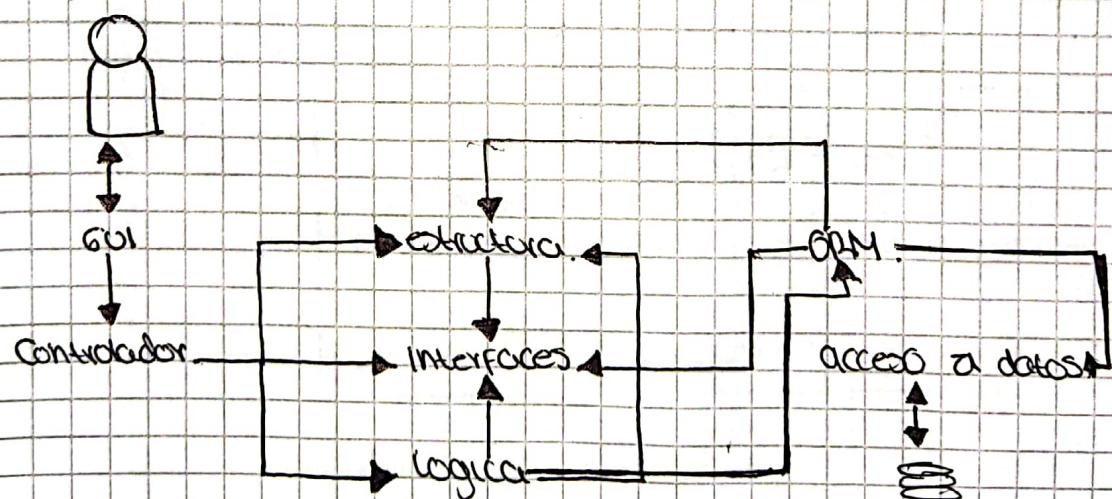


Vista de aplicaciones de patrones de diseño para garantizar alta flexibilidad en el software (2021) UCV.edu.pe. <https://revistas.ucv.edu.pe/index.php/ntd/article/view/1685/1498>

Arquitectura de software académica para la implementación del desarrollo de software en capas.

Se destaca que las arquitecturas bien estructuradas, como las arquitecturas cliente-servidor, orientadas a objetos, basadas en componentes o en capas, ayudan a separar las funciones. Esto significa que si se hacen cambios en una parte del sistema, no afectarán a los demás. Esto hace que sea más fácil mantener y escalar el software, por que cualquier ajuste, ya sea en el negocio, en el control o en la prestación de las vistas, se pueden hacer sin causar problemas en otra partes.

La flexibilidad y escalabilidad que brindan estas arquitecturas son muy importantes para garantizar que los sistemas sean fáciles de mantener y se puedan adaptar a los cambios con el tiempo, aunque hay otros factores, como la utilización de microservicios o el control de transacciones, que también pueden ayudar.



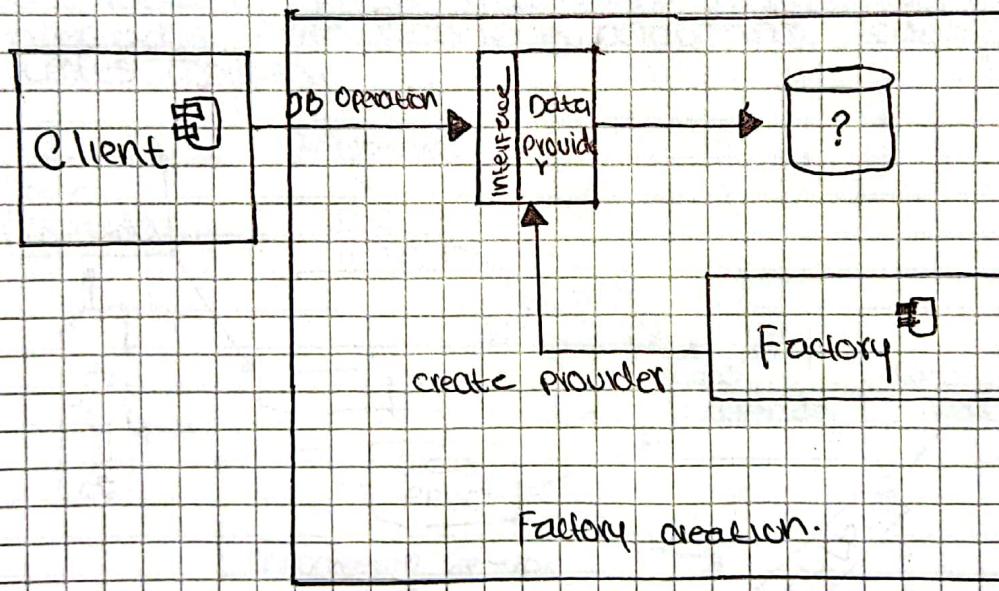
Cerdáceij, Darío G. (n.d.). <https://www.econstor.eu/bitstream/10419/130225/1/837816424.pdf>

Introducción a los patrones de diseño

Los patrones de diseño tienen su origen en la arquitectura tomando como referencia el trabajo de Christopher Alexander. Él decidió que la arquitectura moderna había disminuido en calidad y describió los patrones como respuestas a problemas que se repiten en el entorno, que se pueden utilizar y aplicar en diferentes situaciones.

Alexander y su equipo crearon una colección de patrones arquitectónicos en "A Pattern Language", que ofrece soluciones útiles a problemas de diseño que son comunes en contexto particulares. Este mismo concepto se utiliza en los patrones de software.

Flujo de ejecución



Vista de aplicaciones de patrón de diseño para garantizar alta flexibilidad en el software. (2024). UCV.edu.pe
<https://revistas.ucv.edu.pe/index.php/ntd/article/view/1695/1496>

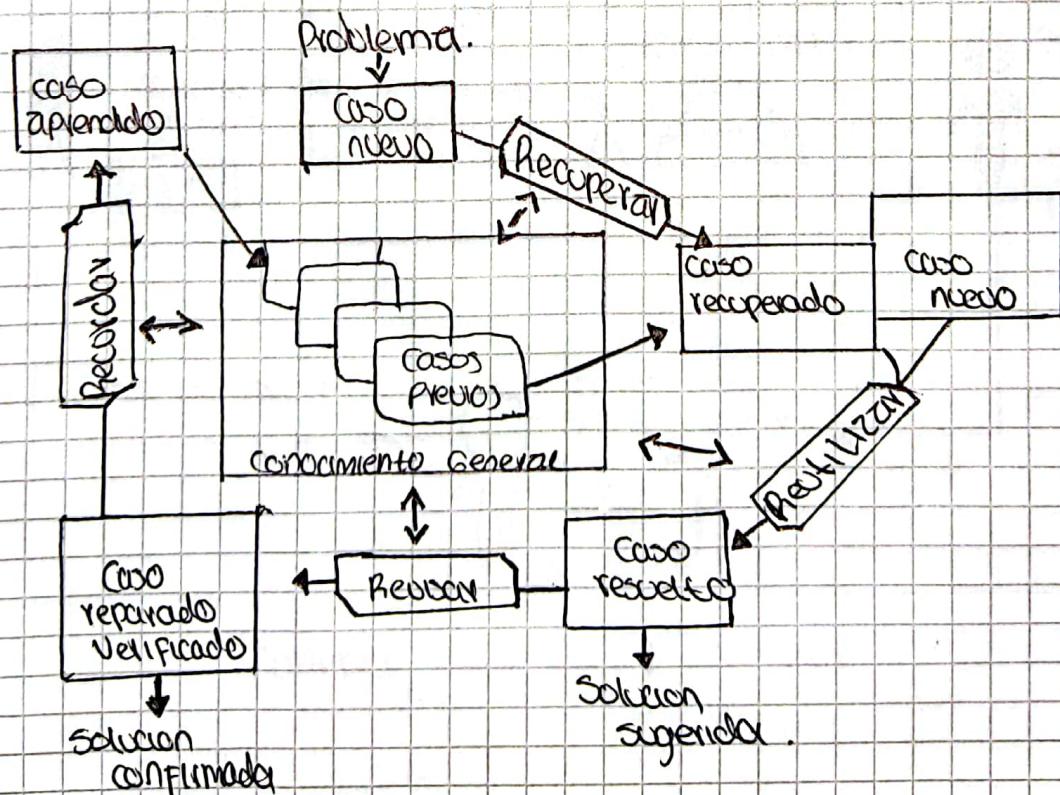
Norma

Modulo de recomendación de patrones de diseño para EGpat

Se busca crear un camino que faciliten la revisión de recursos educativos, así como la gestión y evaluación de patrones de diseño, y que también permitan consultar estos recursos de manera sencilla. Sin embargo, todavía hay problemas relacionados con el tiempo y la complejidad de las consultas.

Aunque hay patrones disponibles en una fuente local, encontrar cuál o cuáles son más apropiados para el recurso educativo que se está creando sigue siendo un proceso complicado.

Este artículo presenta los detalles sobre cómo implementar el módulo para EGpat que al enfrentar un problema, puede recomendar los patrones más adecuados para resolverlo analizando todas las fuentes posibles.

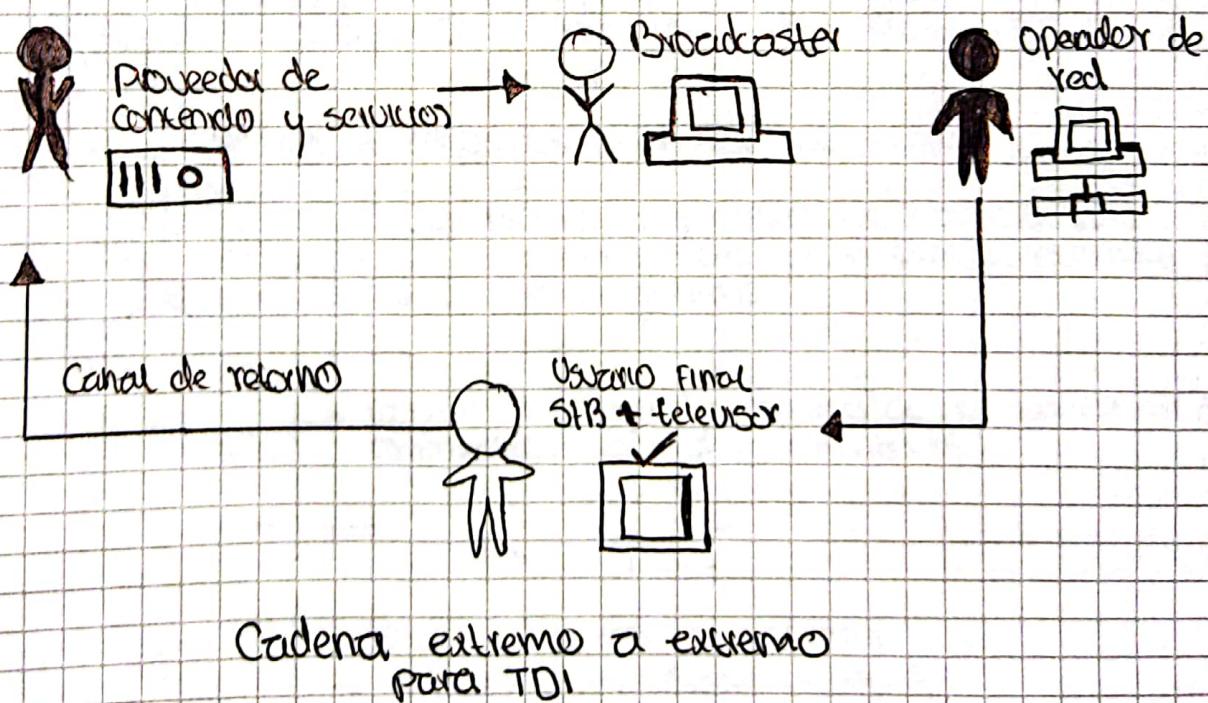


ACCU, A., ULLI CESPEDES, LUB ÁNGEL, AZCUEY, A., & ULLI CESPEDES, CUSÁNGEL. (2021) módulo de recomendación de patrones de diseño para EGpat. Revista cubana de ciencias informáticas, 15(22), 118-137 http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2232-18922020000200008

Arquitectura de software para el soporte de comunidades académicas virtuales en ambientes de televisión digital interactiva

El avance de tecnología ha transformado internet y la televisión digital interactiva (TDI), ofreciendo nuevas formas de aprender y colaborar gracias a herramientas como foros y redes sociales. Se sugiere una estructura que utiliza servicio REST-JSON para aplicar aplicaciones en TDI, especialmente en la educación, aprovechando las ventajas de la web 2.0 y la interactividad. Iniciativas como EDITU y T-Maestro investigan cómo usar tecnologías como IPTV y TDI en el aprendizaje, mejorando la educación y superando los límites del e-learning tradicional.

Este estudio presenta una solución que aplica minería de texto y razonamiento basado en patrones de diseño en aplicaciones interactivas dentro del contexto de la TDI.



1 Campo, W., Chanchi, G., & Arce Negrete, J. (2013). Versión final recibida D.C. Formación Universitaria (Q2), 3-14. <https://db1.org/10.4067/S0718->

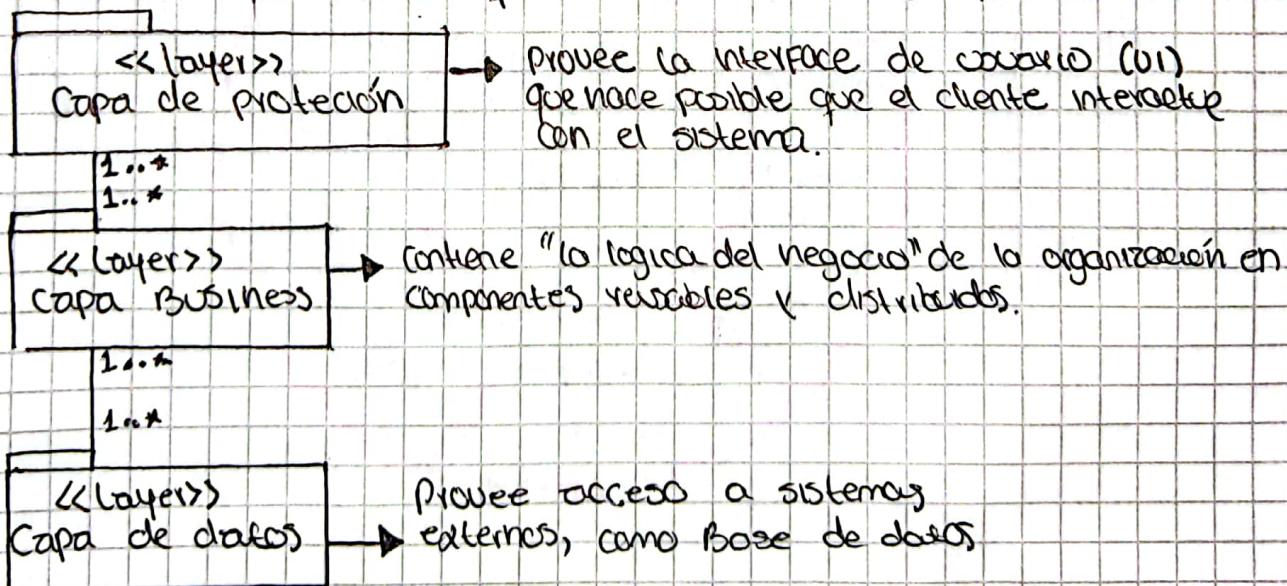
Desarrollo de sistemas de software con patrones de diseño orientado a objetos

el uso de Patrones de diseño, en el desarrollo de software ayuda a reutilizar estructuras lógicas y aumenta la productividad. La programación orientada a objetos (POO) es fundamental para esta reutilización ya que organiza el diseño en torno a componentes que pueden utilizar varias veces.

Los patrones de diseño hace que sea más fácil construir sistemas, disminuyendo los costos y mejorando la calidad del software. Ejemplo como los patrones Informador, Sensor y de 3 capas permite que el sistema sea más modular y flexible, además de ayudar a manejar la complejidad en partes más simples.

Estos patrones también ayudan a que el sistema sea escalable y mejora la comunicación entre los desarrolladores y los analistas. Para implementar estos patrones, es importante usar lenguajes de programación orientados a objetos como .NET, Java o Delphi.

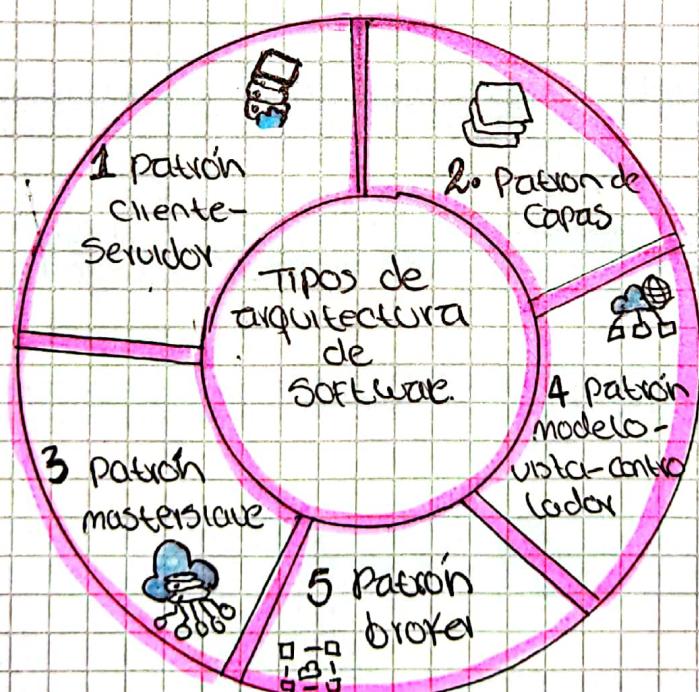
Patrón de arquitectura: Modelo de 3 capas



(S/F-F). EDU.PE Recuperado el 12 de Noviembre de 2024, de <https://cyberbeats.unmsm.edu.pe/backend/api/core/161stteam/q2601647-3651-4e75-9fd1/content>.

Atributos de calidad y arquitectura de software.

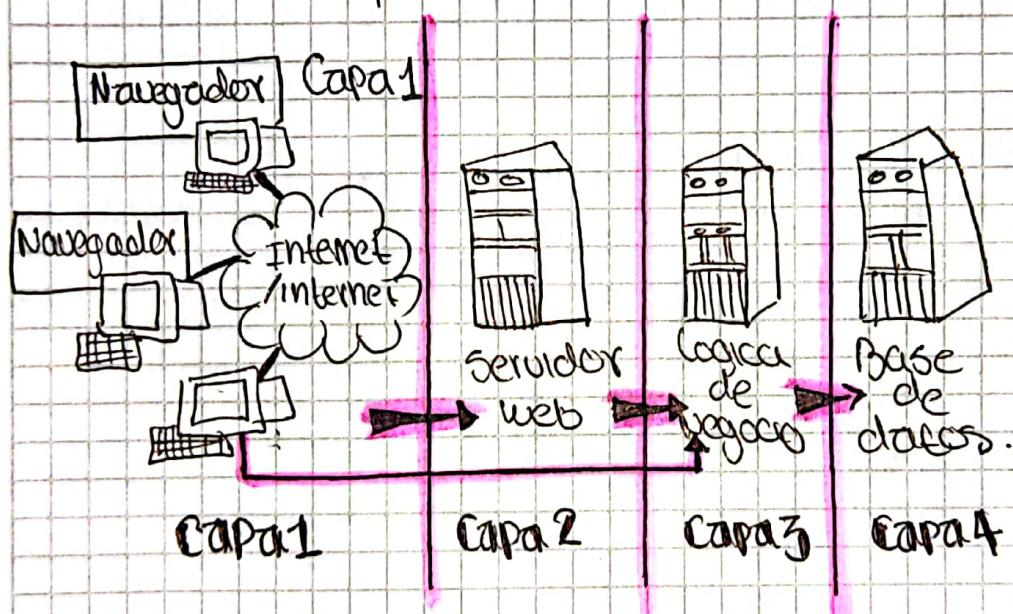
Se exploran los conceptos básicos de la arquitectura de software, los atributos de calidad y cómo crear un modelo de arquitectura que satisfaga estos requisitos. Se enfatiza la importancia de definir la arquitectura desde diferentes ángulos para resaltar las distintas cualidades del sistema y tomar decisiones de diseño que sean equilibradas, asegurando que se logre una calidad sin comprometer demandas a los demás. Diseñar una arquitectura que cumpla con los requisitos tanto funcionales como de calidad es un desafío complicado, ya que las decisiones que se tomen esta etapa afectarán de manera significativa las características del sistema final. Se presenta un método que se basa en analizar y comprobar que la arquitectura propuesta cumplen con las calidad deseadas.



Bastanica, M. (n.d.). Atributos de calidad y arquitectura del software. Retrieved November 12, 2024 from http://www.grsic.upm.es/meetings/conferencias/jisic04/tutoriales/1_tut4.pdf

Arquitectura de Software en el proceso de desarrollo ágil: una perspectiva basada en requisitos significantes para la arquitectura.

(los metodologías) ágiles son conocidas por ser flexibles y fomentar la colaboración, pero a veces puede complicar la definición precisa de los requisitos arquitectónicos al comienzo de un proyecto. Por otro lado la arquitectura de software se enfoca en requisitos no funcionales que son muy importantes. Para solucionar este problema, se creó el concepto de arquitectura ágil que intenta combinar los requisitos arquitectónicos con el enfoque ágil, permitiendo los requisitos importantes para la arquitectura.

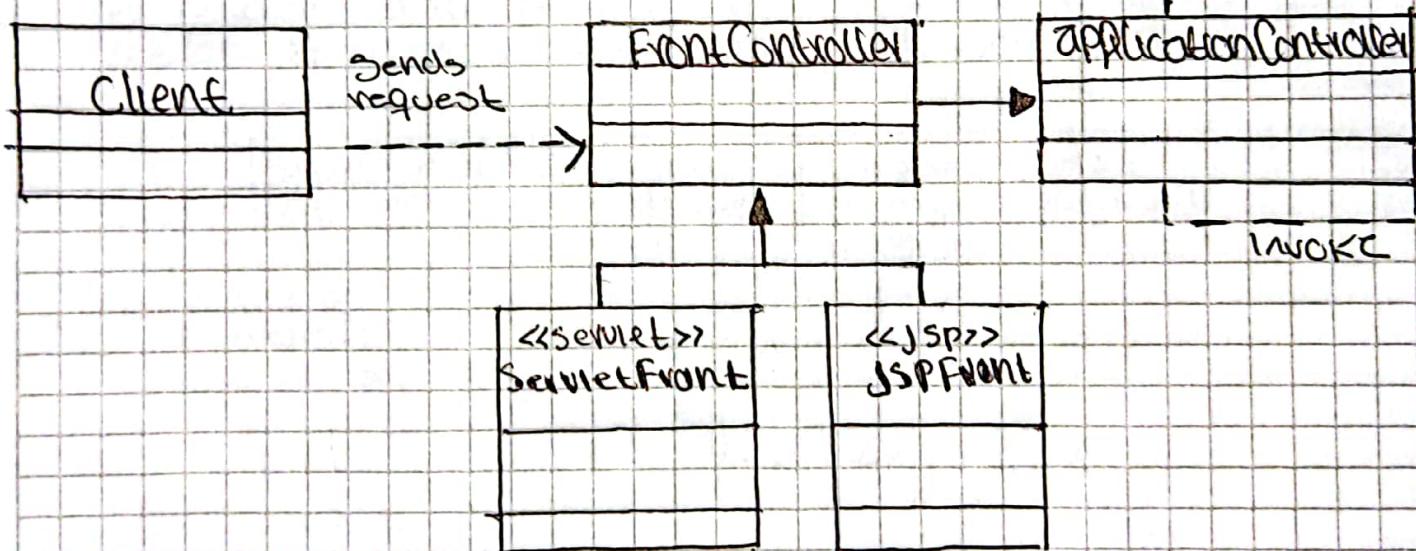


Navarro, M.E. Moreno, M., Aranda, J. Perea, L., & Puerto, J.R. (2018). Arquitectura de software en el proceso de desarrollo ágil: una perspectiva basada en requisitos significante para la arquitectura. <https://seclib.unlp.edu.ar/bitstream/handle/10915/67795/document0>

análisis comparativo de patrones de diseño de software

Los patrones de diseño son métodos que se utilizan a menudo para solucionar problemas que aparecen frecuentemente en la creación de software. Algunos ejemplos son el método plantilla, MVC, MVP, controlador frontal y MVVM. Estos patrones hacen que sea más fácil organizar las aplicaciones y usar el código en diferentes partes. También ayudan a mantener separada la lógica de la interfaz, lo que puede hacer que el desarrollo sea más eficiente, aunque a veces puede complicar las cosas. Además, los patrones garantizan que el código sea válido y utilizable, lo que mejora la forma en que se documenta y se mantiene.

dorache



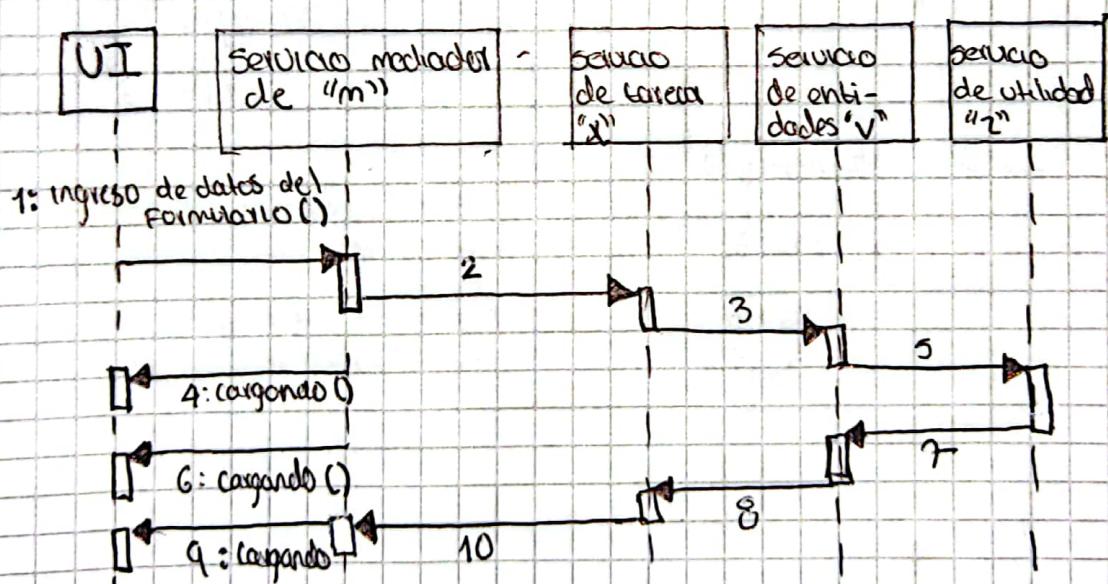
Danilo, O., Patricia, N., & Víncio, M. (2022). Análisis comparativo de patrones de diseño de software. Pab. Del conocimiento: Revista científico-profesional, 7(2), <https://doi.net.unilloja.co/clesorga/articulos/90472929.pdf>.

Norma

Identificación y clasificación de patrones de diseño de servicios web para mejorar QoS.

(Los) servicios web son importantes para concentrar diferentes aplicaciones en las organizaciones de hoy en día. La arquitectura orientada a servicios (SOA) es un modelo que se adapta bien y se utiliza comúnmente con servicios web. Esta tesis sugiere que se puede mejorar la calidad del servicio (QoS) de estos servicios usando patrones de diseño que son bien conocidos.

Se presenta una guía para medir como estos patrones afectan la QoS, tanto de manera individual como en conjunto. La investigación también analiza ontologías web, principios de diseño y antipatrones. Además se aplica esta metodología a los APIs del sistema Saber, mostrando que es efectiva. El trabajo resalta la necesidad de evitar antipatrones que puedan perjudicar el rendimiento y la escalabilidad.



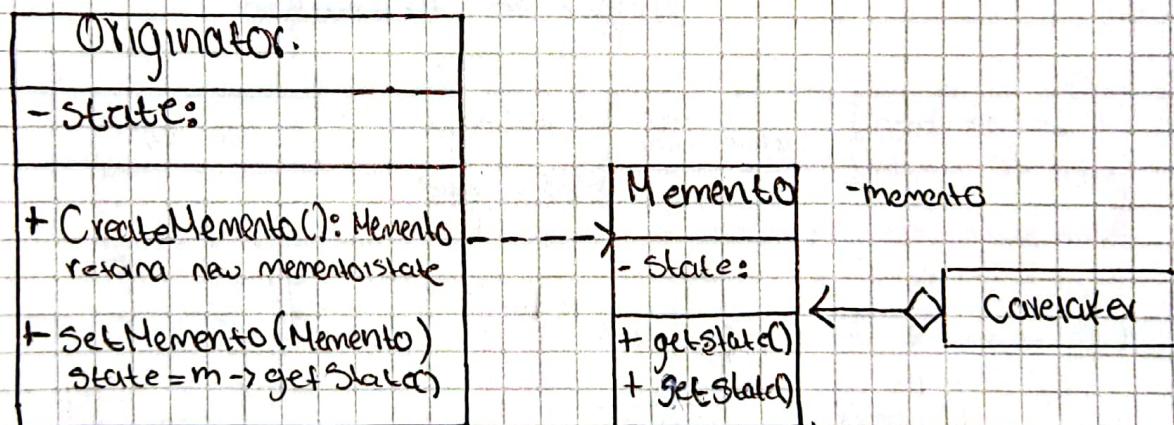
(S/F-9). Educar Recuperado el 17 de noviembre de 2024, de https://seclci.unp.edu.ar/bitstream/handle/10915/94512/Documento_completo.pdf-PDF.pdf?sequence=1&isAllowed=y

Especificación del patrón de diseño Memento a través de un perfil UML

(los patrones de diseño son estratégicos basados en la experiencia de expertos que ayudan a documentar conocimientos y crear una tecnología común en el desarrollo de software)

el catálogo más conocido es el "Gang of Four" (GoF) que significan los patrones en tres tipos: creacionales, estructurales y de comportamiento.

para especificar y validar estos patrones se propone la arquitectura de perfiles UML de patrones de diseño (APP) que organiza los patrones en tres niveles, asegurando coherencia entre diagramas estáticos y dinámicos



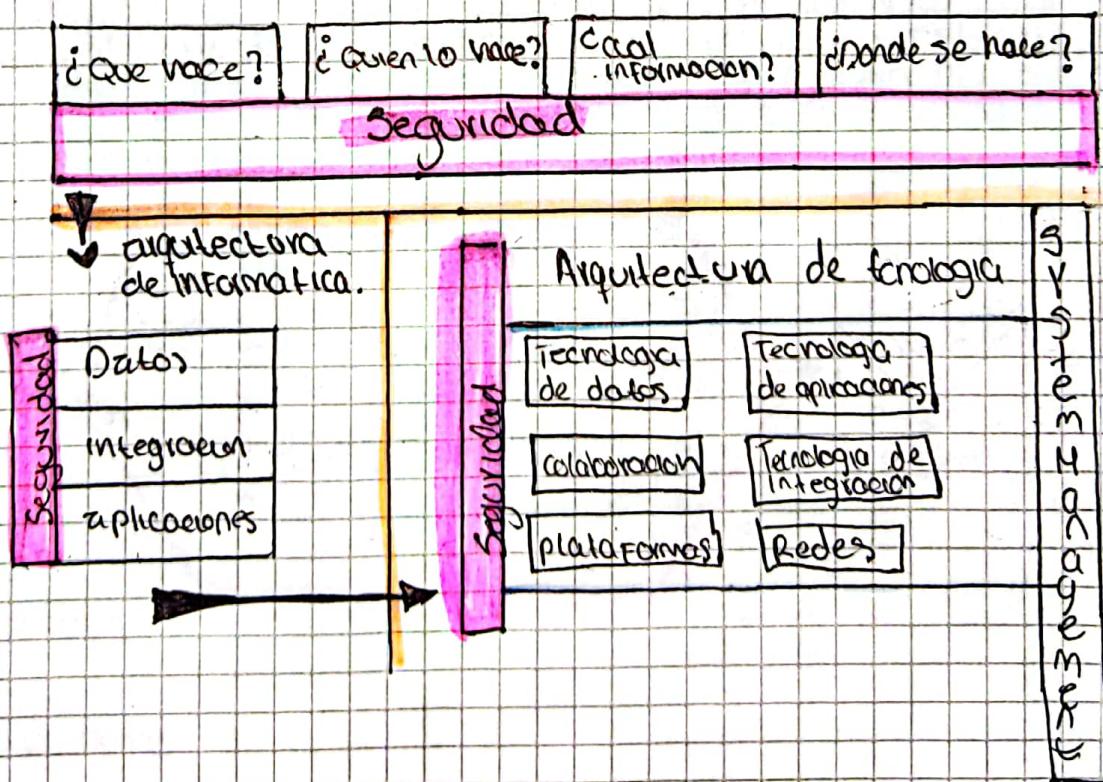
Researchngate.net. Recuperado el 12 de noviembre de 2021 de <https://www.researchngate.net/Profile/Alberto-Cortez-3/publications/1342109503> Especificación del patrón de diseño Memento a través de un perfil UML/cin515ef20bd4628851

Documentación y análisis de los principales frameworks de arquitectura de software en aplicaciones empresariales.

El avance tecnológico ha impulsado nuevos metodologías y herramientas en empresas, destacando la importancia de la arquitectura de software. Se analizó los frameworks más utilizados en el desarrollo de aplicaciones empresariales, resaltando su aplicabilidad.

Una buena arquitectura alinea las estrategias de negocio con la infraestructura tecnológica. La elección correcta de framework es clave. Es fundamental convinar correctamente los componentes de la arquitectura, contra una buena gestión y manteniendo una comunicación constante para el éxito en la implementación.

Arquitectura de Negocio



Sarasti y España, H. F. (2016, April 12). Documentación y análisis de los principales frameworks de arquitectura de software en aplicaciones empresariales. sendie. unlp.edu.ar. <http://sendie.unlp.edu.ar/handle/10915/5783>.

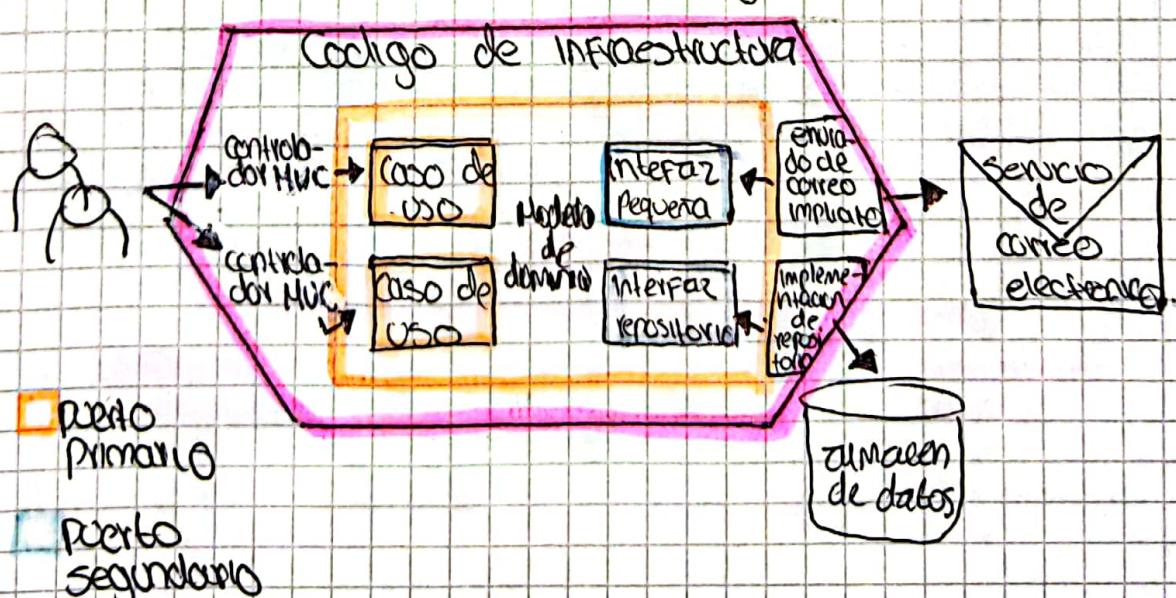
Implementación de una arquitectura de software guiada por el dominio.

La idea principal de este artículo es dividir la lógica de negocio de los aspectos tecnológicos para que el software sea más flexible, fácil de mantener y pueda evolucionar mejor.

El diseño dirigido por el dominio proporciona un método organizado para crear modelos de dominio que refleja la realidad, mientras que la arquitectura hexagonal ayuda a separar el núcleo del negocio de la tecnología, lo que permite hacer cambios en las herramientas sin afectar el dominio.

El artículo sugiere un enfoque innovador para el desarrollo de software destacando la necesidad de separar la lógica de negocio de las tecnologías que se utilizan.

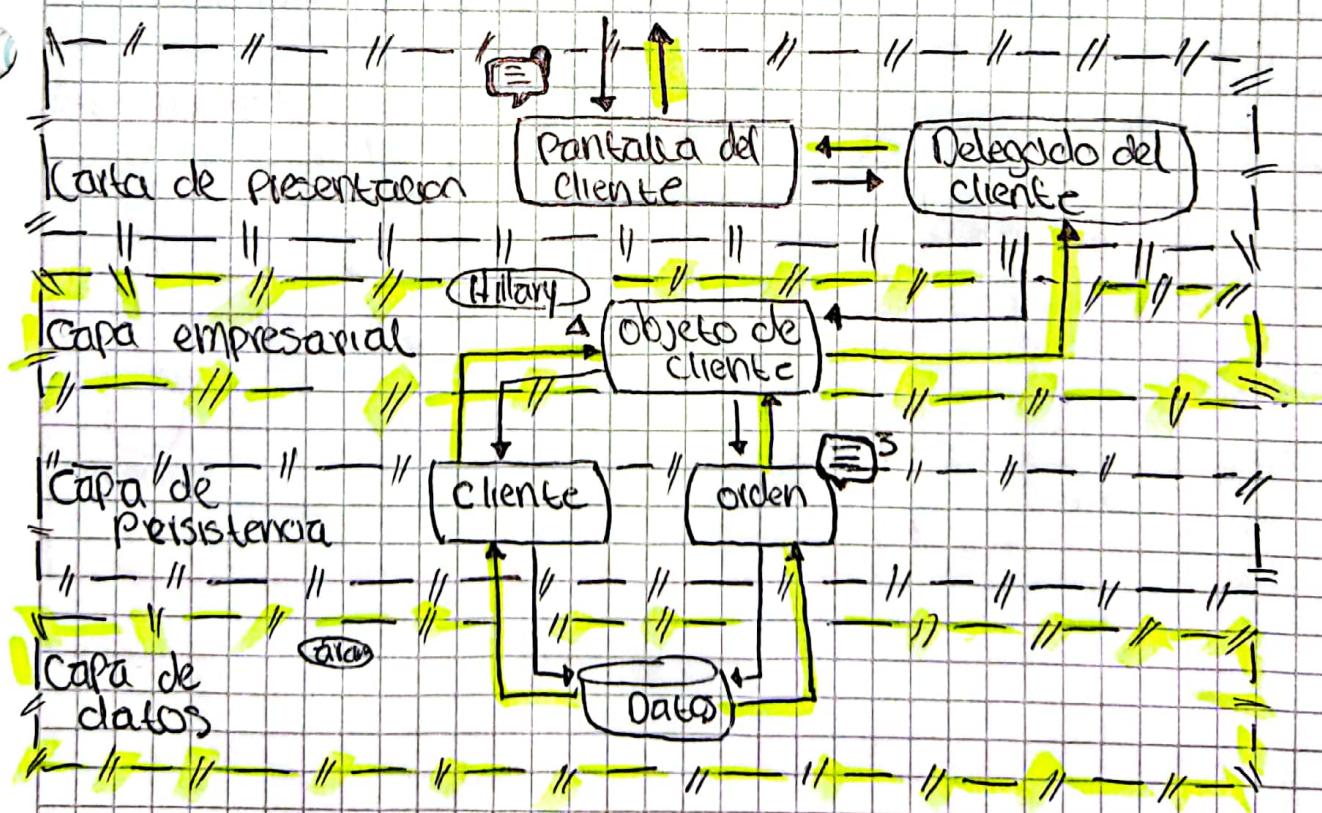
Al aplicar el diseño dirigido por el dominio (DDD) y la arquitectura hexagonal se prefiere construir sistemas más fuertes y sencillos de manera que se adapten a los cambios del mercado y la tecnología sin dañar el núcleo del negocio.



Arquitectura de software, esquemas y servicios

el texto trata sobre cómo cada vez más empresas necesitan aplicaciones que tengan funciones avanzadas que antes no eran tan comunes. Estas funciones incluyen la posibilidad de funcionar sin depender de un sistema operativo o base de datos específico, poder acceder a ellas desde diferentes lugares, integrarse con otros sistemas y manejar muchas interacciones en mismo tiempo.

Se mencionan que otras arquitecturas orientadas a servicios (SOA) Adopción de arquitecturas orientadas a servicio es importante por que ayuda a que los diferentes componentes de una aplicación estén menos conectados entre si, lo que mejora la reutilización y hace que sea más fácil adaptar las aplicaciones sin importar donde estén o qué plataforma se use.



Romero, P.A. (2020). Arquitectura de software, esquemas y servicios. Ingeniería Industrial, 22(1), 1. <https://doi.inet.unioja.es/descarga/articulo/1478855.pdf>.