**Artículo 1:**

**Título:**

**Análisis comparativo de patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad: Una revisión sistemática de la literatura**

**Resumen:**

En el inmenso mundo de las aplicaciones móviles existen diversos patrones de diseño que nos brindan facilidades de crear un desarrollo más eficiente y organizado. El objetivo de la revisión es el de encontrar los principales estudios sobre patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad, y posteriormente determinar criterios de identificación que servirán como herramienta de selección de patrones de diseño de calidad.

**Gráfica:**

**Reflexión:**

Los patrones de diseño han mostrado u mayor impacto en el desarrollo de software. Dado que los patrones de diseño proporcionan soluciones comprobadas, el desarrollo de aplicaciones móviles puede ser un proceso tedioso, ya que cada aplicación debe pasar por los ciclos de desarrollo para garantizar que la aplicación se ajuste a los atributos de calidad estándar. Luego de que el desarrollador compare los principales patrones de diseño de software, finalmente podrá implementar la más conveniente para el proyecto.

**Cita APA:**

Abanto Cruz, J. A., & Gonzales Ramírez, O. F. (2019). Análisis comparativo de patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad: Una revisión sistemática de la literatura.

**Artículo 2:**

**Título:**

**Recomendaciones para la Formación de una Empresa de Desarrollo de Software Competitiva en un País como Colombia**

**Resumen:**

Algunas de las recomendaciones para ser una empresa competitiva, son, buscar que la organización sea reconocida como una compañía transparente con los clientes, la estructura de la organización, debe estar fortalecida con un organismo que oriente el desarrollo tecnológico tener mecanismos que les permitan medir y saber si se están cumpliendo las metas establecidas en los procesos, los productos y proyectos de la compañía.

**Gráfica:**

**Reflexión:**

Al tener una empresa dirigida al desarrollo de software, o al querer formarla, es necesario tener ciertos criterios importantes para de esa manera poder ser una empresa competitiva en todo este campo del desarrollo de software, más en un país como lo es Colombia, el cual en estos tiempos a tomado fuerza en el tema de la tecnología y sus componentes. Todo el artículo hablaba sobre recomendaciones, las nombraban y luego explicaban cada una de ellas, para que la organización de una compañía sea buena y resalte.

**Cita APA:**

Londoño, L. F. L. (2005). Recomendaciones para la Formación de una Empresa de Desarrollo de Software Competitiva en un País como Colombia. *Avances en Sistemas e Informática*, *2*(1), 41-52.

**Artículo 3:**

**Título:**

**Introducción a la Arquitectura de Software**

**Resumen:**

En los primeros años de la construcción de software no existía el diseño del sistema como una etapa independiente a la programación. A principios de la década del 90 algunos investigadores, comenzaron a ver la necesidad de investigar y desarrollar un nivel de abstracción superior al dl diseño, al que llamaron “Arquitectura de software”. La arquitectura del sistema es el resultado de combinar decisiones técnicas, sociales y del negocio.

**Gráfica:**

**Reflexión:**

A partir de la creación de la arquitectura de software, se crearon de igual manera diferentes y variados modelos de desarrollo, como el “Modelo de Cascada”, la arquitectura del sistema, la cual es la que se le sigue a la Ingeniería de Requerimientos. La fase de la arquitectura del sistema, se subdivide en tres etapas: elección del estilo arquitectónico, selección de los patrones de diseño y diseño de componentes; lo cual es importantes de aprender para llevarla a la práctica, también es importante conocer los estilos arquitectónicos, estos son una generalización de los patrones de diseño, como abstracción.

**Cita APA:**

Cristiá, M. (2008). Introducción a la Arquitectura de Software. *Research-Gate.[Online]. Recuperado de: https://www. researchgate. net/publication/251932352 Introduccion a la Arquitectura de Software*.

**Artículo 4:**

**Título:**

**Lenguajes de Patrones de Arquitectura de Software: Una Aproximación al Estado del Arte**

**Resumen:**

Mostrar el estado del arte en un área de la arquitectura de software llamada “Lenguajes de Patrones”, desde sus orígenes los avances actuales y sus aplicaciones en la construcción de arquitecturas de software en diferentes dominios de aplicación. La extensibilidad y aplicabilidad de los lenguajes de patrones, se convierten en una herramienta importante para diseñadores e implementadores de todo tipo de sistemas de información. Es importante evitar la programación al estilo vaquero. Parches de último minuto o trabajo de última noche que pueden traer graves consecuencias.

**Gráfica:**

**Reflexión:**

Para que un software sea bueno y tenga una funcionalidad correcta, es importante evitar hacer las cosas demasiado rápido, porque se pueden presentar especificaciones incompletas, lo cual puede resultar en un re-proceso, haciendo perder tiempo en la elaboración.

La creación de nuevas formas de hacer las cosas ayuda en la productividad y una mayor calidad en los productos de software.

**Cita APA:**

Jimenez-Torres, V. H., Tello-Borja, W., & Rios-Patiño, J. I. (2014). Lenguajes de patrones de arquitectura de software: una aproximación al estado del arte. *Scientia et technica*, *19*(4), 371-376.

**Artículo 5:**

**Título:**

**Integración de arquitectura de software en el ciclo de vida de las metodologías ágiles**

**Resumen:**

Las metodologías ágiles se centran en el trabajo en equipo, la adaptabilidad y colaboración dentro del grupo de software y también entre los miembros del grupo y los usuarios finales. El uso de las metodologías (MA), ha marcado una tendencia en su adopción al desarrollo de proyectos de software (AR), en tanto, es una manifestación de decisiones de etapas muy tempranas del diseño sobre un sistema.

El tratamiento con enfoques diferentes en las primeras etapas, ha sido uno de los factores que ha causado la sensación de las MA y la AS.

**Cita APA:**

Navarro, M.E., Moreno, M.P., Aranda, J., Parra, L., Rueda, J.R., & Pantano, J.C. (2017, September). Integración de arquitectura de software en el ciclo de vida de las metodologías ágiles. In XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires).

**Artículo 6:**

**Título:**

**Evaluación de una Arquitectura de Software**

**Resumen:**

Una arquitectura de software es clave para que las organizaciones puedan avanzar y concentrarse en su función misional, solo si esta está bien diseñada. Es importante realizar evaluaciones tempranas a la arquitectura, mediante algún método, porque eso ayuda a la elección de una arquitectura, mejora la comunicación y permite una mejor interpretación de las historias de usuario. También tener en cuenta el análisis para el atributo de calidad modificabilidad.

**Cita APA:**

Sanabria, E.R., & Rodríguez, S.V. (2021). Evaluación de una arquitectura de software. Prospectiva, 19(2).

**Artículo 7:**

**Título:**

**Arquitectura de software académica para la comprensión del desarrollo de software en capas**

**Resumen:**

El desarrollo de software implica considerar una cantidad variada de aspectos tecnológicos. Entre los más destacados podemos mencionar los relacionados con el acceso a datos, las interfaces, los procesos funcionales, el control de las transacciones, la accesibilidad y la seguridad. Lograr un diseño coherente con los requerimientos planteados, niveles aceptables de flexibilidad, extensibilidad y usabilidad, así como facilitar las actividades de mantenimiento lleva a pensar la concepción del software en capas.

**Cita APA:**

Cardacci, D. G. (2015). Arquitectura de software académica para la comprensión del desarrollo de software en capas. (No. 574). Serie Documentos de trabajo.

**Artículo 8:**

**Título:**

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

**Resumen:**

**Recopilar, comprender y gestionar los requisitos es un aspecto crítico en todos los métodos de desarrollo. Esto también es cierto para las metodologías ágiles en la que la captura de requisitos es realizada en todo el proceso de desarrollo, con requisitos que van evolucionando y cambiando a lo largo del ciclo de vida.**

**Este proceso es opuesto al enfoque de la arquitectura del software, donde los requerimientos deben ser identificados, recabados y comprendidos.**