



UNIVERSIDAD AUTÓNOMA DE CHIAPAS

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN CAMPUS 1

ACTIVIDAD. *Act.1.3 Definición de los Requerimientos de la Arquitectura*

ING. EN DESARROLLO Y TECNOLOGÍAS DE SOFTWARE

Materia: *Taller De Desarrollo 4*

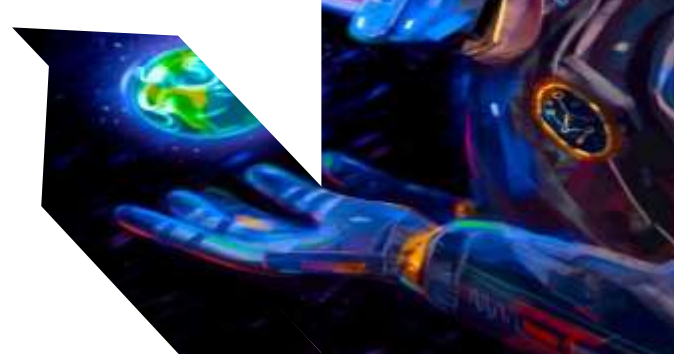
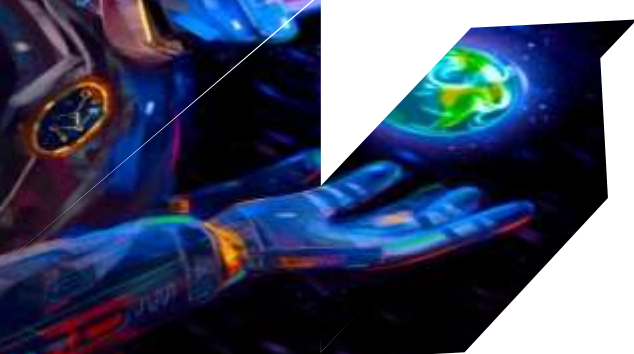
Grado y grupo: *6M*

Matricula: *A211120*

ALUMNO: *Gabriel Omar Fuentes Chacon.*

DOCENTE: *Mtro. Luis Gutiérrez Alfaro.*

Fecha de entrega: *23/08/2023*



## INDICE

Introducción: .....	3
DESARROLLO: .....	4
Definición de los Requerimientos de la Arquitectura .....	4
Requerimientos Funcionales .....	4
Requerimientos No Funcionales .....	6
Casos de Uso .....	7
Conclusión: .....	9
BIBLIOGRAFIA .....	10
ANEXOS .....	10

## Introducción:

Los requerimientos de la arquitectura son los cimientos que dan forma a todo el proceso de desarrollo del software, van más allá de ser simples directrices, son un conjunto de pautas que moldean cada etapa del diseño e implementación del sistema. Estos requerimientos delinean la interacción entre las diversas partes del sistema, cómo se manejan y almacenan los datos, y cómo el software se integra con sistemas actuales o futuros, estos requerimientos son esenciales para proporcionar una base sólida en la creación y organización de un sistema de software específico, su función principal radica en guiar a los desarrolladores en la construcción de una solución que satisfaga las necesidades del cliente y perdure a lo largo del tiempo, cumpliendo con estándares y restricciones técnicas existentes.

La importancia de los requerimientos de arquitectura trasciende la mera dirección del proceso, estos requerimientos son la base sobre la cual un sistema de software asegura su alineación con las expectativas del cliente, su sostenibilidad a largo plazo y su conformidad con los estándares y restricciones técnicas del entorno en el que opera, además dentro de los requerimientos funcionales, se encuentra la vital conexión entre los objetivos comerciales y la ejecución técnica, siendo los encargados de trazar la senda para el diseño y desarrollo del sistema, desde la funcionalidad del usuario, que da forma a la experiencia del usuario y establece la base de la interacción, hasta la integración con sistemas externos, que habilita la colaboración y el aprovechamiento de recursos externos, la usabilidad, la gestión de datos, el rendimiento y la seguridad son aspectos intrínsecos que se entrelazan en la trama de requerimientos funcionales, todos estos componentes aseguran que el sistema sea confiable en situaciones inesperadas, permiten la personalización para adaptarse a preferencias individuales y garantizan la integración fluida con otros sistemas y la interfaz del usuario, por otro lado, los requerimientos no funcionales son esenciales para diseñar sistemas capaces de enfrentar retos venideros, adaptarse a cambios y ofrecer una experiencia de usuario satisfactoria, la usabilidad, la escalabilidad y la seguridad se conjugan para definir el panorama en el que el software opera.

Los casos de uso, por su parte, van más allá de simples listas de funciones. Son ventanas a escenarios reales en los que usuarios y entidades interactúan. A través de ellos, se comunica cómo se espera que el software sea empleado y cómo responderá a las acciones de los usuarios. Desde la definición de requerimientos hasta la validación y pruebas, los casos de uso cumplen un rol crucial en el desarrollo de sistemas robustos y efectivos.

## DESARROLLO:

### Definición de los Requerimientos de la Arquitectura

Estos requerimientos no solo establecen el rumbo del proceso de desarrollo, sino que también son un conjunto de directrices que influyen en cada etapa del diseño y la implementación del software. Definen cómo las diferentes partes del sistema interactúan entre sí, cómo se gestionan y almacenan los datos, y cómo el sistema se integra con otros sistemas existentes o futuros.

Estos Requerimientos constituyen los elementos fundamentales que brindan dirección al diseño y la organización de un sistema de software en particular. Su función esencial es proporcionar una base sólida para que los desarrolladores puedan construir una solución que no solo satisfaga las necesidades del cliente, sino que también sea capaz de perdurar a lo largo del tiempo y cumpla con los estándares y limitaciones técnicas existentes.

Los requerimientos de arquitectura también son esenciales para garantizar que el software resultante sea seguro, eficiente y confiable. Establecen las pautas para la seguridad de los datos y la privacidad, así como para el rendimiento y la escalabilidad del sistema. Además, aseguran que el software pueda adaptarse y evolucionar con el tiempo, ya que la tecnología y las necesidades del negocio cambian constantemente.

Son la base sobre la cual se erige un sistema de software, garantizando su alineación con las expectativas del cliente, su viabilidad a largo plazo y su conformidad con los estándares técnicos y las restricciones impuestas por el entorno en el que opera.

### Requerimientos Funcionales

Si hablamos de los requerimientos funcionales en la arquitectura de software, nos referimos al proceso de diseño y desarrollo del sistema, al momento de poder establecer las acciones y habilidades que el software que este debe ofrecer, los requerimientos establecen una conexión entre las metas del negocio y la realidad técnica, asegurando que el sistema final funcione de manera efectiva, congruente y lógica.

Facilitan la planificación de pruebas y verificaciones específicas para asegurar que cada función se implemente de acuerdo al plan, a su vez, ayuda a identificar de manera temprana problemas o discrepancias entre el diseño y la ejecución, además al describir las interacciones tanto con los usuarios como con otros sistemas y componentes internos, estos requerimientos trazan una clara guía para cómo el sistema se comportará y cómo fluirá la información, siendo como un conjunto esencial de instrucciones que definen las tareas específicas que el sistema debe llevar a cabo para satisfacer los objetivos comerciales y las necesidades de los usuarios, en otras definiciones se centran en la creación de soluciones de software eficaces, por lo que asientan las bases para el diseño y desarrollo del sistema.

Los requerimientos funcionales abarcan distintos puntos para poder llevar a cabo su proceso de manera correcta uno de ellos es la funcionalidad del usuario que es esencial

para dar forma a la experiencia del usuario, un usuario pueda realizar en la interfaz del sistema está diseñada con un propósito específico, la creación de nuevos datos o la edición y eliminación de información, estas funcionalidades son la base de la interacción entre el usuario y el software, siendo un enfoque cuidadoso en estas funcionalidades, nos garantiza que la aplicación sea intuitiva, eficaz para los usuarios y tenga un buen uso y funcionalidad, otro aspecto a tomar en cuenta son los requerimientos de negocios donde estos requerimientos no solo describen los flujos de trabajo y operaciones del negocio, sino que también establecen cómo el sistema debe optimizar, automatizar o mejorar estos procesos, también contamos con el caso de uso que son escenarios prácticos que detallan cómo los usuarios interactuarán con el sistema para lograr objetivos concretos. Cada caso de uso describe una secuencia de acciones que un usuario podría realizar, desde el inicio hasta la conclusión del proceso. Estos casos permiten a los diseñadores y desarrolladores entender las necesidades del usuario desde una perspectiva práctica, lo que ayuda a construir un sistema que responda de manera efectiva a las necesidades reales.

Algunos de otros puntos a tomar en cuenta para llevar a cabo los requerimientos funcionales son los siguientes como el caso de la interfaz para nuestro cliente en este caso el usuario: Estos requerimientos abordan no solo la apariencia visual de la interfaz, sino también su funcionalidad. La disposición de los elementos, la navegación entre pantallas y la facilidad de uso son aspectos clave. La interfaz debe ser intuitiva, fácil de aprender y atractiva para los usuarios.

- **Gestión de Datos:** La validación de datos ayuda a prevenir la entrada incorrecta o maliciosa, mientras que la estructura de la base de datos influye en cómo se relacionan los datos entre sí, es un enfoque sólido en la gestión de datos garantiza la confiabilidad y precisión de la información.
- **Rendimiento:** Es esencial para crear un sistema que funcione de manera eficiente y responda rápidamente, incluso en situaciones de alta carga, contiene respuesta y resultados esperados y las capacidades de procesamiento se definen para asegurar que el sistema brinde una experiencia fluida para los usuarios, sin retrasos frustrantes.
- **Integración con Sistemas Externos:** Estos requerimientos son vitales en un mundo interconectado. La forma en que el sistema interactúa con otros sistemas a través de interfaces como APIs y servicios web impacta directamente en su capacidad para aprovechar recursos externos, compartir datos y brindar una experiencia de usuario coherente en un entorno digital diverso.
- **Seguridad y Autenticación:** Estos requerimientos son críticos para proteger tanto los datos confidenciales como la integridad del sistema. La autenticación asegura que solo usuarios autorizados accedan a la aplicación, mientras que la autorización controla las acciones que pueden realizar. Implementar medidas de seguridad sólidas es esencial para prevenir ataques y garantizar la privacidad de los datos.



Estos requerimientos se enfocan en cómo el sistema debe manejar situaciones inesperadas y errores, en forma en que el software responde a situaciones problemáticas puede afectar significativamente la experiencia del usuario, la personalización, de estos requerimientos permiten a los usuarios adaptar la aplicación a sus preferencias individuales, pudiendo incluir la configuración de colores, la disposición de elementos en la interfaz o incluso la adaptación de flujos de trabajo según las necesidades del usuario, nos proporcionar opciones de personalización puede mejorar la satisfacción del usuario y hacer que la aplicación se sienta más adaptada a sus necesidades específicas.

## Requerimientos No Funcionales

Los aspectos esenciales de un sistema de software que delinean su conducta en términos de atributos no directamente atados a funciones particulares son conocidos como requisitos no funcionales, la arquitectura del software, influida por estos requisitos, se convierte en el cimiento sobre el cual se erige el sistema completo, al cumplir con los requisitos no funcionales, el software se vuelve más apto para enfrentar desafíos futuros, adaptarse a nuevas circunstancias y proporcionar una experiencia de usuario satisfactoria, además tener una identificación precisa de estos requisitos implica un proceso exhaustivo de comprensión de las expectativas de los usuarios y las necesidades del sistema, cada que se identifica estos requisitos, se documentan minuciosamente para establecer criterios de evaluación claros, esta documentación lleva un proceso de no solo actuar como guía para los desarrolladores y arquitectos, sino que también sirve como punto de referencia para asegurar que el software evoluciona de acuerdo con las demandas cambiantes, estos aspectos incluyen la escalabilidad del sistema para afrontar cargas crecientes, la facilidad de mantenimiento a lo largo del tiempo y la adaptabilidad a diferentes entornos y dispositivos.

En este caso se cuenta con una serie de requisitos no funcionales que desempeñan un papel fundamental para asegurar su eficacia y utilidad en diversos escenarios, como lo es rendimiento, un requisito con el que establece la forma en que el sistema debe comportarse en términos de velocidad y capacidad para manejar cargas específicas, esto puede involucrar en criterios como la velocidad máxima de procesamiento, el tiempo necesario para cargar una página web o la cantidad de transacciones ejecutadas por segundo, adentrándonos en el siguiente punto que viene siendo la usabilidad que se refiere a la facilidad con la cual los usuarios pueden interactuar con el sistema de diseño de la interfaz de usuario, accesibilidad y facilidad de aprendizaje, por lo tanto surge la eficiencia que no es más que la optimización de los recursos utilizados por el sistema, como consumo de energía, uso de memoria y capacidad de procesamiento, la seguridad juega un papel muy importante por lo que es esencial para los mecanismos y protocolos necesarios para

resguardar los datos, garantizar adecuada autenticación y autorización, y prevenir intrusiones o accesos no autorizados al sistema, si consideramos integración de estos requisitos no funcionales garantiza que la arquitectura del software cumpla con las expectativas y necesidades cambiantes del sistema y sus usuarios, generando así un sistema sólido y eficaz, surgiendo requisitos como los que son:

- Soporte técnico: Explica los niveles de asistencia técnica que se deben proporcionar a los usuarios, incluyendo documentación, ayuda a la resolución de problemas.
- Escalabilidad: El sistema debe expandirse o contraerse de manera eficiente, en términos de recursos como usuarios, transacciones y almacenamiento, sin que el rendimiento se degrade significativamente.
- Mantenibilidad: Indica cómo el sistema debe ser diseñado para facilitar cambios y actualizaciones futuras, se considera claridad del código, modularidad y documentación adecuada.
- Confiabilidad: Puede involucrar requisitos de tolerancia a fallos y procedimientos de recuperación en caso de desastres, indica la probabilidad de que el sistema funcione sin fallos críticos durante un período específico.
- Cumplimiento normativo: estos requisitos deben ser precisamente establecidos si el sistema debe adherirse a regulaciones específicas o estándares de la industria.

## Casos de Uso

Estos casos de uso trascienden las meras listas de funciones; más bien, ilustran cómo usuarios reales o entidades interaccionan en escenarios reales, estas descripciones detalladas ofrecen una perspectiva clara sobre cómo diversos actores, como usuarios, sistemas externos y dispositivos, se relacionan con el software y cómo este último reacciona ante sus acciones, cada caso de uso sigue un flujo de eventos principal que detalla las acciones ejecutadas por los usuarios y las respuestas generadas por el sistema en consecuencia. Además, tienen la flexibilidad de incluir rutas alternas que describen situaciones inusuales o desvíos en la interacción, la arquitectura del software, los casos de uso representan situaciones específicas en las que los usuarios interactúan con el sistema para alcanzar metas particulares, por lo que contribuye con algunos requerimientos como la definición de requerimientos, esta contribuyen con la identificación y el registro de los requisitos funcionales del sistema desde la perspectiva del usuario, estableciendo así una base sólida para la formulación y desarrollo del software, las pruebas se convierten en la base para la formulación de casos de prueba y escenarios de evaluación que aseguran si el sistema se desenvuelve como se espera, brindando una manera sistemática de probar la funcionalidad.

Los casos de uso desempeñan un papel altamente efectivo en la comunicación, al transmitir cómo se prevé que el software sea empleado y cuál será su comportamiento a

desarrolladores, diseñadores y partes interesadas todo esto brindan una comprensión clara y compartida de las funciones anticipadas, además cuenta con puntos como la validación si bien en los caso de uso permiten que las partes interesadas en validar que los requerimientos y funcionalidades planeadas realmente abarcan sus necesidades y objetivos, generando confianza en la dirección del proyecto, llevándonos a la planificación de la arquitectura en el software los arquitectos al concebir cómo se estructurará el sistema y cómo sus componentes serán organizados para cumplir con los casos de uso establecidos.

Los casos de uso se integran en la documentación del sistema, facilitando su comprensión y permitiendo su uso continuo a lo largo del tiempo, lo cual es esencial para el mantenimiento y evolución del software.



## Conclusión:

En conclusión, en el amplio panorama del desarrollo de software, los requerimientos de la arquitectura desempeñan un papel fundamental como los cimientos que soportan cada fase del proceso, al delinear las interacciones entre diversas partes del software, la gestión de datos y la integración con sistemas actuales o futuros, estos requerimientos trascienden la planificación, además de influir en la dirección del proceso, estos requerimientos son la clave para asegurar la seguridad, eficiencia y confiabilidad del software resultante, los requerimientos funcionales toman la delantera al forjar el puente entre los objetivos empresariales y la ejecución técnica, los requerimientos no funcionales, por otro lado, garantizan que los sistemas puedan enfrentar los desafíos que tendrán en un futuro, la concepción de requerimientos hasta la validación y las pruebas, los casos de uso actúan como el hilo conductor en el tejido del desarrollo de sistemas robustos y efectivos, al final del día, los requerimientos de arquitectura y los casos de uso conforman un sólido andamiaje que permite a los sistemas de software no solo cumplir con las expectativas actuales, sino también enfrentar el futuro con confianza y adaptabilidad.

## BIBLIOGRAFIA

Fernández, L. F. (2006). Arquitectura de software. *Software Guru*, 2(3), 40-45.

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*.

Sommerville, I. (2005). Requerimientos del software. *Ingeniería del software*, 7a ed., PEARSON EDUCACIÓN, Madrid, SPA, 109-110.

Chaves, M. A. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes: Revista de las Sedes Regionales*, 6(10), 1-13.

Soto, S. D., Rivero, L. C., & Olguín, E. L. (2019). El uso de software libre en el control de inventarios: caso de estudio. *Ciencia Administrativa*, 12(1), 2-7.

## ANEXOS

