



El futuro digital
es de todos

MinTIC

Universidad
Industrial de
Santander



Misión
TIC 2022



INTRODUCCIÓN INGENIERÍA SOFTWARE

1. Introducción ingeniería software

1.1. Ciclo de vida del software

Un ciclo de vida de desarrollo de software (SDLC) incluye los procesos de software utilizados para especificar y transformar los requisitos de software en un producto de software entregable.

Está dividido en 4 categorías como se muestra a continuación:

Procesos primarios

Incluye los procesos para análisis, diseño, desarrollo, operación y mantenimiento del Software

Procesos de soporte

Son aplicados de forma intermitente o continua a lo largo del ciclo de vida de un producto software para respaldar los procesos primarios; incluyendo procesos de gestión de la configuración, aseguramiento de calidad, verificación y validación.

Procesos organizacionales

Proporcionan soporte en el proceso de ingeniería de software incluyendo, capacitación, análisis de medición de proceso, gestión de infraestructura, reutilización, mejora de procesos organizacionales y gestión de modelos de ciclo de vida de software.

Procesos de proyectos cruzados

Como reutilización, línea de productos de software; pueden implicar más de un proyecto de software en una organización.

Proceso de ingeniería de software, Fuente: adaptado de [1]

1.2. Modelos de desarrollo de Software

La naturaleza intangible y maleable del software permite una amplia variedad de modelos de ciclo de vida de desarrollo de software, que van desde modelos lineales, en los que las fases del desarrollo de software, se logran secuencialmente con retroalimentación e iteración según sea necesario, seguidas de integración, prueba y entrega de un producto único; a modelos iterativos en los que el software se desarrolla en incrementos de funcionalidad creciente en ciclos iterativos; a modelos ágiles que normalmente implican demostraciones frecuentes de software en funcionamiento a un cliente o representante del usuario que dirige el desarrollo del software en ciclos iterativos cortos que producen pequeños incrementos de software que funciona y que se puede entregar. Los modelos incrementales, iterativos y ágiles, pueden ofrecer subconjuntos tempranos de software de trabajo

Los modelos de SDLC lineales a veces se denominan modelos de ciclo de vida de desarrollo de software predictivo, mientras que los SDLC iterativos y ágiles, se denominan modelos de ciclo de vida de desarrollo de software adaptativo. Cabe señalar, que se pueden realizar diversas actividades de mantenimiento durante un ciclo de vida de producto software (SPLC), utilizando diferentes modelos de SDLC, según corresponda.

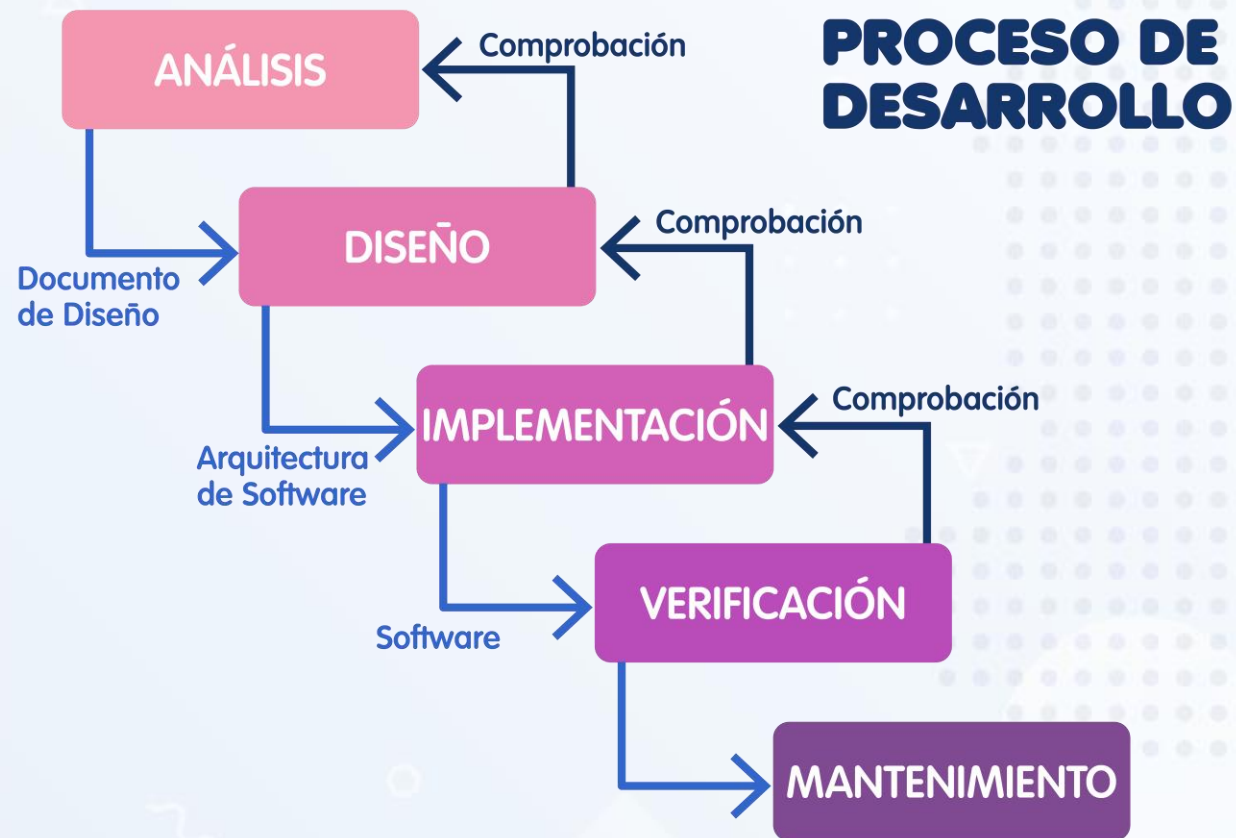
Modelos Básicos de ciclos de vida del software

Ciclo en V



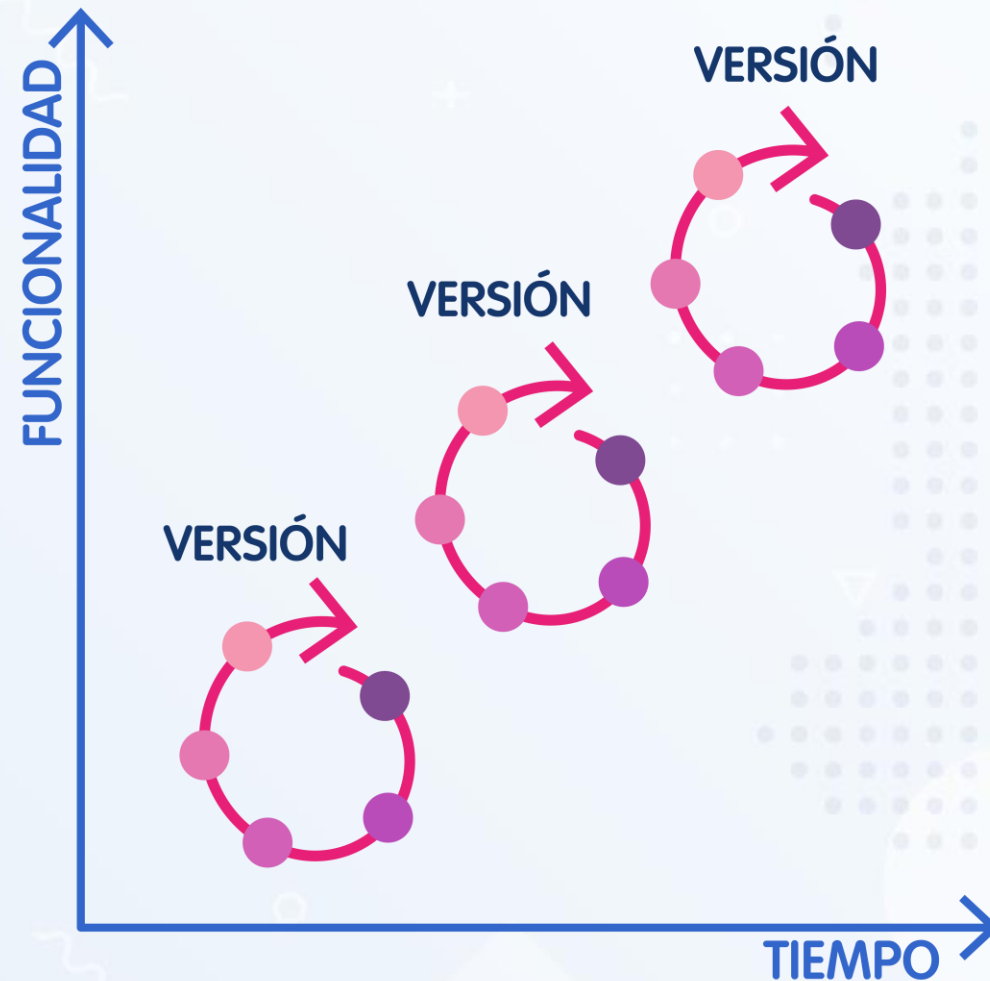
Basado de <http://scrum-qa.blogspot.com/2013/03/modelos-de-desarrollo-de-software.html>

Ciclo en cascada



Recuperado de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>

Ciclo iterativo



Recuperado de: <https://josepablosarco.wordpress.com/2012/03/24/istqb-cap-2-testing-a-traves-del-ciclo-de-vida-del-software-i/>

1.3. Elicitación y análisis de requerimientos

Requerimiento:

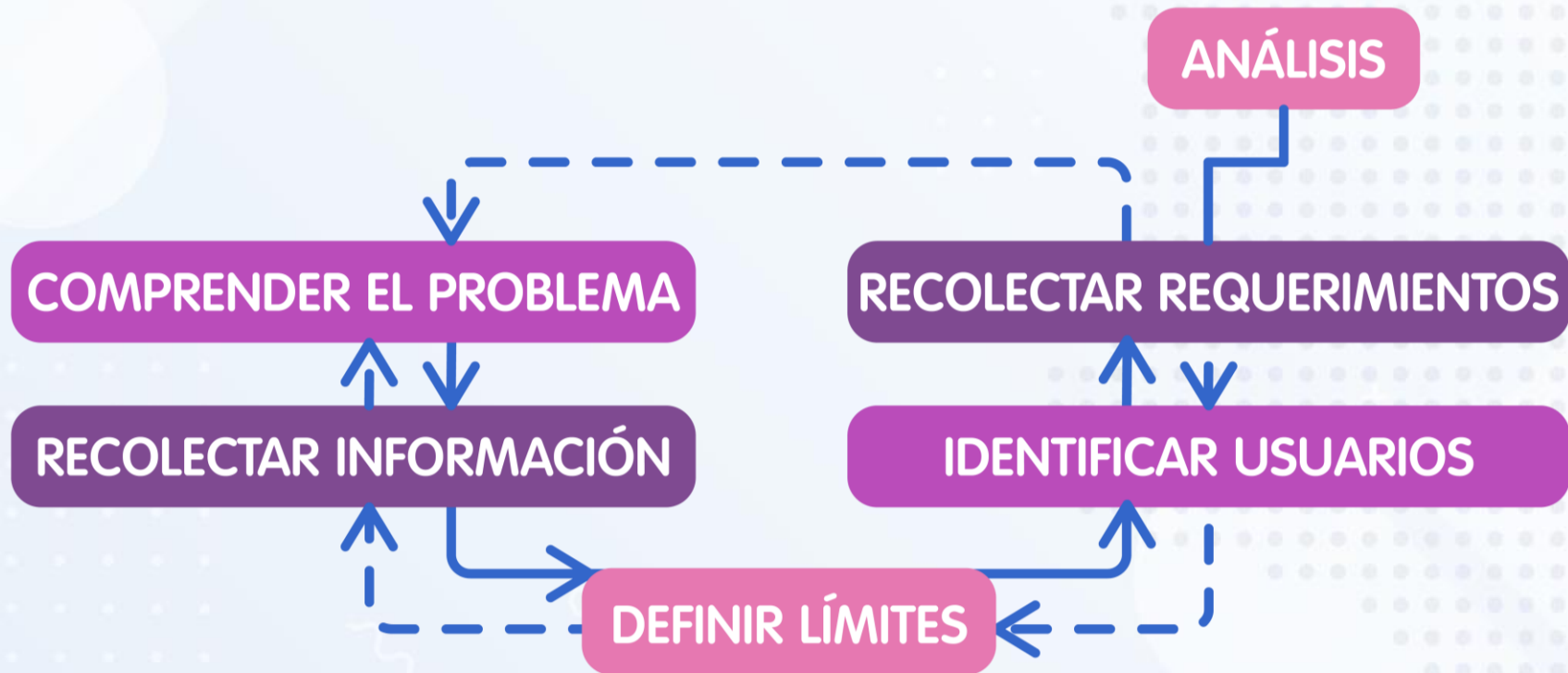
- Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. [Std 610.12-1900, IEEE: 62]
- Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal. [Std 610.12-1900, IEEE: 62].

La elicitación de requerimientos, es el proceso de descubrir los requerimientos para un sistema a través de la comunicación con los clientes, usuarios del sistema y otras personas que tengan algún tipo de interés y conocimiento sobre el producto a desarrollar [Madigan].

El principal resultado de este proceso, son los requerimientos a usar por el equipo de desarrollo de software para crear de forma apropiada un producto. Esto ayuda a los usuarios a entender qué es lo que quieren, qué es lo que necesitan, cuáles son las restricciones y alternativas, ventajas y desventajas de cada una.

Desde el punto de vista de los ingenieros y desarrolladores, la existencia de un buen proceso de elicitación de requerimientos, ayuda a los mismos a resolver el problema correcto, es decir, lo que realmente desea y necesita el cliente, a resolver un problema factible, a ganar la confianza del cliente y su cooperación y a ganar conocimiento sobre el dominio del problema [Tuffley, 2005].

Proceso de definición de requerimientos



1.3.1. Comprensión del problema

Para una correcta identificación de requerimientos, es necesario comprender el problema y cómo éste se relaciona con el entorno.

En esta etapa se responden a las preguntas: ¿qué se quiere hacer?, ¿Qué problema se quiere resolver?, ¿Para que se quiere hacer el sistema?

Sin embargo a menudo se encuentran inconvenientes:

- El cliente no siempre define claramente el problema.
- El analista de requerimientos y los desarrolladores no comprenden la naturaleza del problema
- El analista y los desarrolladores entienden el problema pero no saben cómo llevarlo a cabo.
- El problema es muy amplio, vago, poco factible, o muy volátil.

La Comprensión del dominio de la aplicación se lleva a cabo desde 7 perspectivas:

1. Ambiente operacional. Permite definir el ambiente sobre el cual el sistema estará operando y todos sus componentes.
2. Sistemas de hardware. Estos sistemas comprenden, los sistemas de cómputo, las redes utilizadas y sus protocolos, así como cualquiera de los otros sistemas eléctricos y mecánicos.
3. Sistemas de Software. Estos sistemas comprenden los sistemas operativos, bases de datos, lenguajes, sistemas de manejo de archivos, software de aplicación, sistemas de seguridad, entre otros.
4. Interfaces Hombre-Máquina. Estos sistemas son aquellos con los que los usuarios tendrán contacto directo para llevar a cabo sus labores.
5. Conexiones externas. Estos sistemas son aquellos que provienen del exterior del sistema y que reciben datos del sistema o a quienes el sistema envía datos.
6. Procedimientos operacionales. Estos procedimientos definen las funciones que realiza el sistema actual.
7. Capacidad del Sistema Actual. Este aspecto permite identificar cual es la capacidad de procesamiento y de almacenamiento requeridos por el sistema.

1.3.2. Fuentes potenciales y las formas de obtención de requerimientos.

- Entrevistas
- Documentos que describen sistemas actuales.
- Cuestionarios de usuarios.
- Observación de los usuarios futuros y de su medio ambiente.





1.3.3. Identificación de interesados del sistema

Los interesados deben clasificarse de acuerdo a su actividad y a su perfil.

- Clientes
- Usuarios
- Ingenieros de desarrollo de software
- Ingenieros del cliente
- Administradores o jefes de proyecto de software
- Contratistas externos
- Reguladores externos

1.3.4. Clasificación de requerimientos



Requerimientos Funcionales

Los requerimientos funcionales hacen referencia a la descripción de las actividades y servicios que un sistema debe proveer. Normalmente, este tipo de requerimientos están vinculados con las entradas, las salidas de los procesos y los datos a almacenar en el sistema.

Requerimientos No Funcionales

Describen aspectos del sistema que son visibles por el usuario que no incluyen una relación directa con el comportamiento funcional del sistema; en cambio, incluyen restricciones como el tiempo de respuesta(desempeño), la precisión, recursos consumidos, seguridad, etc.

Ejemplo de requerimientos

Requerimientos de negocio

- Incrementa el porcentaje del mercado en 30 %.
- Ahorra 20% en costos de producción por la automatización instalada.
- Ahorra 40% en costos de mantenimiento

Casos de uso y escenarios

- Yo necesito imprimir una etiqueta de correo para el paquete.
- Yo necesito administrar una cola de reactivos químicos que esperan ser analizados.
- Yo necesito calibrar las máquinas para el control numérico.

Reglas del negocio

- Debe de seguir el estándar de acuerdo con alguna ley o política de la organización.
- El formato del saldo de contabilidad está definido en los manuales de la organización.

Requerimientos funcionales

- Si el voltaje rebasa los 20 v enciende la alarma amarilla.
- El sistema envía un e-mail de confirmación cuando recibe cualquier e-mail.
- El sistema debe ordenar los productos del inventario en orden alfabético.

Requerimientos de interfaces externas

- Las señales de voltaje se leen de los convertidores analógico-digital.
- Los mensajes se envían a través de Internet.
- El software debe controlar el tablero de diagramas eléctricos.
- Los archivos recibidos electrónicamente deben leerse del disco externo
- El usuario debe poder ver páginas web amigables.

Ejemplos de Requerimientos de Calidad

Propiedad	Medida
Velocidad	Transacciones por segundo Tiempo de respuesta a eventos
Tamaño	Número de líneas de código Número de Bytes de Memoria disponible
Facilidad de uso	Tiempo de entrenamiento Número de ayudas
Confiabilidad	Errores permitidos por unidad de tiempo Media de tiempo por fallo Disponibilidad en tiempo
Robustez	Tiempo para restablecer después del fallo. Porcentaje de fallos que causan caída
Portabilidad	Facilidad de transportar a otro S.O o lenguaje.

¿Cómo documentar un requerimiento?

Requerimiento #XXX:	Contabilización de las personas
Descripción:	El programa debe ser capaz de contabilizar las personas que van a asistir a la novena
Responsable:	Organizador
Entradas:	Código personas y un contador
Resultado:	El programa analiza el número de personas que iniciaron la novena y suma 1 al contador.

Documento de Especificación de Requisitos del Sistema (ERS)

Es un documento que define de forma completa, precisa y verificable, los requisitos, el diseño, el comportamiento y otras características de un sistema (IEEE-830):

- No Ambigua. Debe tener una única interpretación.
- Completa. Debe incluir todos los requisitos significativos (Funcionalidad, Ejecución, Diseño, Calidad, Interfaz).
- Fácil De Verificar. Toda la información plasmada en ella se puede verificar.
- Consistente. No deben existir requerimientos contradictorios.
- Fácil de modificar. Debe contar con una estructura y estilo bien definidos. Organización coherente y manejable (Tabla de contenido, Índices, Identificar figuras, etc.).
- Facilidad Para Identificar Origen y Consecuencias De Cada Requisito. Una buena ERS, permite identificar el responsable, origen y destino de cada requisito.
- Facilidad de Utilización. Debe suministrar facilidades que permitan actualizarla en el futuro.

[Ejemplo plantilla ERS](#)

[EEE-STD-830-1998 : ESPECIFICACIONES DE LOS REQUISITOS DEL SOFTWARE](#) Más información de requerimientos: [Aquí](#)

1.4 Arquitecturas de software

Arquitectura es la organización fundamental de un sistema descrita en:

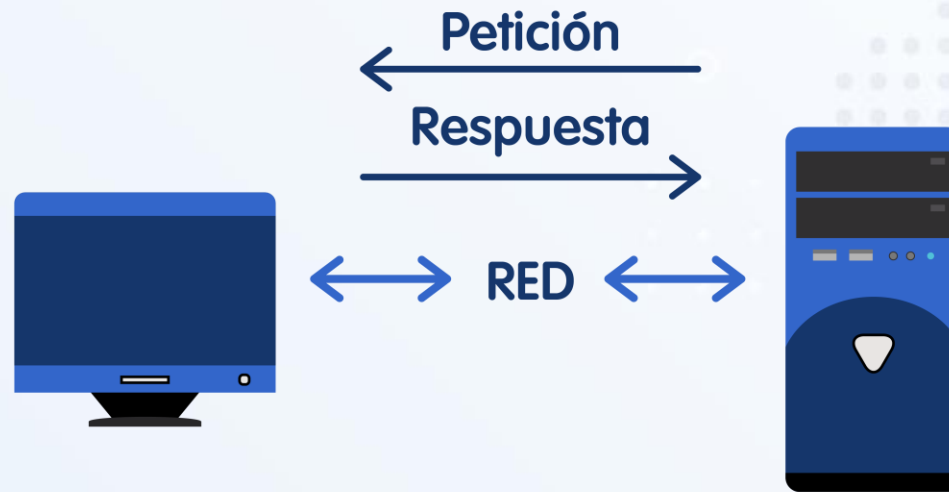
- Sus componentes
- Relación entre ellos y con el ambiente
- Principios que guían su diseño y evolución

La arquitectura software trata el diseño e implementación de la estructura de alto nivel del software. Es el resultado de ensamblar un cierto número de elementos arquitectónicos para satisfacer la funcionalidad y ejecución de los requisitos del sistema; así como, los requisitos no funcionales del mismo: fiabilidad, escalabilidad, portabilidad, disponibilidad, etc. Perry y Wolf (1992) describen una arquitectura software como:

Arquitectura Software = {Elementos, Formas, Fundamento/Restricciones}

Establece los fundamentos para que los analistas, diseñadores, programadores, entre otros, trabajen en una línea común que permita alcanzar los objetivos del sistema.

Arquitectura básica: Cliente/Servidor



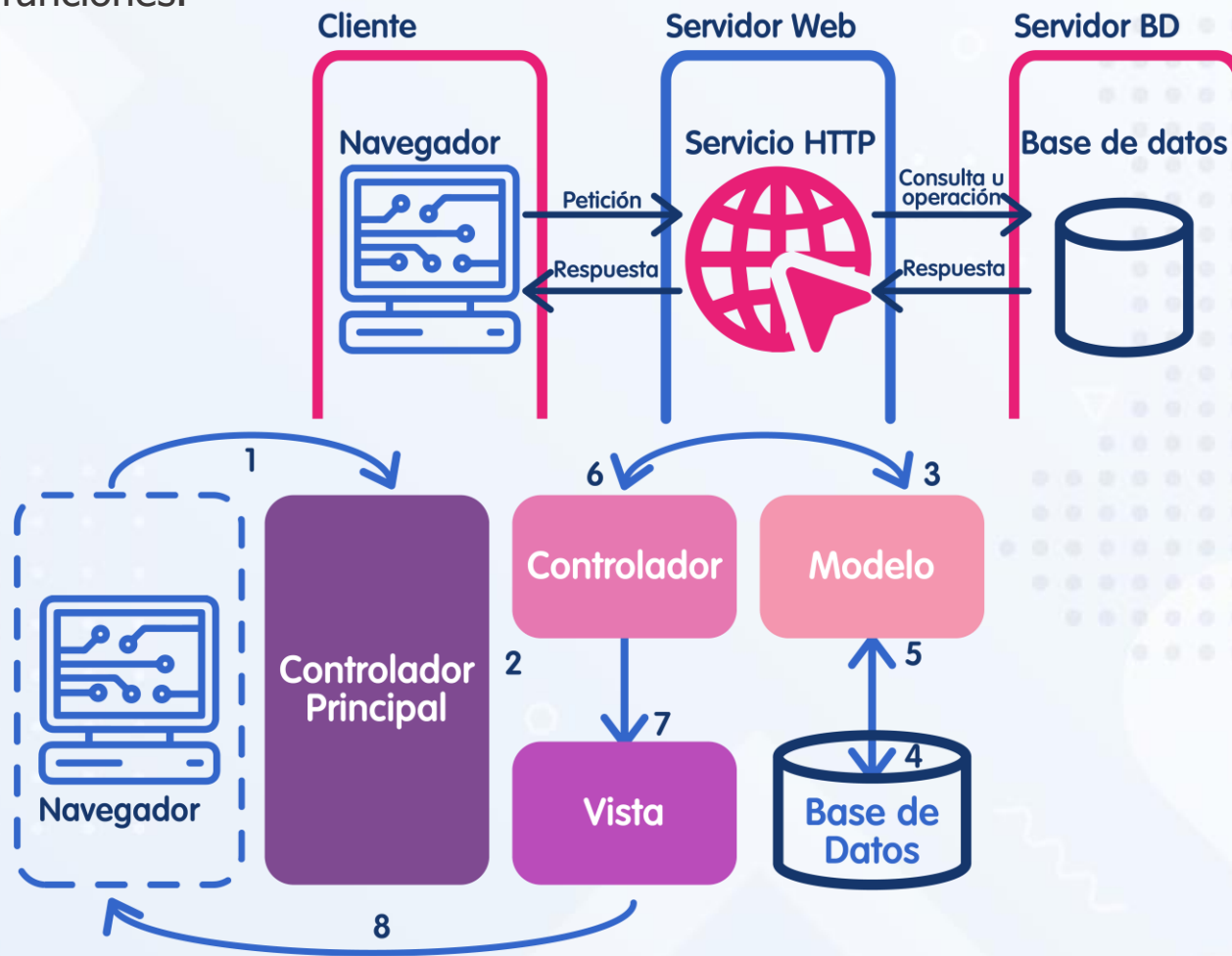
LIVIANO: Lógica de la Aplicación solamente del lado del servidor

PESADO: Lógica de la Aplicación parcial o totalmente del lado del servidor

Cliente "LIVIANO" vs Cliente "PESADO"

Arquitectura de tres capas

Es la especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (capas) con un reparto claro de funciones.



Rol del Arquitecto de Software

En la actualidad, la palabra “arquitectura” muchos la asocian al término de construir y diseñar edificios, pero en realidad esta palabra sobresale en varias ramas, no solo en la de los dibujos, sino que a la vez, en las ramas económicas y empresariales; sin mencionar, en la rama que principalmente está enfocado el siguiente artículo, la cual es la de “Informática”. El auge que está tomando este tipo de definición en el campo de las tecnologías de información, es bastante alta, ya que facilita y comprende no solo la parte de la administración de la solución dentro del negocio, sino que a la vez, tiene una comunicación certera y clara entre los demás integrantes que participan dentro del mismo.

Como AWS y Blockchain posibilitan vencer los desafíos de la interoperabilidad en la Salud

<https://aws.amazon.com/es/blogs/aws-spanish/como-aws-y-blockchain-posibilitan-vencer-los-desafios-de-la-interoperabilidad-en-la-salud>