

Glosario Técnico de Ingeniería de Software

Equipo: EcoEscolar

Ariel Alejandro Mérida Barrera
Angel Miguel Corona Aguilar
Héctor Rafael Mejía Pliego
Javier Eloy Gutiérrez Morales

1. Conceptos Generales de Ingeniería de Software

Ingeniería de Software

Definición: Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software. No se trata solo de escribir código, sino de gestionar todo el proceso productivo para crear sistemas fiables y eficientes que resuelvan problemas reales.

Ejemplo breve: Aplicar principios de ingeniería para diseñar un sistema bancario que sea seguro y escalable, en lugar de programarlo improvisadamente.

Abstracción

Definición: Capacidad de manejar la complejidad de un sistema ignorando los detalles menos relevantes y centrándose en las características esenciales. Permite modelar problemas del mundo real en componentes de software manejables sin perderse en la implementación técnica inmediata.

Ejemplo breve: Un conductor usa el volante y los pedales (abstracción) sin necesidad de entender la mecánica de inyección de combustible del motor.

Modularidad

Definición: Propiedad que permite dividir un sistema complejo en partes más pequeñas e independientes llamadas módulos. Esto facilita el desarrollo en equipo, la localización de errores y la reutilización de código, ya que cada módulo tiene una responsabilidad específica.

Ejemplo breve: En una tienda online, separar el módulo de "Carrito de Compras" del módulo de "Facturación", permitiendo modificar uno sin romper el otro.

Encapsulamiento

Definición: Técnica que consiste en ocultar los detalles internos de funcionamiento de un módulo o clase, exponiendo solo lo necesario a través de una interfaz pública. Protege la integridad de los datos al impedir accesos externos no autorizados o modificaciones directas

Ejemplo breve: Un objeto "CuentaBancaria" permite depositar o retirar dinero mediante métodos, pero impide modificar directamente la variable saldo desde fuera.

Cohesión

Definición: Medida de la fuerza con la que se relacionan las responsabilidades dentro de un mismo módulo. Una alta cohesión es deseable porque indica que el módulo se enfoca en una única tarea bien definida, lo que lo hace más robusto y fácil de entender.

Ejemplo breve: Una función calcularImpuestos() tiene alta cohesión si solo calcula impuestos, y baja cohesión si además envía correos y actualiza la base de datos.

Acoplamiento

Definición: Grado de interdependencia entre distintos módulos de software. Se busca un bajo acoplamiento para que los cambios en un módulo no afecten en cadena a otros, favoreciendo la mantenibilidad y la flexibilidad del sistema.

Ejemplo breve: Si cambiar el diseño de la base de datos obliga a reescribir la interfaz de usuario, existe un alto acoplamiento indeseado entre ambas capas.

Arquitectura de Software

Definición: Es la estructura fundamental de un sistema de software, que incluye los elementos de software, sus relaciones y las propiedades de ambos. Define las decisiones de diseño de alto nivel que son difíciles de cambiar una vez implementadas y actúan como esqueleto del sistema.

Ejemplo breve: Decidir si una aplicación será "Cliente-Servidor" o "Peer-to-Peer" (P2P) es una decisión arquitectónica clave.

Patrón de Diseño

Definición: Solución general y reutilizable a un problema común que ocurre frecuentemente dentro de un contexto de diseño de software. No es código terminado, sino una plantilla o guía sobre cómo resolver un problema arquitectónico o lógico de manera eficiente.

Ejemplo breve: Usar el patrón "Singleton" para asegurar que solo exista una única instancia de la conexión a la base de datos en todo el programa.

Sistema Legado (Legacy System)

Definición: Sistema informático antiguo que sigue siendo esencial para la organización pero que utiliza tecnologías obsoletas. Su mantenimiento es complejo y costoso, pero reemplazarlo conlleva un alto riesgo debido a su importancia crítica.

Ejemplo breve: Un sistema bancario escrito en COBOL hace 30 años que gestiona las cuentas de ahorro y no puede apagarse fácilmente.

Interoperabilidad

Definición: Habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada. Es fundamental en entornos modernos donde diferentes aplicaciones deben trabajar juntas sin conflictos.

Ejemplo breve: Que un sistema de facturación (SAP) pueda enviar datos automáticamente al sistema de logística (Oracle) sin errores de formato.

Middleware

Definición: Software que actúa como un puente entre un sistema operativo o base de datos y las aplicaciones, especialmente en una red. Facilita la comunicación y gestión de datos entre aplicaciones distribuidas que de otro modo no podrían conectarse.

Ejemplo breve: Un servidor de mensajería (como RabbitMQ) que permite que una app web envíe tareas a un servidor de procesamiento en otro lenguaje.

2. Fases del Ciclo de Vida (SDLC)

Análisis de Requisitos

Definición: Fase crítica donde se recopilan, examinan y documentan las necesidades de los usuarios y las restricciones del sistema. El objetivo es comprender exactamente qué debe hacer el software (el "qué") antes de decidir cómo construirlo.

Ejemplo breve: Reunirse con los contadores para definir exactamente qué fórmulas debe aplicar el sistema al calcular la nómina mensual.

Diseño de Software

Definición: Etapa donde se define la arquitectura del sistema, las estructuras de datos, las interfaces y los algoritmos. Transforma los requisitos en un plano técnico o "blueprint" que guiará a los programadores durante la codificación.

Ejemplo breve: Crear un diagrama de clases UML y definir que se usará una arquitectura de microservicios antes de escribir una sola línea de código.

Codificación (Implementación)

Definición: Proceso de traducir el diseño técnico a un lenguaje de programación ejecutable por la computadora. Es la fase de construcción real donde se crean los componentes, librerías y ejecutables del software.

Ejemplo breve: Un programador escribe scripts en Python o Java siguiendo las especificaciones definidas en la fase de diseño.

Pruebas (Testing)

Definición: Proceso de ejecutar el software con la intención de encontrar fallos. Verifica que el sistema cumpla con los requisitos especificados y funcione correctamente bajo diversas condiciones antes de ser entregado al usuario final.

Ejemplo breve: Intentar ingresar una fecha de nacimiento futura en un formulario para asegurar que el sistema rechaza el dato inválido.

Mantenimiento

Definición: Fase más larga del ciclo de vida que ocurre después de la entrega del software. Incluye la corrección de errores no detectados, la adaptación a nuevos entornos (ej. cambios de SO) y la mejora de funcionalidades existentes.

Ejemplo breve: Actualizar una app móvil para que sea compatible con la nueva versión de iOS o para añadir un "modo oscuro".

Estudio de Viabilidad

Definición: Evaluación preliminar realizada al inicio del proyecto para determinar si es posible desarrollarlo técnica, operativa y económicamente. Ayuda a decidir si vale la pena invertir recursos antes de comprometerse con el desarrollo completo.

Ejemplo breve: Analizar si existe tecnología actual para crear un algoritmo de reconocimiento facial con el presupuesto de una startup pequeña.

Prototipado

Definición: Creación de una versión preliminar e incompleta del software para explorar ideas y obtener retroalimentación temprana del usuario. Permite visualizar requisitos y funcionalidades antes de la construcción final, reduciendo riesgos de malentendidos.

Ejemplo breve: Crear pantallas interactivas pero sin lógica real (falsas) para que el cliente confirme si el flujo de navegación es el deseado.

Pruebas Unitarias

Definición: Nivel de prueba de software donde se examinan las unidades más pequeñas comprobables de una aplicación (como funciones o métodos) de forma aislada para asegurar que funcionan correctamente según lo diseñado.

Ejemplo breve: Probar solo la función sumar(a, b) para verificar que 2+2 devuelve 4, sin ejecutar el resto del programa.

Pruebas de Integración

Definición: Fase del testing donde se combinan los módulos de software individuales y se prueban como grupo. El objetivo es detectar fallos en la interacción o interfaces entre componentes que funcionaban bien por separado.

Ejemplo breve: Verificar si el módulo de "Login" envía correctamente el ID del usuario al módulo de "Perfil" tras iniciar sesión.

Despliegue (Deployment)

Definición: Proceso de trasladar el software desde un entorno de desarrollo a un entorno de producción donde los usuarios finales pueden utilizarlo. Incluye instalación, configuración y activación del sistema.

Ejemplo breve: Subir los archivos de una página web a un servidor público (hosting) y configurar el dominio para que sea accesible en internet.

3. Metodología de Desarrollo

Metodología Ágil

Definición: Enfoque de desarrollo iterativo e incremental que prioriza la flexibilidad, la entrega rápida de software funcional y la colaboración constante con el cliente sobre la documentación extensa y los procesos rígidos.

Ejemplo breve: Entregar una versión básica de una app en dos semanas y mejorarla con el feedback del usuario, en lugar de esperar seis meses para lanzar todo junto.

Scrum

Definición: Marco de trabajo ágil utilizado para gestionar el desarrollo de productos complejos. Se basa en ciclos cortos de trabajo llamados "Sprints" (generalmente de 2 a 4 semanas), reuniones diarias y roles definidos (Scrum Master, Product Owner).

Ejemplo breve: El equipo realiza una reunión de 15 minutos cada mañana para sincronizar qué hicieron ayer y qué harán hoy para cumplir la meta del Sprint.

Modelo en Cascada (Waterfall)

Definición: Metodología tradicional lineal donde cada fase del ciclo de vida (análisis, diseño, código, pruebas) debe completarse totalmente antes de pasar a la siguiente. Es rígida y dificulta los cambios una vez iniciado el proyecto.

Ejemplo breve: No se empieza a programar hasta que el documento de requisitos de 200 páginas ha sido firmado y aprobado por el cliente.

DevOps

Definición: Más que una metodología, es una cultura y conjunto de prácticas que une el desarrollo de software (Dev) con las operaciones de TI (Ops). Busca acortar el ciclo de desarrollo y entregar software de alta calidad de forma continua mediante la automatización.

Ejemplo breve: Configurar un sistema que despliega automáticamente el código nuevo en el servidor de producción cada vez que un programador guarda cambios validados.

Kanban

Definición: Método visual para gestionar el flujo de trabajo, originario de la manufactura japonesa. Utiliza un tablero con columnas (ej. "Pendiente", "En Progreso", "Hecho") para visualizar las tareas y limitar el trabajo en curso (WIP).

Ejemplo breve: Mover una tarjeta adhesiva de la columna "En desarrollo" a "Pruebas" en un tablero físico o digital para que todos vean el estado de la tarea.

TDD (Desarrollo Guiado por Pruebas)

Definición: Práctica de ingeniería de software que involucra escribir una prueba fallida antes de escribir el código funcional necesario para pasar dicha prueba. Invierte el proceso tradicional, garantizando una alta cobertura de tests y un diseño más limpio.

Ejemplo breve: Escribir un test que espera que una función devuelva "Hola", ver que falla porque la función no existe, y luego crear la función para que el test pase.

Programación en Pareja (Pair Programming)

Definición: Técnica de desarrollo ágil donde dos programadores trabajan juntos en una sola estación de trabajo. Uno, el "conductor", escribe el código mientras el otro, el "navegante", revisa cada línea mientras se escribe y piensa en la estrategia general.

Ejemplo breve: Dos desarrolladores sentados frente a un monitor; uno teclea la solución y el otro sugiere correcciones de seguridad en tiempo real.

Modelo en Espiral

Definición: Modelo de ciclo de vida impulsado por el riesgo que combina elementos del modelo de cascada y del prototipado iterativo. El proyecto pasa repetidamente por fases de planificación, análisis de riesgo, ingeniería y evaluación.

Ejemplo breve: Desarrollar un software militar complejo en ciclos, donde en cada vuelta se analiza exhaustivamente la seguridad antes de avanzar.

XP (Extreme Programming)

Definición: Metodología ágil centrada en mejorar la calidad del software y la capacidad de respuesta ante los cambios en los requisitos. Enfatiza la simplicidad, la comunicación, la retroalimentación y el respeto, promoviendo prácticas como la programación en pareja.

Ejemplo breve: Un equipo que integra código varias veces al día y mantiene un diseño simple, refactorizando constantemente.

Modelo V

Definición: Extensión del modelo en cascada que enfatiza la relación entre cada fase de desarrollo y su fase de pruebas correspondiente. Visualmente forma una "V", donde la rama izquierda es especificación y la derecha es validación.

Ejemplo breve: Planificar las "Pruebas de Aceptación" al mismo tiempo que se definen los "Requisitos del Usuario", antes de escribir código.

4. Herramientas CASE (Computer-Aided Software Engineering)

Herramienta CASE

Definición: Software diseñado para apoyar, automatizar o asistir en una o más actividades del ciclo de vida de la ingeniería de software. Su objetivo es aumentar la productividad del ingeniero y mejorar la calidad del producto final.

Ejemplo breve: Usar Enterprise Architect para dibujar diagramas o herramientas que generan código automáticamente a partir de un diseño visual.

IDE (Entorno de Desarrollo Integrado)

Definición: Aplicación que proporciona servicios integrales para facilitar la programación. Normalmente combina un editor de código fuente, herramientas de automatización de compilación y un depurador (debugger) en una sola interfaz gráfica.

Ejemplo breve: Visual Studio Code o IntelliJ IDEA, que sugieren autocompletado de código y marcan errores de sintaxis mientras escribes.

Sistema de Control de Versiones (VCS)

Definición: Herramienta que gestiona los cambios en el código fuente a lo largo del tiempo. Permite a múltiples desarrolladores trabajar en el mismo proyecto simultáneamente, guardando un historial de cada modificación y permitiendo volver a versiones anteriores.

Ejemplo breve: Usar Git para recuperar una versión del código de la semana pasada porque la versión actual tiene un error crítico.

Contenedorización

Definición: Tecnología que permite empaquetar una aplicación y sus dependencias en una unidad ligera y aislada llamada contenedor. Asegura que el software se ejecute de manera idéntica en cualquier infraestructura, eliminando el problema de "funciona en mi máquina".

Ejemplo breve: Usar Docker para empaquetar una app Python, asegurando que corra igual en la laptop del desarrollador y en el servidor de la nube.

Rastreador de Problemas (Issue Tracker)

Definición: Herramienta utilizada para gestionar y mantener listas de problemas, tareas, mejoras o errores detectados en el software. Permite asignar prioridades, responsables y estados a cada ticket de trabajo.

Ejemplo breve: Usar Jira o Trello para reportar un bug visual y asignarlo al diseñador para que lo corrija antes del viernes.

Profiler (Analizador de Rendimiento)

Definición: Herramienta de análisis dinámico que mide el comportamiento del programa, como el uso de memoria, la duración de llamadas a funciones y la frecuencia de ejecución. Es vital para la optimización.

Ejemplo breve: Usar una herramienta para detectar qué función específica está consumiendo el 80% de la CPU y haciendo lenta la aplicación.

Linter

Definición: Herramienta de análisis estático que revisa el código fuente para buscar errores de programación, bugs estilísticos y construcciones sospechosas. Ayuda a mantener un código limpio y estandarizado antes de ejecutarlo.

Ejemplo breve: Un programa que subraya en rojo si olvidaste poner un punto y coma o si estás usando una variable que no definiste.

Mockup / Wireframe Tool

Definición: Herramienta de diseño utilizada para crear esquemas visuales de bajo o alto nivel de la interfaz de usuario. Ayuda a definir la estructura y el contenido de las pantallas antes de programar la parte visual.

Ejemplo breve: Usar Figma o Balsamiq para dibujar dónde irán los botones y menús de una aplicación móvil.

5. Calidad del Software

Aseguramiento de Calidad (QA)

Definición: Conjunto de actividades planificadas y sistemáticas aplicadas al proceso de desarrollo para garantizar que el software cumpla con los estándares de calidad requeridos. Se enfoca en prevenir defectos mejorando el proceso, no solo en encontrarlos.

Ejemplo breve: Realizar auditorías de código periódicas para asegurar que el equipo sigue las guías de estilo y estándares de seguridad definidos.

Usabilidad

Definición: Atributo de calidad que mide la facilidad con la que los usuarios pueden utilizar una aplicación para alcanzar sus objetivos de manera efectiva, eficiente y satisfactoria. Un software potente pero difícil de usar tiene baja usabilidad.

Ejemplo breve: Diseñar botones grandes y claros en una app para adultos mayores, asegurando que puedan navegar sin confusión.

Escalabilidad

Definición: Capacidad de un sistema para manejar una cantidad creciente de trabajo o para ser ampliado a fin de acomodar ese crecimiento sin perder calidad en el servicio (como velocidad de respuesta).

Ejemplo breve: Que una tienda online funcione igual de rápido con 100 usuarios simultáneos que con 100,000 durante el Black Friday.

Deuda Técnica

Definición: Metáfora que refleja el costo implícito de retrabajo causado por elegir una solución "fácil" o rápida ahora, en lugar de usar un enfoque mejor que tomaría más tiempo. Como una deuda financiera, se pagan "intereses" en forma de mayor dificultad para mantener el código después.

Ejemplo breve: Copiar y pegar código para terminar rápido hoy, sabiendo que la próxima semana tendrás que refactorizarlo para que no sea un desastre.

Portabilidad

Definición: Facilidad con la que un software puede ser transferido de un entorno (hardware o software) a otro. Un sistema altamente portable requiere pocos o nulos cambios para funcionar en distintas plataformas.

Ejemplo breve: Una aplicación web que funciona correctamente tanto en el navegador Chrome de Windows como en Safari de macOS.

6. Siglas y Acrónimos Técnicos

API (Application Programming Interface)

Definición: Conjunto de definiciones y protocolos que permite que dos aplicaciones de software se comuniquen entre sí. Define las reglas de interacción, permitiendo usar funciones de un sistema externo sin conocer su código interno.

Ejemplo breve: Usar la API de Google Maps para mostrar un mapa dentro de tu propia página web sin tener que programar el mapa desde cero.

UML (Unified Modeling Language)

Definición: Lenguaje estándar visual para especificar, visualizar, construir y documentar los artefactos de un sistema de software. Utiliza diferentes tipos de diagramas para representar estructuras y comportamientos.

Ejemplo breve: Dibujar un "Diagrama de Casos de Uso" para mostrar gráficamente cómo interactúa el usuario con el sistema.

SaaS (Software as a Service)

Definición: Modelo de distribución de software donde las aplicaciones se alojan en la nube por un proveedor y los usuarios acceden a ellas a través de Internet, generalmente mediante una suscripción, sin necesidad de instalación local.

Ejemplo breve: Usar Google Docs o Netflix directamente desde el navegador en lugar de comprar e instalar un programa en el disco duro.

CI/CD (Integración Continua / Despliegue Continuo)

Definición: Práctica de automatizar la integración de cambios de código de múltiples colaboradores en un solo proyecto de software (CI) y la entrega de esas actualizaciones a producción de manera rápida y eficiente (CD).

Ejemplo breve: Cada vez que un desarrollador sube código, un servidor automático corre las pruebas y, si pasan, actualiza la página web en vivo.

SQL (Structured Query Language)

Definición: Lenguaje estándar diseñado para administrar y recuperar información de bases de datos relacionales. Permite definir estructuras de datos e insertar, actualizar, borrar y consultar información.

Ejemplo breve: Escribir una sentencia `SELECT * FROM Usuarios` para obtener una lista de todos los usuarios registrados en la base de datos.

JSON (JavaScript Object Notation)

Definición: Formato ligero de intercambio de datos, fácil de leer y escribir para humanos y fácil de interpretar y generar para máquinas. Es el estándar de facto para enviar datos entre servidores y aplicaciones web.

Ejemplo breve: Enviar datos de un usuario como `{"nombre": "Juan", "edad": 30}` desde el servidor a la app móvil.

REST (Representational State Transfer)

Definición: Estilo arquitectónico para diseñar servicios web en red. Se basa en recursos identificados por URLs y utiliza los métodos estándar del protocolo HTTP (GET, POST, PUT, DELETE) para manipularlos.

Ejemplo breve: Diseñar una API donde acceder a `/productos/1` con una petición GET devuelve la información del producto número 1.

GUI (Graphical User Interface)

Definición: Interfaz de usuario que permite interactuar con dispositivos electrónicos a través de iconos gráficos e indicadores visuales, en lugar de interfaces basadas solo en texto o comandos (CLI).

Ejemplo breve: Windows o Android son GUIs; permiten abrir programas haciendo clic en iconos en lugar de escribir comandos en una pantalla negra.

ORM (Object-Relational Mapping)

Definición: Técnica de programación para convertir datos entre sistemas de tipos incompatibles en lenguajes orientados a objetos. Crea una "base de datos virtual" de objetos que se pueden usar desde dentro del lenguaje de programación.

Ejemplo breve: Usar código como `usuario.guardar()` en lugar de escribir la sentencia SQL `INSERT INTO usuarios VALUES....`

SDK (Software Development Kit)

Definición: Conjunto de herramientas de desarrollo de software que permite a los programadores crear aplicaciones para un sistema concreto, plataforma de hardware, consola de videojuegos o sistema operativo.

Ejemplo breve: Descargar el Android SDK para tener las librerías y emuladores necesarios para crear apps para teléfonos Android.

7. Términos Técnicos

Backend

Definición: Parte del desarrollo web que se encarga de la lógica del servidor, la base de datos y la integración de aplicaciones. Es todo lo que sucede "detrás de escena" y que el usuario final no ve directamente.

Ejemplo breve: El código que verifica si tu contraseña es correcta y recupera tu saldo bancario seguro desde la base de datos.

Frontend

Definición: Parte del software con la que el usuario interactúa directamente. Incluye el diseño visual, los botones, formularios y la experiencia de usuario (UX) en la pantalla del dispositivo.

Ejemplo breve: El diseño de la página web, los colores y las animaciones que ves al entrar a Facebook.

Bug

Definición: Error, fallo o defecto en un programa de computadora o sistema de hardware que provoca que produzca un resultado incorrecto, inesperado o que se comporte de manera no intencionada.

Ejemplo breve: Que la aplicación de calculadora muestre "0" al intentar dividir 10 entre 2 debido a un error en el código.

Refactorización (Refactoring)

Definición: Proceso de reestructurar el código fuente existente sin cambiar su comportamiento externo. El objetivo es mejorar la legibilidad del código y reducir su complejidad para facilitar el mantenimiento futuro.

Ejemplo breve: Reescribir una función muy larga y confusa dividiéndola en tres funciones pequeñas y claras, sin que el usuario note ningún cambio en la app.

Framework

Definición: Estructura o plataforma de trabajo que provee una base sobre la cual los desarrolladores pueden construir programas. Ofrece herramientas y librerías predefinidas para no tener que reinventar la rueda en tareas comunes.

Ejemplo breve: Usar Django o React para crear una web, aprovechando que ya tienen resuelta la gestión de seguridad y conexión a bases de datos.

Algoritmo

Definición: Conjunto ordenado y finito de operaciones o instrucciones que permite hallar la solución a un problema. Es la lógica paso a paso que el software sigue para realizar una tarea.

Ejemplo breve: La serie de pasos lógicos que usa una aplicación de GPS para calcular la ruta más rápida entre dos puntos.

Microservicios

Definición: Estilo arquitectónico que estructura una aplicación como una colección de servicios pequeños, autónomos y débilmente acoplados. Cada servicio implementa una capacidad de negocio específica y se comunica mediante mecanismos ligeros (usualmente HTTP).

Ejemplo breve: Netflix usa microservicios: uno maneja el login, otro las recomendaciones y otro el streaming de video; si uno falla, los otros siguen funcionando.

Caché

Definición: Capa de almacenamiento de datos de alta velocidad que guarda un subconjunto de datos, generalmente transitorios, para que las solicitudes futuras de esos datos se atiendan con mayor rapidez que accediendo a la fuente principal.

Ejemplo breve: El navegador guarda imágenes de los sitios que visitas para no tener que descargarlas de nuevo cada vez que entras a la página.

Latencia

Definición: Tiempo que tarda un paquete de datos en viajar desde un punto de origen hasta un punto de destino. En software, es el retraso entre la acción del usuario y la respuesta de la aplicación.

Ejemplo breve: En un videojuego online, una latencia alta (lag) hace que tu personaje se mueva segundos después de que presionaste el botón.

Balanceador de Carga

Definición: Dispositivo o software que distribuye el tráfico de red o de aplicaciones entre varios servidores. Su objetivo es aumentar la capacidad (usuarios concurrentes) y la fiabilidad de las aplicaciones.

Ejemplo breve: Si 1000 personas entran a una web al mismo tiempo, el balanceador envía 500 al Servidor A y 500 al Servidor B para que ninguno colapse.

Full Stack

Definición: Perfil técnico o enfoque que abarca tanto el desarrollo del lado del cliente (Frontend) como el del lado del servidor (Backend), incluyendo a menudo el manejo de bases de datos y la configuración de sistemas.

Ejemplo breve: Un desarrollador que puede diseñar la página web en HTML/CSS y también programar la base de datos y la API que la alimenta.

8. Términos Especializados del Negocio (Aplicados a Proyectos de Software)

Stakeholder (Interesado)

Definición: Cualquier individuo, grupo u organización que puede afectar, verse afectado o percibirse como afectado por una decisión, actividad o resultado del proyecto de software. Incluye clientes, usuarios, desarrolladores y gerentes.

Ejemplo breve: El gerente de ventas es un stakeholder clave porque define qué reportes necesita que el nuevo sistema genere.

Regla de Negocio

Definición: Directriz que define o restringe algún aspecto del negocio. En software, estas reglas se traducen en lógica de programación que valida que los procesos cumplan con las políticas de la organización.

Ejemplo breve: "Un cliente no puede realizar una compra si tiene facturas vencidas por más de 30 días" es una regla que el software debe validar automáticamente.

Requisito Funcional

Definición: Declaración de los servicios que el sistema debe proporcionar, cómo debe reaccionar a entradas particulares y cómo debe comportarse en situaciones específicas. Define lo que el sistema hace.

Ejemplo breve: "El sistema debe permitir al usuario descargar su factura en formato PDF al hacer clic en el botón 'Descargar'".

Requisito No Funcional

Definición: Restricción sobre los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, proceso de desarrollo, estándares, seguridad o fiabilidad. Define cómo es el sistema.

Ejemplo breve: "El sistema debe cargar la página de inicio en menos de 2 segundos" (Requisito de rendimiento).

MVP (Producto Mínimo Viable)

Definición: Versión de un nuevo producto que permite a un equipo recolectar la máxima cantidad de aprendizaje validado sobre los clientes con el menor esfuerzo posible. Contiene solo las características esenciales para satisfacer a los primeros clientes.

Ejemplo breve: Lanzar una app de taxis que solo permite pedir viajes y pagar en efectivo, dejando el pago con tarjeta y perfiles avanzados para versiones futuras.

Dominio del Problema

Definición: Área de conocimiento o actividad alrededor de la cual gira el software que se está desarrollando. Entender el dominio es crucial para que los ingenieros construyan soluciones útiles para los expertos del negocio.

Ejemplo breve: En un software médico, el "dominio" incluye conceptos como diagnósticos, triaje y posología, que los programadores deben comprender mínimamente.

Historia de Usuario (User Story)

Definición: Descripción breve y sencilla de una funcionalidad del software contada desde la perspectiva de la persona que desea la nueva capacidad (generalmente un usuario o cliente). Sigue el formato: "Como [rol], quiero [acción], para [beneficio]"

Ejemplo breve: "Como administrador, quiero poder bloquear usuarios, para evitar comportamientos abusivos en la plataforma".

Product Backlog

Definición: Lista priorizada y ordenada de todo lo que se conoce que es necesario en el producto. Es la única fuente de requisitos para cualquier cambio a realizarse en el producto y es dinámica, cambiando a medida que el producto evoluciona.

Ejemplo breve: Una lista maestra que contiene todas las tareas, bugs y nuevas funciones que el equipo debe desarrollar en el futuro.

Criterios de Aceptación

Definición: Conjunto de condiciones predefinidas que un producto o funcionalidad debe cumplir para ser aceptado por un usuario, cliente o sistema. Son las reglas del juego para considerar una tarea como "Terminada".

Ejemplo breve: "La función de búsqueda sólo se aceptará si devuelve resultados en menos de 1 segundo y permite filtrar por fecha".

Entregable (Deliverable)

Definición: Cualquier producto, resultado o capacidad de prestar un servicio, único y verificable, que debe producirse para terminar un proceso, una fase o un proyecto. No es solo código, puede ser documentación.

Ejemplo breve: El "Manual de Usuario" o el "Informe de Pruebas de Seguridad" son entregables obligatorios al finalizar el proyecto.

Análisis de Brechas (Gap Analysis)

Definición: Técnica utilizada para comparar el desempeño real o actual de un sistema/proceso con su desempeño potencial o deseado. Identifica qué falta (la brecha) para llegar del estado actual al estado objetivo.

Ejemplo breve: Evaluar qué funcionalidades tiene el sistema actual vs. qué funcionalidades necesita la empresa para cumplir con la nueva ley fiscal.