Tecnológico Nacional de México

Campus Querétaro

Glosario técnico

Producto de la asignatura:
Ingeniería de Software

Desarrolla por:

Alfaro Domínguez Iván
22212379
Díaz García Yaneli Guadalupe
22140752
Hernández Perrusquia Jhoel
21140780
Mata Gallegos Camila Patricia
22140820

Asesor docente:

Villeda Maldonado Julio Alejandro

Fecha de entrega:

6 de febrero de 2026

Grupo:
6A

Conceptos generales de Ingeniería de Software

1. Software

Definición: Conjunto de programas, datos y documentación que permiten a un sistema informático realizar tareas específicas.
Ejemplo:*Una aplicación móvil junto con su base de datos y manual de usuario*

2. Ingeniería de requisitos

Definición: Proceso de identificar, analizar, documentar y validar los requisitos de un sistema.
Ejemplo: *Entrevistar usuarios para definir funciones del sistema.*

3. Modelo de ciclo de vida del software (SDLC)

Definición: Herramienta que define cómo se organiza y gestionan las fases del desarrollo de software.
Ejemplo: *Elegir cascada o ágil para desarrollar un sistema*

a. Modelo en cascada

Definición: Modelo secuencial donde cada fase inicia solo cuando la anterior ha finalizado.
Ejemplo: *Primero análisis, luego diseño y después programación.*

b. Modelo Iterativo

Definición: Modelo que desarrolla el software en ciclos repetitivos, permitiendo mejoras graduales
Ejemplo: *Publicar versiones incrementales de una app.*

c. Modelo espiral

Definición: Modelo que combina iteraciones con análisis de riesgos en cada ciclo de desarrollo.
Ejemplo: *Evaluar riesgos antes de agregar nuevas funciones.*

4. Calidad del software

Definición: Grado en que un sistema cumple con los requisitos y expectativas del usuario..
Ejemplo: *Una app estable, rápida y fácil de usar.*

5. Metodología de desarrollo

Definición: Marco de trabajo que establece cómo se organiza y ejecuta el proceso de desarrollo de software.
Ejemplo: *Usar Scrum para desarrollar un sistema por iteraciones.*

6. Requerimientos

Definición: Condiciones o capacidades que el sistema debe cumplir para satisfacer las necesidades del usuario y del negocio.
Ejemplo: *El sistema debe mostrar espacios disponibles en tiempo real.*

a. Requerimientos funcionales

Definición: Describen qué funciones o servicios debe ofrecer el sistema.
Ejemplo: *Permitir al usuario registrarse e iniciar sesión.*

b. Requerimientos no funcionales

Definición: Definen restricciones de calidad como rendimiento, seguridad, usabilidad o disponibilidad, enfocándose en el "cómo" funciona.
Ejemplo: ***La app debe responder en menos de 3 segundos***

7. UML (Lenguaje Unificado de Modelado)

Definición: Lenguaje gráfico estándar utilizado para modelar, visualizar y documentar sistemas de software.
Ejemplo: *Diagramas de clases y casos de uso.*

8. Caso de uso

Definición: Descripción de cómo un actor interactúa con el sistema para lograr un objetivo específico.
Ejemplo: *Reservar espacio de estacionamiento.*

9. Arquitectura de software

Definición: Estructura general del sistema que define componentes, relaciones y tecnologías utilizadas.
Ejemplo: *"Una arquitectura cliente-servidor con app móvil y servidor en la nube"*

10. Aseguramiento de Calidad de Software (SQA)

Definición: Conjunto de actividades planificadas que ayudan a que el software cumpla con los estándares y procesos de calidad.
Ejemplo: *Auditorías y revisiones durante el desarrollo.*

11. Arquitectura Cliente-Servidor

Definición: Modelo donde el cliente solicita servicios y el servidor los procesa y responde.
Ejemplo: *Una app móvil que se conecta a un servidor en la nube.*

Fases del ciclo de vida

1. Planeación

Definición: Etapa donde se define el alcance, recursos, tiempos y costos del proyecto.
Ejemplo: *Estimar el tiempo de desarrollo de la aplicación.*

2. Análisis

Definición: Fase donde se estudian las necesidades del usuario y se definen los requisitos.
Ejemplo: *Parkly necesita de instalaciones con sensores y controladores.*

3. Diseño del sistema

Definición: Etapa en la que se define la arquitectura y estructura del sistema.
Ejemplo: *Crear los diagramas UML del sistema.*

4. Implementación

Definición: Fase en donde se programa y construye el software según el diseño.
Ejemplo: *Desarrollar las vistas de las interfaces.*

5. Pruebas de software

Definición: Etapa donde se verifica que el software funcione correctamente y sin errores.
Ejemplo: *Probar que los pagos se registren correctamente.*

6. Pruebas unitarias

Definición: Pruebas enfocadas en verificar el correcto funcionamiento de módulos individuales.
Ejemplo: *Probar solo el módulo de inicio de sesión.*

7. Pruebas de integración

Definición: Evalúan la interacción entre diferentes módulos del sistema.
Ejemplo: *Verificar que la app se comunique correctamente con la base de datos.*

8. Despliegue

Definición: Fase en donde el software se instala y se pone en funcionamiento para los usuarios finales.
Ejemplo: *Publicar la app en Google Play.*

9. Operación

Definición: Etapa en la que el sistema se usa de manera cotidiana en su entorno real (condiciones reales).
Ejemplo: *Usuarios usando la app diariamente para ubicar su lugar de estacionamiento.*

10. Mantenimiento de software

Definición: Fase donde se corrigen errores, se mejoran funciones o se adapta el sistema.
Ejemplo: *Actualizar la app con una nueva interfaz.*

a. Mantenimiento correctivo

Definición: Tipo de mantenimiento enfocado en corregir errores detectados.
Ejemplo: *Arreglar un fallo en el inicio de sesión.*

b. Mantenimiento evolutivo

Definición: Mantenimiento que agrega nuevas funcionalidades al sistema.
Ejemplo: *Incluir nuevas opciones de pago.*

c. Mantenimiento Preventivo

Definición: Acciones para evitar fallos futuros y mejorar la estabilidad del sistema.
Ejemplo: *Optimizar el código o mejorar las configuraciones de seguridad.*

Metodología del desarrollo

1. Modelo en cascada (Waterfall)

Definición: Las fases como análisis, diseño, implementación, pruebas, despliegue y mantenimiento se ejecutan de forma lineal y secuencial.
Ejemplo: Si una escuela quiere hacer un sistema para registrar calificaciones, tiene que hacer el análisis de requisitos antes de el diseño del sistema y el desarrollo; no puede pasar a la siguiente fase sin terminar la anterior.

2. Sprint

Definición: es un ciclo de trabajo breve y de duración fija donde un equipo completa una cantidad establecida de trabajo.
Ejemplo: un equipo se propone a hacer un sprint con el objetivo de que un usuario se registre e inicie sesión en una app con una duración de 2 semanas.

3. Scrum

Definición: es un tipo de gestión que se basa en dividir el trabajo en sprints planificados y definir roles en el equipo.
Ejemplo: cuando un equipo quiere realizar una app para una biblioteca y asigna los roles de product owner, scrum master, equipo de desarrollo, etc.; además crea diferentes sprints para cumplir sus objetivos en un tiempo fijo.

4. Kanban

Definición: es una manera de organizar el trabajo utilizando un tablero visual donde las tareas manejan banderas de “por hacer” y “hecho”.
Ejemplo: un estudiante que separa sus tareas en “hacer un glosario técnico - por hacer” y “encontrar un cliente nuevo - hecho”.

5. Lean

Definición: este método busca realizar solo las actividades necesarias y desechar los desperdicios para entregar un resultado más rápido.
Ejemplo: cuando una empresa desarrolla los requerimientos necesarios para que su cliente esté satisfecho y su producto funcione correctamente, sin agregar nada extra.

6. Programación extrema

Definición: es una metodología que se basa en realizar cambios rápidos, pruebas constantes y mucha colaboración entre programadores y cliente para realizar software de calidad.
Ejemplo: una empresa que desarrolla una pequeña característica del software y justo después hace pruebas y se lo muestra al cliente para realizar cualquier cambio que sea requerido antes de pasar a otra característica.

7. DevOps

Definición: este método utiliza desarrollo y operaciones para crear, probar y desplegar software de manera rápida y continua.
Ejemplo: cuando un equipo realiza un cambio pequeño a su aplicación, realiza pruebas automáticamente y una vez aceptado el cambio se publica de manera autónoma el cambio.

8. Desarrollo Rápido de Aplicaciones (RAD)

Definición: se enfoca en crear aplicaciones rápidamente usando prototipos y retroalimentaciones de usuarios.
Ejemplo: un equipo realiza un prototipo del producto y se lo envía a usuarios para que lo prueben y envíen sus comentarios para después utilizarlos y mejorar el prototipo.

9. Modelo en espiral

Definición: es un método que utiliza ciclos para desarrollar el programa por partes, haciendo nuevas implementaciones en cada iteración.
Ejemplo: una escuela crea un sistema de gestión de alumnos donde en el primer ciclo crea un programa básico con nombres y materias, una vez hechas las pruebas y lanzado el programa hacen una segunda vuelta y agregan las calificaciones de cada materia, realizan pruebas y lanzan la actualización para posteriormente seguir haciendo ciclos.

10. Modelo incremental

Definición: en este modelo se hacen entregas por partes del producto en donde en cada una se agregan nuevas funciones al sistema.
Ejemplo: un equipo realiza un sistema de notas, en su primer incremento tienen las funciones de crear y guardar notas, en su segundo incremento tienen las funciones de editar y borrar notas, y en su último incremento tienen la función de compartir notas.

Herramienta CASE a nivel conceptual

1. Toolkit

Definición: es un paquete independiente de herramientas CASE que sirve para apoyar en tareas específicas del desarrollo de software.
Ejemplo: cuando un equipo está diseñando un sistema y cuenta con una herramienta para hacer diagramas UML o para documentar requisitos.

2. Workbench

Definición: es otro conjunto de herramientas pero integradas que trabajan juntas para apoyar una fase completa del desarrollo.
Ejemplo: una empresa tiene su espacio que incluye diagramas UML, modelado de procesos y diseño de base de datos.

3. Upper case

Definición: hace referencia a las herramientas que apoyan en las primeras etapas de desarrollo (análisis y diseño).
Ejemplo: uppercase se utiliza para hacer diagramas de casos de uso, documentar requisitos del sistema, etcétera.

4. Middle case

Definición: hace referencia a las herramientas que apoyan en la parte intermedia del desarrollo, principalmente la implementación y el enlace entre el diseño y el código.
Ejemplo: una herramienta que genera automáticamente la estructura del código y el programador sólo tiene que completar la lógica.

5. Lower case

Definición: se refiere a las herramientas que apoyan en las últimas etapas del desarrollo como la programación, pruebas e implementación.
Ejemplo: al usar un editor de código, ejecutar pruebas automáticas y detectar errores.

6. Reingeniería

Definición: es el proceso de mejorar o reconstruir un sistema que ya existe sin hacerlo desde cero.
Ejemplo: cuando una empresa ya cuenta con un sistema antiguo funcional y lo utiliza para modernizarlo y hacerlo más confiable.

7. Soporte del ciclo de vida

Definición: se refiere a la cantidad de ayuda que brindan las herramientas CASE en todas o algunas etapas desde el análisis hasta el mantenimiento.
Ejemplo: una herramienta solo para diagramas brinda un soporte parcial, un workbench de diseño brinda un soporte de una sola fase pero un environment completo brinda un soporte total del ciclo de vida.

8. Estrategia de implantación

Definición: es la forma en que un sistema nuevo se pone en funcionamiento dentro de una organización, decidiendo cómo y cuándo se reemplaza o convive con el sistema anterior.

Ejemplo: una empresa que cambia su sistema de ventas tiene el sistema viejo y el nuevo funcionando al mismo tiempo, se comparan los resultados y al momento de que el nuevo funcione correctamente, se elimina el viejo.

9. Middleware

Definición: es un software intermediario utilizado en herramientas CASE con clientes que permite que diferentes sistemas se comuniquen entre sí a pesar de tener diferentes tecnologías.
Ejemplo: en una tienda online la página web recibe un pedido, el middleware lo envía al sistema de inventario y el sistema responde si hay stock.

10. Downsizing

Definición: es una estrategia utilizada con herramientas CASE en clientes que consiste en migrar sistemas que estaban en computadoras grandes a equipos más pequeños y económicos.
Ejemplo: una empresa que tenía su sistema en un mainframe, lo migra a servidores y computadoras utilizando bases de datos modernas y reduciendo costos.

Términos de calidad de Software

1. **QA (Quality Assurance)**

Definición: La garantía de calidad se asegura que el producto o servicio cumpla con los estándares de calidad de la organización o industria en la que se está desarrollando. Se enfoca en la prevención de defectos.
Ejemplo breve: Usar un estándar de calidad o definir uno para que se obligue a realizar revisiones de código antes de integrar nuevas funcionalidades.

2. **QC (Quality Control)**

Definición: El control de calidad es un proceso sistemático que consiste en usar técnicas de inspección, medición y pruebas para verificar que los productos cumplan con los estándares de calidad o requisitos establecidos. Se enfoca en el producto final.
Ejemplo breve: Realizar un plan de pruebas sobre la aplicación ya terminada para encontrar errores de interfaz o de botones que no funcionan.

3. **Pruebas de Humo (Smoke Testing)**

Definición: Son un conjunto de pruebas cuyo objetivo es comprobar el funcionamiento de las partes críticas del software. Se utilizan para comprobar que una nueva versión de código no ha perjudicado las funciones existentes.
Ejemplo breve: Verificar tras una actualización, los usuarios aún puedan iniciar sesión.

4. **Pruebas Unitarias**

Definición: Son el conjunto de pruebas de bajo nivel que se encargan de comprobar el funcionamiento correcto de un bloque de código aislado como una función o método.
Ejemplo breve: Comprobar que una función llamada “calcularsuma()” devuelta exactamente el resultado correcto de dos números insertados por el usuario.

5. **Mantenibilidad**

Definición: Es la capacidad del software para ser sometido a cambios de manera efectiva y eficiente, ya sea para como para corregir errores, mejorar el rendimiento o adaptarse a cambios.
Ejemplo breve: Organizar el código en módulos independientes, para que en caso de cambios, solo editar archivos específicos.

6. **Escalabilidad**

Definición: La capacidad del software para manejar una cantidad creciente de usuarios o volúmenes de información, sin sufrir daños en su rendimiento y funcionalidad.
Ejemplo breve: Conseguir que el sistema, que usualmente soporta un tráfico de 1000 usuarios, soporte un tráfico de 10000 usuarios y que la velocidad del servicio sea la misma.

7. **Usabilidad**

Definición: La manera en la que es intuitivo y fácil de usar el software para el usuario final, ya que se busca satisfacer sus necesidades y deseos.

Ejemplo breve: El diseño de un formulario de registro que sea fácil, claro y rápido de entender.

8. **CI/CD (Integración continua / Despliegue Continuo)**

Definición: Prácticas y herramientas que permiten agilizar el ciclo de vida del desarrollo de software, a través de la automatización de la compilación, las pruebas y la implementación, lo que permite enviar cambios de código de forma más rápida y fiable.

Ejemplo breve: Un sistema en el que al subir una actualización en Github, lo compile automáticamente, pase las pruebas y lo publique en el servidor en unos minutos.

9. **TDD (Test Driven Development)**

Definición: Es una metodología de desarrollo de software donde primero se desarrollan las pruebas y luego el código para que esas pruebas pasen.

Ejemplo breve: Escribir una prueba para que falle al buscar un usuario que no existe y realizar el código suficiente para pasar esa prueba.

10. **Trazabilidad**

Definición: La capacidad en que un código como requisito se puede rastrear desde su origen hasta su implementación, junto con las pruebas que lo validan. El punto es brindar mejor calidad y consistencia en el desarrollo del producto.

Ejemplo breve: Implementar una identificación para identificar el pedazo de código que conecte con el requisito del proyecto.

Siglas

1. **ISO (International Organization for Standarization)**

Definición: Una entidad independiente que crea estándares voluntarios para asegurar la calidad, seguridad y eficiencia de productos y servicios.
Ejemplo breve: ISO/IEC 25010, modelo para evaluar el software mediante 8 características.

2. **IEEE (Institute of Electrical and Electronics Engineers)**

Definición: Es la mayor organización técnica profesional del mundo y establece los estándares fundamentales para la documentación y el ciclo de vida de software.
Ejemplo breve: IEEE 829, define una estructura formal para los documentos en todas las fases del ciclo de vida del software.

3. **UI (User Interface)**

Definición: La interfaz del usuario se refiere a cómo se interactúa con el software o dispositivo digital. Incluye botones, iconos, pantallas e imágenes que nos permiten navegar, ingresar y manipular datos en un sistema.
Ejemplo breve: Colocar un icono de lupa en la barra de búsqueda para que sea fácil de identificar.

4. **UX (User Experience)**

Definición: La experiencia de usuario se refiere a la experiencia general que tiene el usuarios con el software o producto. El objetivo de desarrollar el UX es identificar los puntos débiles que podrían obstaculizar la interactividad del usuario con el servicio.
Ejemplo breve: Reducir el número de clicks que un usuario tiene que emplear para comprar un producto, logrando que no se sienta frustrado.

5. **API (Application Programming Interface)**

Definición: Son mecanismos para que dos componentes de un software se comuniquen entre ellos mediante un conjunto de reglas y protocolos.
Ejemplo breve: Un sitio web de viajes que se conecta a una base de datos para mostrar los vuelos disponibles en tiempo real.

6. **REST (Representational State Transfer)**

Definición: Es un estilo de arquitectura de software que impone unas condiciones sobre el funcionamiento de la API basado en HTTP.
Ejemplo breve: Utilizar un método GET para obtener información de un usuario.

7. **IDE (Integrated Development Environment)**

Definición: Una herramienta de software que proporciona facilitadores como desarrolladores, un editor de código fuente, compilación, pruebas que ayuda a los desarrolladores a desarrollar código de software más eficientemente.
Ejemplo breve: Visual Studio Code, Eclipse, IntelliJ

8. **CMMI (Capability Maturity Model Integration)**

Definición: Es un modelo que ayuda a las organizaciones a optimizar la mejora de los procesos. El modelo organiza la madurez de la organización en 5 niveles: Inicial, gestionado, definido, gestionado cuantitativamente, optimizando.

Ejemplo breve: Una empresa alcanza el nivel de 3 de madurez al estandarizar sus procesos de desarrollo en todos sus equipos de trabajo.

9. **RE (Requirements Engineering)**

Definición: Una rama de la ingeniería que se encarga de definir, documentar, analizar, validar y gestionar las necesidades y funcionalidades de un sistema, usualmente de software o hardware.

Ejemplo breve: Realizar entrevistas con los clientes para definir los requerimientos del producto.

10. **SDK (Software Development Kit)**

Definición: Un conjunto de herramientas que permite al programador desarrollar aplicaciones para un sistema en específico.

Ejemplo breve: Android SDK

Términos técnicos

1. **Especificación de requisitos de software(ERS)**
    a. **Definición:** Documento formal que describe el comportamiento, funciones y restricciones de un sistema de software bajo el estándar IEEE 830-1998.
    b. **Ejemplo breve:** El documento técnico de Parkly que define el sistema inteligente para estacionamientos

2. **Requisito funcional(RF)**
    a. **Definición:** Declaración de un servicio o función específica que el sistema debe ser capaz de realizar en respuesta a entradas directas.
    b. **Ejemplo breve:** La capacidad de la app para mostrar un mapa de disponibilidad en tiempo real.

3. **Requisito no funcional(RNF)**
    a. **Definición:** Atributo de calidad o restricción técnica que describe cómo debe operar el sistema, no describen funciones, sino niveles de servicio.
    b. **Ejemplo breve:** El requisito de que la latencia de actualización visual en la app no exceda los 30 segundos.

4. **Aseguramiento de calidad de software(SQA)**
    a. **Definición:** Conjunto de actividades planificadas para evaluar si los procesos de desarrollo y los productos finales cumplen con los estándares establecidos.
    b. **Ejemplo breve:** La revisión sistemática de la documentación y el código con un 80% de cobertura mínima.

5. **Internet de las cosas(IoT)**
    a. Definición: Red de objetos físicos con sensores y tecnología integrados que se conectan para intercambiar datos a través de internet.
    b. Ejemplo breve: Los sensores instalados en los cajones que detectan vehículos físicamente.

6. **Sensores Infrarrojos**
    a. Definición: Dispositivos opto-electrónicos que detectan movimiento o presencia mediante la medición de radiación infrarroja de objetos.
    b. Ejemplo breve: El hardware instalado en cada cajón para determinar si un espacio está libre u ocupado.

7. **Microcontrolador**
    a. Definición: Computadora en un solo chip optimizada para controlar tareas específicas en sistemas integrados, conteniendo CPU, RAM y memoria de programa.
    b. Ejemplo breve: Uso de tarjetas tipo ESP32 o Arduino para procesar datos de ocupación.

8. **Hashing**

a. **Definición:** Algoritmo criptográfico que transforma datos en una cadena única de caracteres para proteger información sensible sin revelarla.
b. **Ejemplo breve:** El almacenamiento de contraseñas de usuarios en la base de datos de Parkly de forma cifrada.

9. Administración de la configuración de software(SCM)

a. **Definición:** Disciplina que controla los cambios en el código y la documentación mediante versiones, asegurando que el equipo trabaje en la versión correcta.
b. **Ejemplo breve:** El proceso de auditar que todas las actualizaciones de la app móvil estén debidamente registradas.

10. Diagrama de clases

a. **Definición:** Representación gráfica de la estructura estática de un sistema, mostrando las clases, sus atributos, métodos y relaciones.
b. **Ejemplo breve:** El esquema técnico que muestra cómo la clase "Reserva" se relaciona con la clase "Usuario".

Términos especializados del negocio

1. **Modelo de Negocio:**
    a. **Definición:** Esquema que describe la forma en que una empresa crea, entrega y captura valor, detallando clientes, socios y finanzas.
    b. **Ejemplo breve:** El plan de Parkly para rentabilizar la gestión de estacionamientos mediante tecnología IoT.

2. **Sociedad por Acciones Simplificada (S.A.S.):**
    a. **Definición:** Forma jurídica empresarial en México que permite una constitución rápida y 100% digital, ideal para startups tecnológicas.
    b. **Ejemplo breve:** La estructura legal elegida para constituir a Parkly ante la Secretaría de Economía.

3. **Software as a Service (SaaS):**
    a. **Definición:** Modelo de distribución de software donde el soporte y la aplicación se alojan en la nube y se acceden vía suscripción.
    b. **Ejemplo breve:** La expansión futura de Parkly como plataforma para múltiples ciudades bajo suscripción.

4. **Propuesta de Valor:**
    a. **Definición:** Declaración de los beneficios únicos que un producto ofrece a sus clientes para resolver sus frustraciones específicas.
    b. **Ejemplo breve:** Ofrecer ahorro de tiempo y reducción de estrés a conductores al encontrar lugar rápido.

5. **Análisis FODA:**
    a. **Definición:** Herramienta estratégica que evalúa Fortalezas, Oportunidades, Debilidades y Amenazas de una empresa.
    b. **Ejemplo breve:** Identificar la IA como fortaleza y la dependencia de internet como una amenaza crítica.

6. **Umbral de Rentabilidad (Punto de Equilibrio):**
    a. **Definición:** Nivel de ventas necesario para que los ingresos igualan exactamente a los costos totales, sin generar pérdida ni ganancia.
    b. **Ejemplo breve:** La proyección financiera que indica que este umbral no se alcanza en el primer año debido a la inversión.

7. **Canales de Distribución:**
    a. **Definición:** Medios digitales o físicos a través de los cuales la empresa entrega su propuesta de valor al cliente.
    b. **Ejemplo breve:** El uso de redes sociales y publicidad digital para atraer conductores a la app.

8. **Estructura de Costes:**
    a. **Definición:** Resumen de todos los gastos (fijos y variables) incurridos para operar el modelo de negocio.

b. **Ejemplo breve:** El pago de salarios a desarrolladores y el mantenimiento de servidores en la nube.

9. **Nicho de Mercado:**
    a. **Definición:** Segmento muy específico y delimitado de la población con necesidades particulares que no están completamente satisfechas.
    b. **Ejemplo breve:** Conductores en zonas urbanas con alto tráfico y escasa disponibilidad de espacios.

10. **Project Charter (Acta de Constitución):**
    a. **Definición:** Documento oficial que autoriza el inicio de un proyecto, definiendo objetivos, presupuesto y responsables.
    b. **Ejemplo breve:** El documento que formaliza el desarrollo de Parkly del 1 de enero al 26 de marzo.