

**UNIVERSIDAD
UCATEC**

ANALISIS Y DESARROLLO DE SISTEMAS I
Ciclos de Vida de los Sistemas
ING. CLAUDIA LOPEZ

La universidad que forma emprendedores...

Ciclos de Vida de los Sistemas

Introducción

Los ciclos de vida de los sistemas son **modelos** que **describen las diferentes etapas** involucradas en el **desarrollo de un sistema** de información, desde su concepción hasta su finalización y mantenimiento. **Cada modelo** ofrece una perspectiva particular sobre cómo **organizar y gestionar el proceso de desarrollo de software**.



Ciclos de Vida de los Sistemas

Fases del ciclo de vida de un Sistema

El ciclo de vida del desarrollo de software (SDLC, por sus siglas en inglés) describe las distintas etapas involucradas en la creación de un software, desde su concepción hasta su mantenimiento. Aunque existen diversos modelos de ciclo de vida, todos comparten un conjunto de fases fundamentales que se adaptan a las necesidades específicas de cada proyecto.



Ciclos de Vida de los Sistemas

Fases del ciclo de vida de un Sistema

1. Planificación

- **Definición de objetivos:** Se establecen los objetivos claros y medibles del sistema.
- **Alcance del proyecto:** Se delimitan los límites del proyecto y las funcionalidades a incluir.
- **Estimación de recursos:** Se determinan los recursos necesarios (tiempo, presupuesto, personal).
- **Planificación del proyecto:** Se crea un plan detallado con las actividades, responsables y fechas de entrega.

2. Análisis

- **Recopilación de requisitos:** Se identifican las necesidades de los usuarios y las funcionalidades del sistema.
- **Modelado del sistema:** Se crean modelos (diagramas de flujo, casos de uso, etc.) para representar el sistema.
- **Análisis de factibilidad:** Se evalúa la viabilidad técnica, económica y operativa del proyecto.

Ciclos de Vida de los Sistemas

Fases del ciclo de vida de un Sistema

3. Diseño

- **Diseño arquitectónico:** Se define la estructura general del sistema.
- **Diseño detallado:** Se especifican los componentes, interfaces y módulos del sistema.
- **Diseño de base de datos:** Se define la estructura de la base de datos si es necesario.

4. Desarrollo

- **Codificación:** Se escribe el código fuente del sistema utilizando un lenguaje de programación adecuado.
- **Construcción:** Se ensamblan los componentes del sistema para formar un producto funcional.

5. Pruebas

- **Pruebas unitarias:** Se verifican los componentes individuales del sistema.
- **Pruebas de integración:** Se verifica la interacción entre los componentes.
- **Pruebas de sistema:** Se evalúa el sistema completo en su conjunto.
- **Pruebas de aceptación:** Los usuarios validan que el sistema cumple con sus requisitos.

Ciclos de Vida de los Sistemas

Fases del ciclo de vida de un Sistema

6. Implementación

- **Despliegue:** Se instala el sistema en el entorno de producción.
- **Migración de datos:** Se transfieren los datos existentes al nuevo sistema.
- **Capacitación de usuarios:** Se capacita a los usuarios en el uso del sistema.

7. Mantenimiento:

- **Corrección de errores:** Se identifican y corrigen los defectos en el sistema.
- **Mejora:** Se realizan modificaciones para mejorar el rendimiento o añadir nuevas funcionalidades.
- **Adaptación:** Se realizan ajustes para adaptarse a cambios en el entorno o en los requisitos del usuario.

Ciclos de Vida de los Sistemas

Inicialización de análisis de sistemas

La inicialización del análisis de sistemas marca el punto de partida en cualquier proyecto de desarrollo de software. Es la fase en la que se establecen las bases para todo el proyecto, definiendo los objetivos, identificando los usuarios y recopilando los requisitos iniciales.

¿Por qué es importante la inicialización?

- **Claridad de objetivos:** Permite definir claramente qué se quiere lograr con el sistema.
- **Entendimiento del problema:** Ayuda a comprender a fondo la necesidad que el software debe satisfacer.
- **Base sólida para el proyecto:** Proporciona una base sólida sobre la cual se construirá el resto del sistema.
- **Prevención de errores:** Al identificar los requisitos al principio, se evitan cambios costosos en etapas posteriores.

Ciclos de Vida de los Sistemas

Inicialización de análisis de sistemas

Pasos clave en la inicialización:

1. Identificación del problema:

- ❖ ¿Cuál es el problema específico que el sistema debe resolver?
- ❖ ¿Qué necesidades insatisfechas existen actualmente?
- ❖ ¿Cómo se manifiesta el problema en la organización?

2. Definición de los objetivos:

- ❖ ¿Qué se espera lograr con el nuevo sistema?
- ❖ ¿Cuáles son los beneficios esperados?
- ❖ ¿Cómo se medirá el éxito del proyecto?

3. Identificación de los usuarios:

- ❖ ¿Quiénes son los usuarios finales del sistema?
- ❖ ¿Cuáles son sus roles y responsabilidades?
- ❖ ¿Qué nivel de experiencia tienen con tecnología?

Ciclos de Vida de los Sistemas

Inicialización de análisis de sistemas

Pasos clave en la inicialización:

4. Recopilación de requisitos:

- ❖ ¿Qué funcionalidades debe tener el sistema?
- ❖ ¿Qué datos debe procesar y almacenar?
- ❖ ¿Cuáles son las restricciones técnicas y de negocio?

5. Análisis de factibilidad:

- ❖ ¿Es técnicamente posible desarrollar el sistema?
- ❖ ¿Es económicamente viable?
- ❖ ¿Es factible desde el punto de vista organizacional?

Ciclos de Vida de los Sistemas

Técnicas de recopilación de datos

Entrevistas: Conversaciones individuales o grupales con los usuarios clave.

Cuestionarios: Encuestas estructuradas para recopilar datos de un gran número de personas.

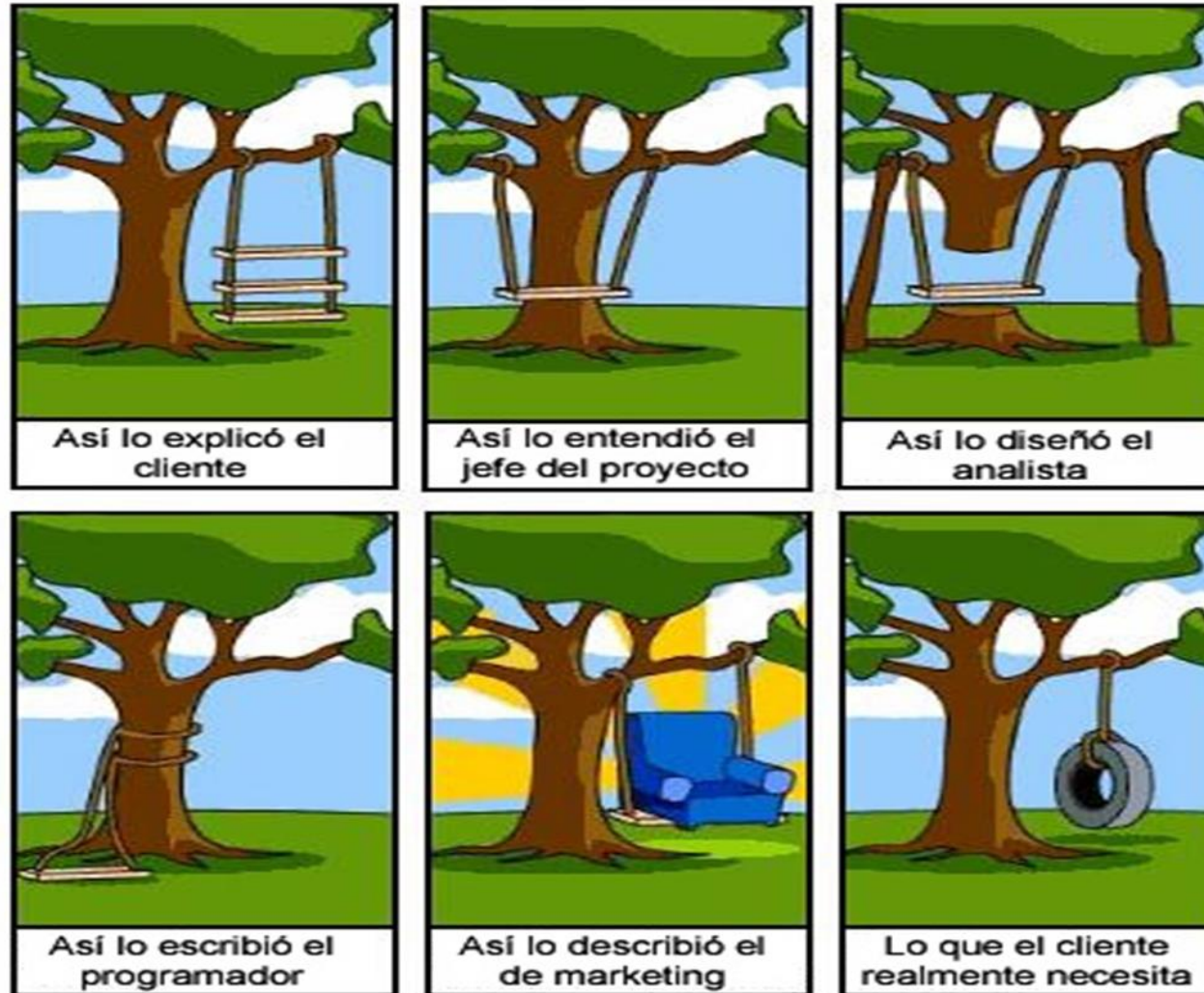
Observación: Análisis directo de las actividades de los usuarios en su entorno de trabajo.

Revisión de documentos: Estudio de documentación existente (manuales, reportes, etc.).

Jornadas de trabajo: Sesiones grupales para generar ideas y consenso.



Ciclos de Vida de los Sistemas



Ciclos de Vida de los Sistemas

Identificación del problema

La **identificación del problema** es el paso inicial y fundamental en cualquier proyecto de desarrollo de software. Consiste en definir con claridad y precisión **cuál es la situación problemática que se busca solucionar mediante el desarrollo de un sistema informático.**

¿Por qué es importante identificar correctamente el problema?

Enfoque claro: Permite establecer un objetivo claro y conciso para el proyecto.

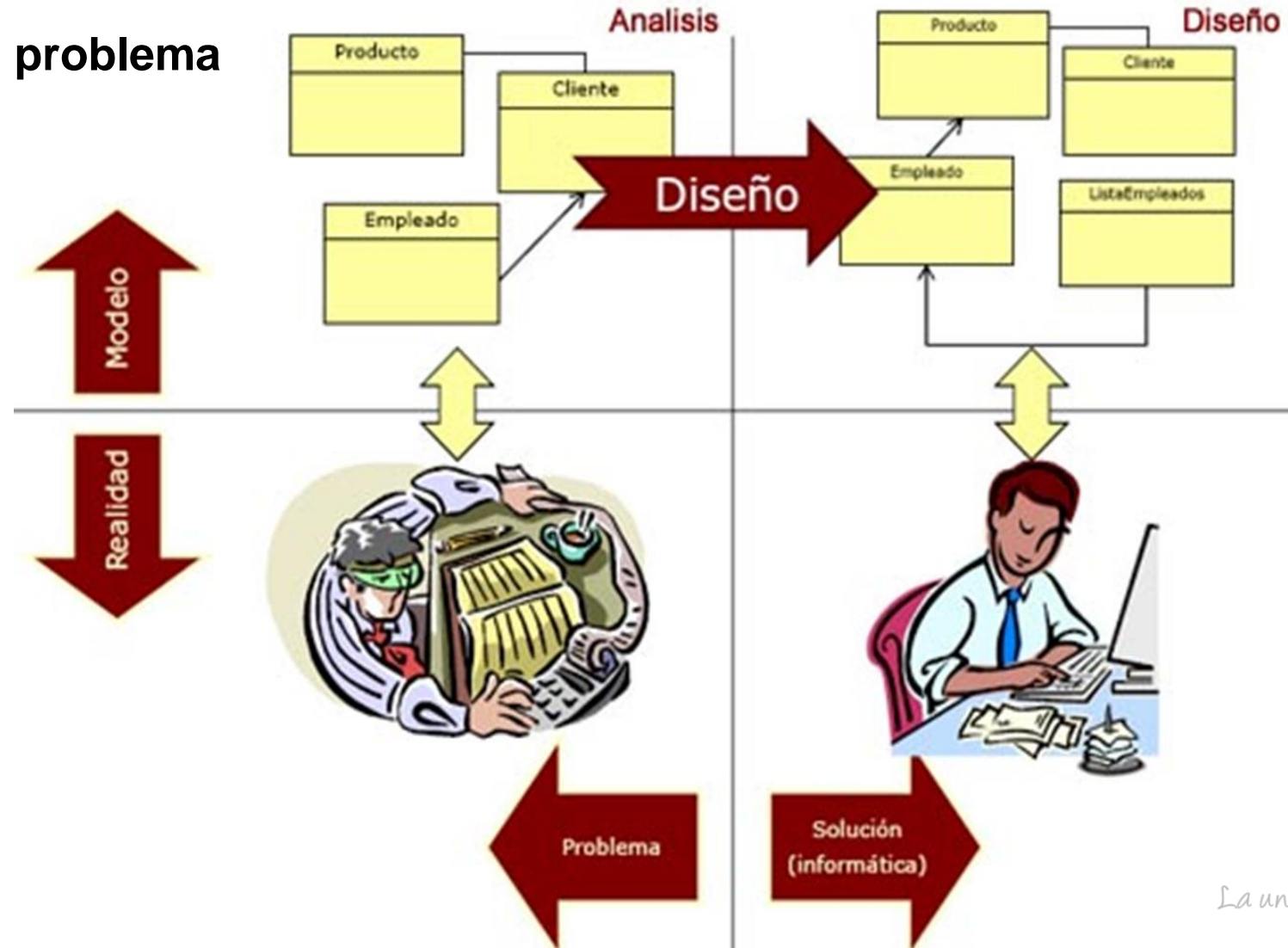
Soluciones efectivas: Ayuda a desarrollar soluciones que realmente aborden las necesidades del usuario.

Prevención de errores: Evita que se desarrollen funcionalidades innecesarias o que no resuelvan el problema principal.

Ahorro de recursos: Al enfocarse en el problema correcto, se optimizan los recursos y se reduce el tiempo de desarrollo.

Ciclos de Vida de los Sistemas

Identificación del problema



Ciclos de Vida de los Sistemas

Técnicas para identificar el problema

Entrevistas: Conversar con los usuarios clave para entender sus necesidades, frustraciones y expectativas.

Observación: Analizar los procesos actuales para identificar ineficiencias, cuellos de botella y áreas de mejora.

Análisis de documentos: Revisar documentos existentes como manuales, reportes y registros para obtener información relevante.

Cuestionarios: Aplicar encuestas para recopilar datos de un gran número de usuarios.

Jornadas de trabajo: Realizar reuniones grupales para generar ideas y consenso.

Ciclos de Vida de los Sistemas

Preguntas clave para identificar el problema

- ¿Cuál es la situación actual?
- ¿Cuáles son las principales dificultades o limitaciones?
- ¿Qué consecuencias negativas genera el problema?
- ¿Qué beneficios se obtendrían al resolver el problema?
- ¿Quiénes se ven afectados por el problema?

Ciclos de Vida de los Sistemas

Comprensión básica del sistema

¿Qué es un sistema?

En el contexto del análisis y desarrollo de software, un sistema es un conjunto de elementos interrelacionados que trabajan juntos para lograr un objetivo común. Estos elementos pueden ser tanto físicos (hardware, dispositivos) como lógicos (software, datos)

¿Por qué es importante comprender un sistema?

Antes de poder modificar o mejorar un sistema, es esencial comprender cómo funciona. Un análisis exhaustivo del sistema actual permite:

Identificar fortalezas y debilidades: Saber qué funciona bien y qué necesita ser mejorado.

Definir requisitos: Establecer claramente qué se espera del nuevo sistema.

Prevenir errores: Evitar cambios que puedan generar problemas inesperados.

Optimizar recursos: Utilizar los recursos de manera eficiente.

Ciclos de Vida de los Sistemas

Elementos clave de un sistema:

Entradas: Datos o información que ingresan al sistema.

Procesos: Actividades que transforman las entradas en salidas.

Salidas: Resultados o información que se generan.

Retroalimentación: Mecanismo que permite ajustar el sistema en función de las salidas.

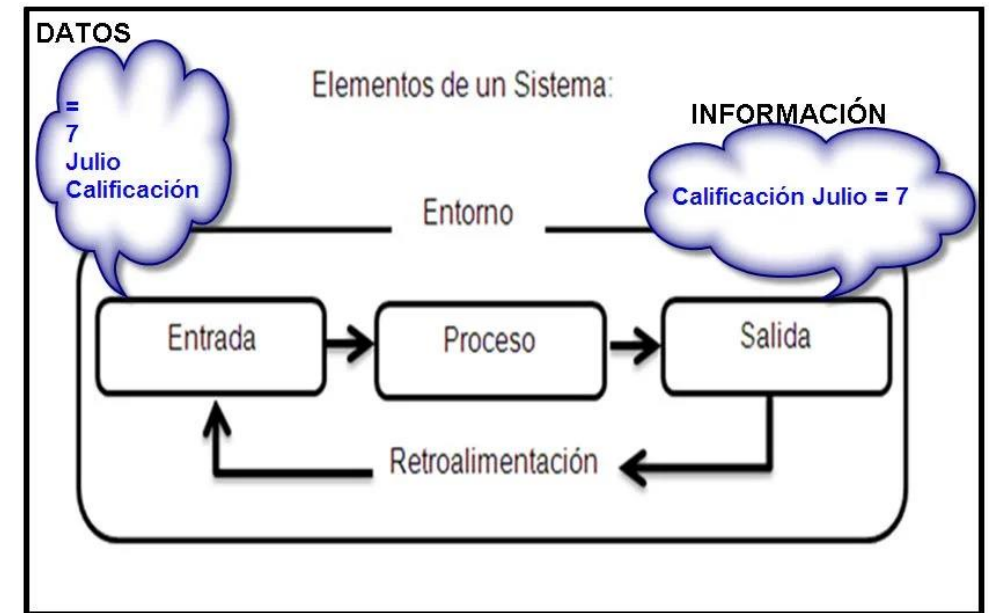
Entorno: Contexto en el que opera el sistema.

Niveles de comprensión de un sistema:

Nivel superficial: Conocimiento básico de las funciones y componentes del sistema.

Nivel intermedio: Comprensión de las relaciones entre los componentes y los procesos.

Nivel profundo: Conocimiento detallado de la lógica interna del sistema y su interacción con el entorno.



Ciclos de Vida de los Sistemas

Técnicas para comprender un sistema:

Observación directa: Ver cómo funciona el sistema en la práctica.

Entrevistas: Hablar con los usuarios y expertos del sistema.

Documentación: Analizar manuales, diagramas y otros documentos relacionados.

Modelado: Crear representaciones gráficas del sistema (diagramas de flujo, UML).

Ciclos de Vida de los Sistemas

Análisis de las necesidades del sistema

El análisis de las necesidades del sistema es una etapa crucial en el desarrollo de software. Consiste en identificar, documentar y validar los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios y los objetivos del negocio.

¿Por qué es importante?

Enfoque claro: Permite definir con precisión qué se espera del sistema.

Prevención de errores: Evita desarrollar funcionalidades innecesarias o que no resuelvan el problema.

Satisfacción del cliente: Garantiza que el sistema final cumpla con las expectativas de los usuarios.

Base sólida para el diseño y desarrollo: Proporciona una guía clara para las siguientes etapas del proyecto.

Ciclos de Vida de los Sistemas

Análisis de las necesidades del sistema

Etapas del análisis de necesidades

- 1. Identificación de los usuarios:** Determinar quiénes utilizarán el sistema y cuáles son sus roles.
- 2. Recopilación de requisitos:**
 - **Funcionales:** Qué debe hacer el sistema (ej: generar reportes, realizar cálculos).
 - **No funcionales:** Características del sistema como rendimiento, seguridad, usabilidad.
- 3. Validación de requisitos:** Verificar que los requisitos sean claros, completos y consistentes.
- 4. Priorización de requisitos:** Establecer un orden de importancia para los requisitos.

Ciclos de Vida de los Sistemas

Análisis de las necesidades del sistema

Técnicas de recopilación de requisitos:

Entrevistas: Conversaciones directas con los usuarios para conocer sus necesidades.

Cuestionarios: Encuestas para obtener información de un gran número de usuarios.

Observación: Analizar el trabajo de los usuarios para identificar problemas y oportunidades.

Revisión de documentos: Estudiar documentos existentes como manuales, reportes, etc.

Jornadas de trabajo: Reuniones grupales para generar ideas y consenso.

Ciclos de Vida de los Sistemas

Planteamiento del sistema de información

El planteamiento del sistema de información es el proceso inicial y fundamental para definir las bases de un proyecto de software. Consiste en identificar la necesidad específica que se busca cubrir, establecer los objetivos del sistema y definir el alcance del proyecto.

¿Por qué es importante el planteamiento?

Claridad de objetivos: Permite establecer un rumbo claro para el desarrollo del sistema.

Enfoque en las necesidades: Garantiza que el sistema se adapte a las necesidades reales de los usuarios.

Optimización de recursos: Evita el desarrollo de funcionalidades innecesarias.

Base sólida para el proyecto: Proporciona una guía clara para las etapas posteriores.

Ciclos de Vida de los Sistemas

Planteamiento del sistema de información

Etapas del planteamiento

Identificación del problema: Definir claramente la situación problemática que el sistema debe resolver.

Definición de objetivos: Establecer los objetivos específicos que se desean alcanzar con el sistema.

Alcance del proyecto: Determinar los límites del sistema, qué incluirá y qué quedará fuera.

Identificación de los usuarios: Definir quiénes utilizarán el sistema y sus roles.

Recopilación de requisitos: Identificar las funcionalidades y características que el sistema debe tener.

Ciclos de Vida de los Sistemas

Planteamiento del sistema de información

Técnicas para el planteamiento

Entrevistas: Conversaciones directas con los usuarios para conocer sus necesidades.

Cuestionarios: Encuestas para obtener información de un gran número de usuarios.

Observación: Analizar el trabajo de los usuarios para identificar problemas y oportunidades.

Revisión de documentos: Estudiar documentos existentes como manuales, reportes, etc.

Jornadas de trabajo: Reuniones grupales para generar ideas y consenso.

Ciclos de Vida de los Sistemas

Planteamiento del sistema de información

¿Por qué es importante un buen planteamiento?

Un planteamiento sólido es la base de un proyecto exitoso. Un planteamiento deficiente puede llevar a:

Desviaciones del proyecto: El sistema desarrollado no cumple con las necesidades reales.

Costos adicionales: Se deben realizar cambios significativos en el sistema después de su desarrollo.

Insatisfacción del cliente: El sistema no cumple con las expectativas de los usuarios.

Ciclos de Vida de los Sistemas

Planteamiento del problema

El planteamiento del problema es el punto de partida de cualquier proyecto de desarrollo de software. Consiste en definir de manera clara y concisa cuál es la situación problemática que se busca resolver a través de la tecnología.

¿Por qué es importante?

Enfoque claro: Permite establecer un objetivo preciso para el proyecto.

Soluciones efectivas: Ayuda a desarrollar soluciones que realmente aborden las necesidades.

Prevención de errores: Evita que se desarrollen funcionalidades innecesarias.

Optimización de recursos: Al enfocarse en el problema correcto, se optimizan los recursos y se reduce el tiempo de desarrollo.

Ciclos de Vida de los Sistemas

Planteamiento del problema

¿Cómo identificar el problema?

Entrevistas: Conversar con los usuarios clave para entender sus necesidades, frustraciones y expectativas.

Observación: Analizar los procesos actuales para identificar ineficiencias, cuellos de botella y áreas de mejora.

Análisis de documentos: Revisar documentos existentes como manuales, reportes y registros para obtener información relevante.

Cuestionarios: Aplicar encuestas para recopilar datos de un gran número de usuarios.

Jornadas de trabajo: Realizar reuniones grupales para generar ideas y consenso.