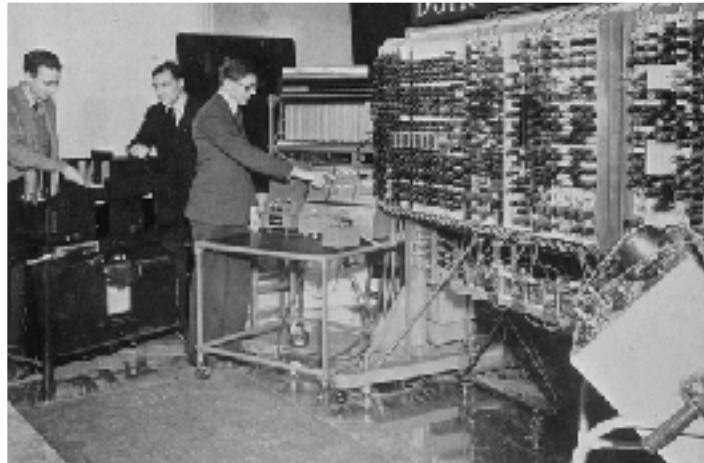


Programación básica

TEMA II. ANÁLISIS Y DISEÑO

La programación ha recorrido un largo camino desde finales de los cuarentas, cuando los programadores colocaban interruptores eléctricos para crear programas.

Hoy existen herramientas y metodologías de computación que nos facilitan mucho la tarea de programar, pero que no nos eximen de **entender** el problema y **pensar** en posibles soluciones al mismo.



Objetivo del tema

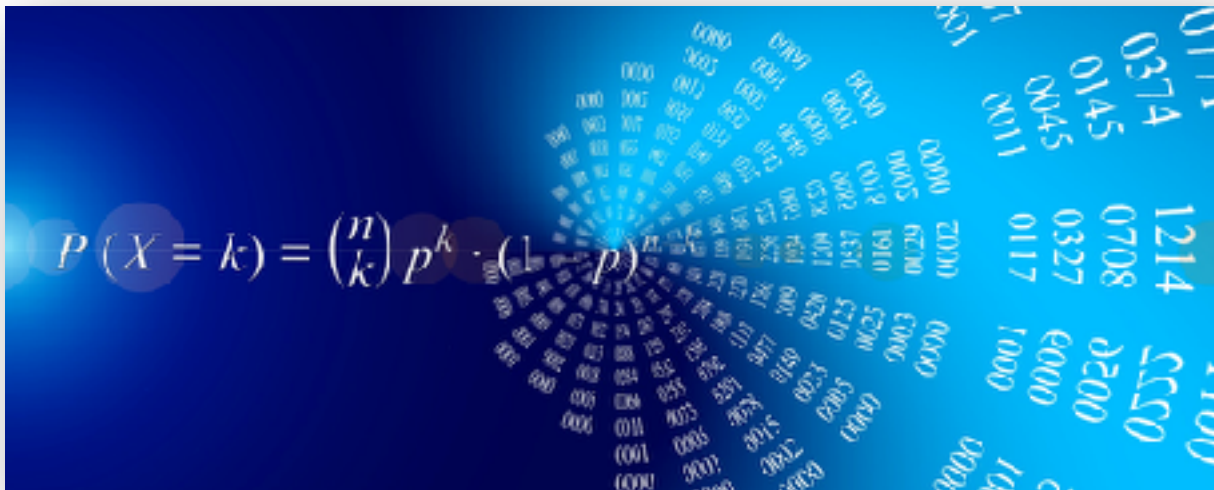
El alumno analizará problemas, construirá algoritmos y diseñará pseudocódigos para resolver problemas.

Objetivo general

Nivel de objetivo:

TERCER NIVEL: Se tendrá que **dominar a la perfección**

✓ **Analizar problemas** de física y matemáticas para que de ser posible se **diseñe la solución por medio de un programa de cómputo.**



Objetivo de la clase

El alumno comprenderá los métodos y conceptos necesarios para analizar problemas

Temas a tratar

- Definición, planteamiento y modelado del problema.
 - Problemas no computables y problemas intratables
- Ingeniería de software
 - Ciclo de vida del software
- Análisis
 - Caja negra

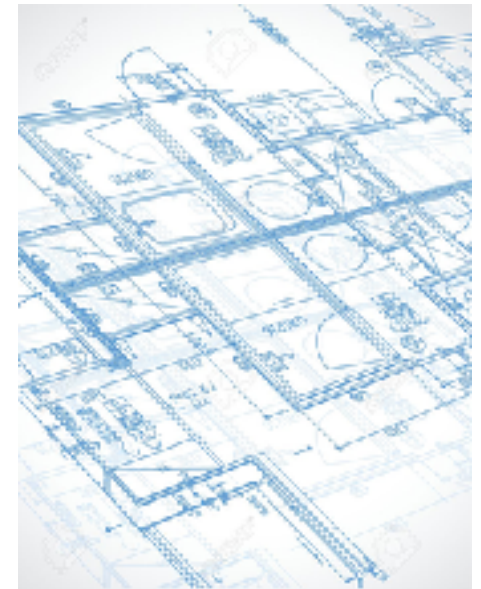
Definición, planteamiento y modelado del problema

Un programa es como los materiales utilizados para construir un edificio. Mucho del trabajo intelectual involucrado en la construcción se hace desde antes de comenzar a construirlo.

El lugar, apariencia y la función del edificio están determinados mucho antes de que se coloque el primer ladrillo. Así sucede en la programación. El diseño de un programa y su lógica son determinados antes de escribir el programa.

Un programa solo hace lo que el programador indica.

Un programa es un conjunto de instrucciones ordenadas y agrupadas de forma lógica durante la etapa de programación



¿Qué es un problema?



$$(3^x)^x - 3^x - 6 = 0 \rightarrow 3^{2x} - 3^x - 6 = 0$$

$$3^x = a \Rightarrow a^2 - a - 6 = 0 \Rightarrow a = \frac{-(-1) \pm \sqrt{1 + 24}}{2}$$

$$a = \frac{1 \pm 5}{2}$$

$$a_1 = 3$$

$$a_2 = -2$$

$$3^x = 3$$

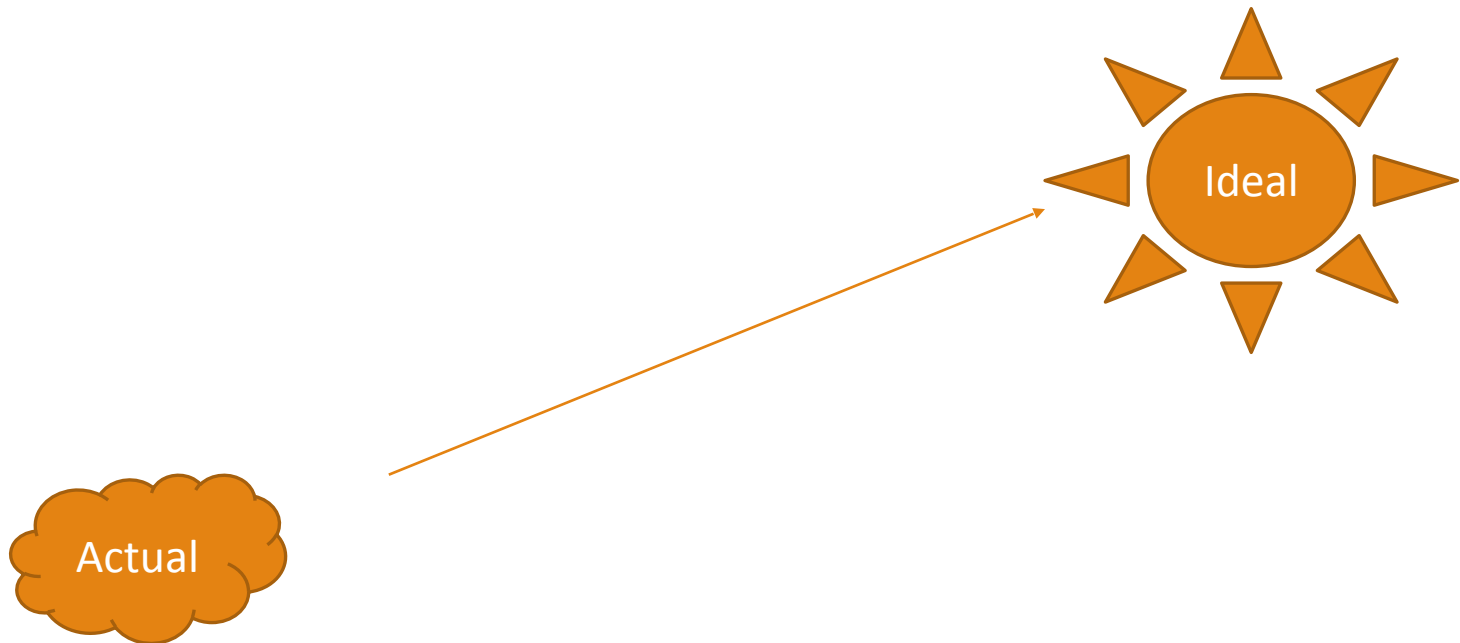
$$x = 1$$

¿Qué es un problema?

Un problema es un conjunto de hechos o circunstancias que dificultan la consecución de algún fin.

Un Problema es una cuestión o punto discutible que requiere de una solución.

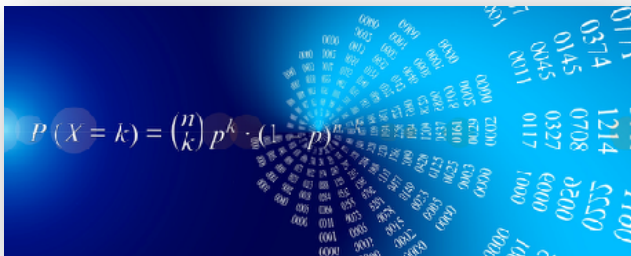
Problema es una cuestión que se propone con la finalidad y ánimo de aclararla o resolverla utilizando una metodología determinada.



Problemas convergentes y divergentes

Problemas divergentes.

Cuando se encuentran soluciones que se contradicen entres si, existe un sin número indefinido de soluciones.



Problemas convergentes.

Problemas lógicos o estructurados ya que tienen respuestas únicas y definidas.

Problemas no computables y problemas intratables

Un problema no computable, es un problema que no puede ser resuelto por medio de una computadora. En general son problemas que no pueden ser expresados en forma lógica, como una serie de instrucciones o de forma matemática, cálculos matemáticos que la computadora no puede realizar por su complejidad.

Hay otros problemas que si bien son computables demandan un consumo enorme en tiempo, hardware o es complicado ingresar la información. Estos problemas se llaman intratables



Adaptación o interpretación del problema

Hay problemas que tienen que ser adaptados o interpretados para que se conviertan en problemas computables y tratables.

Ejemplo: Tenemos que permitir la actualización y consulta de los sentimientos de una persona.

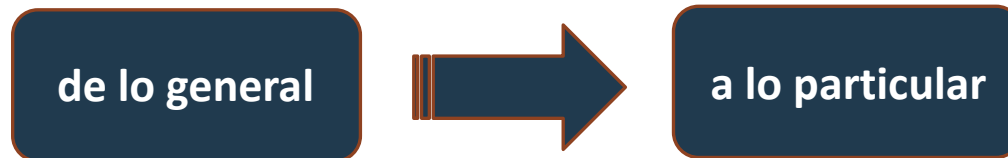
Para este problema no hay una solución única: hay muchas y cada una está relacionada con un contexto de uso. ¿De qué sentimientos estamos hablando? Y en cada caso ¿de qué tipo de consulta estamos hablando: hay que imprimir un listado una vez por mes, diariamente, se trata de una consulta en línea, etc.? Para cada contexto hay una solución diferente



¿Cómo comenzar a analizar un problema?

De manera general para la solución de problemas se sugiere la técnica de cuatro etapas:

1. Entender el problema.
2. Configurar un plan
3. Ejecutar el plan
4. Mirar hacia atrás (Retroalimentación)





Así lo explicó el cliente



Así lo entendió el jefe del proyecto



Así lo diseñó el analista



Así lo escribió el programador



Así lo describió el de marketing



Lo que el cliente realmente necesita

1. Entender el problema.

¿Entiendes todo lo que dice?

¿Puedes replantear el problema en tus propias palabras?

¿Distingues cuáles son los datos?

¿Sabes a qué quieres llegar?

¿Hay suficiente información?

¿Hay información extraña?

¿Es este problema similar a algún otro que hayas resuelto antes?

2. Configurar un plan.

Se puede utilizar una se las siguientes estrategias (Una estrategia se define como un artificio ingenioso que conduce a un final).

Ensayo y error

Usar una variable

Hacer un diagrama o figura

Identificar sub-metas

Usar propiedades de los números

Resolver un problema equivalente

Buscar una fórmula

Usar un modelo



3. Ejecutar un plan.

Implementar la o las estrategias que escogiste hasta solucionar completamente el problema o hasta que la misma acción te sugiera tomar un nuevo curso.

Concédete un tiempo razonable para resolver el problema. Si no tienes éxito solicita una sugerencia o haz el problema a un lado por un momento (¡puede que **se te prenda el foco** cuando menos lo esperes!).

No tengas miedo de volver a empezar. Suele suceder que un comienzo fresco o una nueva estrategia conducen al éxito.

4. Mirar hacia atrás. (retroalimentación)

¿Es tu solución correcta? ¿Tu respuesta satisface lo establecido en el problema?

¿Adviertes una solución más sencilla?

¿Puedes ver cómo extender tu solución a un caso general?

Comúnmente los problemas se enuncian en palabras, ya sea oralmente o en forma escrita. Así, para resolver un problema, uno traslada las palabras a una forma equivalente del problema en la que usa símbolos matemáticos, resuelve esta forma equivalente y luego interpreta la respuesta.

Ingeniería de Software

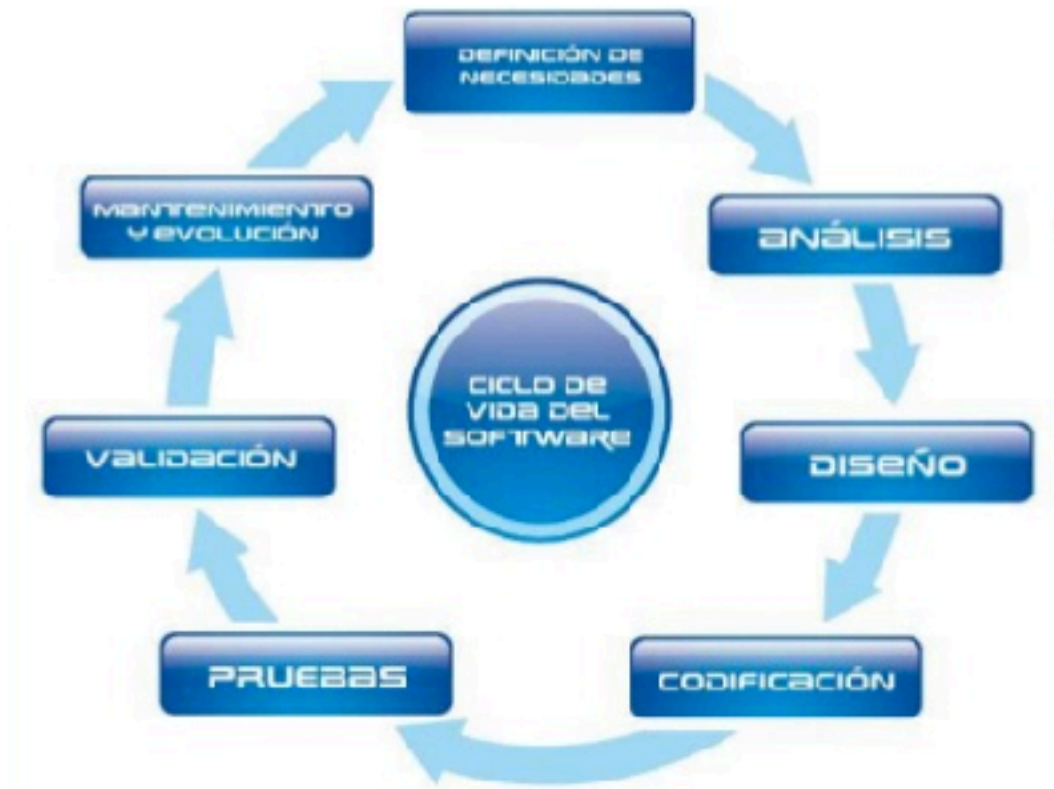
Se define como el uso y establecimiento de principios de ingeniería sólidos, a fin de obtener un software que sea económicamente fiable y funcione eficientemente.

Desarrolla un conjunto de métodos, herramientas y técnicas que se utilizan en el **desarrollo de los programas informáticos**



Ciclo de vida del software

Es un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando desde la definición hasta la finalización de su uso.



Ciclo de vida del software

- **Definición de necesidades.** En esta etapa el analista luego de un minucioso y detallado estudio de los sistemas de una organización, detecta un problema o una necesidad que para su solución y/o satisfacción es necesario realizar un desarrollo de software.
- **Análisis.** En esta etapa se debe entender y comprender de forma detallada cual es la problemática a resolver, verificando el entorno en el cual se encuentra dicho problema, de tal manera que se obtenga la información necesaria y suficiente para afrontar su respectiva solución. Esta etapa es conocida como la del QUÉ se va a solucionar.
- **Diseño.** Una vez que se tiene la suficiente información del problema a solucionar, es importante determinar la estrategia que se va a utilizar para resolver el problema. Esta etapa es conocida bajo el CÓMO se va a solucionar.
- **Codificación.** partiendo del análisis y diseño de la solución, en esta etapa se procede a desarrollar el correspondiente programa que solucione el problema mediante el uso de una herramienta computacional determinada.

Ciclo de vida del software

- **Pruebas.** Los errores humanos dentro de la programación son muchos y aumentan considerablemente con la complejidad del problema. Cuando se termina de escribir un programa de computadora, es necesario realizar las debidas pruebas que garanticen el correcto funcionamiento de dicho programa bajo el mayor número de situaciones posibles a las que se pueda enfrentar.
- **Validación.** Esta etapa tiene como objetivo la verificación de que el sistema desarrollado cumple con los requisitos expresados inicialmente por el cliente.
- **Documentación.** Es la guía o comunicación escrita en sus diferentes formas, ya sea en enunciados, procedimientos, dibujos o diagramas que se hace sobre el desarrollo de un programa. La importancia de la documentación radica en que a menudo un programa escrito por una persona, es modificado por otra. Por ello la documentación sirve para ayudar a comprender o usar un programa o para facilitar futuras modificaciones (mantenimiento).





Ciclo de vida del software

- **Mantenimiento y evolución.** Una vez instalado un programa y puesto en marcha para realizar la solución del problema previamente planteado o satisfacer una determinada necesidad, es importante mantener una estructura de actualización, verificación y validación que permitan a dicho programa ser útil y mantenerse actualizado según las necesidades o requerimientos planteados durante su vida útil. Para realizar un adecuado mantenimiento, es necesario contar con una buena documentación del mismo.



El ciclo de vida:



- ✓ **Nace**  **Cuando se detecta la necesidad de crear u optimizar un sistema .**
- ✓ **Crece**  **Cuando se diseña o se crea el sistema como tal, Diseño de base de datos, Diseño de pantalla entre otros.**
- ✓ **Se Reproduce**  **Cuando se implementa, se prueba y se corrigen las fallas existentes.**
- ✓ **Muere**  **Cuando el sistema NO satisface las necesidades, o la tecnología usada pasa a ser obsoleto o el sistema no se adapta a las nuevas tecnologías.**

Análisis

De acuerdo a los datos y requerimientos definir las necesidades de forma clara y concisa.

Por lo tanto, la etapa del análisis consiste en conocer qué es lo que se tiene (el punto de partida) y que está solicitando el usuario (el punto final).

Caja negra

Una estrategia para el análisis puede ser enfocarme en el que necesito para resolver el problema y que resultado espero. No preocuparme por el cómo lo voy a resolver todavía. Para ello es importante identificar dos grandes conjuntos dentro del sistema: el conjunto de entrada y el conjunto de salida.



Repaso

- ¿Qué es un problema computable?
- ¿Qué es un problema intratable?
- ¿Qué es ingeniería de software?
- Describa el ciclo de vida del software y sus etapas
- ¿Por qué es importante el análisis de un problema?
- ¿Qué es caja negra?

Hasta hace no mucho tiempo se utilizaba el término algoritmo para referirse únicamente a formas de realizar ciertos cálculos, pero con el surgimiento de la computación, el término algoritmo pasó a abarcar cualquier método para obtener un resultado.