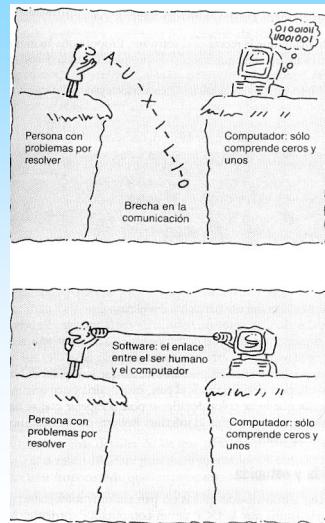


Software:

- Se puede definir como las instrucciones que indican al computador lo que debe hacer.



CLASIFICACION DEL SOFTWARE DE COMPUTADORA:

- software de aplicación: herramientas para elevar la productividad de los usuarios de computadoras en la resolución de problemas.
- Software de sistemas: coordina las operaciones de hardware y lleva a cabo las tareas ocultas que el usuario del computador rara vez observa.

SOFTWARE DE APLICACION

Este software puede adquirirse de dos maneras básicas:

- 1-. Programación a la medida
- 2-. Comprando paquetes preescritos.

CLASIFICACION DE PAQUETES PREESCRITOS: Se pueden clasificar en 4 grupos básicos:

- *Paquetes generales de negocios*: paquetes generales, que en la mayoría de las organizaciones realizan o llevan a cabo: inventarios, etc.
- *Paquetes específicos para una industria*: son para satisfacer las necesidades específicas de la industria como agricultura, ganadería, bienes raíces, transporte, etc.
- Paquetes de productividad organizacional: Esta categoría un tanto general incluye software pensado para satisfacer las necesidades de la organización más que las de los destinados a resolver problemas industriales.
- Paquetes de productividad personal: La mayor parte de los paquetes de productividad personal se ha escrito para ejecutarse en micros.

SOFTWARE DE SISTEMAS

El software de sistemas lleva a cabo tareas fundamentales que todos los usuarios de una computadora requiere.

Hay 3 tipos de software de sistemas:

- Sistemas Operativos
- Programas de Utilería
- Traductores de Lenguajes

Sistemas Operativos

El sistema operativo controla los procesos de la computadora, actuando como interfaz que conecta al usuario.

Un sistema operativo puede realizar 6 funciones básicas:

- Programar trabajos: El sistema operativo puede determinar la secuencia en que se ejecutaran los trabajos.
- Manejar recursos de hardware y software: El sistema operativo puede hacer que se ejecute el programa de aplicación del usuario cargandole en la memoria principal y luego hacer que las distintas unidades de hardware operen según lo especifica la aplicación.
- Mantener la seguridad del sistema: El sistema operativo puede exigir al usuario que teclee una *contraseña*: un grupo de caracteres que identifica al usuario como alguien autorizado para acceder al sistema.

- Permitir a múltiples usuarios compartir los recursos: El sistema operativo puede manejar la planificación y ejecución de los programas de aplicación de muchos usuarios al mismo tiempo, en lo que se conoce como multiprogramación.

- Manejar interrupciones: Una interrupción es una técnica empleada por el sistema operativo para suspender temporalmente el procesamiento de un programa para permitir la ejecución de otro programa.

- Mantener registros de uso: El sistema operativo puede llevar la cuenta del tiempo durante el cual cada usuario usó cada unidad del sistema: la CPU, el almacenamiento secundario y los dispositivos de entrada y salida.

PROGRAMA DE UTILERIA

Un programa de utilería, también conocido como *utilería*, es una rutina que permite al usuario realizar ciertas operaciones de procesamiento de datos básicas que no son exclusivas de las aplicaciones de ese usuario. Las utilerías permiten a los usuarios copiar archivos, borrar archivos, ordenar el contenido de un archivo, etc. Otras utilerías permiten un control de datos entre los usuarios y las computadoras.

Niveles de lenguajes de un computador

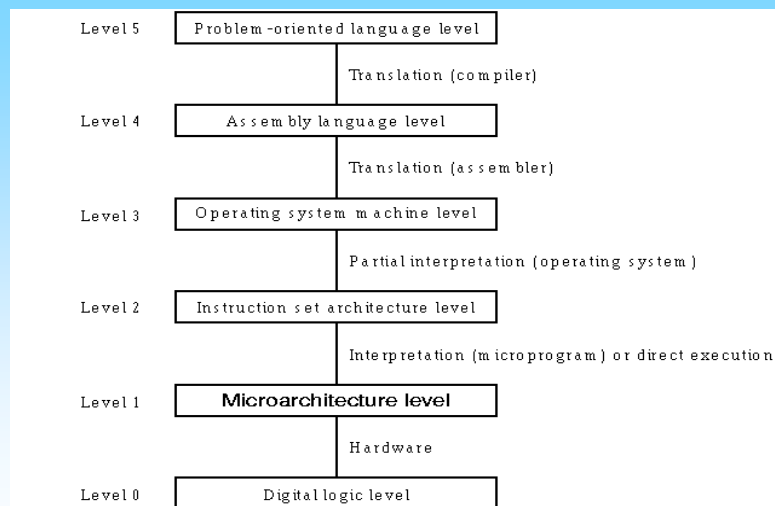


Figure 1-2. A six-level computer. The support method for each level is supported is indicated below it (along with the name of the supporting program).

TRADUCTORES DE LENGUAJE

El software también puede clasificarse en términos de generaciones:

- Lenguajes de la primera generación-lenguaje de máquina: Las primeras computadoras se programaban en lenguaje de máquina, o lenguaje de primera generación: una serie de ceros y unos que la CPU puede interpretar y ejecutar. El lenguaje de máquina es importante porque es el único lenguaje que la computadora entiende. Sin embargo, el lenguaje de máquina es difícil de usar y los científicos en computación desarrollaron lenguajes de programación que se parecen más a la comunicación humana.
- Lenguaje de segunda generación-los ensambladores: Los primeros traductores se llamaban ensambladores, y permitían al programador codificar los programas lenguaje ensamblador o lenguaje de segunda generación. Un lenguaje ensamblador emplea *nombres mnemónicos* (como DIV, para dividir) para las operaciones que deben realizarse y nombres *simbólicos* para datos que se procesan.

- Lenguaje de la tercera generación- los compiladores e intérpretes: Este lenguaje produce múltiples instrucciones del programador objeto con una sola instrucción del programa fuente. Eso implica que los programadores tienen que producir menos líneas de código. Además, la sintaxis de los lenguajes de la tercera generación se parece más al lenguaje del usuario. Ej: Basic, C, Java, Pascal.
- Lenguaje de cuarta generación-los lenguajes naturales: El término lenguaje de cuarta generación, se abrevia 4GL, se refiere a un lenguaje que permite al programador o usuario indicarle a la computadora que debe hacer, en vez de cómo hacerlo. También se usa el término lenguaje natural porque la sintaxis de la 4GL puede ser muy similar a la forma como hablamos normalmente.
- Lenguaje de consulta de base de datos
- Lenguaje de modelado: un lenguaje de modelado está diseñado especialmente para la construcción de modelos matemáticos y sea más fácil que cuando se usa un lenguaje orientado a los problemas.
- Lenguaje de muy alto nivel: El término lenguaje de muy alto nivel suele usarse para describir lenguajes de programación, como APL, que es conciso y potente(pero no necesariamente fácil para el usuario), en mayor grado que los lenguajes

Diseño y desarrollo de sistemas Como Crear Programas.

La Programación de computadores es una actividad relacionada con el antiguo procesos de resolución de problemas:

- a) Compresión del problema
- b) Desarrollo de un plan para resolver el problema
- c) Realización del plan
- d) Evaluación de la solución

El proceso de programación también puede describirse en cuatro pasos.

- a) Definición del problema
- b) Desarrollo, refinamiento y pruebas al algoritmo
- c) Escritura del programa
- d) Pruebas y depuración del programa

Aplicación a un ejemplo concreto:

Se desea desarrollar un juego para adivinar números. En este juego el computador elige al azar un numero entre 1-1000 y permite al jugador un máximo de siete oportunidades para adivinar el numero. Después de cada intento incorrecto el computador indica al jugador si su conjetura fue demasiado alta o demasiado baja.

A) Definición del problema: Escribir un programa que despliegue un juego para adivinar números.

b) Desarrollo, refinamiento y pruebas al algoritmo:

Algoritmo es un conjunto de instrucciones paso a paso que al completarse resuelven el problema original. Generalmente se escribe en pseudocódigo.

Refinamiento por pasos: Dividir el problema.

Iniciar el juego.

Repetir el intento hasta que se adivine el numero o concluyan siete intentos

Finalizar el juego.

Refinamiento1

iniciar el juego

 presentar las instrucciones

 elegir un numero entre 1 y 1000

Repetir el intento hasta que se adivine el numero o concluyan siete intentos

 Leer la conjetura del usuario

 responder a la conjetura

Fin de la repetición

Finalizar el juego

 Presentar el mensaje de terminación

Refinamiento2

iniciar el juego

 presentar las instrucciones

 elegir un numero entre 1 y 1000

 Asignar cero al contador

Repetir el intento hasta que se adivine el numero o concluyan siete intentos

 Leer la conjetura del usuario

 responder a la conjetura

 Si conjetura = numeroRandom, indicarlo y salir

 En caso contrario, si conjetura < numeroRandom indicar que la conjetura es demasiado pequeña

 En caso contrario, si conjetura > numeroRandom indicar que la conjetura es demasiado grande

 Sumar 1 al contador

Fin de la repetición

Finalizar el juego

 Presentar el mensaje de terminación

Estructuras de control:

Controlan el orden de ejecución de las instrucciones, se emplean tres estructuras básicas: Secuencia, selección y repetición.

Pruebas al algoritmo. Se eligen números de conjetura que sean mayores, menores o igual al numero elegido. Por Ejemplo si numeroRandom = 200

iniciar el juego

presentar las instrucciones

elegir un numero entre 1 y 1000

Asignar cero al contador

Repetir el intento hasta que se adivine el numero o concluyan siete intentos

Leer la conjetura del usuario

responder a la conjetura

Si conjetura = numeroRandom, indicarlo y salir

En caso contrario, si conjetura < numeroRandom indicar que la conjetura es demasiado pequeña

En caso contrario, si conjetura > numeroRandom indicar que la conjetura es demasiado grande

Sumar 1 al contador

Fin de la repetición

Finalizar el juego

Presentar el mensaje de terminación

c) Escritura del programa

Los enunciados en el algoritmo se traducen literalmente a líneas de código en el lenguaje de programación que mejor se ajuste a las necesidades de programador.

Formato general de los programas:

1. Encabezado del programa
2. Declaraciones y definiciones
3. El cuerpo del programa.

Texto del programa escrito en lenguaje C

```
#include <stdio.h>
```

```
int numero, contador, random;
```

```
main()
```

```
{  
    printf ("\nBienvenido al juego de adivinanzas. Escogeré un número");  
    printf ("\nentre 1 y 1000; Ud. intentará adivinar cual es.");  
    printf ("\nTiene siete oportunidades");  
    int contador = 0;  
    random = rand();  
    while (contador < 7 && numero != random)  
    {  
        printf ("\nIngrese un numero: ");  
        scanf ("%d", &numero);  
        if (numero == random)  
            printf ("Correcto");  
        if (numero > random) printf ("Muy grande");  
        if (numero < random) printf ("Muy pequeño");  
        contador++;  
    }  
    printf ("%d", random);  
    if (contador == 7) printf ("\nlo engañe siete veces");  
}
```

→ Encabezado

→ Declaraciones

Cuerpo del
programa

d) Pruebas y depuración del programa:

Introducir el programa en la memoria del computador, guardarlo como archivo y traducirlo al lenguaje máquina (compilarlo) y hacerlo ejecutable.

ACTIVIDAD PRACTICA

Análisis de Sistemas

Los programas no existen en un vacío, forman parte de lo que se denominan sistemas de información. La programación es solo una parte de un proceso mas amplio de diseño, implementación y administración de sistemas de información.

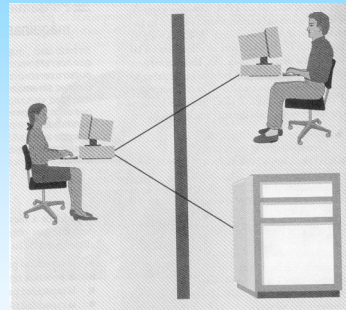
Este es el ámbito del Analista de sistemas el cual es el profesional de la computación que es el principal responsable del desarrollo y mantenimiento de un sistema.

Ciclos de vida de un sistema:

- Investigación
- Análisis
- Diseño
- Desarrollo
- Implantación
- Mantenimiento
- Retiro.

Inteligencia artificial:

Alan Turing y maquinas inteligentes: Puede considerarse que una maquina es inteligente si puede pasar por un ser humano en una prueba ciega.



I.A.: Campo de las ciencias de la computación dedicada a hacer que computadoras perciban, razonen y actúen en forma que hasta ahora, han estado reservadas para el ser humano.

Pueden pensar las máquinas? Test de Turing, tema de preguntas abiertos.
Si se logra engañar a sus interrogadores pasa la prueba. Hasta la fecha ningún programa y computador lo ha logrado

Conversación con PC Therapist

01:32:25 Juez 6: ¡Hola, terrícola!

01:33:03 Terminal 5: Hola, juez, ¿está listo para divertirse? ¿Cómo le va, juez?

01:33:35 Juez 6: Oiga, ¿cómo está el clima allá arriba?

01:34:45 Terminal 5: Nubes oscuras con rayos luminosos ocasionales (más bien, destellos de *flash*).

01:35:26 Juez 6: El problema con la carrera de ratas es que aunque uno gane seguirá siendo una rata. Lo ingenioso puede ser frágil, ¿no? ¿Qué importancia tiene eso para usted?

01:36:51 Terminal 5: No comprendo la pregunta... pero el ingenio de un poeta es una cosa y el de los demás es otra.

01:37:37 Juez 6: Cómo puede decir que no comprende la pregunta... pero que el ingenio de un poeta es una cosa y el de los demás es otra.

01:38:04 Terminal 5: ¿Está repitiendo lo que dije sólo para divertirse?

Lo siento. En ocasiones soy repetitivo... quizá sea porque no puedo encontrar mi organizador electrónico.

Reimpreso de *AI Magazine*, verano de 1992.

Una de las primeras aplicaciones de la inteligencia Artificial fue el desarrollo de juegos como la Dama o el Ajedrez:

- a) Búsqueda: Procesar todas las posibilidades
- b) Heurística: Corresponden a reglas que a diferencia de los algoritmos están basados en la experiencia
- c) Reconocimiento de patrones
- d) Aprendizaje de máquinas

Se ha tratado con diversas definiciones de IA, confluyendo en 2 enfoques:

- a) Intentar simular los procesos mentales humanos.
- b) Diseñar máquinas inteligentes independientes de la forma de pensar de los seres humanos.

Actualmente los computadores han tenido gran éxito en el manejo de la sintaxis de los lenguajes naturales no así de la semántica

Bases de Conocimientos y Sistemas expertos.

Bases de conocimientos se puede definir como una colección de información que incluyen hechos y un sistema para determinar y modificar las relaciones entre estos hechos.

Sistema experto: Programa de software diseñado para replicar el proceso de toma de decisiones de un experto humano.

Se han aplicado en diversas áreas: Ingeniería, matemática, Medicina. Por ejemplo MYCIN fue el primer sistema experto diseñado para capturar la experiencia de los médicos, superando en varias oportunidades a expertos en el diagnóstico de enfermedades.

Ej 2: AARON es un sistema experto que utiliza más de 1000 reglas de anatomía y comportamiento humano para crear dibujos de personas y objetos abstractos.

Reconocimiento de patrones:

Implica la identificación de patrones recurrentes en los datos de entrada con el objetivo de comprender y clasificar esa entrada:

- a) Análisis de imágenes.
- B) Reconocimiento óptico de caracteres
- c) Reconocimiento automático de la voz

Redes Neuronales:

Sistemas de cómputos distribuidos y paralelos inspirados en la estructura del cerebro humano. El computador sobresale en procesos de lógica lineal, diseñados para procesar secuencialmente la información digital con una CPU. El cerebro humano consta de miles de millones de neuronas conectadas cada una a miles de otras neuronas en una estructura distribuida con paralelismo masivo. Otras características:

- Es robusto y tolerante a fallas.
- Es flexible, se ajusta a nuevos ambientes mediante el aprendizaje
- Puede manejar información difusa
- Es pequeño, compacto y consume poca energía.

Aplicaciones:

- Procesamiento de imágenes y voz
- Reconocimiento de patrones
- Planeamiento
- Predicción
- Control y optimización
- Filtrado de señales