LICENCIATURA EN INFORMATICA

PROGRAMADOR UNIVERSITARIO

INGENIERÍA EN INFORMÁTICA

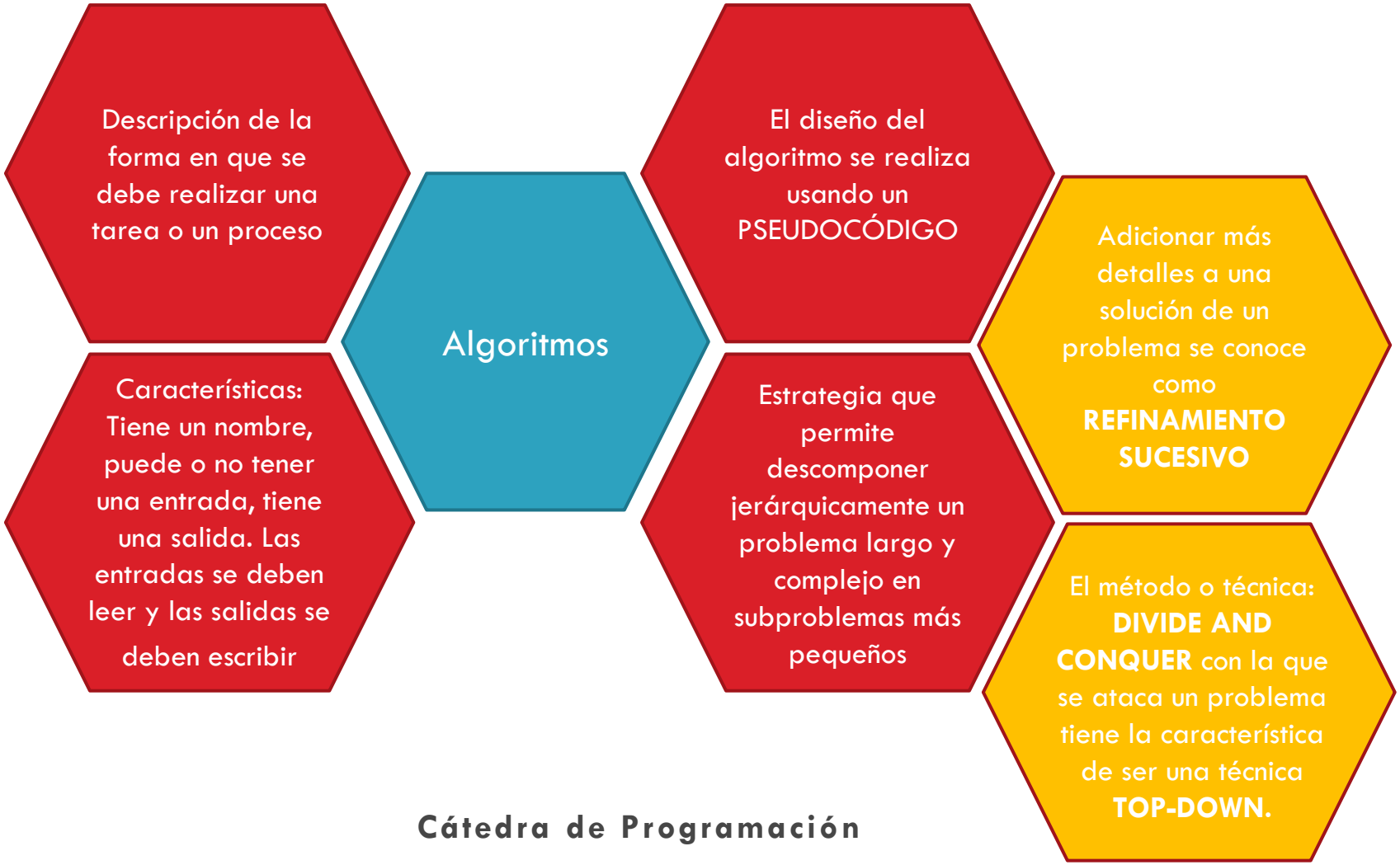
PROGRAMACIÓN

Unidad N° 1: El proceso de programación: Etapas en la resolución de problemas con computadora. Algoritmos. Formas de reducción de complejidad de problemas del mundo real.

2022

Lic. Mariela A. Velázquez

Repasemos lo visto



Comenzamos!!!

El proceso de programación

Comencemos con algunas preguntas …

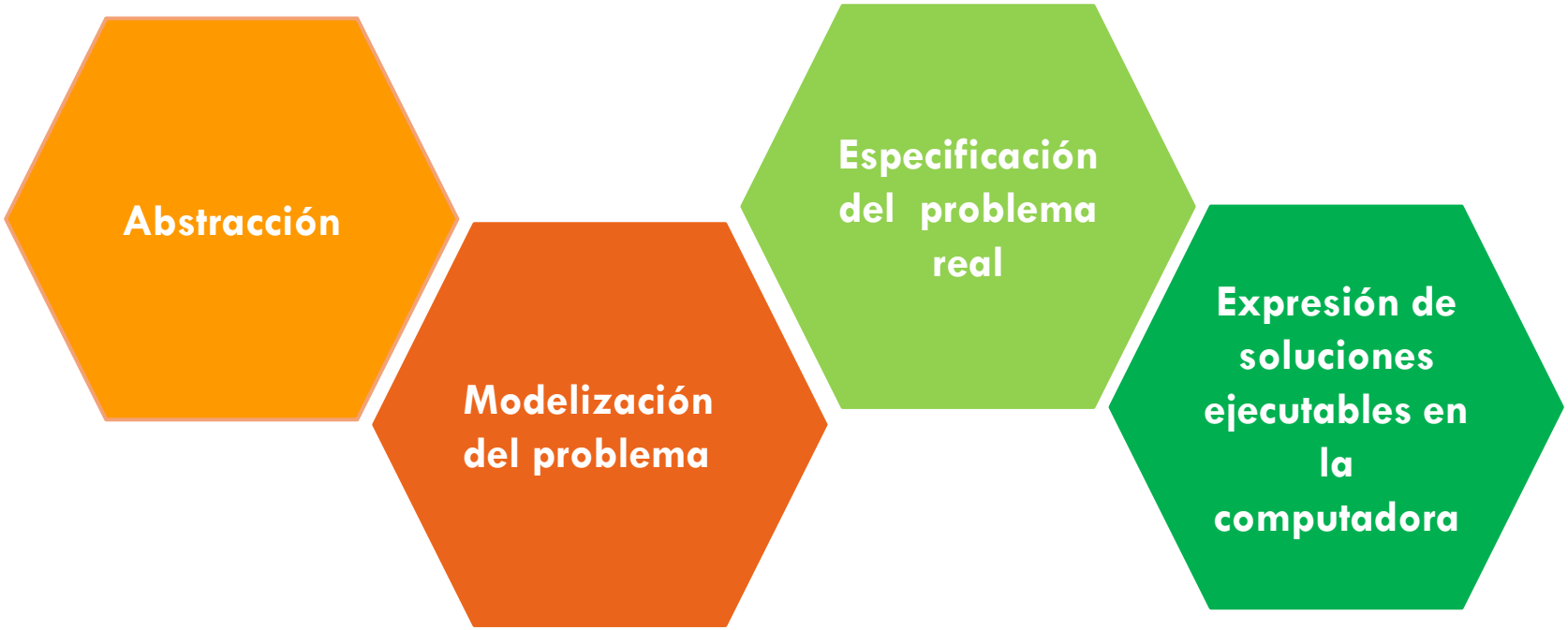
¿Cómo se resuelve un problema del mundo real con una computadora?

¿ Cómo se expresa la solución al problema planteado?

¿Cómo se reduce la complejidad de los problemas ?

El proceso de programación

El programador debe realizar algunos procesos intelectuales



Ejemplo

- Analicemos el proceso a través de un ejemplo

En una Universidad privada se ha establecido un programa que permita retener a los estudiantes en la casa de altos estudios. Para esto se tiene en cuenta las siguientes condiciones; el estudiante debe cursar 4 materias en cada período:

- Si el promedio obtenido en el último período es mayor o igual a 9, se hará un descuento del 20% sobre la cuota.
- Si el promedio es menor a 9 debe pagar la cuota completa.

Además se debe calcular el total a pagar por cada estudiante según el medio de pago:

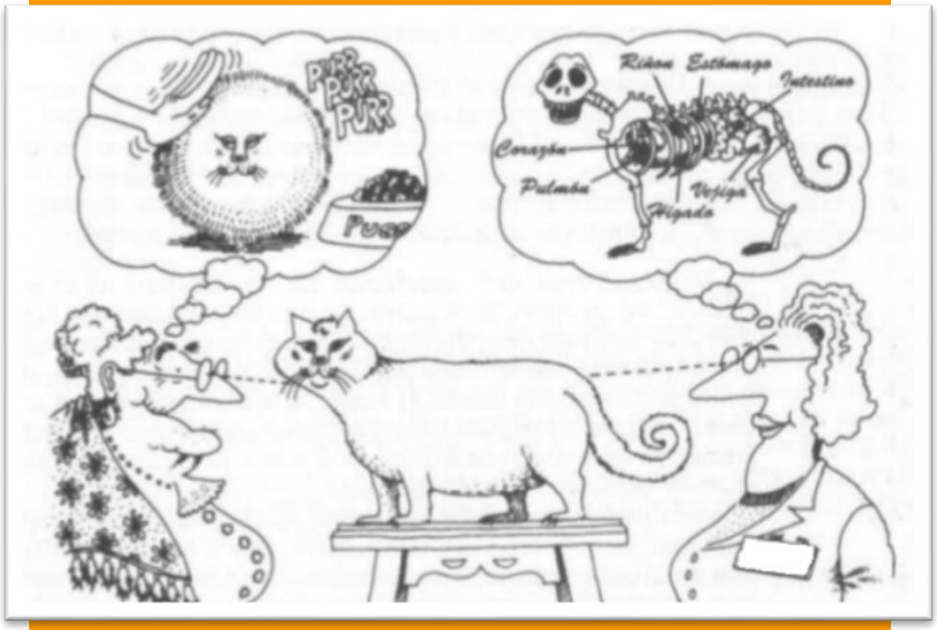
- Si paga con transferencia bancaria, se hace un descuento del 5%.
- Si paga con débito, la cuota no recibe modificaciones.
- Si paga con tarjeta de crédito, se aplica un interés del 10%.

La institución desea saber cuanto debe pagar cada alumno, y cuanto es el monto total recaudado en un numero no determinado de estudiantes.

Diseñe una solución modular que resuelva el problema dado.

Abstracción

Interpretar los aspectos esenciales de un problema y expresarlo en términos precisos.



Obtengamos los aspectos esenciales del problema

- ¿Qué debo calcular?

  La cuota que deberá pagar cada alumno y la recaudación total.

- ¿De qué depende el calculo?

  Del promedio del alumno y el método de pago.

- ¿Tiene importancia el periodo?

  Para realizar el calculo no es relevante el periodo en el que se encuentra.

- ¿El proceso se realizará más de una vez?

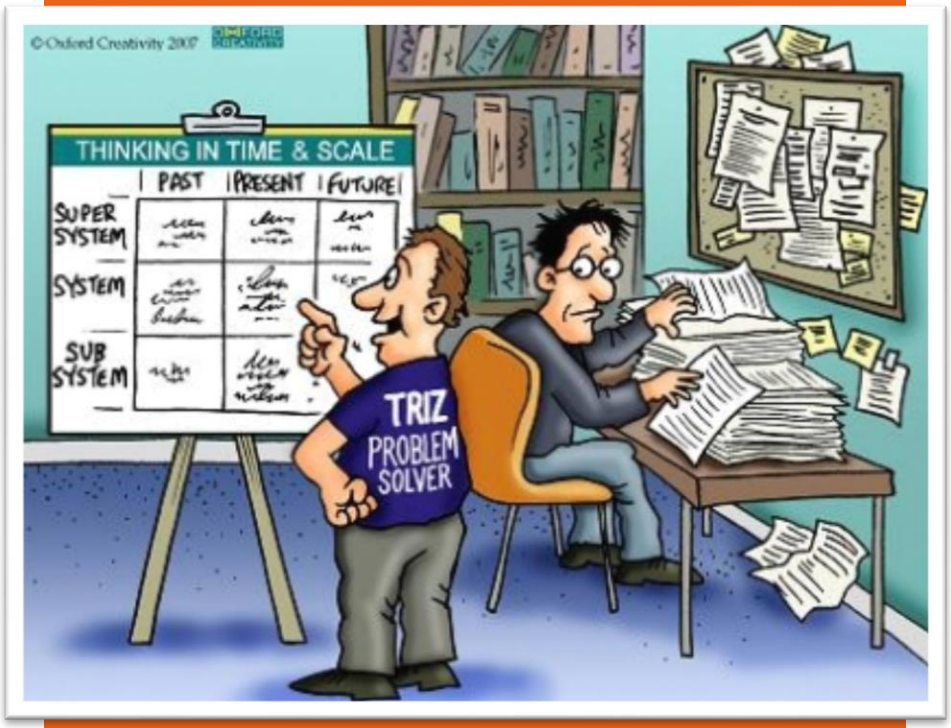
  Si, un número indeterminado.

Modelización del problema

¿Cuáles serán los datos necesarios para procesar?

- **Promedio del alumno**
- **Medio de pago**
- **Valor de la cuota**
- **DNI: dato que lo utilizo para marcar el final de una secuencia.**

Simplificar su expresión, encontrando sus aspectos principales, los datos que se habrán de procesar y el contexto.



Especificación del problema real

Determinar en forma clara y concreta el objetivo que se desea.

```
FUNCION CUOTAS(promedio, valorCuota, medioPago): real+, real+, caracter → real+
      SI (promedio >=9) ENTONCES
                  cuotaFinal ← valorCuota *0,8
      SINO
                  cuotaFinal ← valorCuota
      FIN_SI

      SEGUN (medioPago)
                  'T': cuotaFinal ← cuotaFinal * 0,95
                  'C': cuotaFinal ← cuotaFinal + (cuotaFinal*0,1)
      FIN_SEGUN

      RETORNA (cuotaFinal)
FIN_Función
```

Algoritmo Universidad
ENTRADA: dni: entero+, nota1,nota2, nota3, nota4, valorCuota: reales+, medioPago: carácter {T= transferencia, D= debito, C= credito}
SALIDA: totalRecaudado, cuotaTotal: real+
Vble AUX: promedio: real+

```
A0- totalRecaudado ← 0
A1- LEER(dni, valorCuota)
A2- MIENTRAS(dni <>0)
                  LEER(nota1, nota2, nota3, nota4, valorCuota, medioPago)
                  promedio ← (nota1+ nota2+ nota3+ nota4)/4
                  cuotaTotal ← CUOTAS(promedio, valorCuota, medioPago)
                  ESCRIBIR (cuotaTotal)
                  totalRecaudado ← totalRecaudado + cuotaTotal
                  LEER(dni)
      Fin_mientras
A3- ESCRIBIR (totalRecaudado)
A4- PARAR
```

Expresión de soluciones ejecutables en la PC.

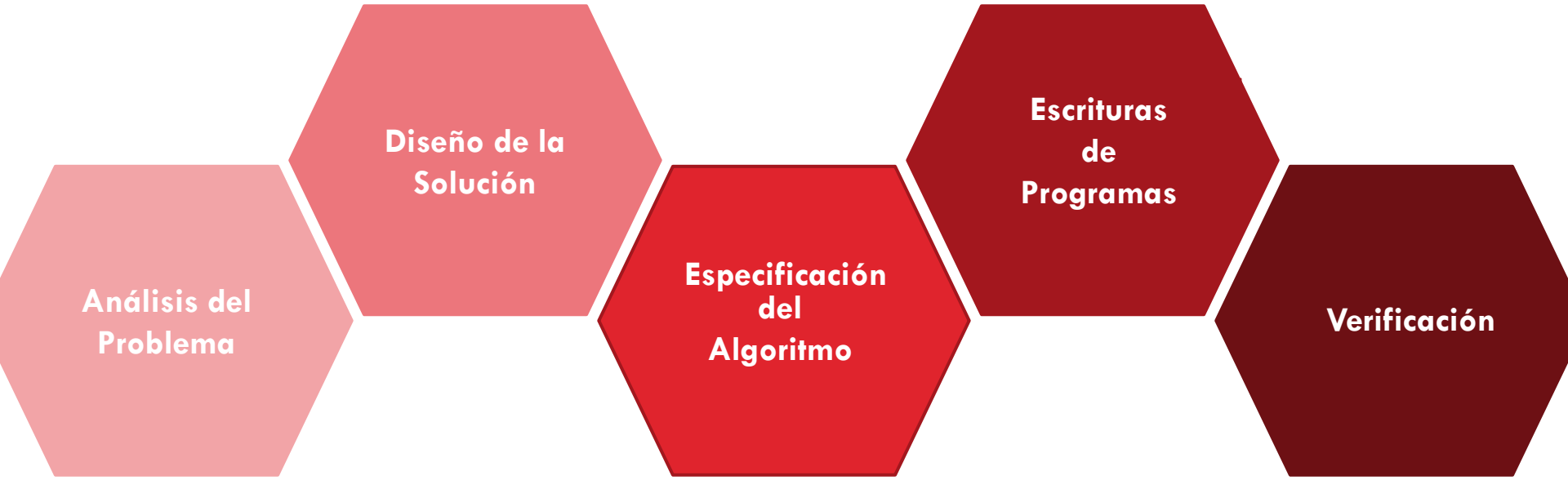
```c
#include <stdio.h>

float Cuotas(float promedio, float valorCuota, char medioPago);

int main()
{
    int dni;
    float nota1, nota2, nota3, nota4, valorCuota;
    char medioPago; // T= transferencia, D= debito, C= credito

    float totalRecaudado=0, cuotaTotal;
    float promedio;

    printf("Ingrese el dni del alumno:");
    scanf("%d", &dni);

    while(dni != 0)
    {
        printf("\n Ingrese las notas de las 4 materias:");
        scanf("%f %f %f %f", &nota1, &nota2, &nota3, &nota4);
        promedio = (nota1 + nota2 + nota3 + nota4)/4;

        printf("\n Ingrese el valor de la cuota:");
        scanf("%f", &valorCuota);
        fflush(stdin);
        printf("\n Ingrese el medio de pago:");
        scanf("%c", &medioPago);

        cuotaTotal = Cuotas(promedio, valorCuota, medioPago);
        totalRecaudado = totalRecaudado + cuotaTotal;

        printf("El alumno debe pagar: %f", cuotaTotal);

        printf("\nIngrese el dni del alumno:");
        scanf("%d", &dni);
    }

    printf("El total recaudado es: %f", totalRecaudado);
    return 0;
}
```

```c
float Cuotas(float promedio, float valorCuota, char medioPago)
{
    float cuotaFinal;


    if(promedio>=9)
    {
        cuotaFinal = valorCuota * 0.8;
    }
    else{
            cuotaFinal = valorCuota;
        }

    switch (medioPago)
    {
        case 'T':
          cuotaFinal = cuotaFinal * 0.95;
          break;

        case 'C':
          cuotaFinal = cuotaFinal + (cuotaFinal * 0.1);
          break;

    }

    return cuotaFinal;
}
```

Realizar una solución ejecutable en una computadora usando un lenguaje de programación.

Profundicemos este proceso…

Etapas de Resolución de Problemas



Análisis del problema.

La importancia del contexto:

La definición del contexto es importante para analizar y diseñar la solución usando computadoras.

Impone restricciones y consideraciones.

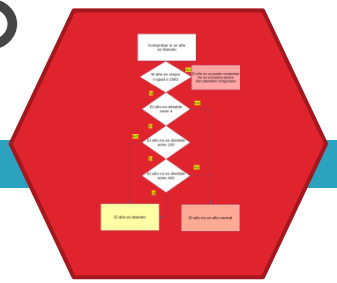
Diseño de la solución

Descomposición – Modularización:

Se usará la metodología top-down (arriba-abajo) de descomposición de problemas para desarrollar el sistema de software.

Se obtendrán módulos que deberán estar ligados entre si para obtener la solución final.

Especificación del Algoritmo



- Cada uno de los módulos habrá de tener su propio algoritmo.

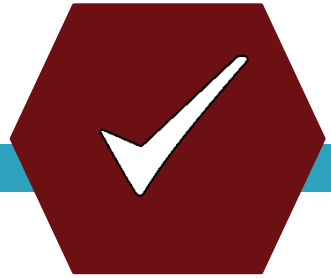
- La elección del algoritmo es importante , de ella depende la eficiencia de la solución.

Escritura de programas

```
#include<stdio.h>

int main()
{
    printf("Hola Mundo");
    return 0;
}
```

- Un algoritmo es una especificación simbólica que debe convertirse a un programa real sobre un lenguaje de programación concreto.

Verificación



Antes de dar por finalizada cualquier labor de programación, es fundamental preparar un conjunto de datos representativos del problema que permiten probar el programa cuando se ejecute, y así verificar resultados. Para esto se realizan:

- Pruebas (testing)
- Depuración
- Alternativas de diseño y estilo

Importante! : Cuanto mas exhaustivas sean las pruebas mayor seguridad se tendrá que el funcionamiento del programa es correcto, por lo tanto menor posibilidad de errores.

La buena programación…

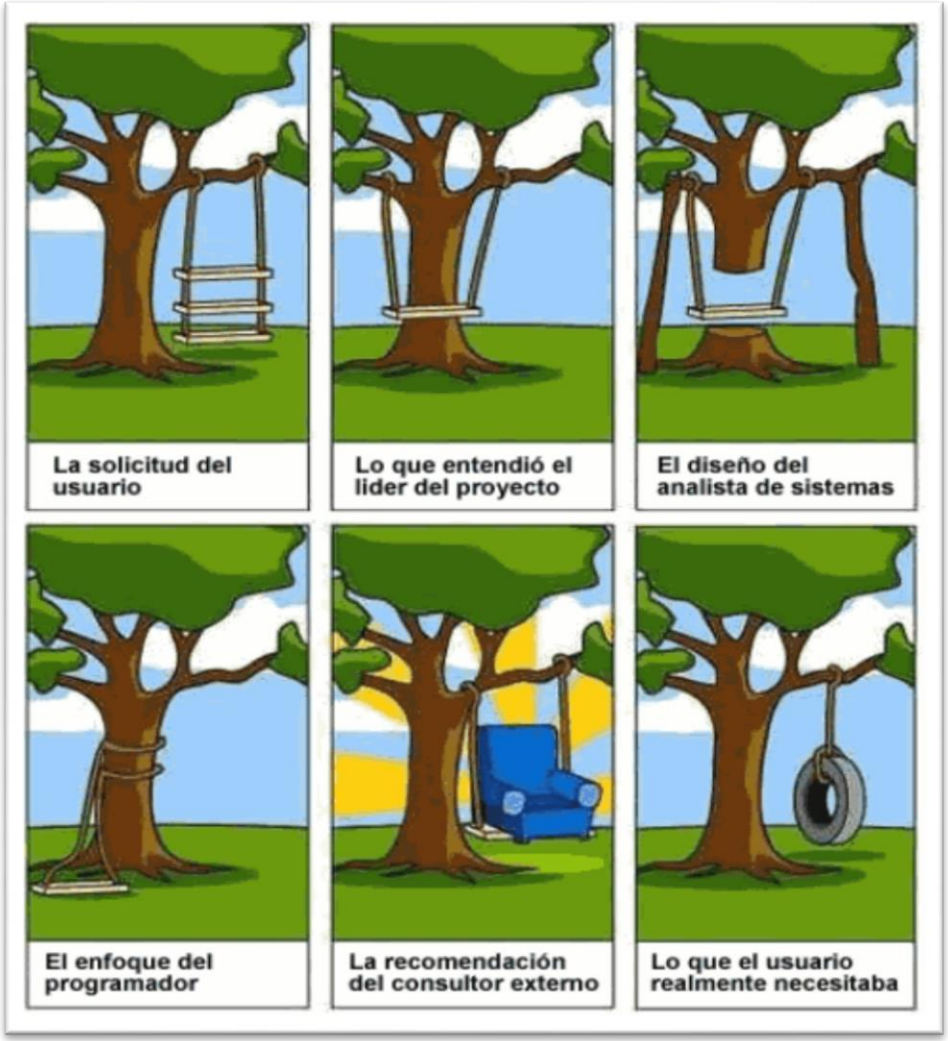
Un buen estilo hace que un programa sea fácil de leer e interpretar.

Los principios básicos : sentido común, lógica directa, expresión natural, nombres con significado, comentarios útiles, entre otros.

Tres criterios básicos a tener en cuenta

Correctitud…

Resultados deseados.



Tres criterios básicos a tener en cuenta

Claridad

Tres criterios básicos a tener en cuenta

Eficiencia…. rentabilidad en función de tiempo y espacio

```c
#include <stdio.h>

int main()
{
    printf("Hasta la próxima clase!!\n");

    return 0;
}
```