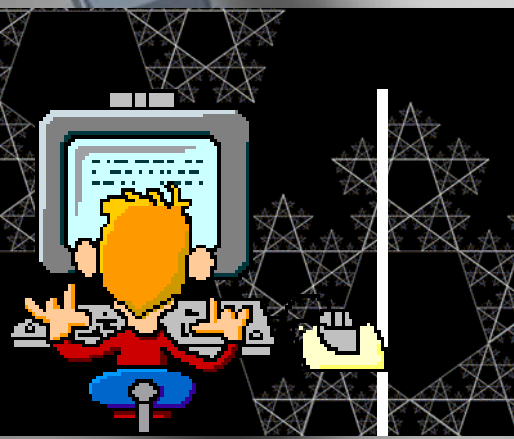




# ***Programación I***

***Ing. Carlos R. Rodríguez***

***Tecnicatura  
Universitaria en  
Programación***



UNIVERSIDAD TECNOLÒGICA NACIONAL  
*Facultad Regional Mendoza*



Soy **Carlos Rodríguez**, ingeniero de profesión y docente por vocación, y seré su guía para aprender a **descifrar** los *misterios* de la **programación** entendida como el **diseño y construcción** de una **solución** informática a un **problema**.

Los **pasos** que nos aseguran la **solución** del **problema** constituyen el **algoritmo**.



Para ello usaremos diversas **herramientas** que funcionan tanto en sistemas operativos **libres** (**Linux**) como en sistemas **propietarios** (**Mac, Windows**).





Aprenderemos a ***pensar creativamente*** en la solución de problemas.

Así, una vez aprendamos ***métodos de solución*** para problemas informáticos, los aplicaremos para casi cualquier **problema**, lo cual nos será de gran utilidad.

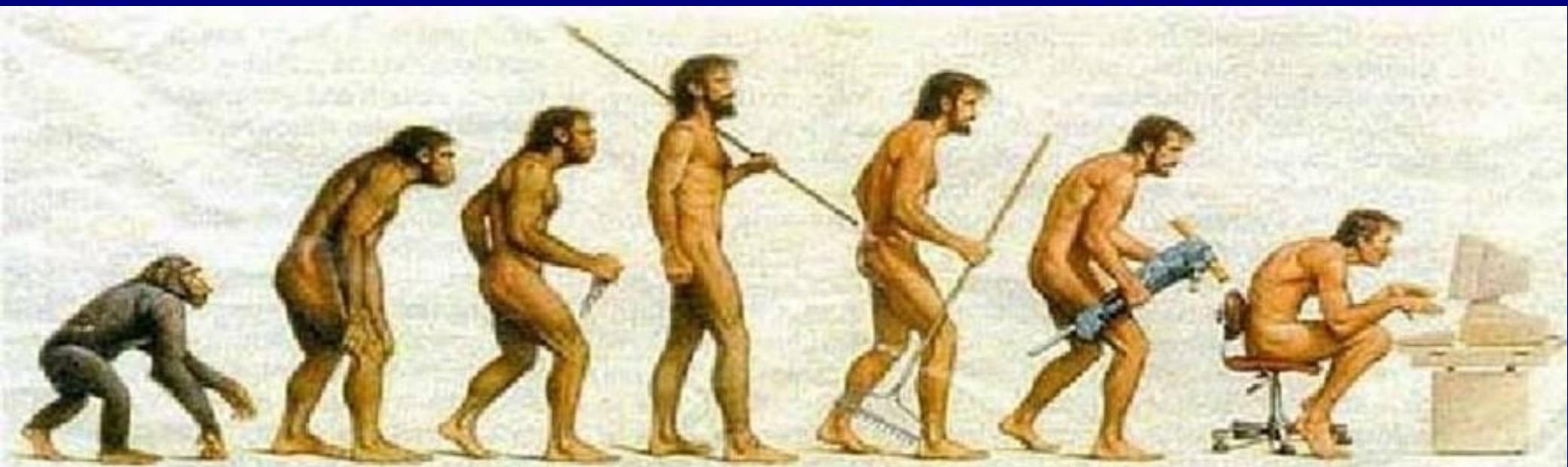






*“Un problema abstracto permite establecer formalmente la relación deseada entre la entrada de un algoritmo y su salida.”*

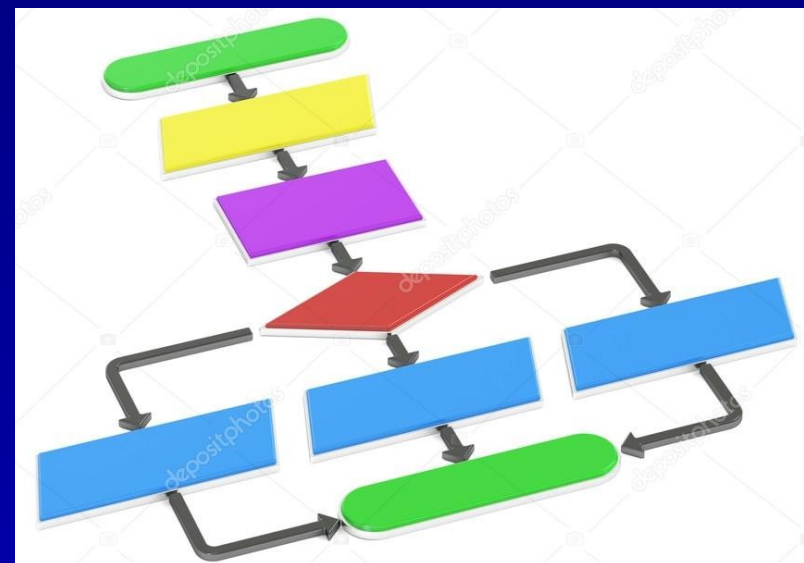
- Fuente: [http://es.wikipedia.org/wiki/Problema\\_computacional](http://es.wikipedia.org/wiki/Problema_computacional)





*“Una solución algorítmica a un problema abstracto consiste de un algoritmo que **por cada** instancia del problema **calcula al menos una** solución **correspondiente** –en caso de haberla– o **expide un certificado de que no existe solución alguna**.”*

- Fuente: [http://es.wikipedia.org/wiki/Problema\\_computacional](http://es.wikipedia.org/wiki/Problema_computacional)





*“Un problema abstracto se convierte en un problema concreto cuando las instancias y soluciones están codificadas en forma de lenguajes formales.*

```
#include <program1.h> // Archivo de traducción n d
/**
 * Enunciado: Euclides estableció un algoritmo que lleva su nombre y que permite calcular
 * el máximo común divisor (mcd) entre dos números naturales
 * a y b (siendo a > b) de la siguiente forma:
 * Si b = 0 entonces mcd(a,b) = a y el algoritmo termina.
 * En otro caso, mcd(a,b) = mcd(b,r) donde r es el resto de dividir a entre b.
 * Para calcular mcd(b,r) se utilizan estas mismas reglas.
 * Ejemplifique para a = 54321 y b = 432 y otros 2 valores de su elección.
 */
funcion entero mcd(entero a, entero b) {
    entero resul;
    si(b ES 0) entonces
        resul = a;
    sino
        resul = mcd(b,a % b);
    finSi
    regresa(resul);
}

principal // Unidad de programa princ
    entero uno, dos; // Limpia la pantalla.
    limpiar;
    repetir
        leerM(uno,"Primer valor:");
        hasta(uno > 0);
    repetir
        leerM(dos,"Segundo valor:");
        hasta(dos > 0 Y uno > dos);
    mostrar << "El mcd entre " << uno << " y " << dos << " es:" << mcd(uno,dos) << salto;
    pausa; // Pausa antes de finalizar
    finPrincipal
```

◦ Fuente: [http://es.wikipedia.org/wiki/Problema\\_computacional](http://es.wikipedia.org/wiki/Problema_computacional)





Es la puerta de acceso a un mundo fascinante donde el conocimiento está – *literalmente* – **al alcance de las manos** y **es gratis** (aunque quizás nos demande bastante esfuerzo). Y como el conocimiento es poder, eso nos volverá más libres.

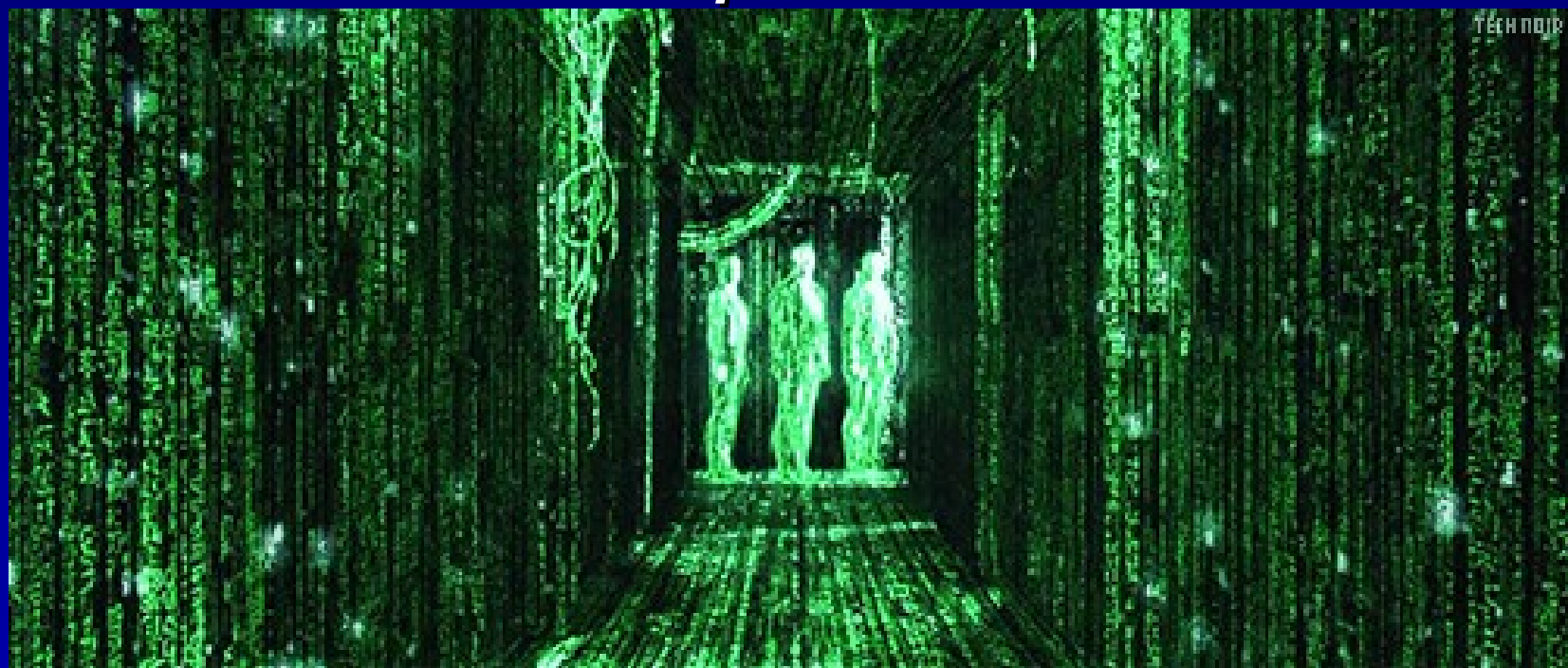






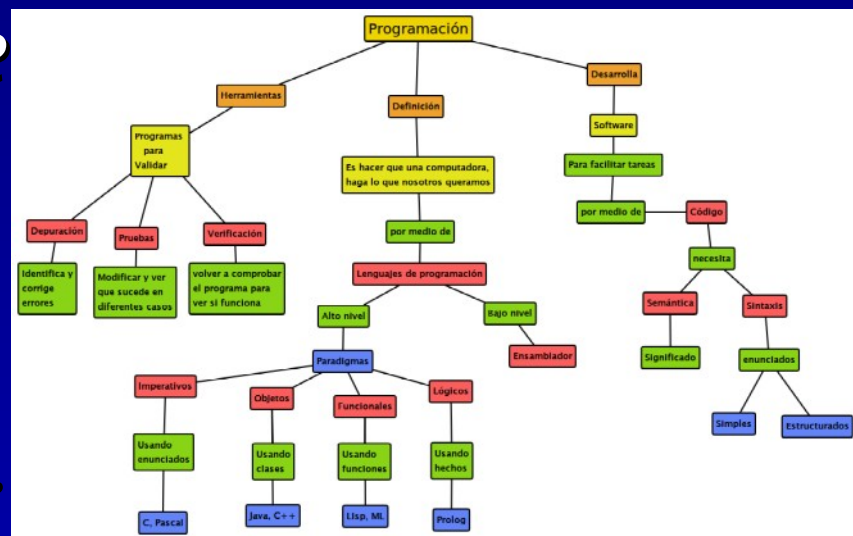
Así, **investigamos y aprendemos *haciendo***, y veremos que puede – *y debe* – ser **divertido**.

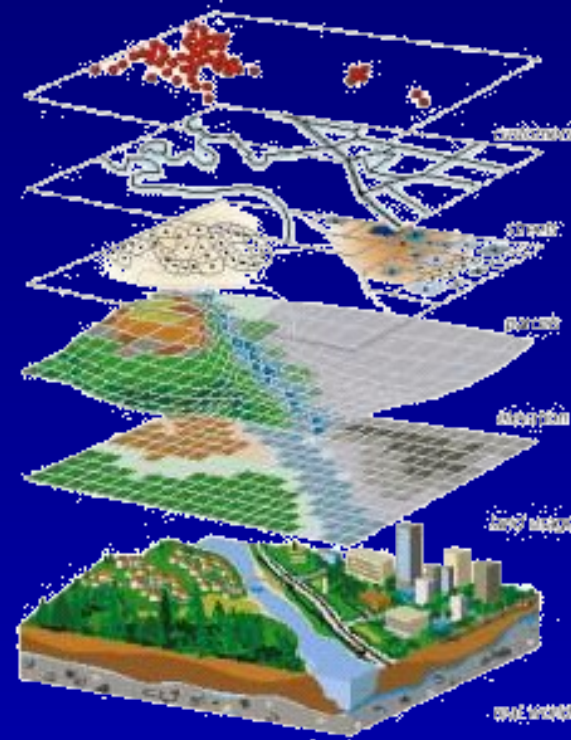
Daremos pues, unos cuantos pasos en el camino del **conocimiento**. *¡Toda una aventura!*





"Se propone un formato de **aula taller**, para abordar conceptos que básicamente persiguen el objetivo de **incrementar** en el alumno su **natural habilidad** para utilizar el **pensamiento** elaborado **lógicamente** a fin de resolver **problemas** a través de la descomposición de los mismos en **subproblemas** más simples (**diseño descendente**).









Se la impulsa en el diseño de **soluciones** a problemas planteados.

- Porque **no existe una única "verdad" (paradigma)** ni un único camino - sino quizás muchos - para alcanzarla.

## ¿SABÍAS QUE?

Tu cerebro tiene la habilidad de cambiar la dirección del tren, **solo tienes que pensarlo**



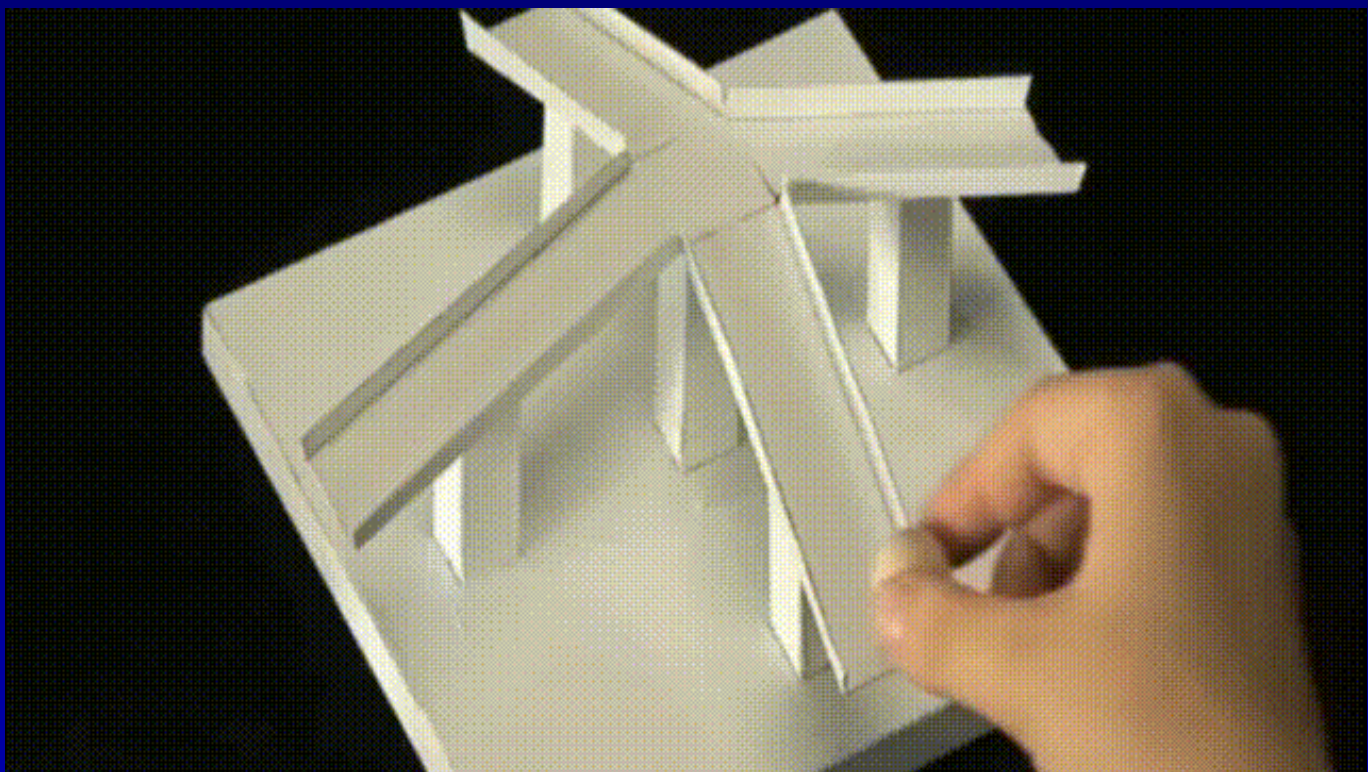


Normalmente **hay varios** puntos de vista posibles.

- Es bueno **pensar** *antes de decidir*.

- Usar varios enfoques.

- Lo que **vemos** puede ser **una realidad**, pero **no siempre es completa**.

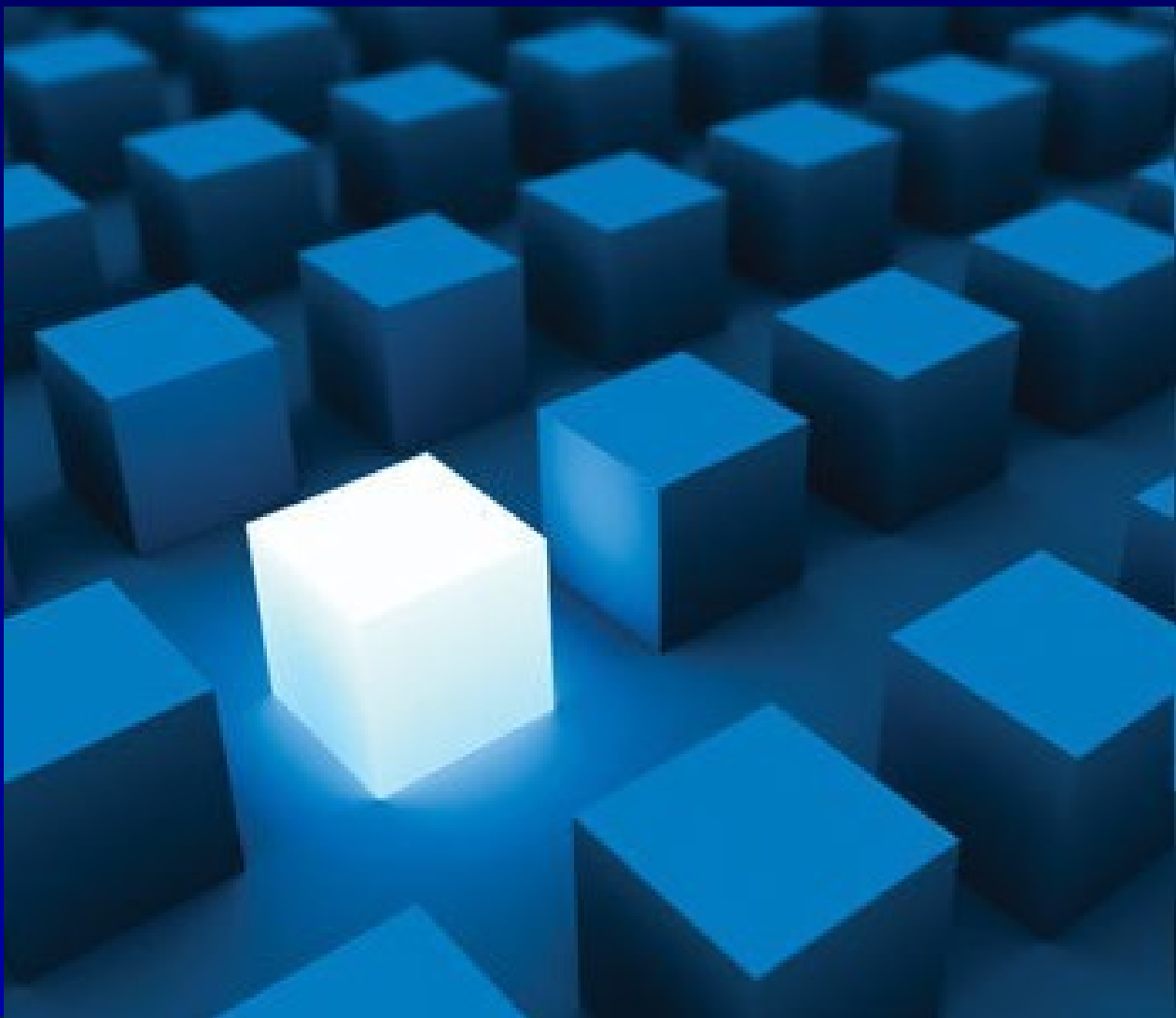




En Programación - y  
*en otras áreas* - es  
necesario  
establecer  
principios de buen  
diseño,  
***minimizando:***

- El riesgo de error.
- El esfuerzo para lograr la

***solución***

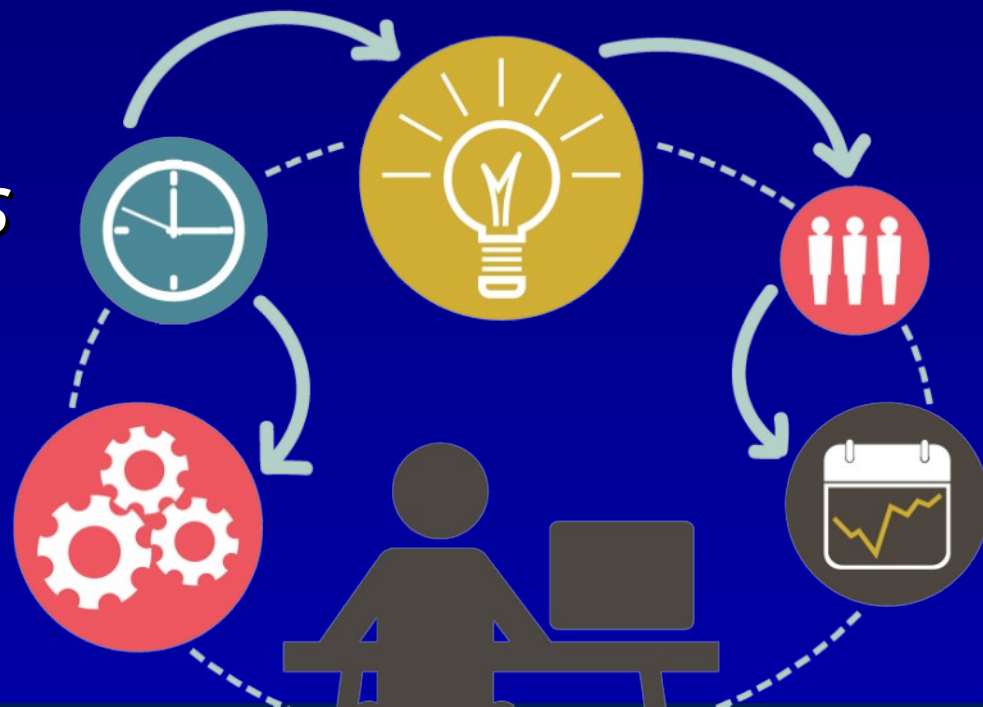






## Capacidad de **Planificación**:

- El desarrollo de software es una **tarea compleja** que *vincula múltiples elementos en el ensamblado de la solución a un problema.*
- Requiere de la **descomposición de tareas** en *subtareas más sencillas*, con sus **recursos** y un **orden de precedencia** entre ellas.





Para que una **idea** sea **comunicable** es necesario que esté **bien formada** y justificada.

- Para comunicarla a otros es necesario además **organizar los contenidos** en forma **lógica**, respetando las **precedencias** y - en general - una **complejidad** creciente.





Buscamos generar:

- **Autoconfianza.**
- Perseverancia.
- Creatividad en la búsqueda de soluciones.
- **Gusto por el trabajo autónomo y en grupo.**







Por ejemplo:

- Disposición positiva para **cooperar** en *trabajos grupales*.
- **Respeto** por las *opiniones diferentes*.





## Desarrollo de la curiosidad:

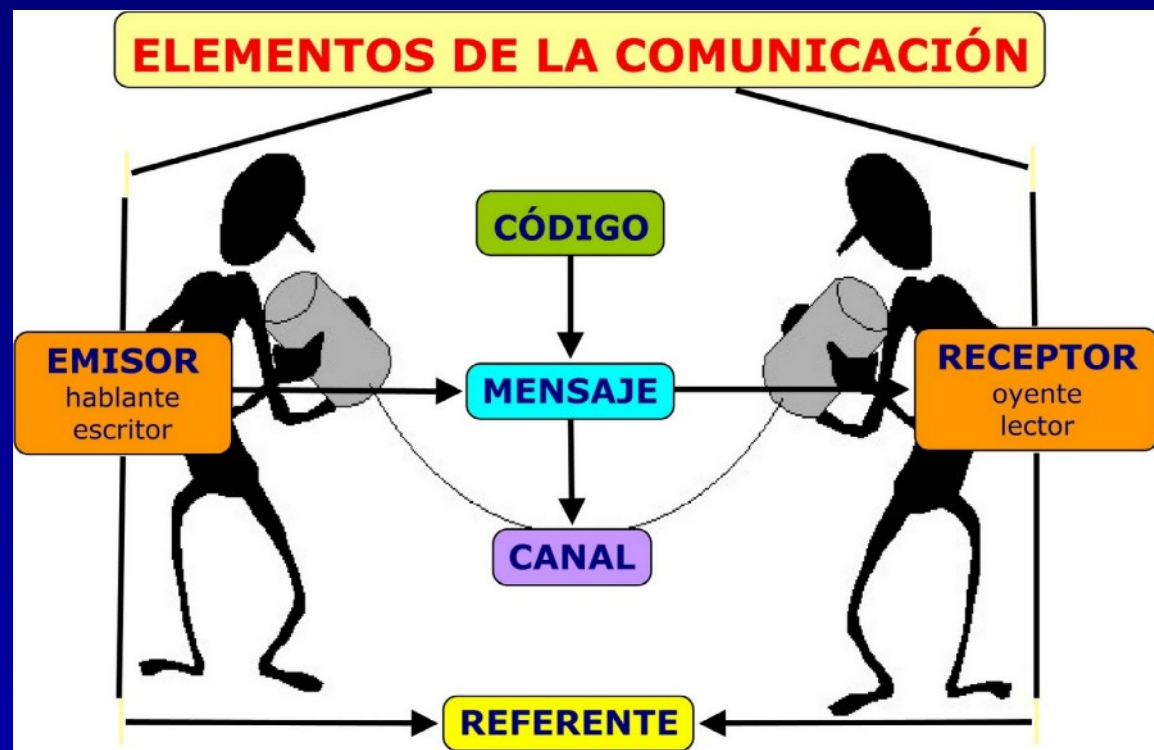
- Disposición positiva hacia la **indagación** y los **desafíos** de la **programación**.
- Desarrollo de la **visión autocrítica** de problemas y **soluciones**.
- Aporte de **diferentes ciencias**.
- Cuidado y uso **racional** de recursos.





## Desarrollo de habilidades:

- Uso de **vocabulario preciso**.
- **Elaboración** de la argumentación.
- **Análisis reflexivo y crítico** de cualquier argumentación.
- Claridad, calidad y pertinencia en la **presentación de resultados**.

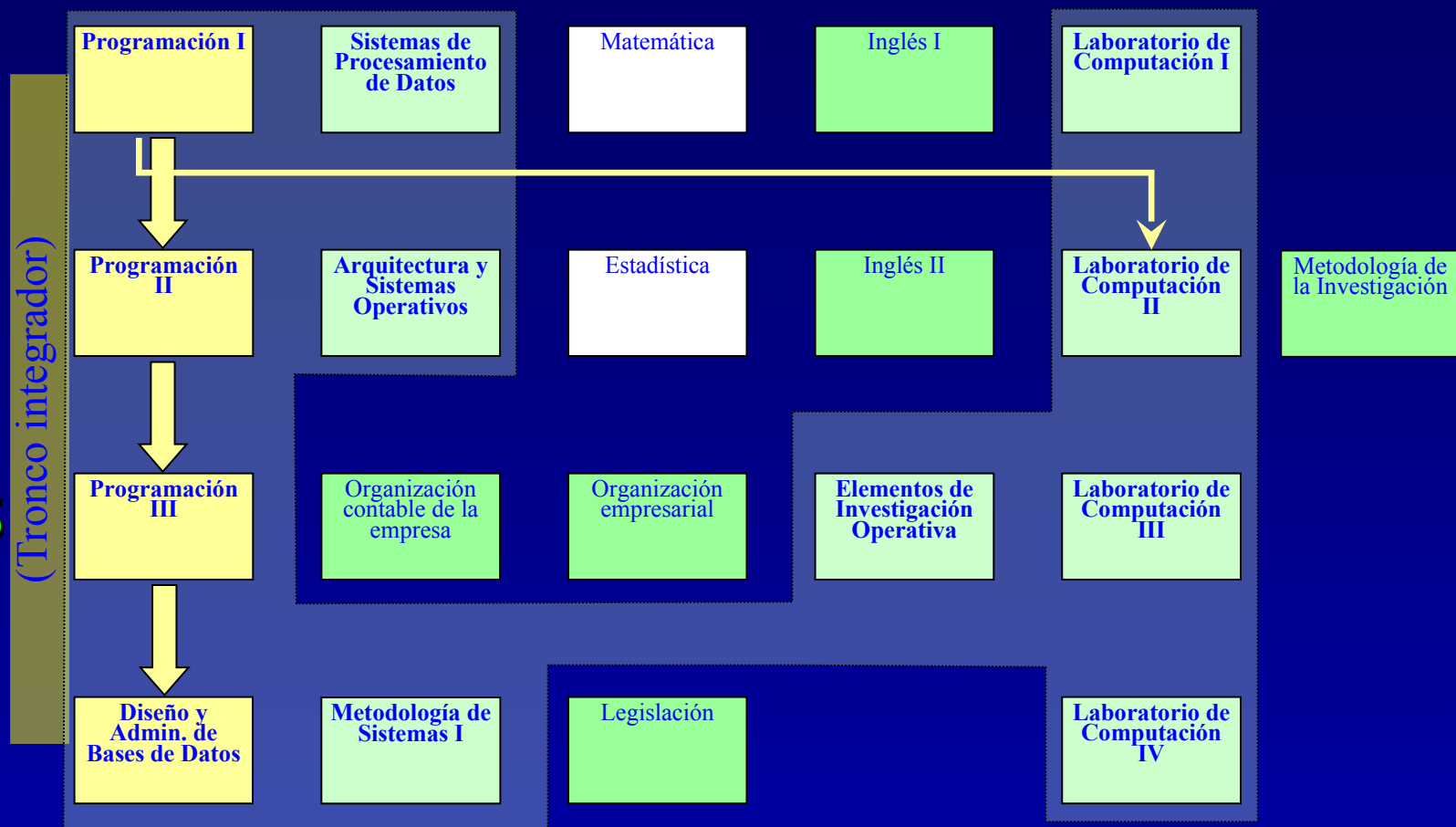






Forma parte  
del tronco  
integrador,  
según el  
diseño  
curricular.

Son **6** horas  
cátedra  
semanales y **15**  
semanas,  
equivalentes a  
**90** horas  
cátedra.





## Evaluación:

### – **Continúa**

- **Prácticos** (aprobados, permiten la promoción por las evaluaciones puntuales)


### – **Puntual**

- **A carpeta abierta**
  - Nivel de complejidad creciente con cada evaluación
  - Teórico – práctico
- **Globalizadora**





## Evaluación Parcial 1:

|  |             |  |            |                                 |
|---|-------------|--|------------|---------------------------------|
| Semana<br>N°  | Uni-<br>dad | Contenidos                                 | Objetivos  | Actividades                     |
|   |             | selectiva                                  |            |                                 |
| 5   | 1 a 3       | Secuencias,<br>expresiones y<br>decisiones | Evaluación | Prácticos 1<br>a 8<br>aprobados |

| < septiembre > |     |     |     | < 2021 > |     |     |
|----------------|-----|-----|-----|----------|-----|-----|
| dom            | lun | mar | mié | jue      | vie | sáb |
| 29             | 30  | 31  | 1   | 2        | 3   | 4   |
| 5              | 6   | 7   | 8   | 9        | 10  | 11  |
| 12             | 13  | 14  | 15  | 16       | 17  | 18  |
| 19             | 20  | 21  | 22  | 23       | 24  | 25  |
| 26             | 27  | 28  | 29  | 30       | 1   | 2   |





## Evaluación Parcial 2:

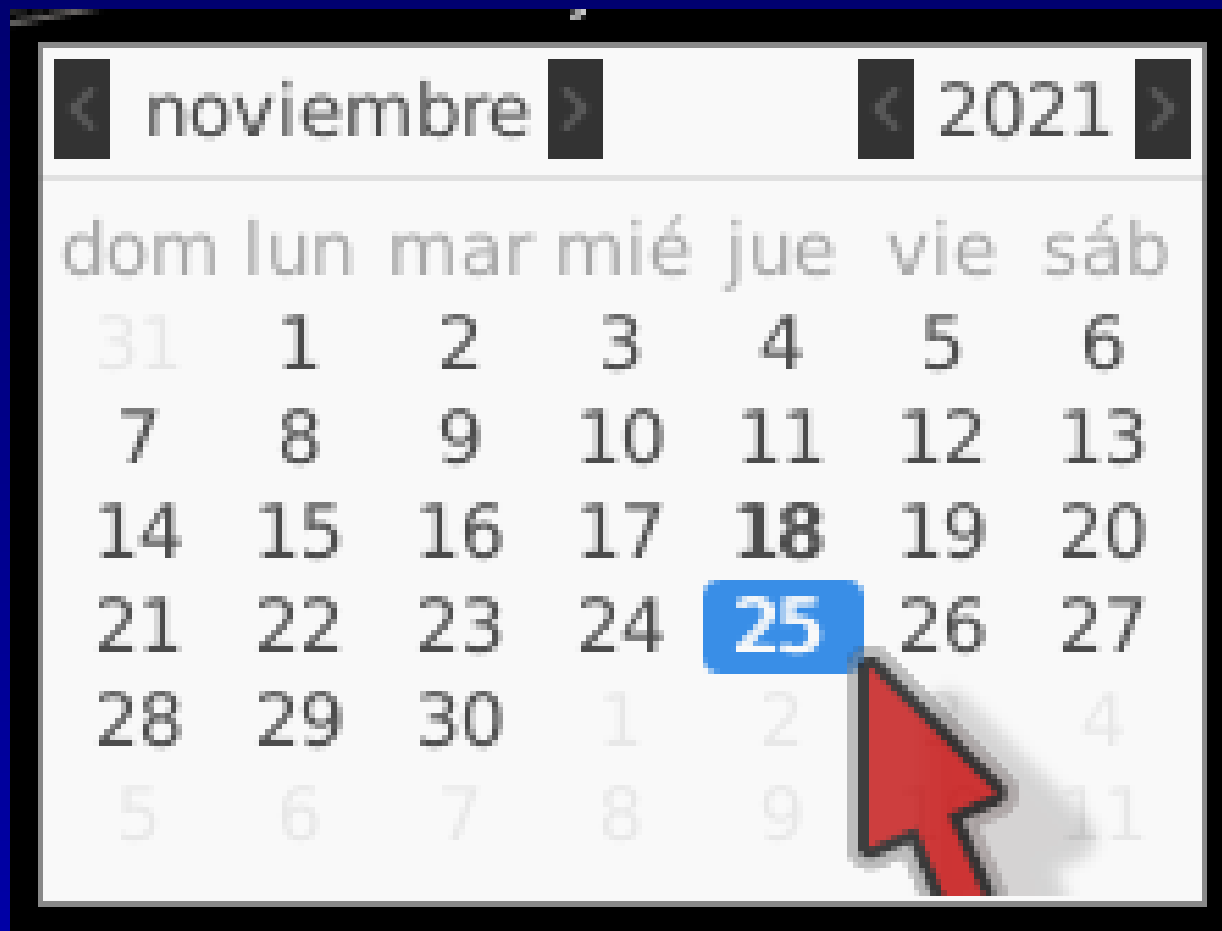
|    |       |                                  |            |                            |
|----|-------|----------------------------------|------------|----------------------------|
| 10 | 4 a 6 | Bucles y variables dimensionadas | Evaluación | Prácticos 9 a 19 aprobados |
|----|-------|----------------------------------|------------|----------------------------|

| < octubre > |     |     |     |     |     |     | < 2021 > |  |
|-------------|-----|-----|-----|-----|-----|-----|----------|--|
| dom         | lun | mar | mié | jue | vie | sáb |          |  |
| 26          | 27  | 28  | 29  | 30  | 1   | 2   |          |  |
| 3           | 4   | 5   | 6   | 7   | 8   | 9   |          |  |
| 10          | 11  | 12  | 13  | 14  | 15  | 16  |          |  |
| 17          | 18  | 19  | 20  | 21  | 22  | 23  |          |  |
| 24          | 25  | 26  | 27  | 28  | 29  | 30  |          |  |
| 31          | 1   | 2   | 3   | 4   | 5   | 6   |          |  |



## Evaluación Global:

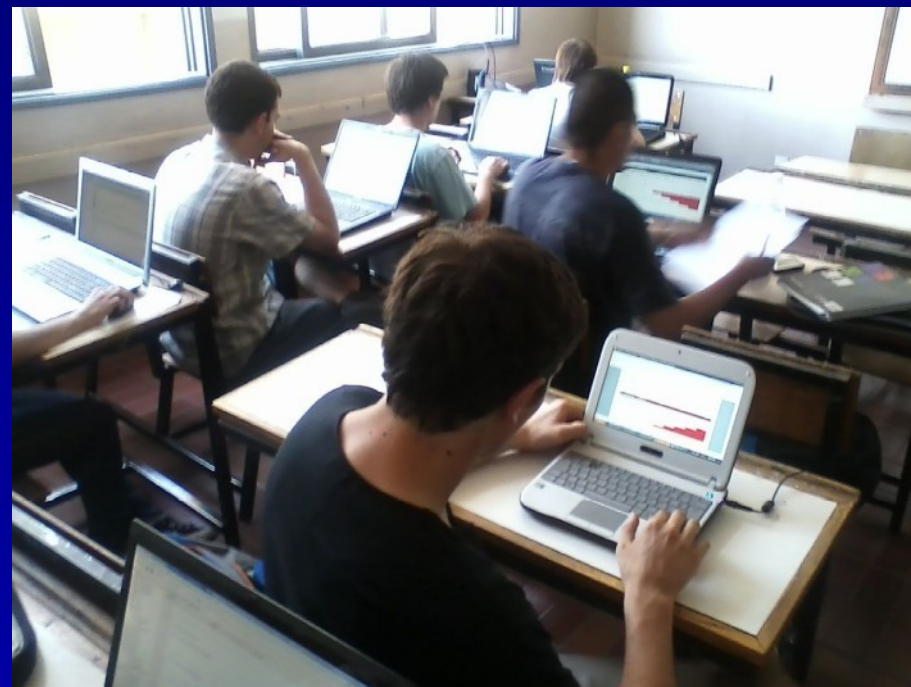
|    |       |       |            |                                   |
|----|-------|-------|------------|-----------------------------------|
| 14 | 1 a 7 | Todos | Evaluación | Prácticos<br>20 a 23<br>aprobados |
|----|-------|-------|------------|-----------------------------------|





Cálculo de la Nota (con los 2 *parciales* y el *global* aprobados con 60% o más):

$$\text{NotaFinal} = \frac{\text{Parcial}_1 + (\text{Parcial}_2 * 2) + (\text{Global} * 4)}{7}$$







Estados posibles al terminar el cursado:

- **AD - Promocionado**

- Cuando la **nota final** (*redondeada al entero más próximo*) es **mayor o igual a 6** (seis).

- **AND - Regular**

- ▮ Cuando la **nota final** es **menor o igual a 5** (cinco) y **mayor o igual a 4** (cuatro)

- **Libre**

- ▮ Cuando la **nota final** es **menor a 4** y no se ha aprobado el recuperatorio

