

Fábrica Digital

El Rule

Laboratorio de Aplicaciones Culturales

Septiembre 2017

- ☐ **Arquitectura de Software**
- ☐ **Software Libre**
- ☐ **Campos de Aplicación**
- ☐ **Lenguajes de Programación**
- ☐ **Herramientas y Tecnologías**
- ☐ **Aplicaciones Web**
- ☐ **Aplicaciones Interactivas**
- ☐ **Conceptos Básicos de Programación**

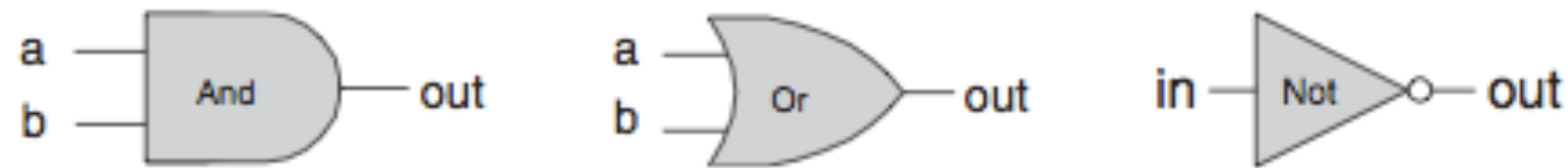
- Conceptos básicos
- Paradigmas de programación
- Algoritmos y estructuras de datos
- Metodologías de desarrollo

Un binario, o ejecutable es sólo una lista de instrucciones no-ambiguas en lenguaje máquina que están destinadas a ser ejecutadas por un procesador.

Una aplicación es uno o más binarios junto con elementos adicionales como imágenes, audio, vídeo, tipografía, bases de datos y texto.

Álgebra de Boole

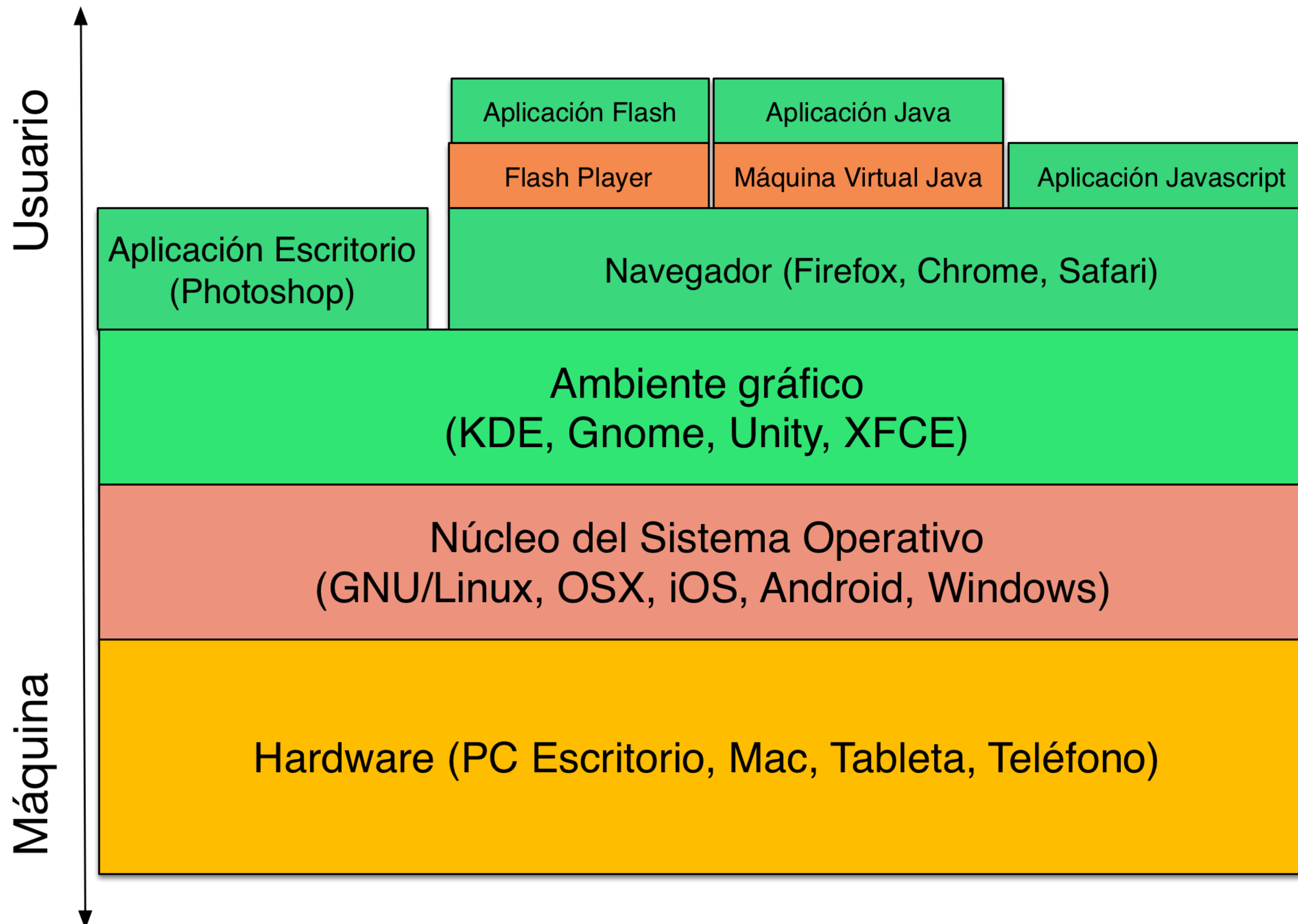
El álgebra de Boole proporcionó las bases para que los pioneros en ingeniería electrónica construyeran **compuertas lógicas** que emulaban las operaciones del álgebra booleana: conjunción, disyunción, negación y todas las operaciones derivadas de éstas.



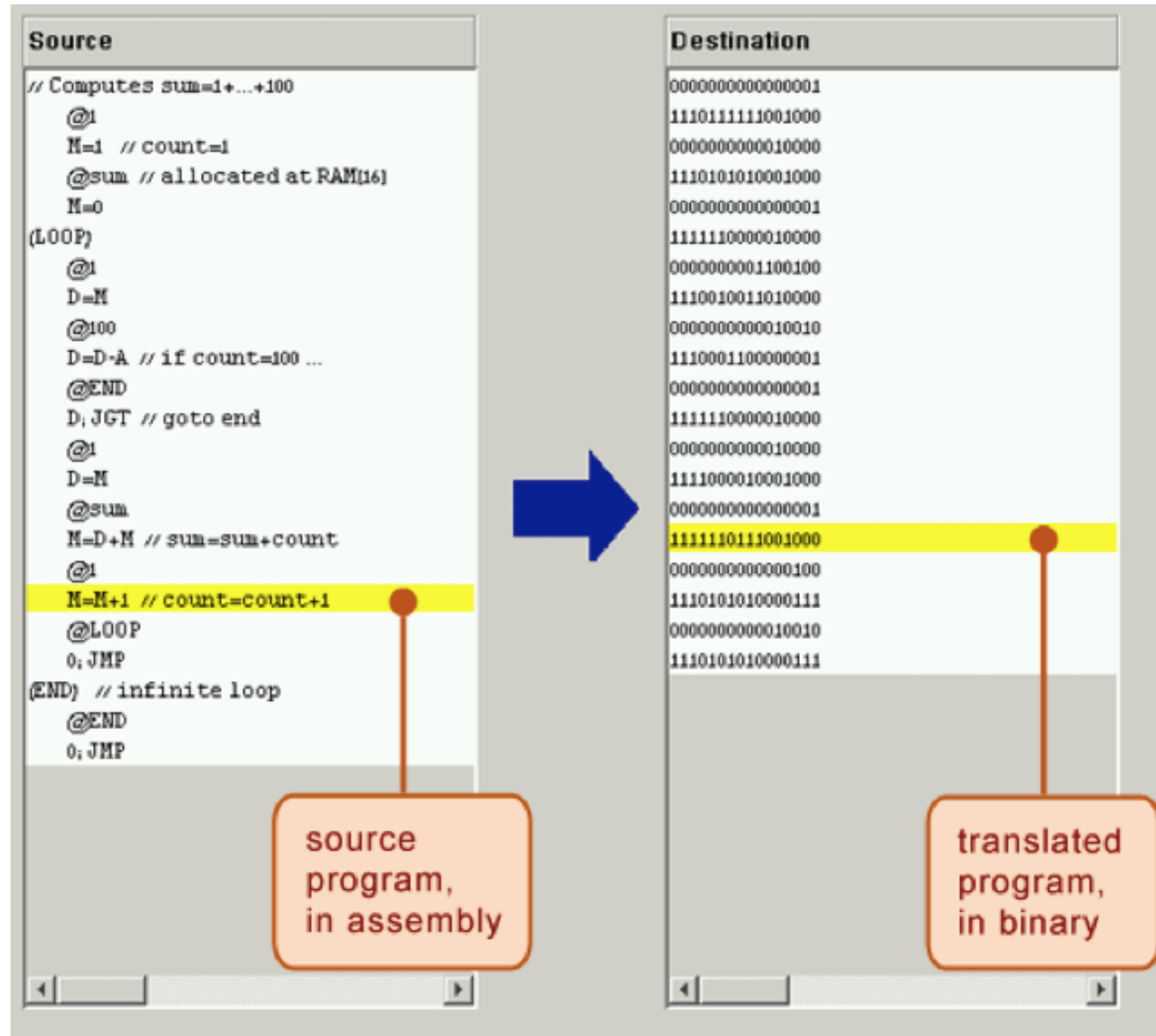
x	y	$x \wedge y$	$x \vee y$
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	1

x	$\neg x$
0	1
1	0

Lenguaje Máquina



Lenguaje Ensamblador



Ejemplo de código en ensamblador y su traducción en binario de 16 bits.

Paradigmas de programación

Imperativo			Declarativo		
Guiados por ejecución de órdenes o comandos, ¿Qué hacer y cómo hacerlo?			Basado en declaración de reglas, condiciones y restricciones. No se describe un flujo de datos fijo.		
Procedimental	C	Cobol	Haskell	Lisp	Funcional
	Pascal	Fortran	ML	Scheme	
Orientado a Objetos	C++	Smalltalk	GHC	Prolog	Lógico
	Java	C#	Datalog	Answer Set Programming	

Código C++ (Imperativo — Orientado a Objetos)

```
double maxiOsc::noise() {  
    //White Noise  
    //always the same unless you seed it.  
    float r = rand()/(float)RAND_MAX;  
    output=r*2-1;  
    return(output);  
}  
  
void maxiOsc::phaseReset(double phaseIn) {  
    //This allows you to set the phase of the oscillator to anything you like.  
    phase=phaseIn;  
}  
  
double maxiOsc::sinewave(double frequency) {  
    //This is a sinewave oscillator  
    output=sin (phase*(TWOPI));  
    if ( phase >= 1.0 ) phase -= 1.0;  
    phase += (1./(maxiSettings::sampleRate/(frequency)));  
    return(output);  
}
```

Código Haskell (Declarativo — Funcional)

```
prop_ordered xs = ordered (qsort xs)  
    where ordered []      = True  
          ordered [x]    = True  
          ordered (x:y:xs) = x <= y && ordered (y:xs)  
  
prop_permutation xs = permutation xs (qsort xs)  
    where permutation xs ys = null (xs \\ ys) && null (ys \\ xs)  
  
prop_maximum xs      =  
    not (null xs) ==>  
        last (qsort xs) == maximum xs  
  
prop_append xs ys      =  
    not (null xs) ==>  
    not (null ys) ==>  
        head (qsort (xs ++ ys)) == min (minimum xs) (minimum ys)
```

Algoritmos

Un **algoritmo** es un conjunto de instrucciones no ambiguas que permiten ejecutar una tarea o resolver un problema de modo general.

- Una receta de cocina es un algoritmo.
- Las instrucciones para ensamblar un mueble son un algoritmo.
- El método para hacer una división o una multiplicación de números con varias cifras son algoritmos.

Algoritmos

Factorial $n = n!$
(n es un número natural)
 $n! = n (n-1) (n-2) \dots 3 \cdot 2 \cdot 1$

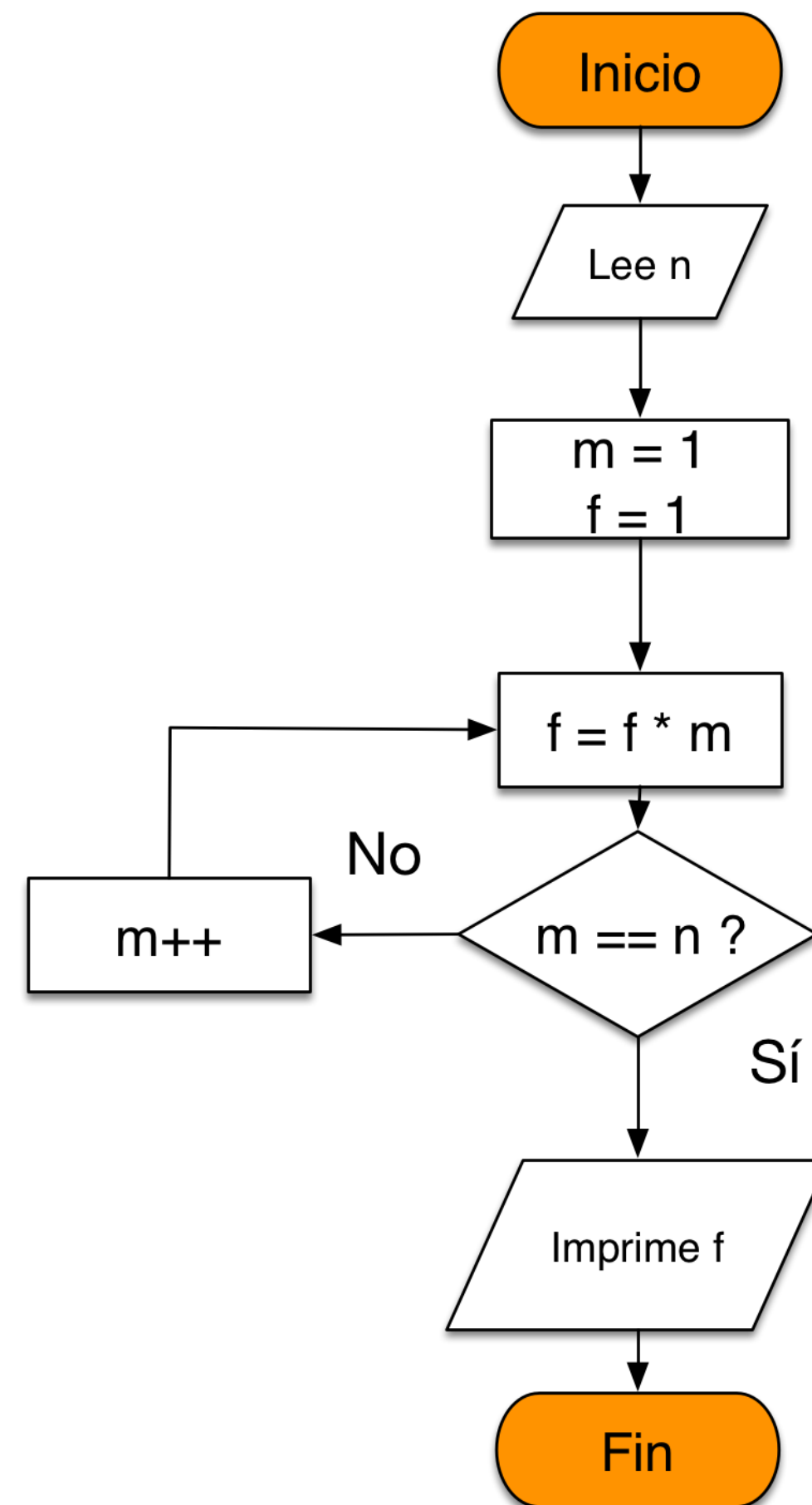


Diagrama de flujo
para calcular el factorial ($n!$)
de un número n

Estructuras de Datos

¿Cómo abstraer información de modo eficiente?

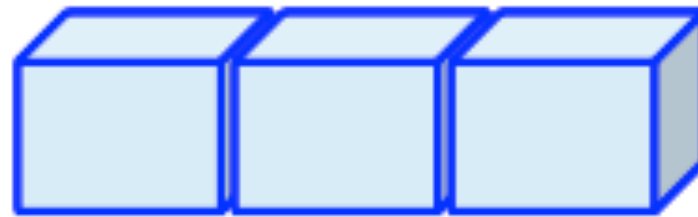
Las estructuras de datos son espacios de memoria que almacenar información de modo eficiente para ser utilizados en programas.

Estructuras de Datos

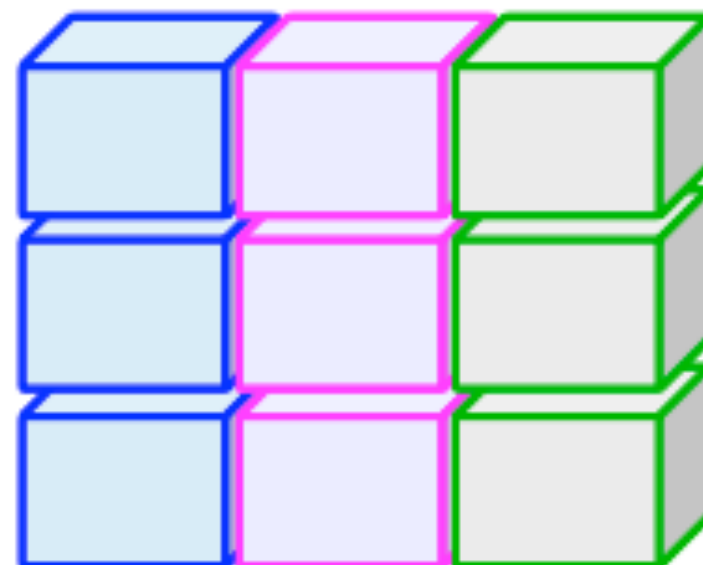
Primitivas	Compuestas	Abstractas
Booleano	Arreglo	Lista
Entero	Tupla	Pila
Caracter	Union	Cola
Punto Flotante	Punto Flotante	Árbol
Puntero		Grafo

Estructuras de Datos

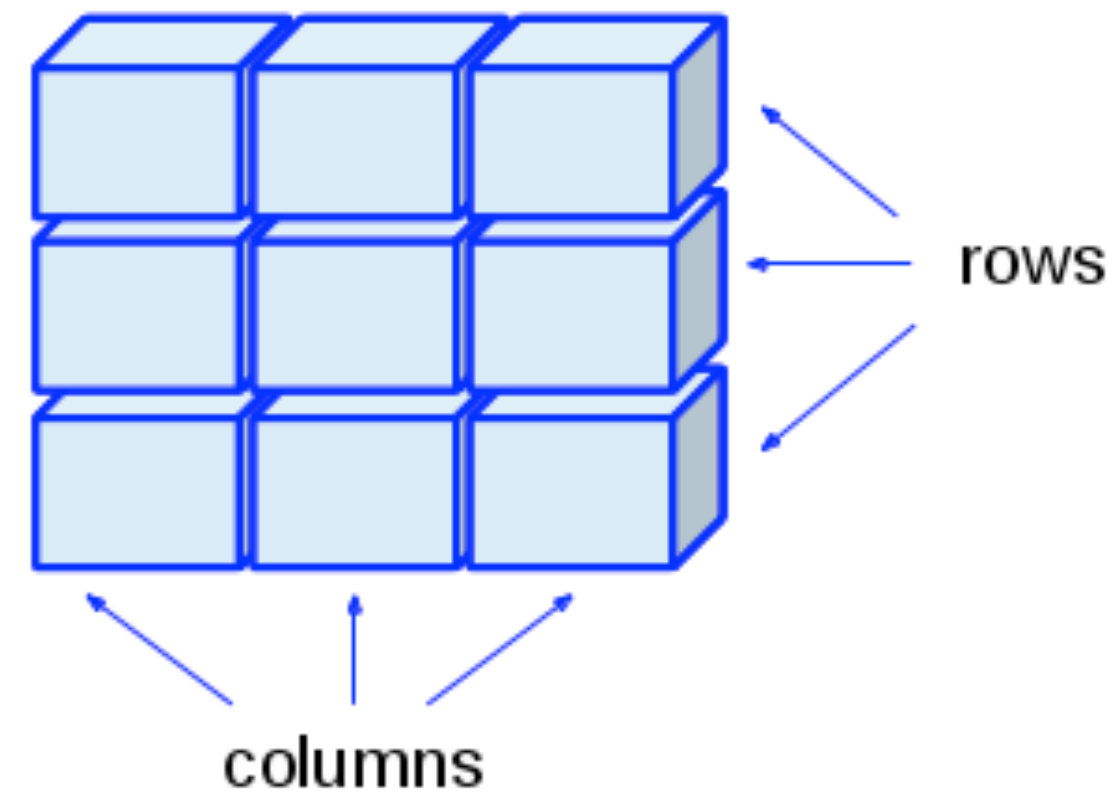
Vector



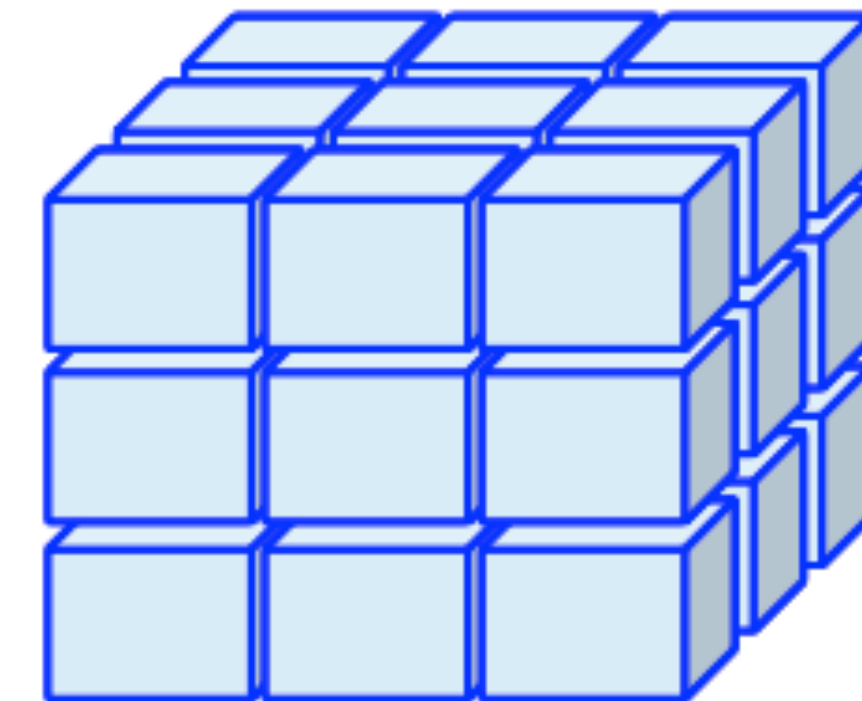
Data Frame
(Table)



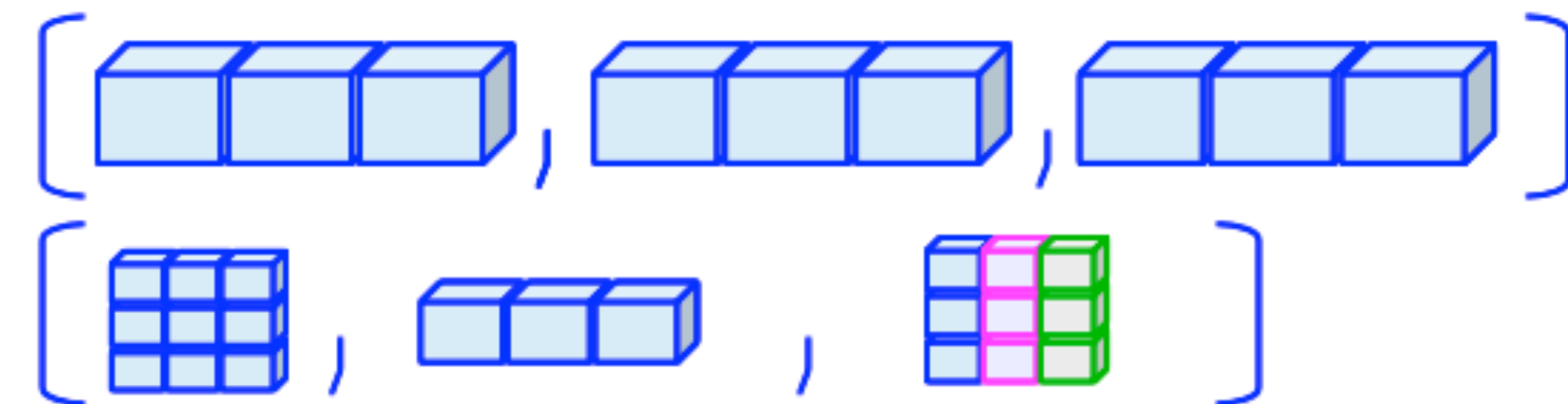
Matrix



Array



Lists

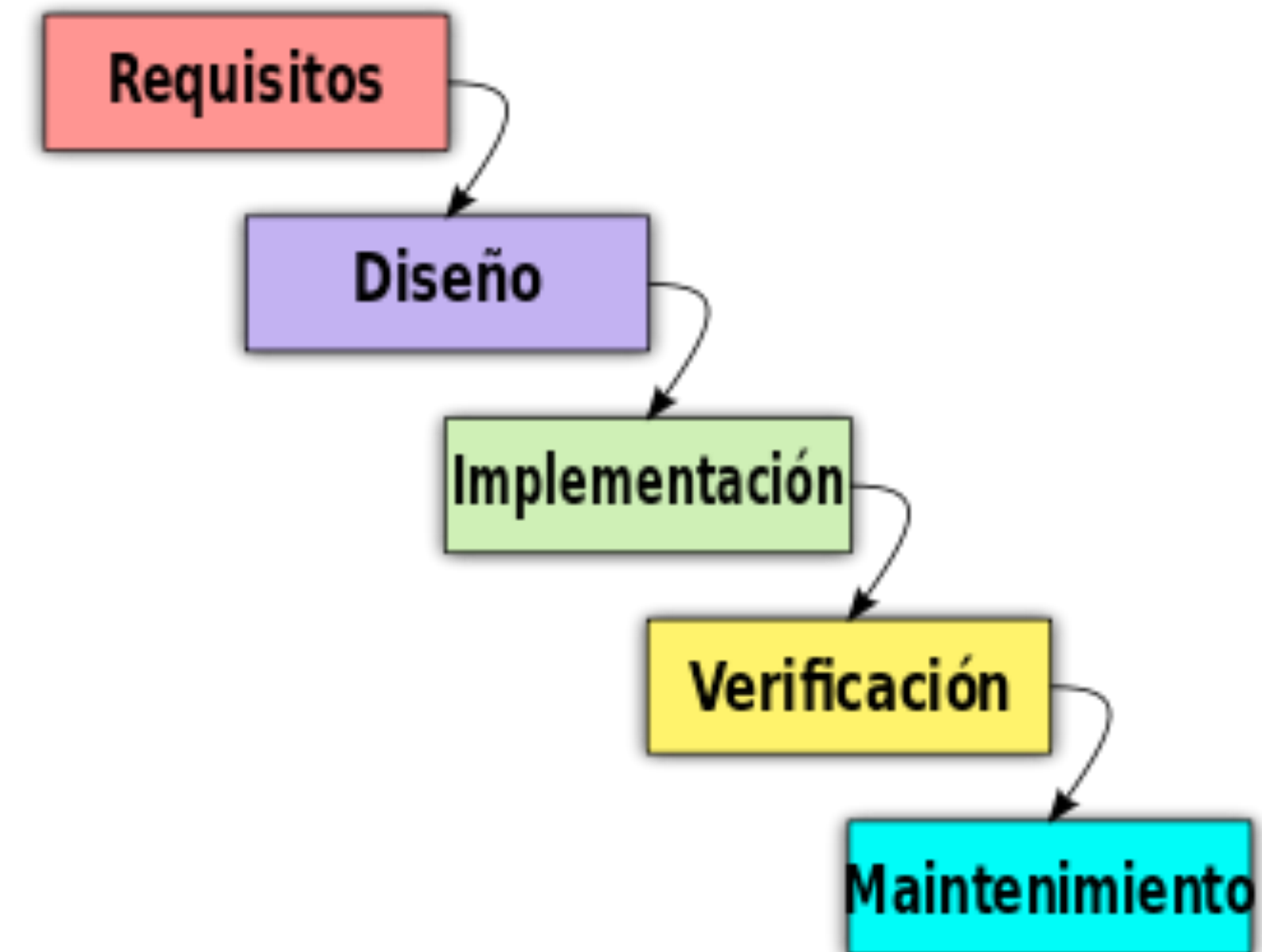


Metodologías de Desarrollo

- Especificaciones — ¿Qué hará el software?
- Diseño e implementación — Definir módulos e implementarlos.
- Validación — Pruebas de funcionamiento y aprobación.
- Evolución — Cambiar el sistema de acuerdo a necesidades en los procesos.

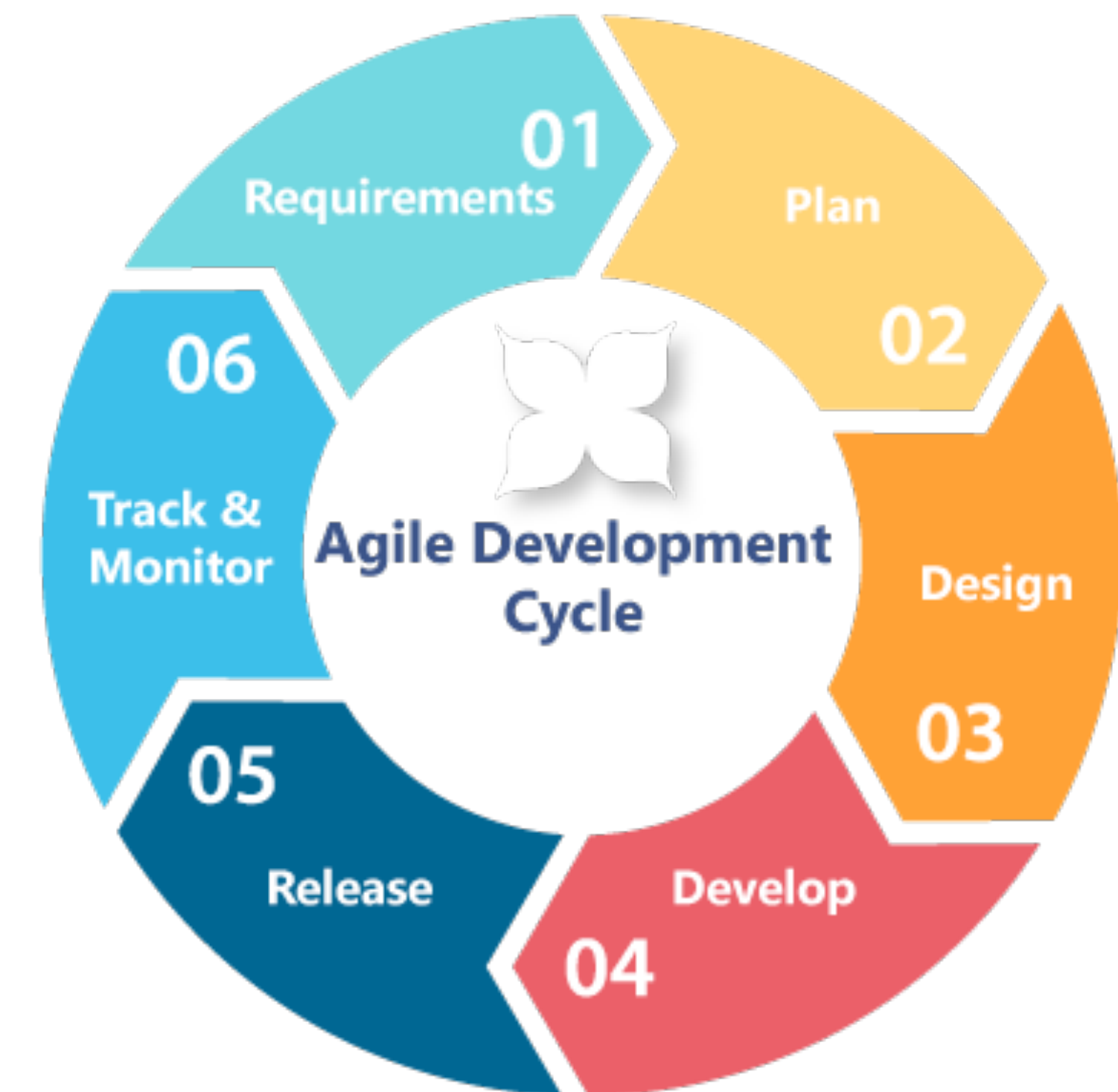
Modelos de Desarrollo

- Guiados en un plan inicial — Los avances son evaluados de acuerdo a ese plan.(Modelo en cascada)



Modelos de Desarrollo

- Ágiles e Incrementales — La planeación es incremental e iterativa, es fácil cambiar los procesos para reflejar cambios.



- Free Software Foundation
- GNU Public Licence y otras licencias
- Creative Commons
- Open Source



VS

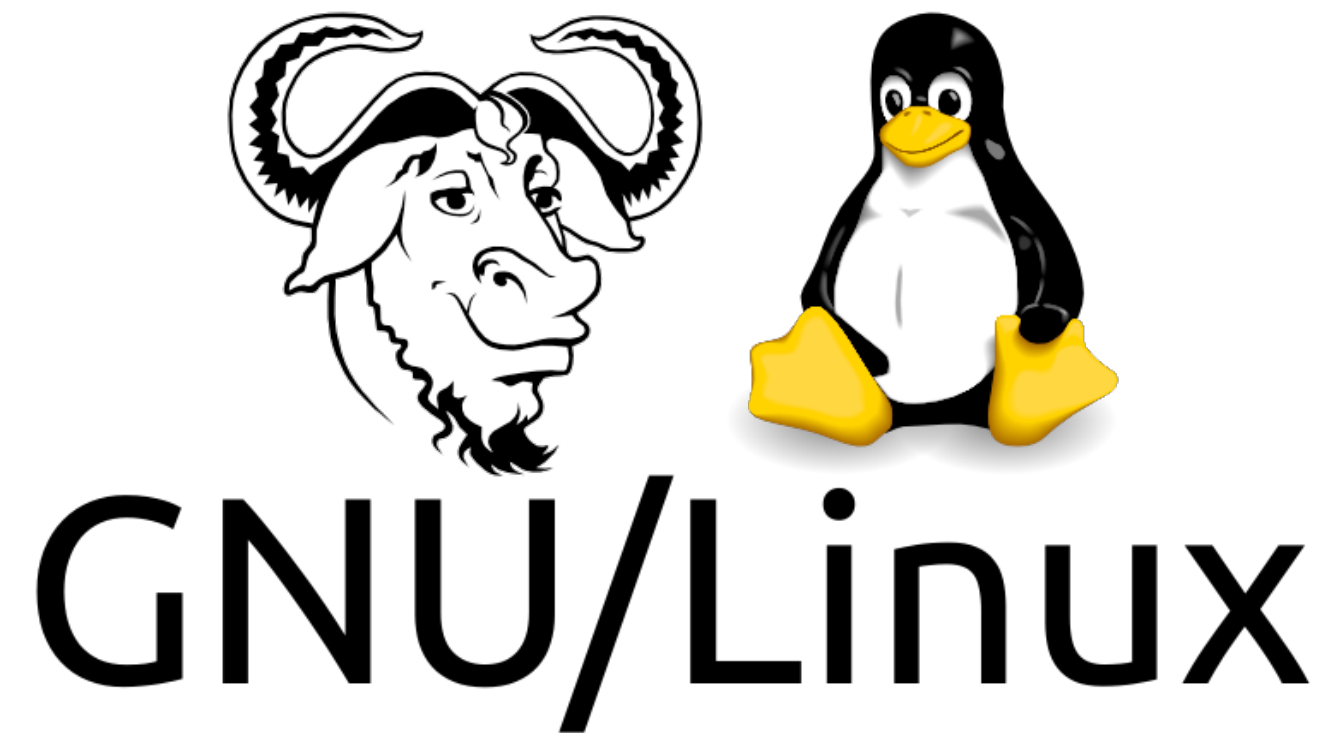


- Educación — Moodle*, openedX*
- Ofimática — OpenOffice, LibreOffice, Evolution



* Sistemas de Gestión de Aprendizaje - Learning Management System (LMS)

Sistemas Operativos — GNU/Linux, FreeBSD, ReactOS



* Sistema de Gestión de Aprendizaje - Learning Management System (LMS)

Ilustración, CAD, edición de imágenes — Blender, Inkscape, Gimp



Publicación y administración de contenido — Wordpress,
Drupal, Django CMS



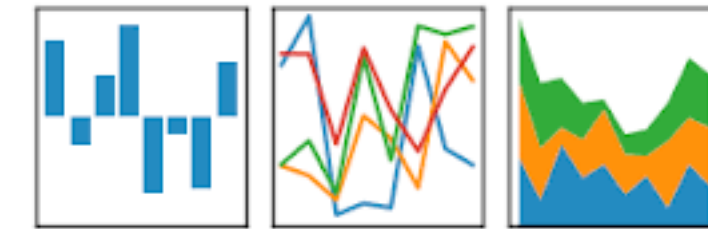
Internet — Apache HTTP server, Apache Hadoop, nginx



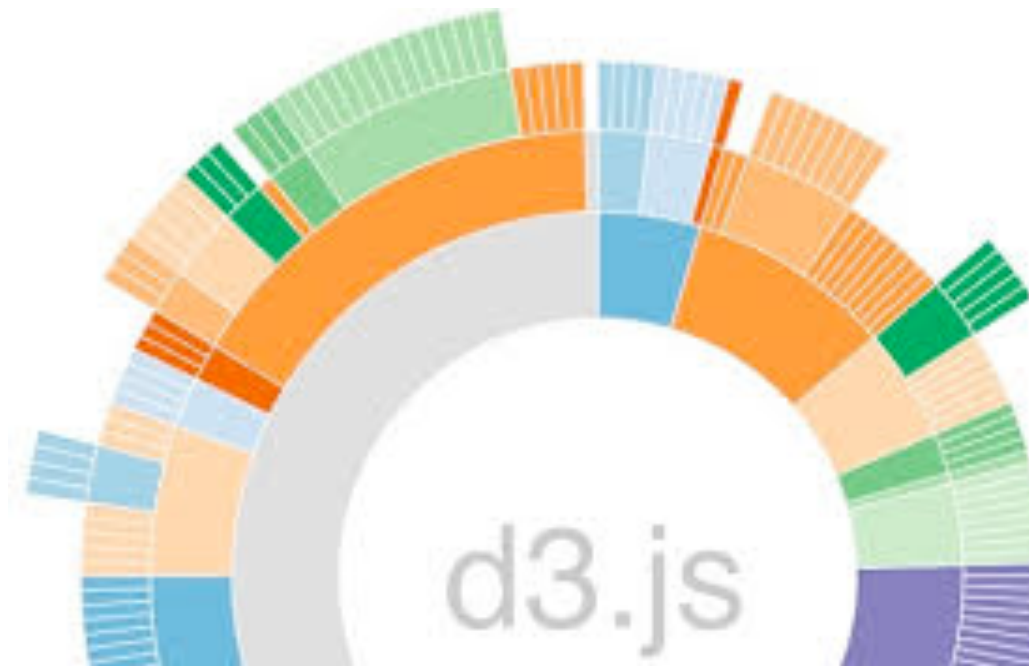
Ciencia — R, Numpy, SciPy, Pandas



pandas
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



Desarrollo — Git, Node.js, Cordova, d3.js, Eclipse,
Angular.js, three.js Django, Emacs, Bootstrap, Boost,
openFrameworks, Kivi,

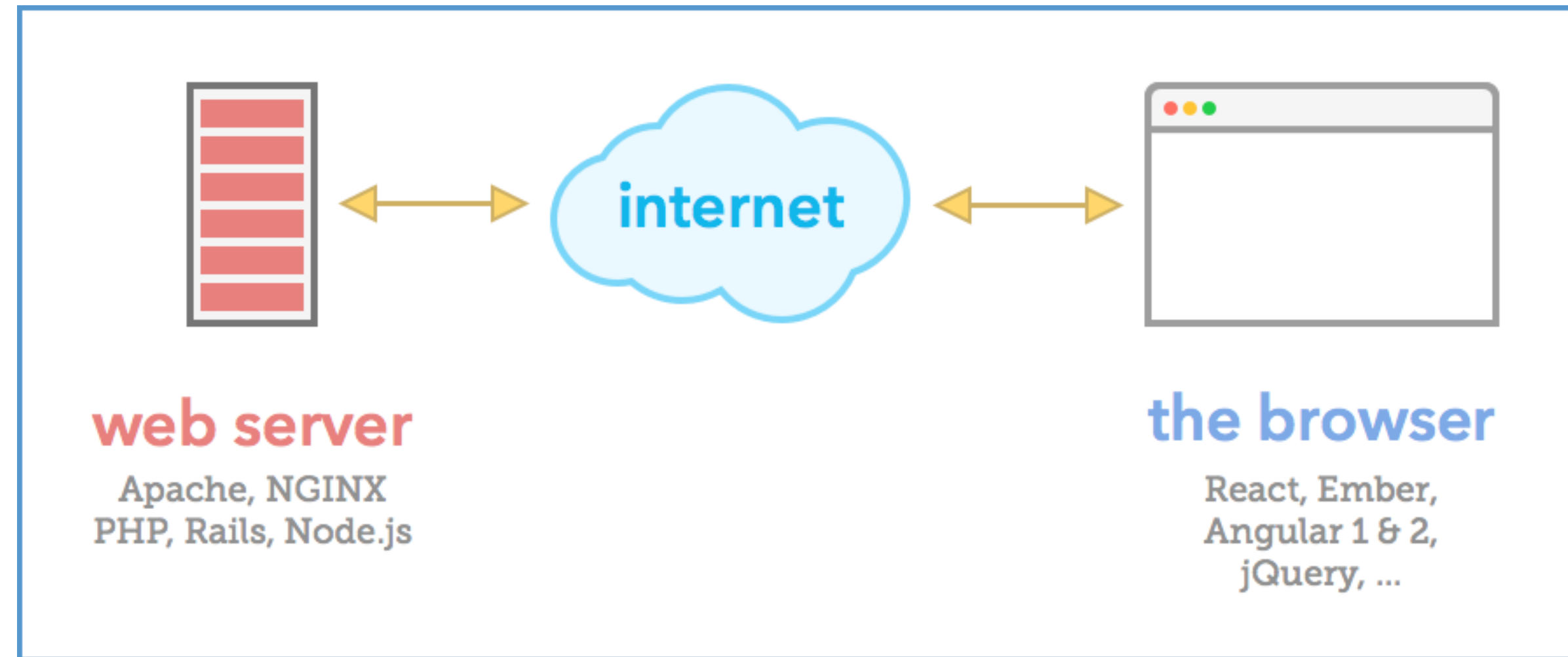


- Hardware
- Sistemas Operativos
- Librerías
- Entornos de Trabajo (Frameworks)
- Entornos de Desarrollo (IDE)
- Manejo de Versiones

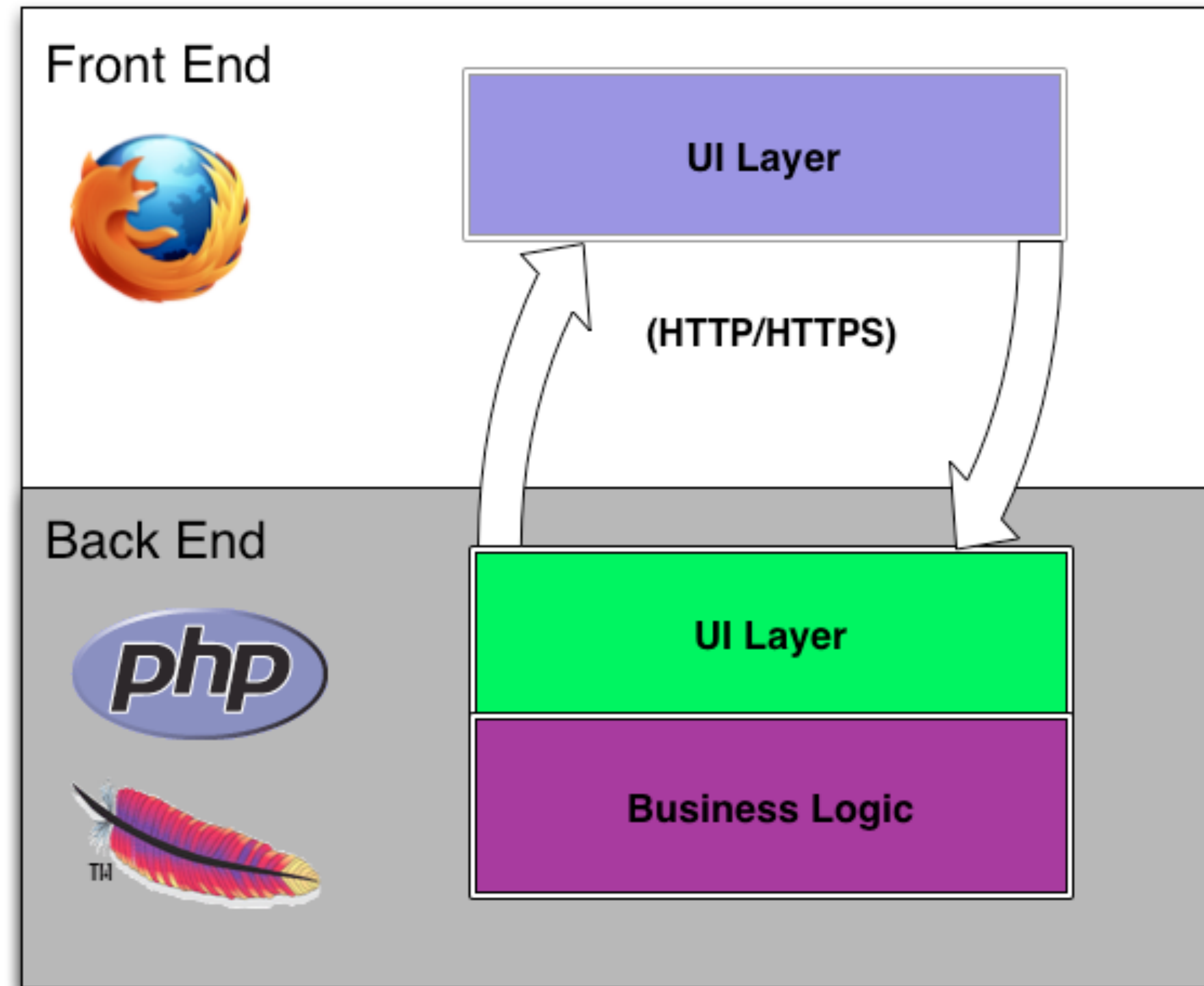
Herramientas del Laboratorio

- openFrameworks / C++
- Django / Python
- Javascript, HTML, CSS
- Git

- Navegador
- Cliente/Servidor - Front/Backend (MVC)
- Javascript
- Java, Flash, PHP, Ruby, Python
- Frameworks



- HTML
- CSS
- Javascript



- Php
- Javascript
- C++
- Java
- Python
- C++

- Processing, Pure Data
- p5.js, paper.js, Juce, Cinder, openFrameworks, Kivy
- Touch Designer, vvvv, Quartz Composer
- Hardware: Arduino, Raspberry PI, Kinect, Leap Motion, otros sensores
- Entornos de desarrollo



Arquitectura

- Definición de funcionalidades
- Abstracción de datos
- Planeación
- Modularización
- Programación
- Pruebas
- Documentación

Programación

- Algoritmos
- Sintaxis
- Semántica
- Pragmática
- Estilo