

## Clase 10

**Alumna: Laura Loreiro**

14/04/2021

### Playground

#### Lenguajes de programación

Para poder traducir lo que nuestra app haga, tenemos que ser específicos, los sistemas carecen de inteligencias para interpretar cosas ambiguas. Son muy importantes las instrucciones que le damos, conformadas de pasos llamados "Algoritmos", por medio de un lenguaje formal, lenguajes de programación. Existen muchos, hay dos grupos, específicos, resuelven problemas puntuales y generales, que permiten generar app casi independiente del contexto. Hay de altos nivel, los que se encuentran mas cercanos al lenguaje natural, 0 y 1, ej: javascripts y bajo nivel, son utilizados para instrucciones muy específicas. Los lenguajes van cambiando con el tiempo.

Los lenguajes de tipado fuerte y débil se distinguen según si permiten o no violencia de los tipos de datos una vez declarados.

#### Tipado débil

En estos, no indicamos el tipo de variable. Podemos asignar, ej, un valor entero a una variable que antes tenía una cadena. También podemos operar con variables de distintos tipos.

Ventaja, mas rápido de desarrollar; desventaja, podemos cometer muchos errores si no tenemos cuidado.

#### Tipado Fuerte

Se nos obliga a indicar el tipo de dato al declarar la variable. Y no puede ser cambiado una vez definida la variable.

Ventaja, código más expresivo, cometeremos menos errores. Desventaja, mas estricto a la hora de programar y hay que escribir mas código.

#### Tipado estático

La comprobación de tipificación se realiza durante la compilación y no durante la ejecución. Comparado con el tipo dinámico, el estático permite que los errores de tipificación sean detectados antes y que la ejecución del programa sea mas eficiente y segura.

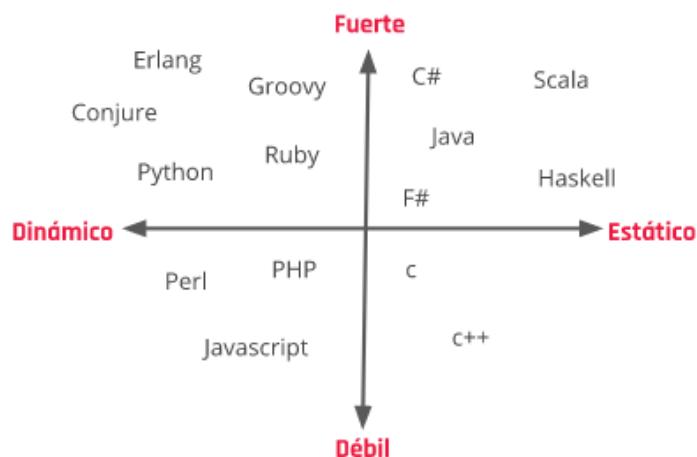
### Tipado Dinámico

La comparación de tipificación se realiza durante su ejecución en vez de durante la compilación. Comparado con el estático, este es más flexible, a pesar de ejecutarse más lentamente y ser más propenso a contener errores de programación.

## Tipado **dinámico** y tipado **estático**



## Posicionamiento de cada **lenguaje de programación**



### Frameworks

Marco de trabajo, es una estructura previa/ esqueleto que se puede aprovechar para desarrollar un proyecto.

Es una especie de plantilla, un esquema conceptual, que simplifica la elaboración de una tarea, ya que solo es necesario complementarlo de acuerdo a lo que se quiere realizar.



### Paradigmas de la programación

En Oslo, en los 60', Ole-Johan Oahl y Kristen Nygaard, trabajan en simulaciones de naves para descubrir como se comportan cada una de sus partes al modificar sus cualidades, en esa época la forma de programar, era estructurada, y los lenguajes de programación seguía este modelo o paradigma, donde uno debía escribir funciones y el código se ejecutaba línea tras línea. Las simulaciones era complejas y abordarlas con el modelo de programación estructurada, sumada complejidad en lugar de facilitarlo. Pensaron ver el problema desde otra perspectiva, con otro modelo, cambiaron el código con otras reglas, pudieron representar cada objeto. Así nació el paradigma de programación orientada a objeto.

### ¿qué es un paradigma?

Es una forma de pensar bajo un modelo preestablecido.

## ✓ Paradigma estructurado

Sigue una línea de pensamiento donde se suele ejecutar una instrucción a la vez y uno se rige en un acotado set de instrucciones.

Este paradigma es muy utilizado para el desarrollo de sistemas.



### Ejemplo:

Una función "esPar" recibe un número y devuelve el mensaje "verdadero", si el número es par, y "falso", si es impar.



## ✓ Paradigma de programación orientado a objetos

El código puede agruparse de tal forma que llegue a representar una entidad y que interprete mensajes. La fortaleza del paradigma de la programación orientada a objetos yace en utilizar abstracciones y crear entidades.

### Ejemplo:

Un código representa un carrito de compra.



Otro código representa un producto con su precio.



Luego, puedo agregarle la responsabilidad al carrito que vaya agregando productos para luego preguntarle el costo total.



## ✓ Paradigma funcional

El paradigma de programación funcional se basa en un concepto muy simple y es el de las funciones matemáticas.

La fortaleza de este paradigma radica en que siempre que a la función X se le pasa el valor A, esta siempre va a devolver el valor B. La fortaleza de este paradigma radica en que siempre que a la función X se le pasa el valor A, esta siempre va a devolver el valor B.



Esta propiedad de devolver el mismo valor se le conoce como inmutabilidad, y es característico de este paradigma.

### Ejemplo:

La solución funcional al problema de si un número es par o no es muy similar al estructurado, debemos crear una función "esPar" que reciba un número y nos diga si es par o impar.



## ✓ Paradigma lógico

En lugar de desarrollar pasos e instrucciones, utiliza reglas lógicas para consultar al sistema y el mismo infiere que hacer en base a las reglas lógicas establecidas.

Paradigma de programación lógica → Instrucciones → Reglas lógicas

### Ejemplo:

Reglas lógicas:

Toda persona cuyo saldo sea negativo es deudor.

A todo deudor se le aplica una tasa de interés del 10%

Con este set lógico podríamos preguntar:

**¿Cuál es la tasa de interés de Juan?**

El sistema responde analizando si Juan es una persona, si es deudor o no y si aplica o no la tasa de interés.



## ✓ Paradigma de programación con lenguaje específico de dominio

Los lenguajes que encontramos acá tratan de resolver problemáticas superespecíficas.

### Ejemplo:

Cuando queremos consultar una base de datos de un supermercado para saber qué productos tenemos en la categoría de electrodomésticos.



## ✓ Multiparadigma

A lo largo de la evolución de la programación, con nuevos desafíos y paradigmas han habido lenguajes que han modificado su estructura para poder permitir dar soluciones en distintos paradigmas.

### Ejemplo:



En JavaScript se puede escribir código tanto con el paradigma estructurado como con programación orientada a objetos e incluso utilizar el paradigma funcional.

## ¿Esto quiere decir que mientras más paradigmas tenga un lenguaje es mejor?

No, un lenguaje es una herramienta y hay distintas herramientas para distintas soluciones. Siempre debemos analizar el contexto, tiempos, con que equipo contamos, ¿hay presupuesto? ¿Cuáles son las herramientas que disponemos para trabajar?  
¿Qué queremos lograr?





## Del código al ejecutable

Un programa puede estar compuesto de archivos de varios programas, ¿cómo los interpreta? Por medio de la compilación ¿el código máquina? La máquina que se compila, una similar arquitectura, y cpu. Para que se compile correctamente, ¿hay otra forma? Máquina virtual, el código fuente, intérprete, hace un análisis línea por línea, ¿podemos elegir cómo va a ser compilado? No, depende de como el lenguaje de programación fue ..

La diferencia radica en la performance y rendimiento, intermedia entre la física y ...

El primer compilador de la historia, el A-0 , fue desarrollado en 1952 por la científica en computación Grace Hopper.

## Código fuente

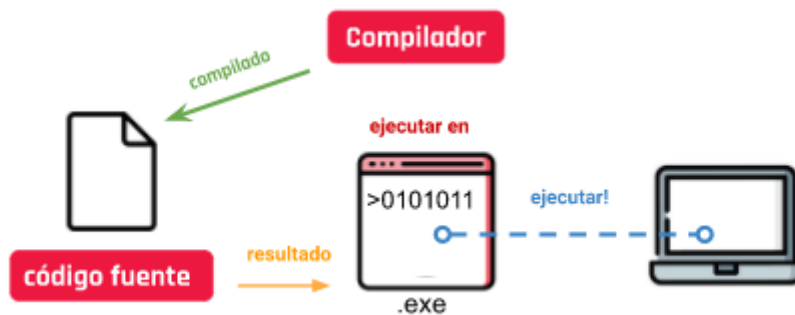
Es la colección de instrucciones de computadora escritas usando un lenguaje de programación legible por humanos.

## Código máquina

Es una secuencia de sentencias en lenguaje de máquina o binario. Es el resultado obtenido después de que el compilador convierta el código fuente en un lenguaje que pueda ser comprendido por el procesador.

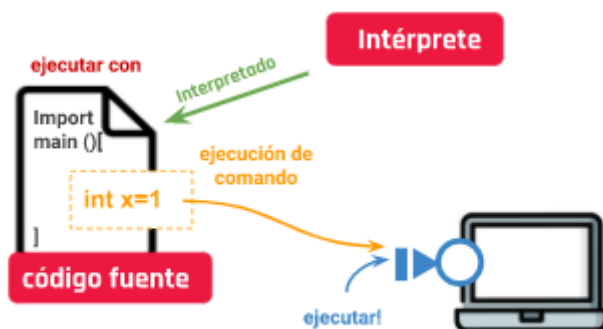
## Compilador

Es una aplicación traduce (compila) el código fuente en un código que el procesador puede comprender ejecutar. Este código de máquina se almacena en forma de archivo ejecutable.



### Intérprete

Traduce el código fuente línea por línea y lo ejecuta directamente. El proceso de traducción funciona mucho mas rápido que en un compilador, pero la ejecución es más lenta y se necesita una gran cantidad de memoria.

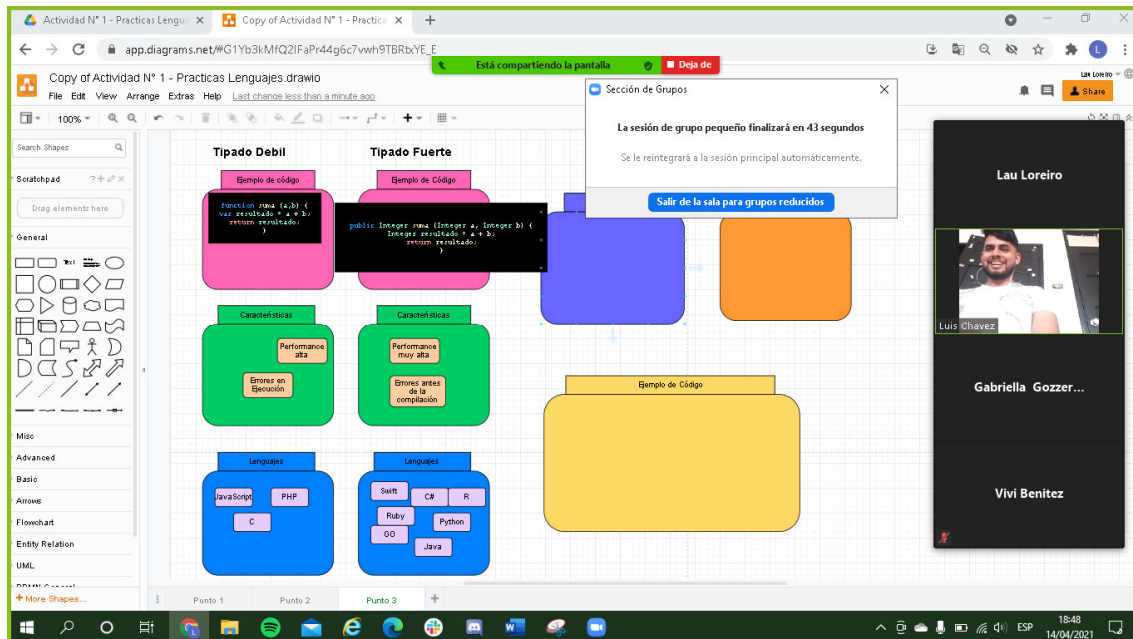


### Clase sincrónica

Una máquina virtual, usa una aplicación que simula hardware, entonces el so piensa que está instalado sobre un hardware.

### Actividad





## Frameworks

Es una estructura, una plantilla, facilita el código al desarrollador, porque hay funcionalidades ya desarrolladas.

<https://github.com/topics/framework>

<https://www.tiobe.com/tiobe-index/>

