



UNIVERSIDAD
DE LIMA

Lenguajes de Programación

Criterios de Diseño

Hernan Quintana (hquintan@ulima.edu.pe)

Los lenguajes de programación son diferentes.

Ruby vs Python

Smalltalk



Ruby

Rapidez y flexibilidad.

Orientación a objetos

Se utiliza en Web apps

Algol



Python

Una única manera obvia de hacer las cosas.

Orientación a objetos

Se utiliza en Web apps y Data science.

Criterios de Diseño

- Nos permiten tener puntos de comparación entre lenguajes de programación.
- Podemos dividirlos en 4:
 1. Eficiencia
 2. Regularidad
 3. Seguridad
 4. Extensibilidad

1. Eficiencia

a) Eficiencia de un programa

```
int i = 10;  
String s = "hola";
```

```
i = 10  
s = "hola"
```

b) Eficiencia del programador

```
if (x > 0 && y < 10)
{
    num = x + y;
}
```

```
if x > 0 and y < 10:
    num = x + y
```

Readability y Reliability

2. Regularidad

a) Generalidad

- No permitir casos especiales en la disponibilidad o uso de constructos.
- Permite combinar constructos similares en uno general.
- Ejem: En C no se pueden comparar structs con el operador `==`.

```
struct Persona {  
    char nombre[50];  
    int edad;  
    char genero;  
};
```

```
struct Persona p1;  
strcpy(p1.nombre, "Juan");  
p1.edad = 25;  
p1.genero = 'M';
```

```
struct Persona p2;  
strcpy(p1.nombre, "Juan");  
p1.edad = 25;  
p1.genero = 'M';
```

```
if (p1 == p2) ???? X
```

b) Baja Ortogonalidad

- Se refiere a la falta de consistencia en el uso de las construcciones del lenguaje de programación.
- Ejemplo:

```
int i = 10;  
int j = 2;  
int res = i * j; // SI se puede  
  
int[] arrI = {1 , 2};  
int[] arrJ = {3 , 4};  
int[] arrRes = arrI * arrJ; // NO se puede
```

c) Uniformidad

- Se nos permite utilizar de forma uniforme las distintas construcciones del lenguaje.

```
struct Perro  
{  
}; // obligatorio
```

```
int sumar(int param1, int param2)  
{  
  
} // no es necesario
```

3. Seguridad

- El lenguaje debe proporcionarnos construcciones que nos permitan programar sin errores.
- Existen lenguajes que dan mayor control pero a costa de mayor complejidad (mayor probabilidad de cometer errores).

C++

```
Perro* perro = new Perro();  
perro->ladrar();
```

Java

```
Perro perro = new Perro();  
perro.ladrar();
```

4. Extensibilidad

- Posibilidad de crear nuevas estructuras con el lenguaje de programación.

```
#include<stdio.h>
#define si if

int main()
{
    int x = 5;
    si (x > 0) {

    }
}
```