

# Lenguajes de Programación

## Principios de diseño

Ma. Laura Cobo

Universidad Nacional del Sur  
Departamento de Ciencias e Ingeniería de la Computación  
2018

## Características deseables en el software

De cualquier software se espera, básicamente, que sea:

- confiable
- extensible
- reusable
- robusto
- eficiente
- etc.

## Características deseables en el software

Los lenguajes de programación también son software pero al tener otro tipo de propósito se espera que posea otras características:

- facilidad de lectura y escritura (legibilidad)
- seguridad
- costo

## Principios

En la evaluación del lenguaje es importante tener en cuenta las facilidades que provee el lenguaje como asistencia al programados (pragmatismo)

- Diseño de programas
- Documentación de programación
- Debugging



## Discusión

El lenguaje puede ser evaluado en termino de al menos estos criterios:

- Simplicidad
- Seguridad
- Traducción
- Costo
- Legibilidad
- Sintaxis
- Estructura de los programas

La gente tiende a disentir con respecto al conjunto de propiedades que un lenguaje debe tener para considerarse un “gran” lenguaje. También suele especificar con mas detalle los criterios anteriores.

# Evaluación de lenguajes

## Criterios a tener en cuenta para una evaluación general

Una lista simplificada podría incluir:

- **simplicidad**: el lenguaje debe ser lo mas simple posible. Relacionado al numero de constructores y la definición de sus posibles combinaciones.
- **uniformidad**: los conceptos básicos deben ser utilizados en forma consistente y universal.
- **ortogonalidad**
- **abstracción**: evitar la repetición de los que se indica
- **claridad**
- **ocultamiento de información**
- **modularidad**
- **seguridad**
- **expresividad**
- **eficiencia**

## Evaluación de lenguajes

Adicionalmente también se podría considerar:

- **automatización:** automatización de tareas tediosas, mecánicas o relacionadas al manejo de errores
- **defensa en profundidad:** tener defensas provistas para que si un error no es capturado en un determinado lugar pueda resolverse en otro.
- **portabilidad:** dependencia de la arquitectura
- **consistencia sintáctica:** facilidades similares deben verse de manera similar



## Evaluación del diseño lenguajes

### Tamaño

Este aspecto es discutible y difícil de evaluar ya que no hay una definición estandarizada de tamaño.

¿Cómo se determina el “tamaño” de un lenguaje de programación?

El tamaño es una medida pero no la meta del lenguaje de programación.



## Evaluación del diseño lenguajes

### Tamaño

Este aspecto es discutible y difícil de evaluar ya que no hay una definición estandarizada de tamaño.

¿Cómo se determina el “tamaño” de un lenguaje de programación?

El tamaño es una medida pero no la meta del lenguaje de programación.

## Evaluación del diseño lenguajes

### **Bonzai - Baobarb**

La verdad tiene que estar entre los monstruos de la complejidad y las obras maestras minimalistas

Una perspectiva de un buen diseño de lenguajes indica que deben ser pequeños. Sin embargo muchos lenguajes exitosos son grandes y complejos.

El exceso de tamaño tiene algunas consecuencias dañinas causando sorpresas hasta en el usuario mas experimentado. El uso de un subconjunto de las facilidades del lenguaje puede provocar que involuntariamente se utilicen propiedades que se desconocen.

Tampoco deben ser extremadamente simples, ya que el lenguaje no debería resistirse a adiciones útiles en un principio.  
(Ej. agregado de arreglos de longitud variables y facilidades elementales de módulo en Pascal)

# Evaluación de lenguajes

## Consistencia

La consistencia es el camino para obtener lenguajes elegantes, “*aprendibles*” y con resultados convincentes.

El lenguaje se basa en un conjunto de ideas. El conjunto debe ser de cardinalidad baja y cada idea debe ser “poderosa”

Por ejemplo:

- diseño por contrato
- herencia
- excepciones
- clases
- parametrización
- ligadura dinámica
- Tipado estático
- garbage collection

# Evaluación del diseño de lenguajes

## Unicidad

Este principio en el diseño de lenguajes establece que el lenguaje debe proveer una buena manera de expresar cada operación de interés. **Debe evitar proveer dos maneras.**

Este principio está asociado a las posibilidades de evolución del lenguaje luego de su diseño inicial.



## Evaluación de lenguajes de programación

La evaluación de lenguajes puede hacerse desde dos puntos de vistas diferentes:

- haciendo foco en el lenguaje
- haciendo foco en el uso del lenguaje o en las necesidades de quien usa el lenguaje

Saber que tan bien un lenguaje soporta un dominio de aplicación en general no es suficiente.

Si la evaluación se hace considerando el segundo foco de atención, se pueden tener en cuenta los siguientes criterios:

# Evaluación de lenguajes de programación

## Criterios de diseño e implementación del lenguaje

1. ¿Está el lenguaje formalmente definido?
2. ¿Es un lenguaje compilado o interpretado?
3. ¿Qué tan simple es implementar su traductor?
4. etc.

## Criterios sobre factores humanos

1. ¿Qué tan fácil es aprender el lenguaje?
2. Facilidad de lectura y escritura
3. etc.

## Criterios de ingeniería de software

1. Portabilidad
2. Eficiencia
3. etc.

## Criterios del dominio de aplicación