

Lenguajes de Programación

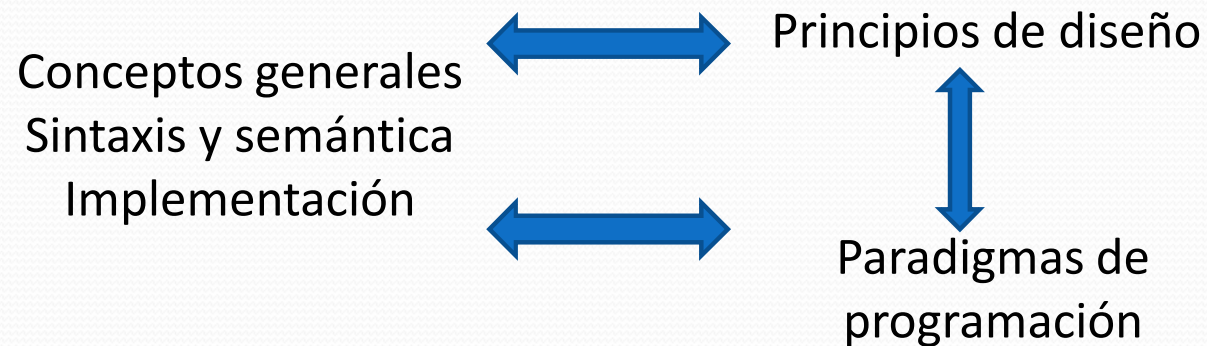
Introducción – Criterios de evaluación

Ma. Laura Cobo

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

Organización general del curso

- desde los conceptos a la ejemplificación sobre lenguajes
(enfoque vertical)
- buscar los conceptos sobre lenguajes de programación específicos
(enfoque horizontal)



Bibliografía

- Scott, Michael L. *“Programming Languages Pragmatics”* 4^{ta}. Edición. Morgan Kaufmann Publishers 2015
- Sebesta, Robert W. *“Concepts of Programming Languages”*. 11^{ava}. Edición. Editorial Addison Wesley. 2016

Vamos a ir utilizando separatas de diferentes libros para los diferentes temas

Objetivos generales del curso

Comparar y contraponer:

- principios teóricos - detalles técnicos
- semántica – sintaxis
- evolución - historia

Razones para estudiar conceptos de lenguajes de programación

- Incrementar la capacidad para expresar ideas.
- Mejorar la calidad y cantidad de conocimiento previo para la apropiada selección de lenguajes.
- Aumentar la habilidad para aprender nuevos lenguajes.
- Mejor comprensión de la importancia de la implementación.
- Inclusión de los avances en las formas de cómputo.

Razones para lenguajes de programación

- Comprender aspectos oscuros de los lenguajes.
- Elegir modos alternativos de expresar cosas.
- Realizar buenos debuggers, ensambladores, linkeadores y herramientas relacionadas.
- Simular facilidades útiles en un lenguaje que carece de la misma
- Hacer mejor uso de la tecnología de los lenguajes de programación a medida que aparecen.

Lenguajes

Un lenguaje en general, es el medio del que se vale cualquier individuo para comunicar ideas y experiencias a otros individuos. Estas ideas y experiencias están “almacenadas” de alguna manera el individuo emisor y “almacenadas” tal vez con otra representación en el receptor.

El **lenguaje de programación** en general tiene la misma idea, solo que el receptor es una computadora, por ello es una de las principales herramientas en el proceso de desarrollo del software.

El arte de diseñar lenguajes

Hay una gran variedad de lenguajes y continúan apareciendo nuevos ¿por qué?

- Evolución
- Propósito especial
- Preferencia personal

¿qué convierte a un lenguaje en exitoso?

- Poder expresivo
- Facilidad de uso para novatos (baja curva de aprendizaje)
- Facilidad de implementación
- Estandarización
- Código abierto
- Excelentes compiladores
- Inercia, economía, quien lo promueve

Introducción a lenguajes de programación

- Originalmente las computadoras eran grandes y costosas, eran programadas en lenguajes maquina (expresado como números en hexadecimal)
- A medida que se comenzaron a escribir programas más largos, se volvió claro que era necesario utilizar un lenguaje menos propenso a errores (notación con abreviaturas nemotécnicas – lenguaje ensamblador). La traducción quedó a cargo del ensamblador.
- Con la evolución de las computadoras y el desarrollo de diseños se volvió frustrante reescribir los programas para cada nueva máquina. También se volvió humanamente complicado entender los programas en lenguaje ensamblador.
- Se hizo necesario contar con lenguajes independientes de la máquina.

Lenguajes

Hay diferentes taxonomías para los lenguajes:

Generaciones (primera, segunda, tercera, cuarta, quinta)

Paradigma

Dominio de aplicación

Tipo de traductor

Otra Clasificación

Declarativa:

Funcional	Lisp/Scheme, ML, Haskell
Dataflow (flujo de información)	Id, Val
Lógica, basada en restricciones	Prolog, aspectos programables de las hojas de cálculo (Excel, Lotus, etc.)
Basada en templates	XSLT

Imperativa:

Von Newmann	C, Ada, Fortran,
Scripting	Perl, Python, PHP,
Orientada a objetos	Smalltalk, Eiffel, C++, Java

Desarrollo de Lenguajes de Programación

El **lenguaje de programación** surge de la vinculación entre las diferentes metodologías de diseño y las arquitecturas de la computadoras.

Otro de los factores que terminan teniendo especial relevancia es el dominio de aplicación pretendido.

Características deseables en el software

De cualquier software se espera, básicamente, que sea:

- confiable
- extensible
- reusable
- robusto
- eficiente
- flexibilidad
- 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

Readability (facilidad de lectura)

La facilidad para leer e interpretar programas es fundamental.

Características que contribuyen a la legibilidad:

- **Simplicidad**
 - Se obtiene en gran medida de combinar un número de pequeño de constructores primitivos y un uso limitado (ni mucho-ni poco) del concepto de ortogonalidad.
- **Ortogonalidad**
 - Básicamente significa que un conjunto pequeño de constructores primitivos, puede ser combinado en número relativamente pequeño a la hora de construir estructuras de control y datos. Cada combinación es legal y con sentido.
- **Estructuras de control y de datos**
 - Contar con estructuras legibles (sólo uso de goto – agregado del tipo boolean)
- **Consideraciones sobre la sintaxis**
 - Formas de los identificadores – palabras reservadas – forma y significado (por ejemplo el modificador static de JAVA)

Writability (facilidad de escritura)

La facilidad para crear programas de un dominio. Este criterio depende del dominio de aplicación del lenguaje.

- **Simplicidad y Ortogonalidad**
 - Mucha ortogonalidad genera una menor detección de errores. Si el lenguaje no es simple solo se aprende y utiliza una pequeña porción de él.
- **Soporte para la abstracción**
 - Capacidad de definir y usar estructuras u operaciones complicadas de manera que sea posible ignorar muchos de los detalles.
- **Expresividad**
 - El lenguaje posee formas relativamente convenientes de expresar ciertas operaciones, por ejemplo `contador++` en lugar de `contador = contador + 1` o el uso del `for` en lugar del `while` equivalente.

Reliability (fiabilidad)

- Chequeo de tipos
 - Cuanto antes se encuentren errores menos costoso resulta realizar los arreglos que se requieran.
- Manejo de excepciones
 - La habilidad para interceptar errores en tiempo de ejecución, tomar medidas correctivas y continuar.

Costo

- Aprender (a escribir programas cercanos a aplicaciones particulares)
- Usar (entrenar programadores para el uso del lenguaje)
- Compilar
- Ejecutar
- Sistema de implementación del lenguaje (disponibilidad de compiladores gratuitos)
- Confiabilidad (escasa confiabilidad conlleva altos costos)
- Programas de mantenimiento