



Sistema Nacional de Educación Superior Tecnológica



Dirección General de Educación Superior Tecnológica

Unidad II Introducción al diseño de los lenguajes de programación

M.C. Juan Carlos Olivares Rojas

Agenda

2.1 Visión del problema

2.2 Consideraciones preliminares

2.3 Objetivos y filosofías del diseño de los lenguajes de programación

2.4 Diseño detallado

2.5 Caso de estudio



2.1 Visión del problema

- ¿Cuál es el propósito de un lenguaje?
- Los lenguajes de computación pueden ser de propósito general o específicos.
- C, C++, Java, Pascal, etc. Son lenguajes de programación de propósito general
- SQL, PROMELA, Actionscripts son lenguajes específicos



¿Por qué tal diversidad de lenguajes?

- Los lenguajes de programación son como los carros, existen para todos los gustos y/o usos.
- ¿Quién cargaría una tonelada de papas en un auto deportivo?
- Los lenguajes de propósito general son como los autos sedán, sirven para casi todo



Visión del problema

- Se debe identificar que es lo que se piensa hacer con el lenguaje, ya que puede ser sólo la estructuración de contenido Web, visualizar información o bien realizar la conversión de un documento.
- HTML es lenguaje de representación visual
- OWL es lenguaje de descripción de elementos
- C es un lenguaje programación



2.2 Consideraciones preliminares

- Debemos tomar en cuenta las palabras reservadas del lenguaje, los operadores, los tipos de datos.
- Debemos considerar el objetivo del lenguaje, si es un lenguaje de enseñanza, si es un lenguaje para profesionales, si el código desarrollado va a ser mejor.



2.3 Objetivos y filosofías del diseño de los lenguajes de programación

- Algunos usos de los lenguajes de programación son:
 - Comunicación humana
 - Prevención y Detección de errores
 - Usabilidad
 - Portabilidad
 - Independencia de la máquina



Filosofías

- Se debe lograr una correcta comunicación entre emisor y receptor
- Es más importante que un programa sea leíble que escribible, ya que un programa generalmente se escribe una vez y se lee muchas veces (documentación, mantenimiento, etc.)
- La tendencia actual es separa la interfaz de la implementación



Filosofías

- Tratar de hacer lenguajes para múltiples arquitecturas de computadoras (máquinas virtuales)
- Control de apuntadores
- Control de tipo de datos robustos
- Simplicidad por eficiencia



2.4 Diseño detallado

- Considerar características como:
 - Patrones de diseño
 - Paquetes (bibliotecas, APIs, componentes)
 - Excepciones
 - Validaciones
 - Marco de trabajo
 - Utilerías auxiliares (preprocesador, enlazador)
 - Inclusión de otros lenguajes



2.5 Caso de estudio

- Explicar el lenguaje que se va a desarrollar en el curso:
 - ¿Por qué se va a desarrollar (problemática)?
 - Vocabulario del lenguaje (léxico palabras clases que hacen)
 - Reglas de estructura (gramática, sintaxis)
 - Semántica
 - Si existe código intermedio
 - Si se mejora ese código
 - El código objeto final



¿Preguntas?

