



# Unidad I Introducción a la programación de Sistemas

M.C. Juan Carlos Olivares Rojas

# Agenda

- 1.1 ¿Qué es y que estudia la programación de sistemas?
- 1.2 Herramientas desarrolladas con la teoría de programación de sistemas
- 1.3 Lenguajes
  - 1.3.1 Lenguajes naturales.
  - 1.3.2 Lenguajes artificiales.
  - 1.3.3 Proceso de la comunicación.
- 1.4 Traductor y su estructura.
  - 1.4.1 Ensambladores.
  - 1.4.2 Compiladores.
  - 1.4.3 Interpretes.
- 1.5 Generadores de código para compiladores (compilador de compilador).



# 1.1 ¿Qué es y qué estudia la programación de sistemas?

- Programa: conjunto de instrucciones que ejecuta una computadora para realizar una actividad.
- Sistema: conjunto de elementos autónomos que trabajan en armonía para alcanzar un objetivo en común.



# Tipos de sistemas

- Sistemas físicos: equipo, maquinaria, objetos reales, Hardware
- Sistemas abstractos: ideas, hipótesis, conceptos, planes, Software
- Sistemas abiertos y cerrados dependiendo del ambiente en que se ejecutan.



# Características de un sistema

- Están bien delimitados, por lo tanto se conocen sus fronteras.
- Un sistema puede interactuar con su medio ambiente a través de una interfaz de entradas y salidas que recibe el nombre de parámetros del sistema.
- Un sistema puede ser componente de otro sistema



# ¿Cuál es la diferencia entre un técnico y un ingeniero?

- A parte del grado académico, el tipo de programas que realizan.
- Generalmente un técnico realiza software de aplicaciones (ABCD) o aplicaciones de gestión (nóminas, control de inventarios, etc.)
- Un ingeniero es más creativo, más ciencia e investigación; mientras que un licenciado aplica los conocimientos.



# Clasificación del software

- Software de sistemas: software que ayuda (sirve de base) a otro software.
- Procesan estructuras de datos complejas
- Tienen una fuerte dependencia con el hardware
- Utilizan una fuerte gestión de procesos (conurrencia, planificación, etc.)



# Clasificación del software

- Software de aplicación (gestión)
- Software de tiempo real
- Software empotrado
- Software de ingeniería y científico
- Software basado en Web
- Software de inteligencia artificial



# Áreas afines a la programación de sistemas

- Teoría de la computación (lenguajes y autómatas).
- Lenguajes de programación
- Arquitecturas de computadoras
- Algorítmica
- Ingeniería del software



## 1.2 Herramientas desarrolladas con la teoría de programación de sistemas

- El caso más sencillo de programación de sistemas es la construcción de compiladores para ejecutar lenguajes de programación.
- Pero no sólo se aplica en lenguajes de programación, sino también se aplica en cualquier programa que se tenga que hacer un análisis o extracción de información



# Software de sistemas

- Editores de texto inteligentes (IDEs con autocompletar, revisores ortográficos, etc)
- Impresoras estéticas (impresión de gran calidad sin un editor visual, Latex, etc.)
- Intérpretes (Shells de sistemas operativos o de alguna aplicación como un SMBD)
- Búsqueda de información que no es tan común en base a patrones, etc.



# 1.3 Lenguajes

- Conjunto de palabras y reglas que permiten comunicar información entre dos entidades.
- Lenguaje son las cadenas que pueden generarse a través de una gramática
- El lenguaje que entienden las máquinas (lenguaje formal) es muy diferente del lenguaje que entendemos los humanos



# Repaso de lenguajes y autómatas

- Símbolo: representación abstracta de alguna entidad
- Alfabeto: conjunto finito de símbolos
- Cadena: yuxtaposición de símbolos de un alfabeto que representan a un objeto
- Lenguaje: conjunto de cadenas válidas que se pueden formar a través de un alfabeto



## 1.3.1 Lenguajes naturales

- El lenguaje natural es inherentemente ambiguo, por lo que se necesita crear un lenguaje que permita eliminar esas ambigüedades.
- Es mejor crear otro lenguaje, denominado de alto nivel que es el encargado de mediar entre la abstracción humana y la abstracción de lenguaje de máquina



## 1.3.2 Lenguajes artificiales

- Los lenguajes artificiales son aquellos que los humanos hemos creado para comunicarnos
- Las computadoras sólo saben 0 y 1
- Un lenguaje artificial permite implementar un algoritmo en una computadora para resolver un problema.



# Lenguajes de bajo nivel

- Una abstracción más entendible del lenguaje máquina es el uso de lenguajes ensambladores en donde cada instrucción o mnemónico es traducido a una instrucción máquina.
- ADD AX, 5
- LOAD A, 5



# Lenguajes máquina

- El lenguaje máquina es dependiente de cada tipo de arquitectura de computadoras por lo que el código no es fácilmente portable a otras arquitecturas.
- Los lenguajes de alto nivel son más portables en lo que respecta al código fuente pudiendo llevarse a otras arquitecturas de computadoras sin mayor problema.



# Clasificación de Chomsky

- Lenguajes sin restricciones (gramática 0)
- Lenguajes dependientes del contexto (tipo 1)
- Lenguajes independientes del contexto (tipo 2)
- Lenguajes regulares (tipo 0 no presentan ambigüedades, lo ideal para representarlas en una computadora)



## 1.3.3 Proceso de la comunicación

- Para entablar una comunicación se necesita que tanto el emisor como el receptor conozcan el mismo lenguaje o en su defecto tengan un traductor.
- En este sentido, los humanos escribimos algoritmos en un lenguaje formal que una computadora pueda transformar a un lenguaje entendible por ella.



# 1.4 Traductor y su estructura

- Un traductor es un mediador entre dos entidades: emisoras y receptoras
- Los mediadores enmascaran la complejidad y heterogeneidad de los lenguajes
- Un traductor convierte un lenguaje de entrada (código fuente) a uno de salida (código objeto)

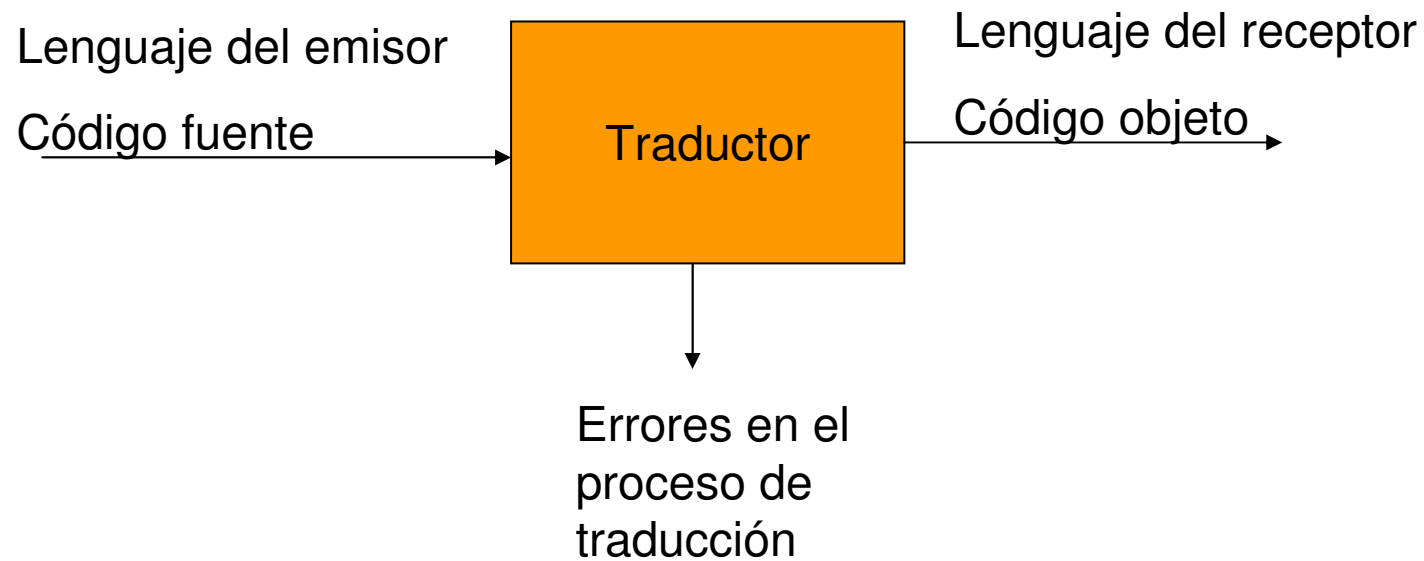


# Traductor

- La traducción puede ser sencilla (literal) o compleja (revisar el contexto) dependiendo del tipo de lenguaje de entrada y salida.
- Traducción español a inglés si se hace de manera literal es una mala traducción, se necesita de al menos otra revisión (pasada) para hacer una buena traducción.



# Traductor



# 1.4.1 Ensambladores

- Ensamblador es el traductor que se encarga de convertir instrucciones de bajo nivel a instrucciones de una máquina en general

- //Encabezados

00 MOV AX, 58d 4F0188

03 CMP 0 3A00

05 JMP etiqueta 9918

...

Etiqueta:

18 MUL AX, FF 4401FF



## 1.4.2 Compiladores

- Es el traductor que se encarga de convertir un lenguaje de alto nivel a código máquina.
- La característica de este traductor radica en el hecho de que necesita revisar todo el código fuente para poder realizar la traducción.
- Ejemplo: la traducción de un libro, discurso, o artículo técnico o de investigación



# Compiladores

- Ejemplos de compiladores: C, C++, Pascal, etc.
- Entre más pasadas se de a un código fuente mayor es la optimización que se puede hacer. El problema radica en el tiempo y en los recursos para hacerlo
- Antes de compilar un programa fuente se sigue una etapa de preprocesamiento.

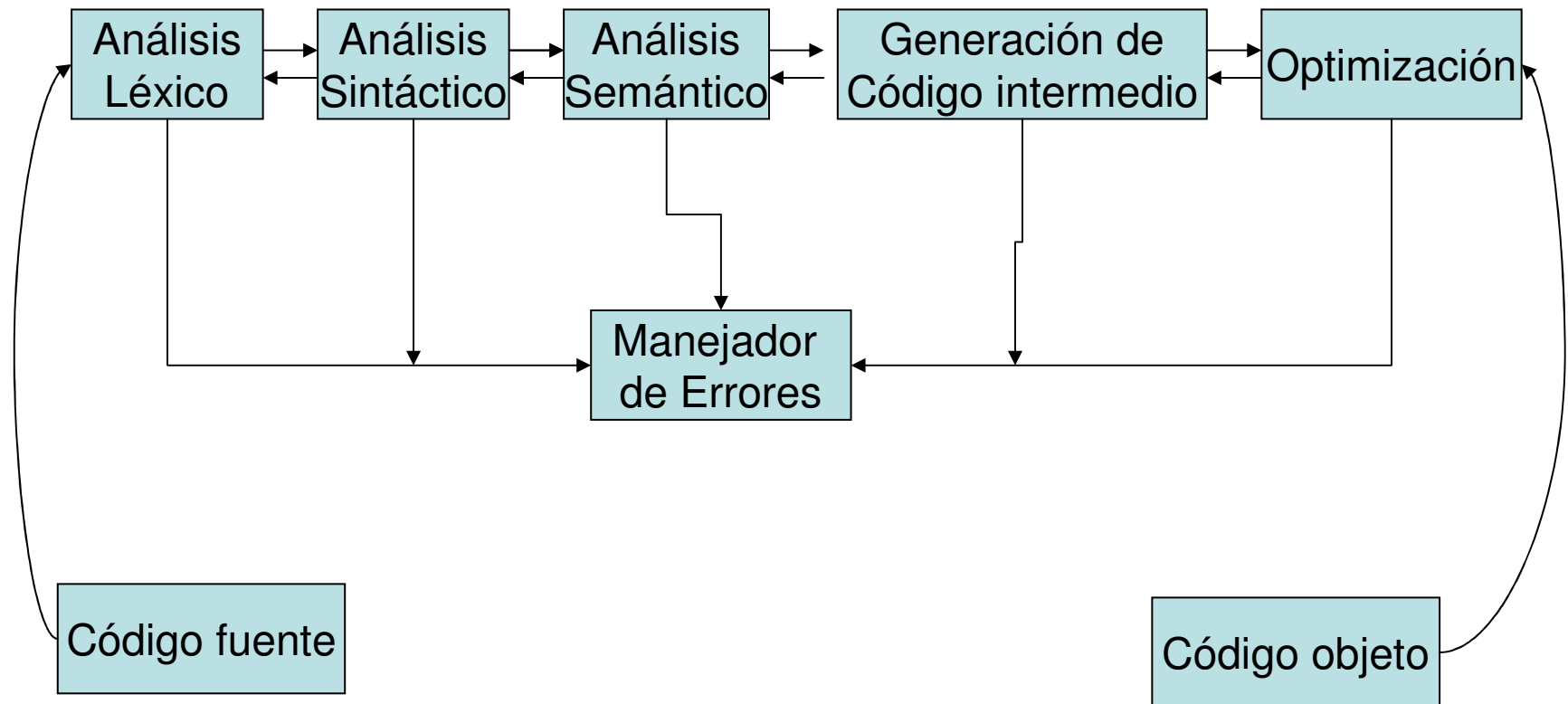


# Preprocesadores:

- Macros (expansión de funciones)
- Inclusión de archivos (bibliotecas)
- Procesadores racionales
- Extensiones al lenguaje (inclusión de ensamblador en C)



# Etapas de un compilador



## 1.4.3 Intérpretes

- Se ejecutan línea por línea, instrucción por instrucción.
- Lenguajes interpretados: PHP, PERL, BASIC
- En algunas ocasiones se necesita de una traducción rápida de algunas instrucciones, como en el Shell, instrucciones SQL, etc.



# ¿Java es compilado o interpretado?

- Java al igual que otros lenguajes como C# son lenguajes híbridos. Por una parte se compila un programa fuente para generar código objeto para una máquina virtual (bytecode o MSIL) para posteriormente ejecutarse de manera interpretada en las diferentes máquinas virtuales de cada plataforma. A este compilador se les llama jitter de JIT (Just in Time)



# 1.5 Generadores de código para compiladores (compilador de compilador)

- Los dos primeros lenguajes de alto nivel desarrollado fueron FORTRAN y COBOL.
- Desarrollar FORTRAN tardó alrededor de 14 años. Desarrollar nuestro compilador tardará menos de 6 meses
- Son herramientas que auxilian algún aspecto del proceso de traducción



# Herramientas auxiliares para programación de sistemas

- Cargadores y editores de enlace
- Generadores de analizadores léxico
- Generadores de Analizadores sintácticos
- Traductores dirigidos por sintaxis
- Generadores automáticos de código
- Dispositivos para el análisis de flujo de datos



# ¿Preguntas?

