

Sintaxis → Semántica

Compilador, para el lenguaje máquina

- Sintaxis = texto del programa

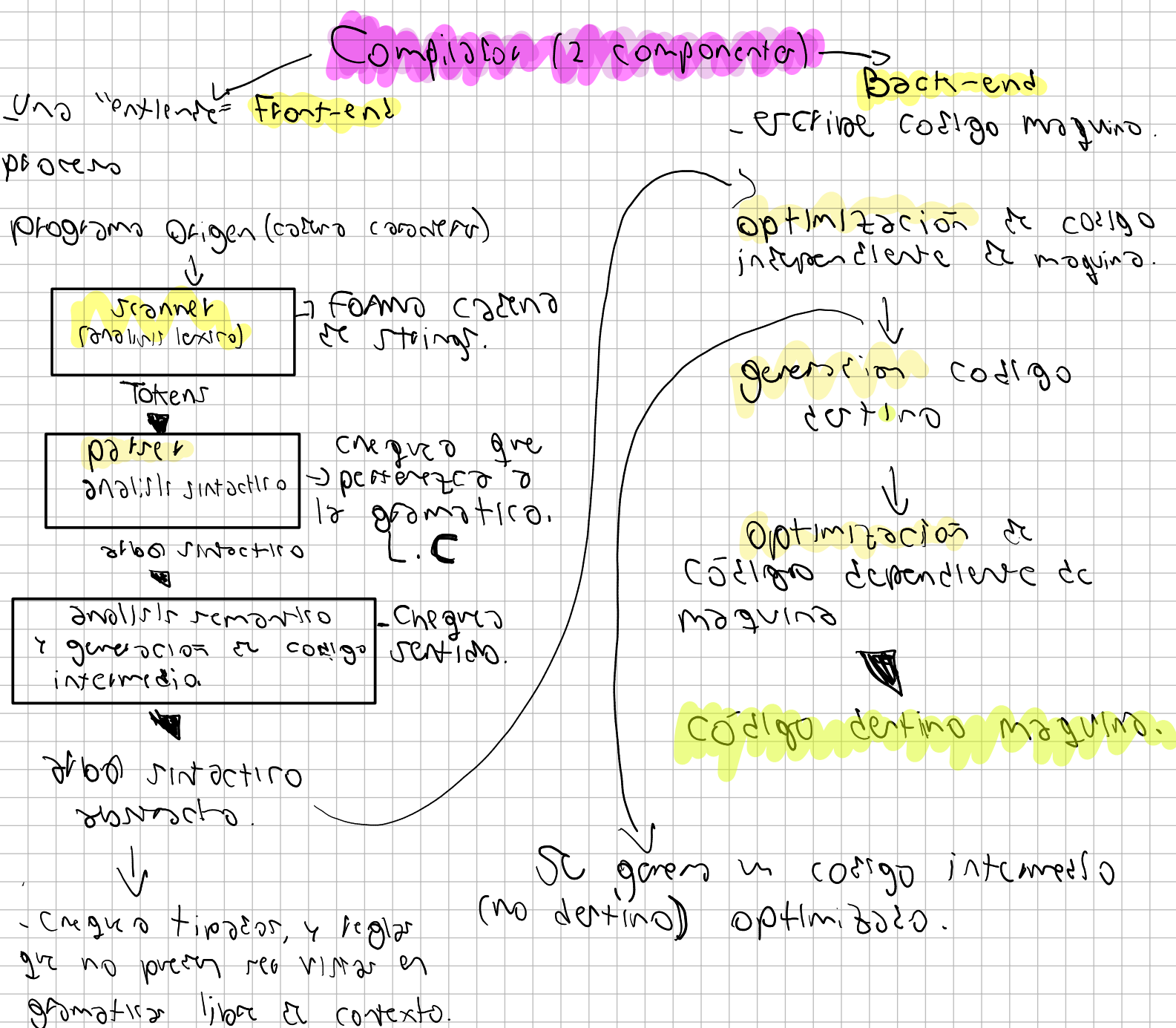
- Semántica = cómo se hace → código máquina.

- La implementación de un lenguaje transforma la sintaxis a ensamblar.

- Existen intérpretes que combinan traducción y ejecución.

- El compilador traduce a un programa en lenguaje máquina.

- Existen compiladores de lenguaje a otro lenguaje. (transpilador)



Comprobaciones semánticas del compilador:

- comprobación de tipos.
- declaración de variables.
- uso de identificación en contexto adecuado.
- comprobar argumentos.
- Si hay fallo, se genera error.

En tiempo de ejecución.

- Valor de reglar de no llimitar.
- división por 0.
- Si hay error se levanta una excepción.

Tipado fuerte

- Tiene tipado fuerte si siempre se detectan los errores de tipo.

- en tiempo de compilación o ejecución.

- fuerte = Java - ML - Haskell
- débil = Fortran - Pascal - C/C++ - Lisp.

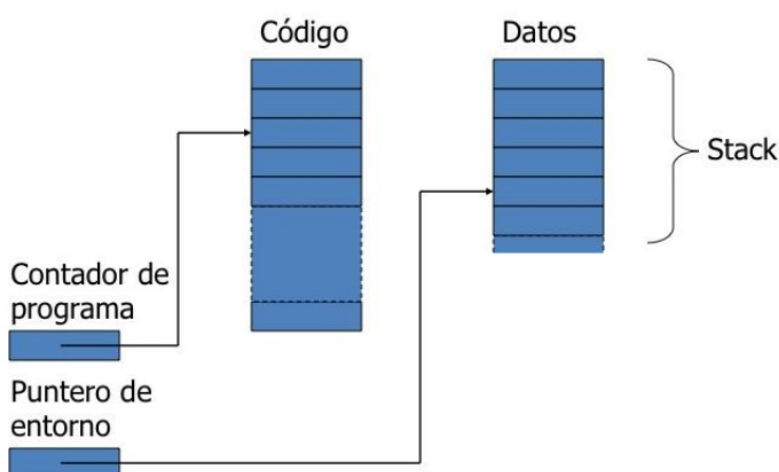
- hace que sea más seguro pero más lento por las comprobaciones dinámicas.

- En general los lenguajes escapan de la expresividad que tienen las gramáticas LC.

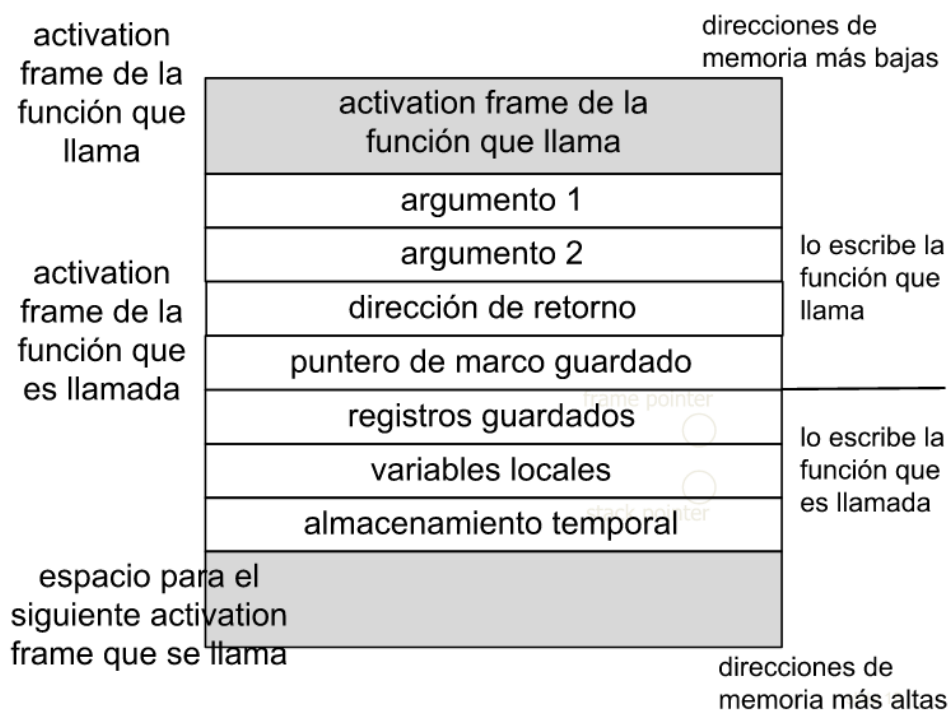
Semántica Operacional.

- representación abstracta de la ejecución de un programa, como secuencia de transiciones entre estados.

Los estados son una descripción abstracta de la memoria.



- cada bloque agrega un **activation record** o stack para las variables locales y puntero de retorno.



estructura de un activation record.

Cap 1, 2, 3 y 5 de la guía.