Bueno, tenemos entonces, el código fuente que son las instrucciones escritas en un lenguaje de programación legible por humanos y el código máquina que es el resultado obtenido luego de la compilación, lenguaje de máquina o binario.

----- SIGUIENTE DIAPO-----

Lo que yo voy a explicar es **cómo hace la máquina** para poder interpretar los distintos lenguajes que podríamos utilizar en los distintos archivos.

Esto lo logra por medio de una traducción llamada compilación**. La compilación** toma tooodo el código fuente, entero, y lo transforma en el programa ejecutable escrito en código máquina. La traducción la realiza de una sola vez y al mismo tiempo se encarga de **detectar errores**, por lo que, si llegara existir alguno en mi código, el programa **no** va a compilar y voy a tener que corregirlos para volver a solicitar al compilador que haga la traducción.

El ejecutable, valga la redundancia, va a poder ser ejecutado en cualquier otra máquina siempre y cuando tenga una arquitectura del CPU y sistema operativo similar a donde fue compilado. En caso contrario vamos a tener que hacer una nueva compilación para esta otra plataforma.

Tenemos otras dos formas para que los programas sean entendidos y ejecutados por una máquina independientemente de la arquitectura.

Estas son, máquinas virtuales e intérpretes.

**Máquina virtua**l, es un concepto intermedio entre el compilador y el intérprete que fue popularizado por Java. La idea acá es escribir los programas en un lenguaje de alto nivel, pero en vez de compilarlo a lenguaje de máquina, se lo traduce a un código intermedio que es ejecutado por un intérprete conocido como máquina virtual.

Y después tenemos el **intérprete**. Este proceso, también traduce el código fuente a código máquina, tal como el compilador, pero este, hace un análisis línea por línea en cada sistema en donde se ejecuta el código, traduciendo el programa a medida que se va necesitando, entonces, no genera ningún código ejecutable, sino que se mantiene cargado en memoria hasta que finalice el programa.

Emm bueno, a pesar de que no podamos elegir cómo va a ser ejecutado nuestro código, ya que esto va a depender de cómo el lenguaje de programación fue diseñado. Obviamente podemos encontrar ventajas y desventajas de uno sobre el otro.

En cuanto a la **velocidad**, por ejemplo, un programa compilado va a ser ejecutado más rápido que uno interpretado porque al entregar el ejecutable, la traducción ya está hecha, entonces la computadora solo va a tener que leerla. Mientras que el interpretado, como expliqué antes se irá traduciendo línea por línea a medida que sea necesario.

Por otro lado, un programa interpretado va a ser **multiplataforma**, mientras que uno compilado va a correr únicamente en las que sean similares a la máquina donde se realizó la compilación.

Y otra de las diferencias va a ser que, en un programa compilado, al generar un ejecutable, el código fuente va a estar únicamente en manos de su propietario, mientras que un programa interpretado lo dejará siempre a disposición de cualquiera.