

Yago Pego Martínez ([yago.pego.martinez@alumnos.upm.es](mailto:yago.pego.martinez@alumnos.upm.es))

Evaristo de Vega Galindo ([evaristo.devega.galindo@alumnos.upm.es](mailto:evaristo.devega.galindo@alumnos.upm.es))

**ESPECIFICACIONES**

**Enunciado:**

Escribir y compilar un programa que se llame “hello\_world.f90”, que abra una pantalla y que imprima “hello world”. La ventana deberá permanecer abierta hasta que se introduzca retorno por el teclado.

**Objetivo:**

Aprender el uso del compilador, editor y consola.

**INTRODUCCIÓN TEÓRICA**

1. **Arquitectura von Neumann:**

La **arquitectura von Neumann** (también conocida como modelo de von Neumann o arquitectura Princeton) se trata de una arquitectura de computadoras basada en la descrita por el matemático-físico John von Neumann y un grupo de compañeros de la Escuela Moore de Ingeniería en Filadelfia. Esta estructura se puede dividir en cinco partes:

- Una unidad de procesamiento (CPU), que incluye una unidad aritmético-lógica y los registros del procesador.

- Una unidad de control que, a su vez, contiene un registro de instrucciones y un contador de programa.

- Una memoria para almacenar tanto datos como instrucciones.

- Un almacenamiento masivo externo.

- Mecanismos de entrada y de salida.

La CPU (***Central Processing Unit***) es la encargada de procesar los datos y controlar la máquina. Dentro de la CPU, la unidad de control se encarga de leer las instrucciones y de enviar las órdenes a los componentes del procesador para que ejecuten las instrucciones; y la unidad aritmético-lógica es la encargada de realizar todas estas operaciones.

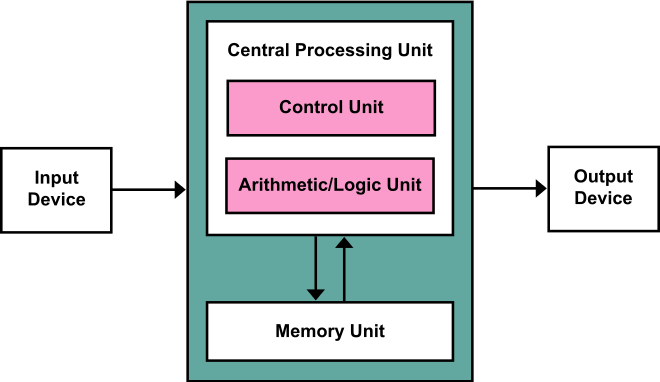
La **unidad de control** (UC), que organiza la implementación consistente de algoritmos de decodificación de instrucciones que provienen de la memoria del dispositivo, responde a situaciones de emergencia y realiza funciones de dirección general de todos los nodos de computación. Por lo general, el DO (dispositivo de operación) y la UC conforman la estructura llamada CPU. Cabe señalar que el requisito es consistente, el orden de la memoria es fundamental a la hora de la ejecución de la instrucción.

La **memoria** del dispositivo es aquella que contiene y memoriza las instrucciones y los datos durante un periodo de tiempo.

Los **dispositivos de entrada y de salida** permiten la comunicación entre un sistema de procesamiento de información, tal como la computadora y el mundo exterior, y posiblemente un humano u otro sistema de procesamiento de información.

Por lo general, las computadoras son máquinas de arquitectura von Neumann cuando:

1. Tanto los programas como los datos se almacenan en una memoria en común. Esto hace posible la ejecución de comandos de la misma forma que los datos.
2. Cada celda de memoria de la máquina se identifica con un número único, llamado dirección.
3. Las diferentes partes de la información (los comandos y los datos) tienen diferentes modos de uso, pero la estructura no se representa en memoria de manera codificada.
4. Cada programa se ejecuta de forma secuencial que, en el caso de que no hay instrucciones especiales, comienza con la primera instrucción. Para cambiar esta secuencia se utiliza el comando de control de transferencia.



1. **Evolución de Fortran:**

**Fortran** (previamente FORTRAN, del inglés *The IBM Mathematical* ***For****mula* ***Tran****slating System*) es un lenguaje de programación de alto nivel que está especialmente adaptado al cálculo numérico y a la computación científica. Desde que nació en el año 1957 para el equipo IBM 704, el FORTRAN ha estado en uso continuo hasta la actualidad, siendo la ingeniería y la ciencia sus principales aplicaciones.

Los organismos que actualmente regulan este lenguaje son la ANSI (*American National Standards Institute*) y la ISO (*International Standards Organization*).

Desde su origen, ha habido diversas actualizaciones en el sistema Fortran, por lo que se puede afirmar que es este un lenguaje de programación en constante evolución. A continuación se muestran las versiones aprobadas hasta nuestros días:

- En 1953, John Warner Backus propuso a su equipo de programadores (Richard Goldberg, Sheldon F. Best, Harlan Herrick, Peter Sheridan, Roy Nutt, Robert Nelson, Irving Ziller, Lois Haibt y David Sayre) una alternativa al entonces lenguaje ensamblador para programar el computador IBM 704.

- En octubre de 1956, aparece el primer manual FORTRAN y, en abril de 1957, el primer compilador de FORTRAN.

- El primer **FORTRAN** tenía hasta treinta y dos sentencias. Algunas de ellas son: “dimension”, “equivalence”, asignaciones, la sentencia “if”, “write”, “read”…

- **FORTRAN II** aparece en 1958. Su principal mejora fue la introducción de la programación por procedimientos (*procedural programming*), a partir de parámetros. “Subroutine”, “function”, “end” y, más tarde, “complex” y “double precisión”, son sentencias que aparecen por primera vez con FORTRAN II.

- **FORTRAN IV** nace en 1962 para el IBM 7030 primero, y, más tarde para IBM 7090, IBM 7094 e incluso IBM 1401 (1966). Nace como resultado de la gran demanda del cliente. Añade nuevas sentencias como “logical” *data type*.

- **FORTRAN 77**: aparecen las primeras versiones en 1977. Para corregir los defectos del anterior FORTRAN 66, surgen nuevas sentencias como “implicit”, “character”, “parameter”, “save”, “sqrt”…

- En **Fortran 90** se pueden observar ya los numerosos avances en programación que se dieron desde los anteriores estándares. Se puede destacar la introducción de los identificadores de hasta treinta y un caracteres (antes eran seis), los comentarios entre líneas, los procedimientos recursivos (*recursive procedures*), acumulador de operaciones (*operator overloading*)… Es además —¿por qué no decirlo?— la versión que hemos utilizado en el presente “Hito 1”.

- **Fortran 95** es considerado poco más que una pequeña revisión del anterior estándar. Aun así da lugar a la aparición de nuevas extensiones: “forall” y “where”, que facilitan la vectorización, permite definir procedimientos puros o elementales y expandió la habilidad para el uso de la inicialización en datos.

- **Fortran 2003** supone una profunda renovación en cuanto a la introducción de nuevos recursos. Entre las principales revisiones cabe citar la interoperabilidad con el lenguaje de programación C, las mejoras en la manipulación de datos y un soporte adaptado para el uso internacional.

- **Fortran 2008**, como el del 95, es una ligera revisión del anterior, aprobada en septiembre de 2010.

Desde el año 2015 se está trabajando por conseguir una nueva revisión del lenguaje Fortran, que se espera que aparezca en 2018. Se esperan mejoras en la interoperabilidad entre Fortran y C, nuevos recursos disponibles y la eliminación de pequeñas deficiencias para facilitar la tarea de programación.

1. **Representación real (4), (8). Coma flotante. IEEE.**

Fortran tiene cinco tipos de datos intrínsecos: “integer”, “real”, “complex”, “logical” y “character”. Cada uno de estos tipos puede estar caracterizado adicionalmente por un “kind”, que, básicamente, define la representación interna de las variables: para las numéricas, la precisión y el rango, y, para las dos últimas, aspectos específicos sobre almacenamiento. Por consiguiente, es un concepto abstracto el que modela los límites para la representación de estos datos, que se expresa como un conjunto de números reales, que denotan los bytes ({1, 2, 4, 8}, por ejemplo, para los tipo “integer”). Concretamente, para cada tipo existe un número de bytes ya predeterminado: (4) para los “real”, en los que nos centraremos en esta parte del informe.

Los datos de tipo real no tienen representación exacta y poseen al menos dos “kinds” o identificadores de clase: el predeterminado (4) y una con mayor precisión (“double precisión”).

Valor de la variable

Variable

En el programa se escribirá:

- real (kind=4)::a=8.0 → real (4)::a=8.0 → real::a=8.0

Identificador de tipo

Número de bytes

Identificador de clase

Los cuatro bytes permiten la utilización de cantidades comprendidas entre ±10-7 y ±1038.

- Con doble precisión: real (8)(…)

“(8)” debe aparecer

Estos ocho bytes permiten la manipulación de datos en el intervalo (10-16, 10308).

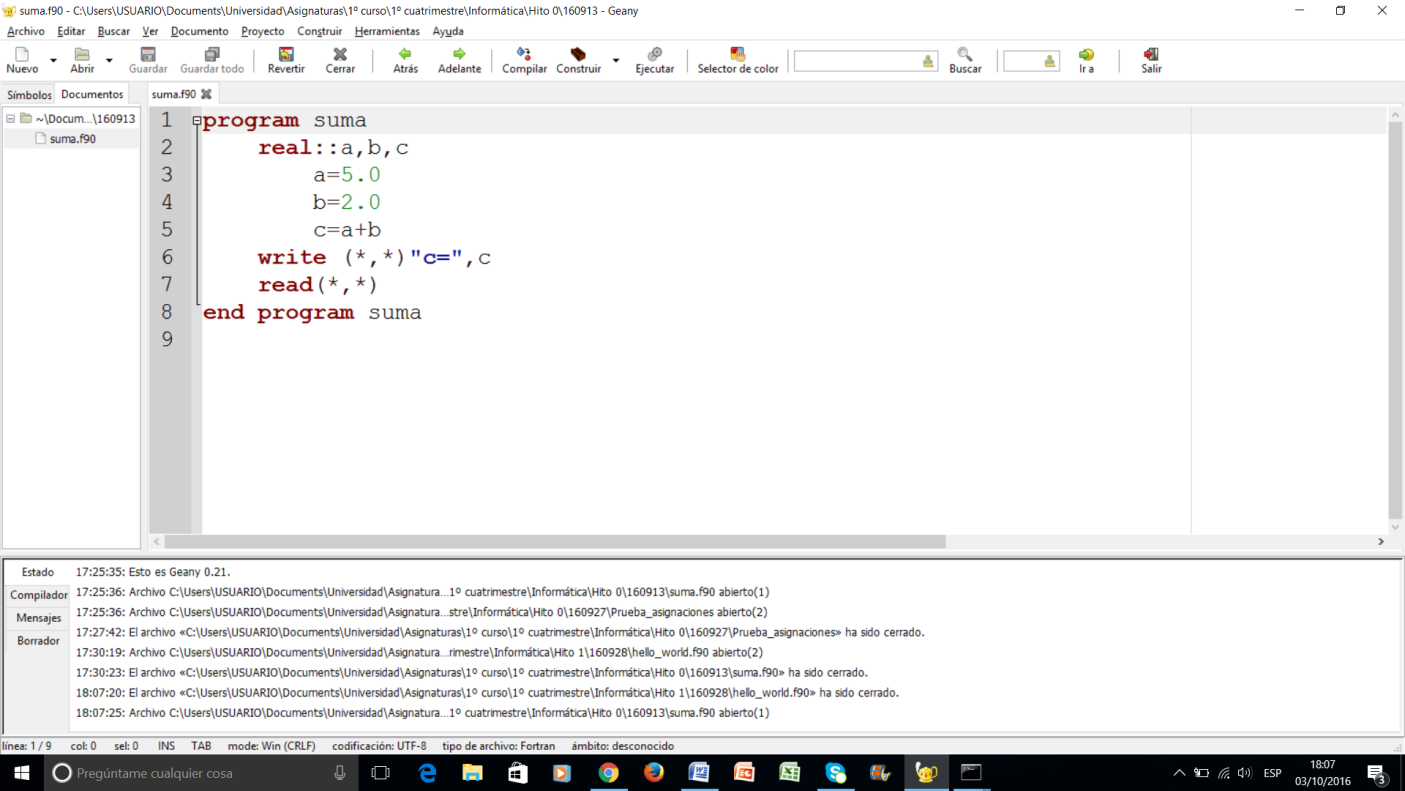
Los datos numéricos también pueden ser escritos de la siguiente manera:

sea el número ‘-8,32’,

de manera predeterminada: “real (4):: a = -8.32(e0)” ≡ “real:: a = -832e-1”

con doble precisión: “real (8):: a = -8.32d0” ≡ “real (8):: a = -832d-1”

Con identificadores de tipo real se pueden realizar las operaciones de suma, resta, multiplicación y división, entre otras. A continuación, ejemplo de suma en Fortran 90.



El concepto de **coma flotante** (‘floating point’ en inglés) se refiere a la forma de notación científica utilizada en los microprocesadores con la cual se pueden representar números racionales extremadamente grandes o pequeños de una manera muy eficiente y compacta, de modo que se puedan realizar las principales operaciones aritméticas.

Entre otras funciones, da la posibilidad de operar con irracionales (√2, π, e…) de forma, según se requiera, más o menos precisa.

El estándar para la representación en coma flotante es el IEEE 754.

El **Instituto de Ingeniería Eléctrica y Electrónica** es una organización profesional sin ánimo de lucro que acoge a ingenieros de todo el mundo dedicados a la estandarización y el desarrollo de áreas técnicas.

Su nacimiento se remonta al año 1884, cuando personalidades como Thomas Alva Edison, Alexander Graham Bell o Franklin Leonard Pope dieron forma a este proyecto. Es en 1963 cuando, al fusionarse con IRE (Institute of Radio Enginners) y AIEE (American Institute of Electrical Engineers), toma el nombre, IEEE, que lleva en la actualidad.

La subdivisión IEEE 754 es la que se dedica al campo de la computación y programación y a partir sobre todo de la segunda mitad del siglo XX ha potenciado el uso estandarizado de la coma flotante y mejorado notablemente la CPU Y FPU.

1. **Sintaxis y semántica:**

SINTAXIS

del latín tardío *syntaxis*, y este del griego σύνταξις *sintaxis*, de συντάσσειν *syntássein*, ‘**disponer conjuntamente**’, ‘**ordenar’**.

1. f. *Gram*. Parte de la gramática que estudia el modo en que se combinan las palabras y los grupos que estas forman para expresar significados, así como las relaciones que se establecen entre todas esas unidades.

**2. f. *Inform*. Conjunto de reglas que definen las secuencias correctas de los elementos de un lenguaje de programación.**

(<http://dle.rae.es/?id=XzfiT9q>)

SEMÁNTICA

del griego σημαντικός *sēmantikós*, ‘significativo’

1. adj. Perteneciente o relativo a la semántica.

2. f. Significado de una unidad lingüística. La semántica de esa palabra se presta a discusión.

3. f. *Ling*. Disciplina que estudia el significado de las unidades lingüísticas y de sus combinaciones.

(<http://dle.rae.es/?id=XVRDns5>)

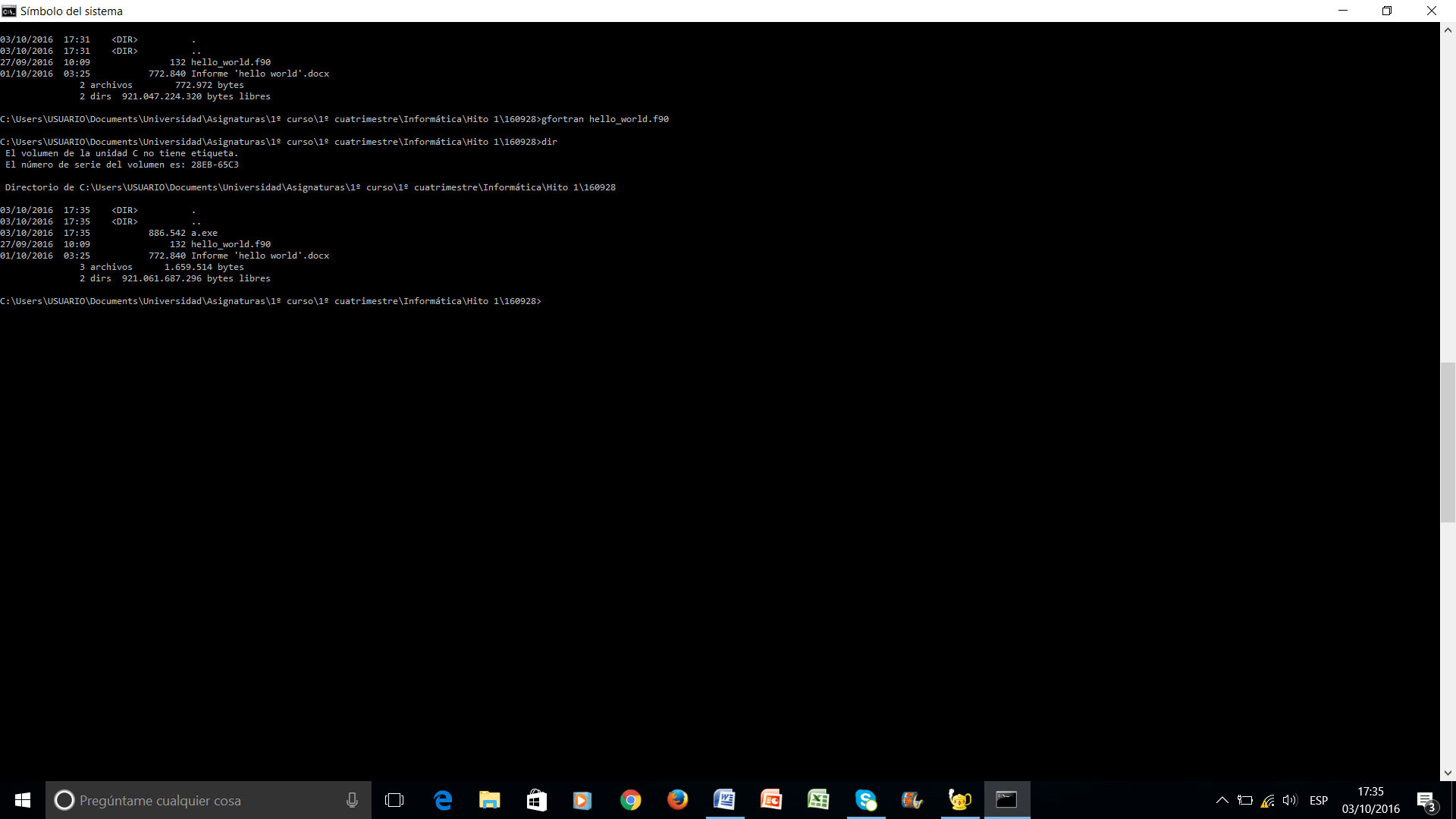
**En la**[**Teoría de lenguajes de programación**](https://es.wikipedia.org/wiki/Teor%C3%ADa_de_lenguajes_de_programaci%C3%B3n)**, la semántica es el campo que tiene que ver con el estudio riguroso desde un punto de vista matemático del significado de los**[**lenguajes de programación**](https://es.wikipedia.org/wiki/Lenguajes_de_programaci%C3%B3n)**.**

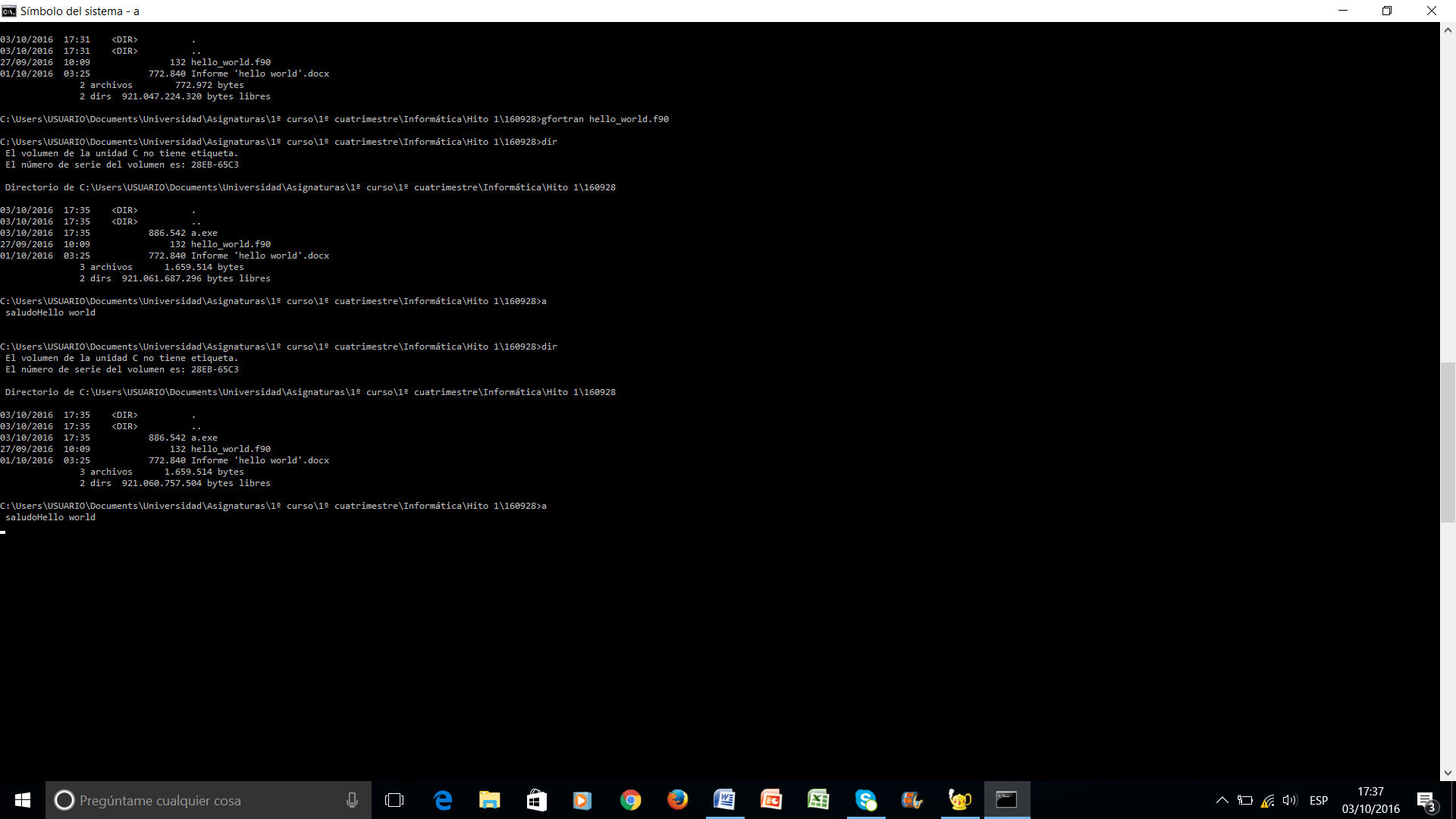
(<https://es.wikipedia.org/wiki/Sem%C3%A1ntica_de_lenguajes_de_programaci%C3%B3n>)

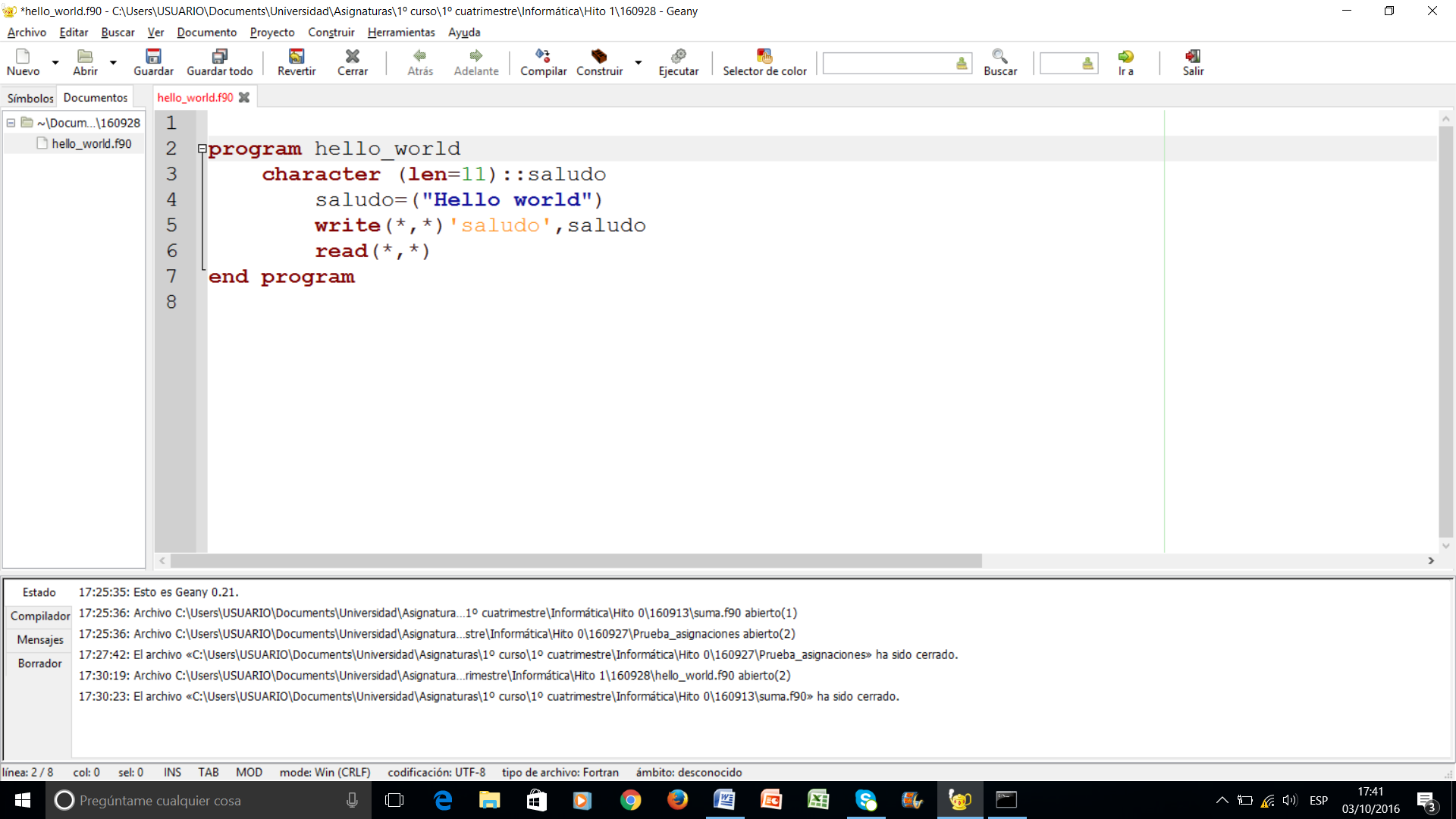
**BIBLIOTECA DE VARIABLES**

La única variable utilizada en el programa “hello world” es de tipo “**character**” (letras, números signos de puntuación…).

**RESULTADOS**

***Compilación*** *del programa “hello world” en ‘Símbolo del sistema’ (command prompt)*

***Ejecución*** *del programa “hello world” en ‘Símbolo del sistema’ (command prompt)*

*Escritura del programa “hello world” en el editor Geany*

**CONCLUSIÓN**

Con este primer hito, que nos ha servido como una pequeña iniciación a la programación, hemos podido aprender, y comprender, el manejo de la consola o símbolo del sistema para compilar, construir o ejecutar programas con extensión “.f90”, es decir, a través de Fortran 90.

A partir además del editor *Geany* hemos creado el programa “hello world”, haciendo uso por primera vez de la variable carácter (“character” en inglés), que permite, al ser el archivo ejecutado, la aparición de las mencionadas palabras. Esto nos hace ver que Fortran puede ser utilizado además con fines lingüísticos, no solo matemáticos, como se le presupone.

Por otro lado, la parte del informe que más nos ha enseñado ha sido la introducción teórica, pues, a base de buscar información, hemos aprendido cómo funcionan los ordenadores (arquitectura de von Neumann), algo de historia sobre el lenguaje de programación Fortran y las posibilidades que ofrece la variable real.

*Evaristo de Vega Galindo*

*Yago Pego* *Martínez*