HITO 5

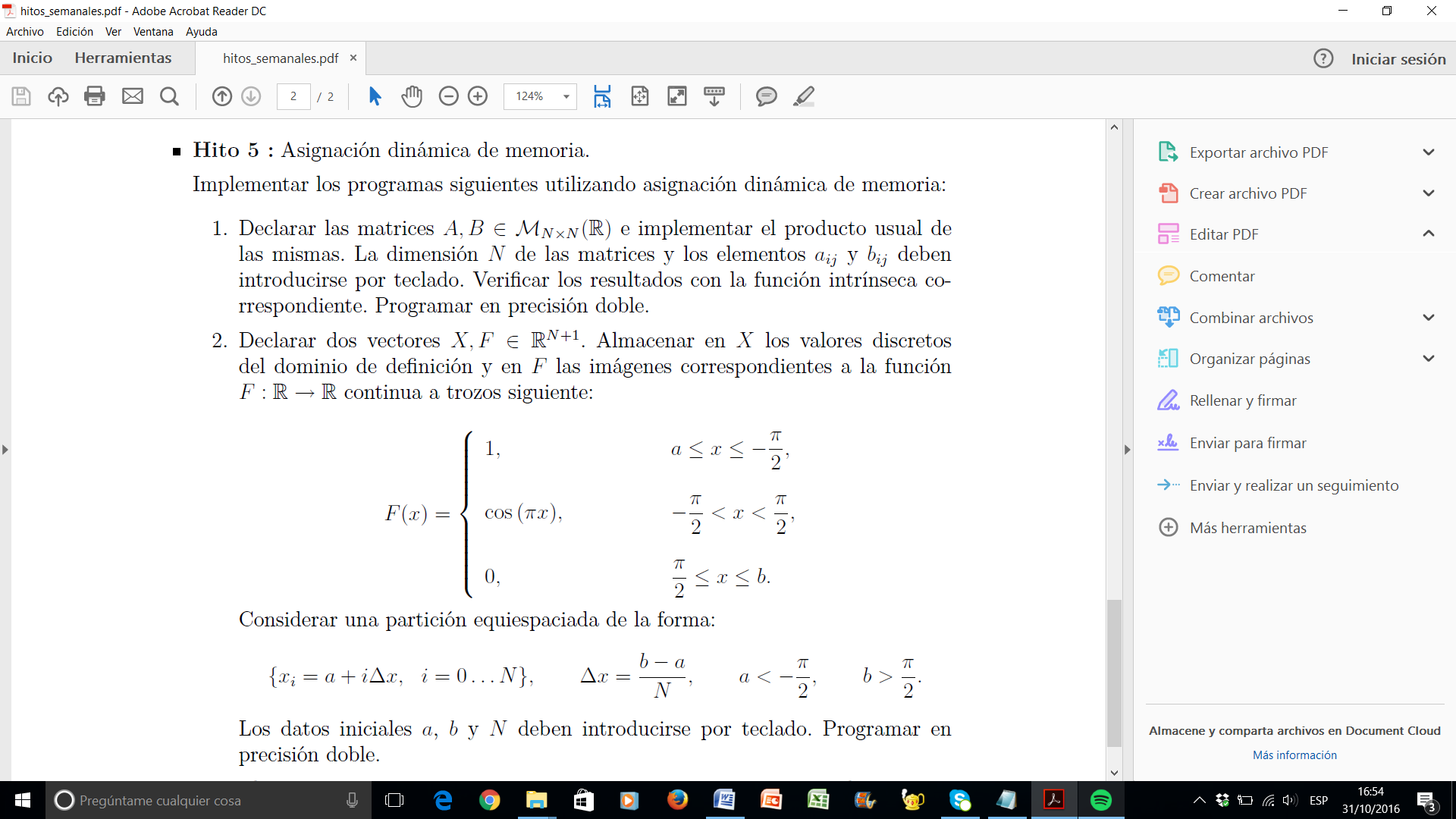
asignación dinámica de memoria

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))

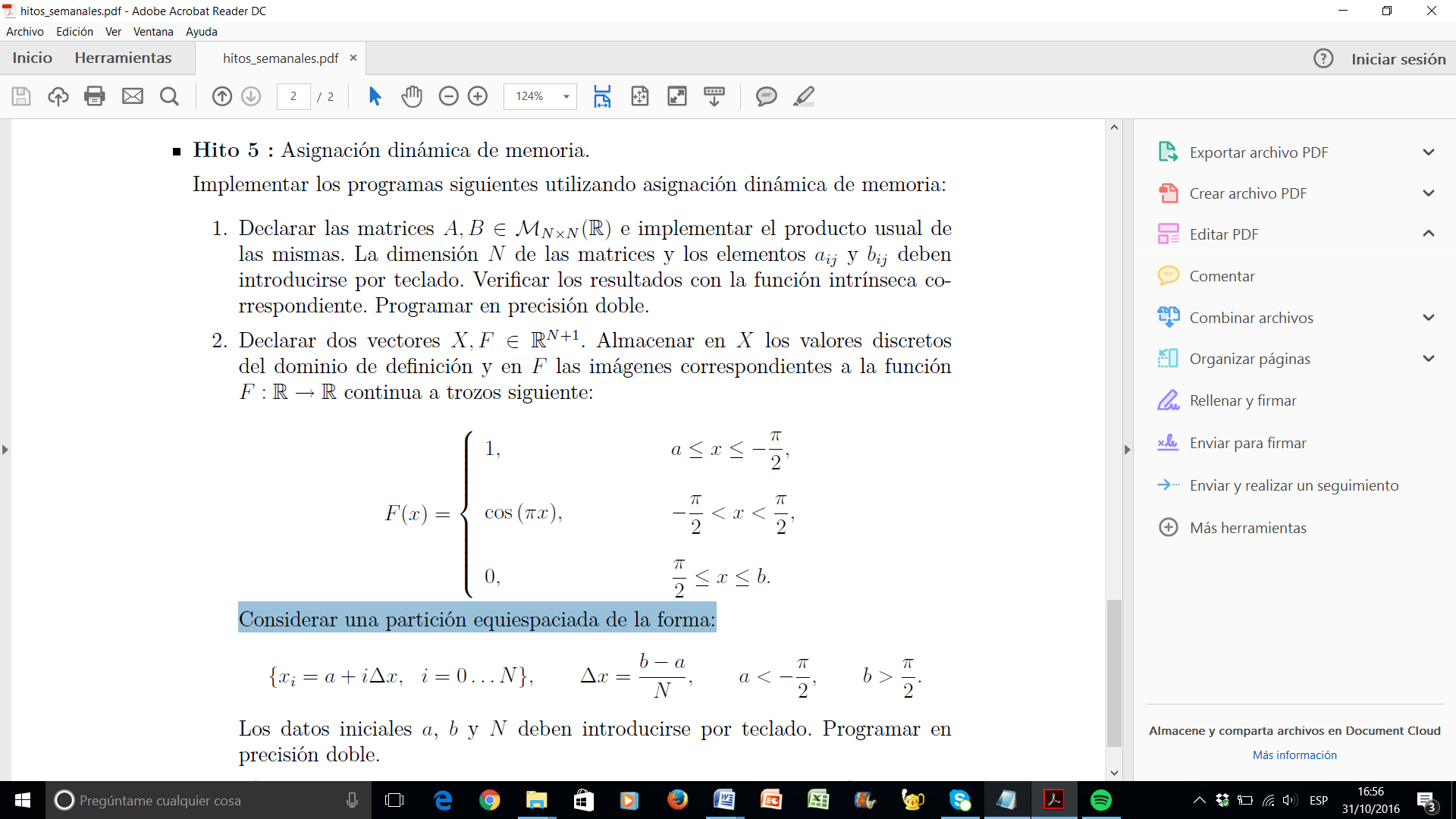
**ESPECIFICACIONES**

**Enunciado**: Implementar los programas siguientes utilizando asignación dinámica de memoria:

1. Declarar las matrices *A*, *B* ∈ *MN ×N* (*R*) e implementar el producto usual de las mismas. La dimensión N de las matrices y los elementos *aij* y *bij* deben introducirse por teclado. Verificar los resultados con la función intrínseca correspondiente. Programar en precisión doble.

2. Declarar dos vectores *X*, *F* ∈ *RN +1* . Almacenar en *X* los valores discretos del dominio de definición y en *F* las imágenes correspondientes a la función *F : R* → *R* continua a trozos siguiente:

Considerar una partición equiespaciada de la forma:



Los datos iniciales *a*, *b* y *N* deben introducirse por teclado. Programar en

precisión doble.

3. El mismo enunciado anterior pero introduciendo *a*, *b* y *Δx* por teclado.

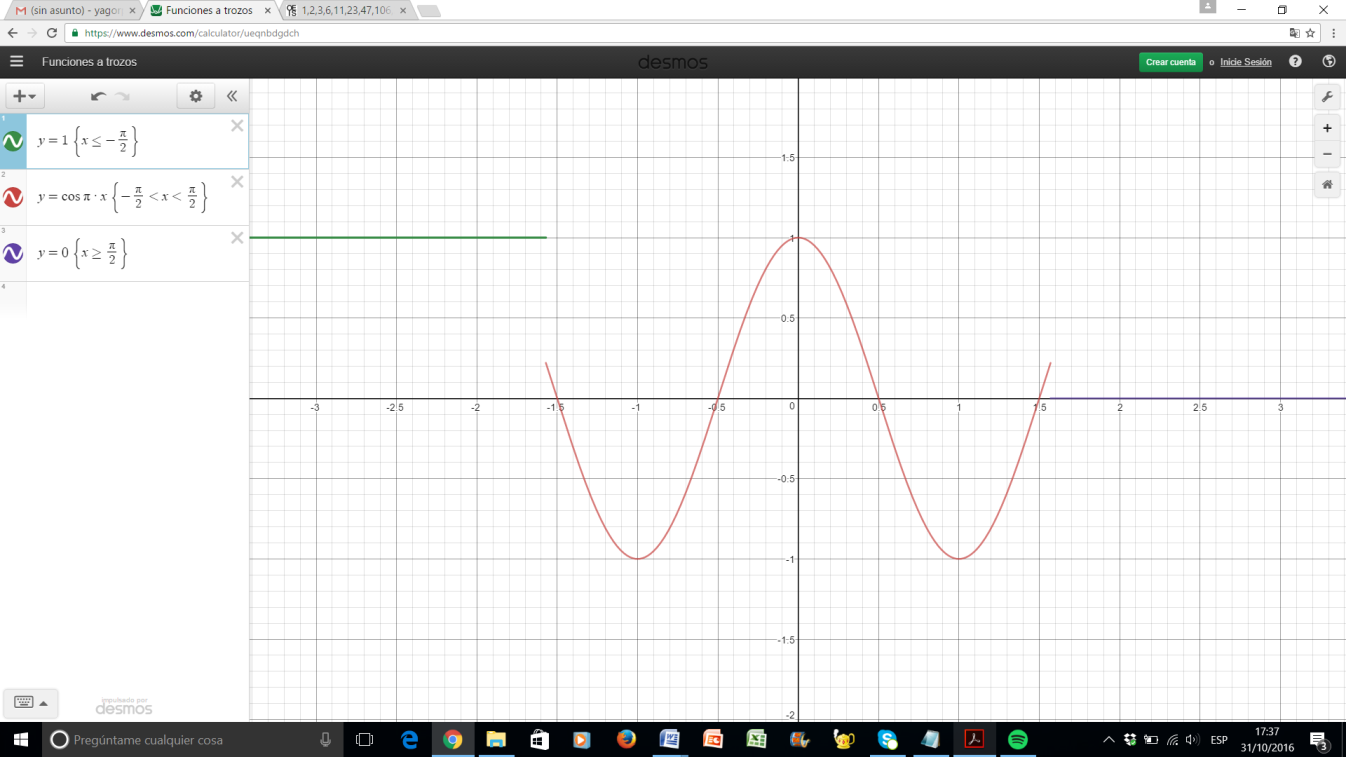
**Objetivo:** aplicar una nueva y alternativa forma de asignación de memoria: la dinámica. Trabajar con las sentencias *allocate* y *deallocate*, con matrices y vectores de dimensión introducida por teclado y representar funciones a trozos.

**FUNDAMENTOS TEÓRICOS**

Trabajamos en este hito de dos maneras: en el primer programa, con matrices, y en el segundo, con una función a trozos. Aunque son aparentemente distintos, programando hemos seguido un método similar para ambos, utilizando variables que se han de declarar como *allocatable*.

El primer programa nos pide realizar el producto usual de dos matrices del mismo orden (mismo número de filas y columnas). Las matrices, conjuntos de elementos ordenados por filas y columnas, tienen una forma peculiar para la multiplicación. Matemáticamente se deben multiplicar cada una de las filas de la primera matriz por cada una de las columnas de la segunda. El resultado será una matriz con tantas filas como la primera matriz y tantas columnas como la segunda.

El producto de matrices, por tanto, no cumple la propiedad conmutativa: AB BA. Sí cumple sin embargo la asociativa — A(BC) = (AB)C — y la distributiva — A(B + C) = AB + AC —.

El segundo y tercer programa se resumen en escribir en pantalla las abscisas y ordenadas que tiene la anterior función en cualquier punto de la recta real.

Esta es la citada función que, como podemos ver, es discontinua en los puntos y .

**BIBLIOTECA DE VARIABLES**

- En el primer programa:

“**i**”, “**j**” y “**k**” son variables enteras que se han identificado como las filas y columnas de las diferentes matrices. Funcionan con el bucle “do”.

“**N**” es la variable entera que nos permite introducir por teclado la dimensión que queremos dar a las matrices del programa.

“**alloc\_error**” y “**dealloc\_error**” son variables enteras que permiten asignar y liberar, de forma dinámica, memoria a las variables que se consideren.

“**suma**”, real de precisión doble, no es más que una variable que nos facilita y sirve de “enlace” para calcular la matriz resultado.

“**A**”, “**B**”, “**C**” y “**D**” son variables reales de precisión doble que actúan como matrices de dimensión no definida, es decir, de dimensión asignable. “A” y “B” son, en este caso, las dos matrices que se multiplican (en ese orden), “C” es la matriz resultado utilizando el bucle “do”, y “D” es equivalente a “C”, solo que aparece de nuevo en pantalla, al realizarse la comprobación del producto con la función intrínseca “matmul”.

- Comunes en el segundo y tercer programa:

“**N**” y “**D**” son las variables enteras que en ambos programas se refieren al número de particiones de la función en el intervalo [a, b] y a la dimensión de los vectores (N+1), respectivamente. En el último programa, se podrá dar el caso de que se amplíe el número de particiones y, por tanto, la dimensión de los vectores.

“**a**” y “**b**” son variables reales y en doble precisión en el segundo, y enteras en el tercero (para facilitar los cálculos). No son más que los puntos que marcan los puntos de inicio y final del intervalo.

“**delta\_x**” es una variable real de precisión doble (en el segundo), y entera (en el tercero), que se refiere a la distancia que existe entre cada partición del intervalo. Se relaciona con las anteriores variables de la forma: delta\_x . En el último programa cuando la anterior ecuación se cumpla únicamente con “N” no entero, se creará un nuevo incremento de x: “delta\_x\_2”.

“**i**” es la variable entera que, con el bucle “do”, nos da las abscisas y sus correspondientes ordenadas, según la dimensión definida.

“**pi**” es la variable real de precisión doble sobre la que se carga un valor lo más semejante a ∏. “pi” es igual a cuatro veces la inversa de la tangente de uno.

“**x**” y “**f\_x**” son variables reales de doble precisión y *allocatable*s. Se les podrá dar la dimensión que se considere, o la obtenida tras introducir el valor de “delta\_x” en el tercer programa.

“**x\_i**” y “**f\_x\_i**” son los distintos componentes del vector según la dimensión y la abscisa y ordenada, respectivamente, que marquen las particiones.

“**alloc\_error**” y “**dealloc\_error**”, como en el anterior programa, son variables enteras que tienen la misma función: asignar o liberar memoria de forma dinámica.

- En el tercer programa:

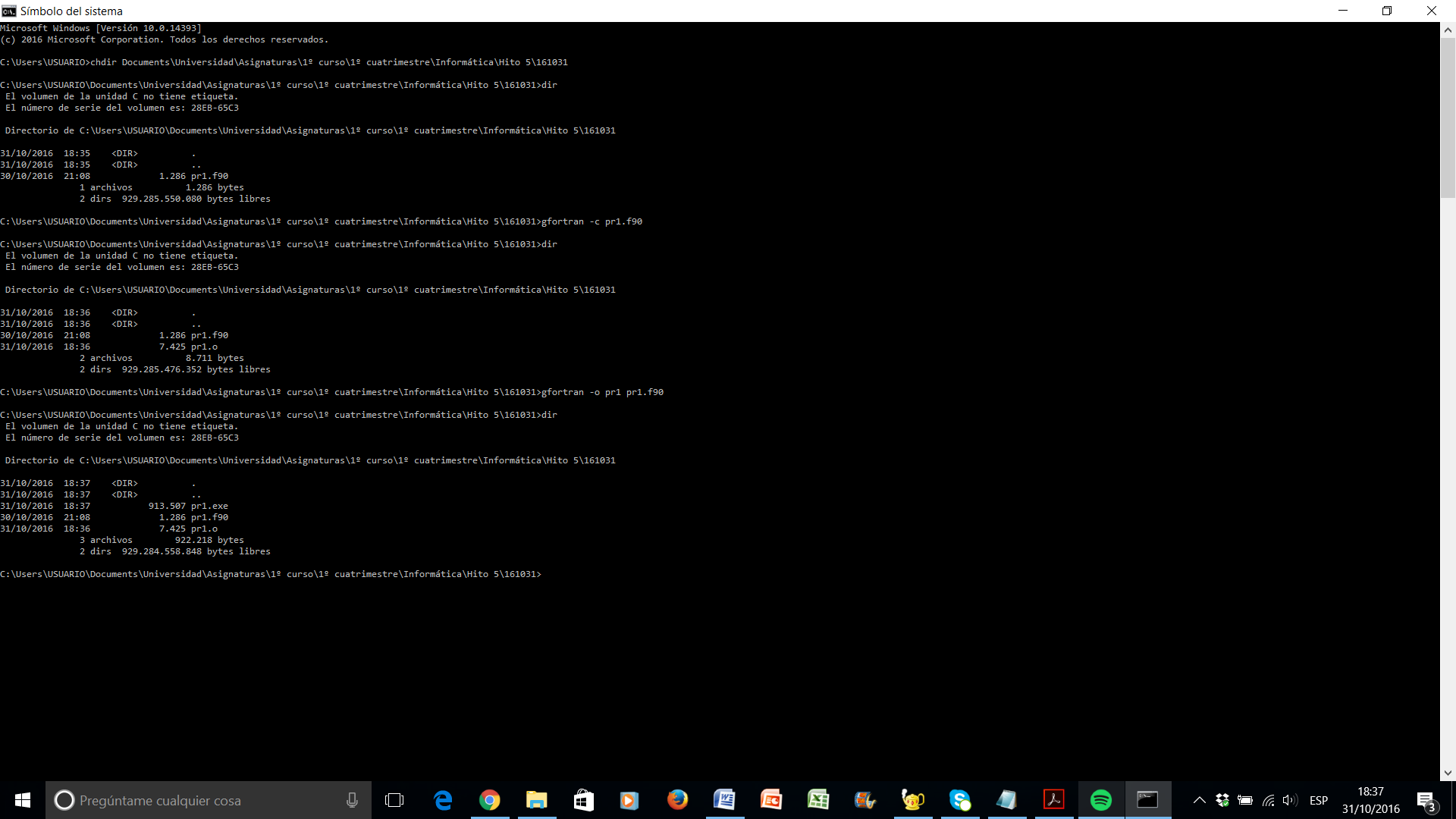
“**resto**” es la variable real de precisión doble que distingue a ambos programas. En el tercero, es la diferencia entre el punto “b” y el “a” y el número de particiones obtenidas “N” por el “delta\_x” escogido.

“**N\_2**” y “**D\_2**” son el nuevo número de particiones y dimensión de los vectores, respectivamente, una vez, que exista el resto y se vuelva a dividir el intervalo [a, b].

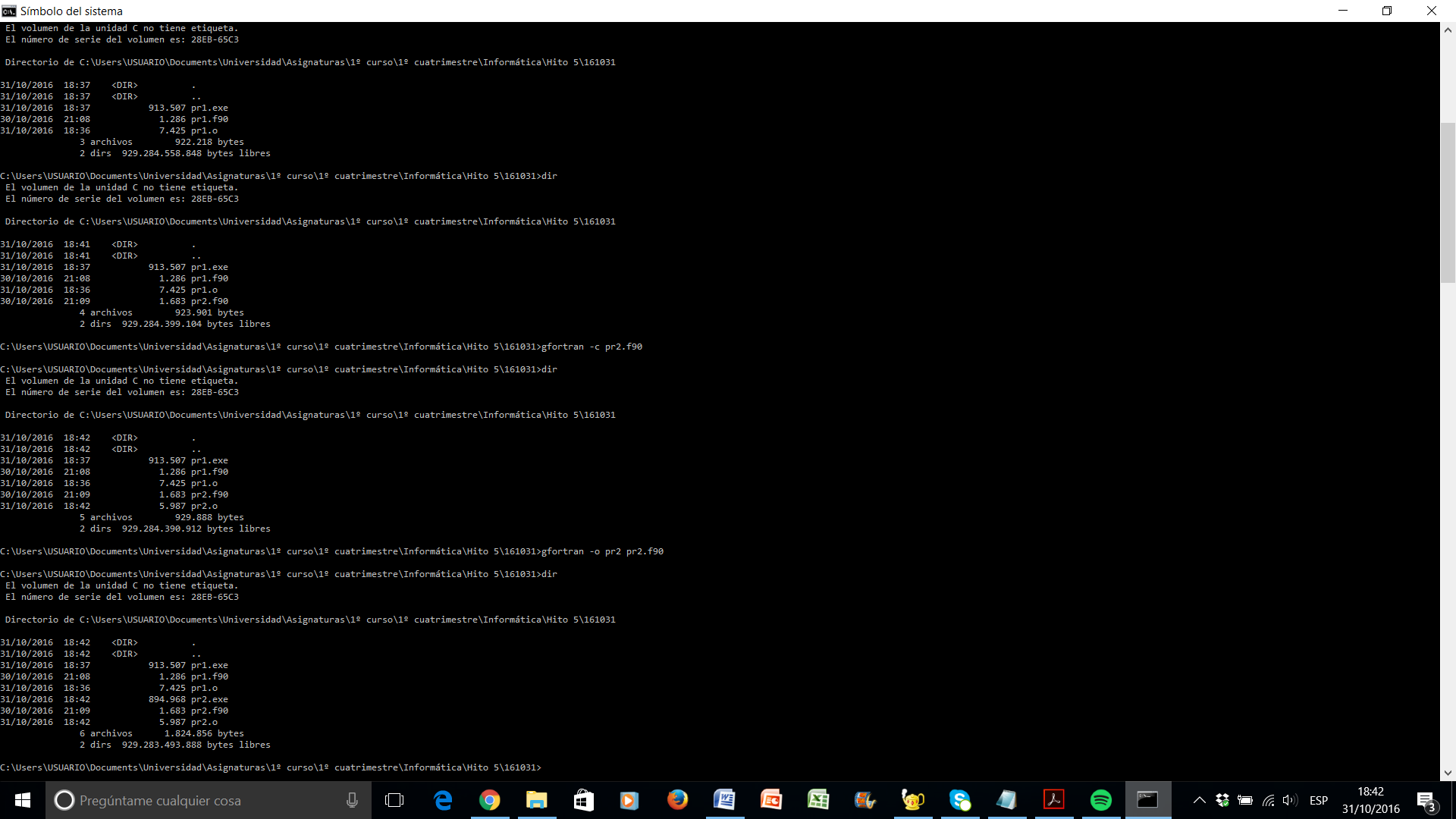
“**x\_j**” y “**f\_x\_j**” son variables reales de precisión doble que actúan como las nuevas componentes de los vectores dominio e imagen, una vez que haya habido resto.

“**x\_2**” y “**f\_x\_2**” son variables reales y de doble precisión. Se les asignará una dimensión determinada a partir de la sentencia “allocate”. Aparecerán en pantalla cuando el resto en el tercer programa sea mayor que cero.

“**j**” es una variable entera que tiene la misma función que “i” para las nuevas particiones. Funciona con el bucle “do”.

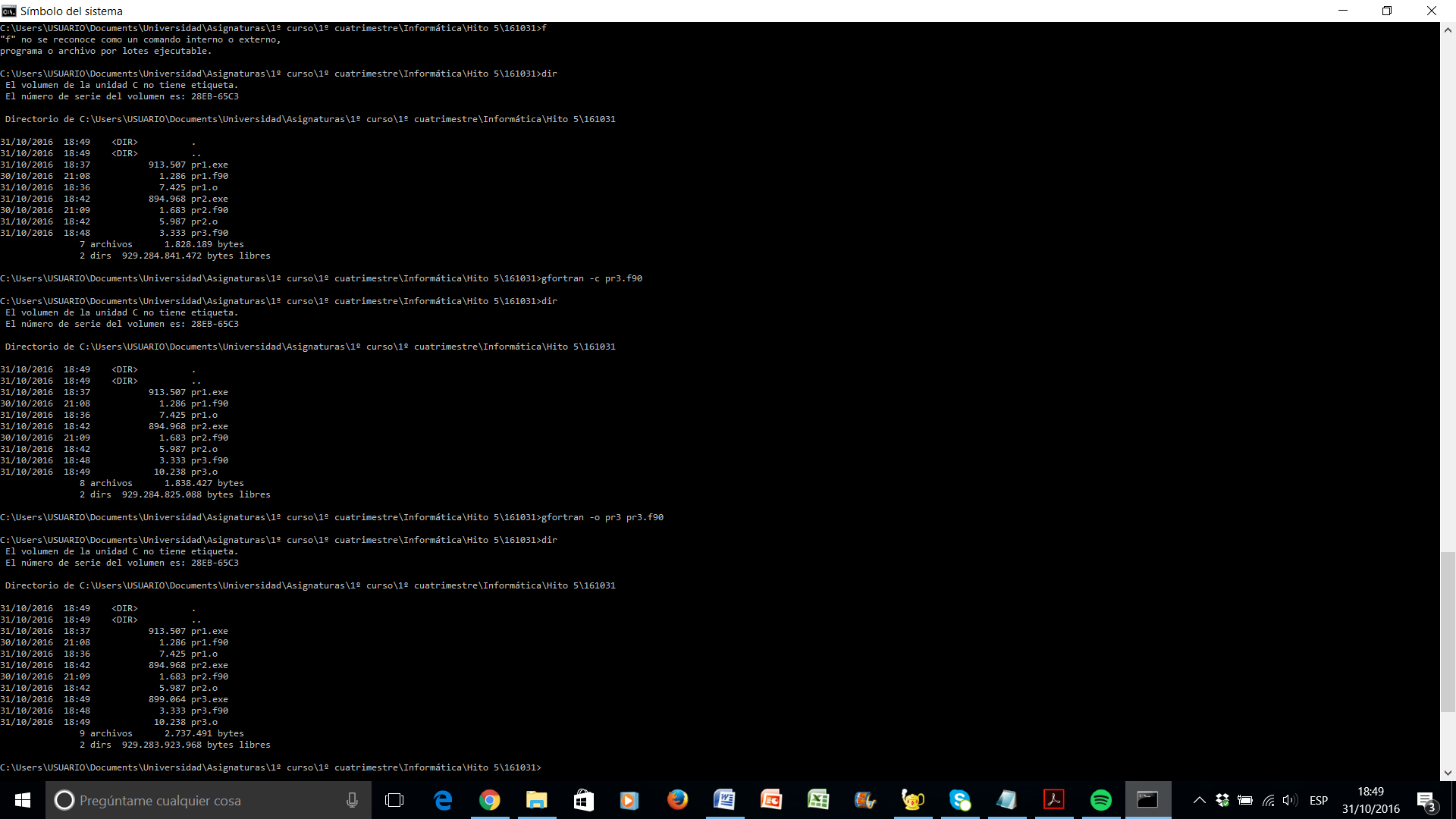
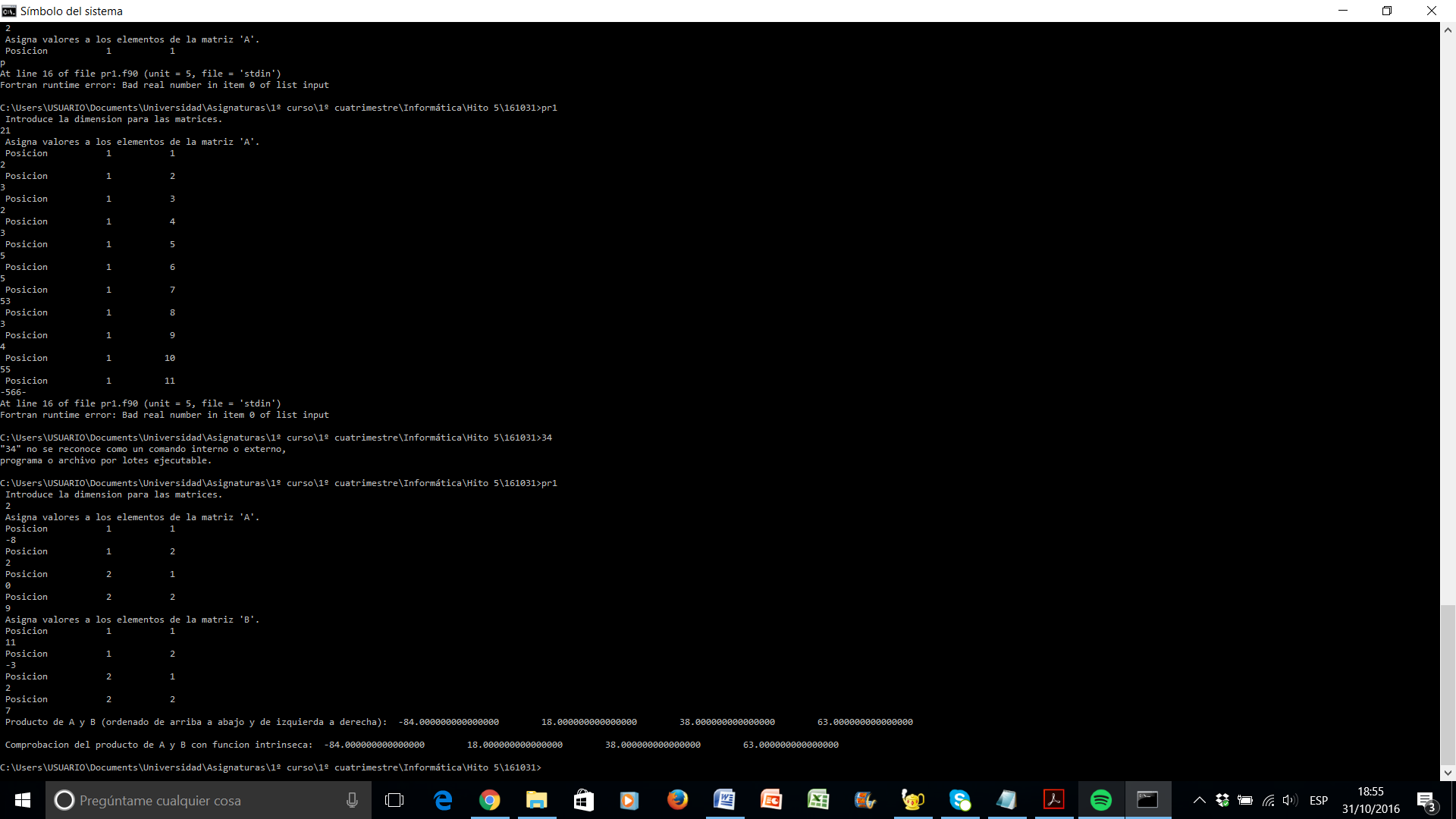
**RESULTADOS**

Compilación y ejecución del primer programa

****

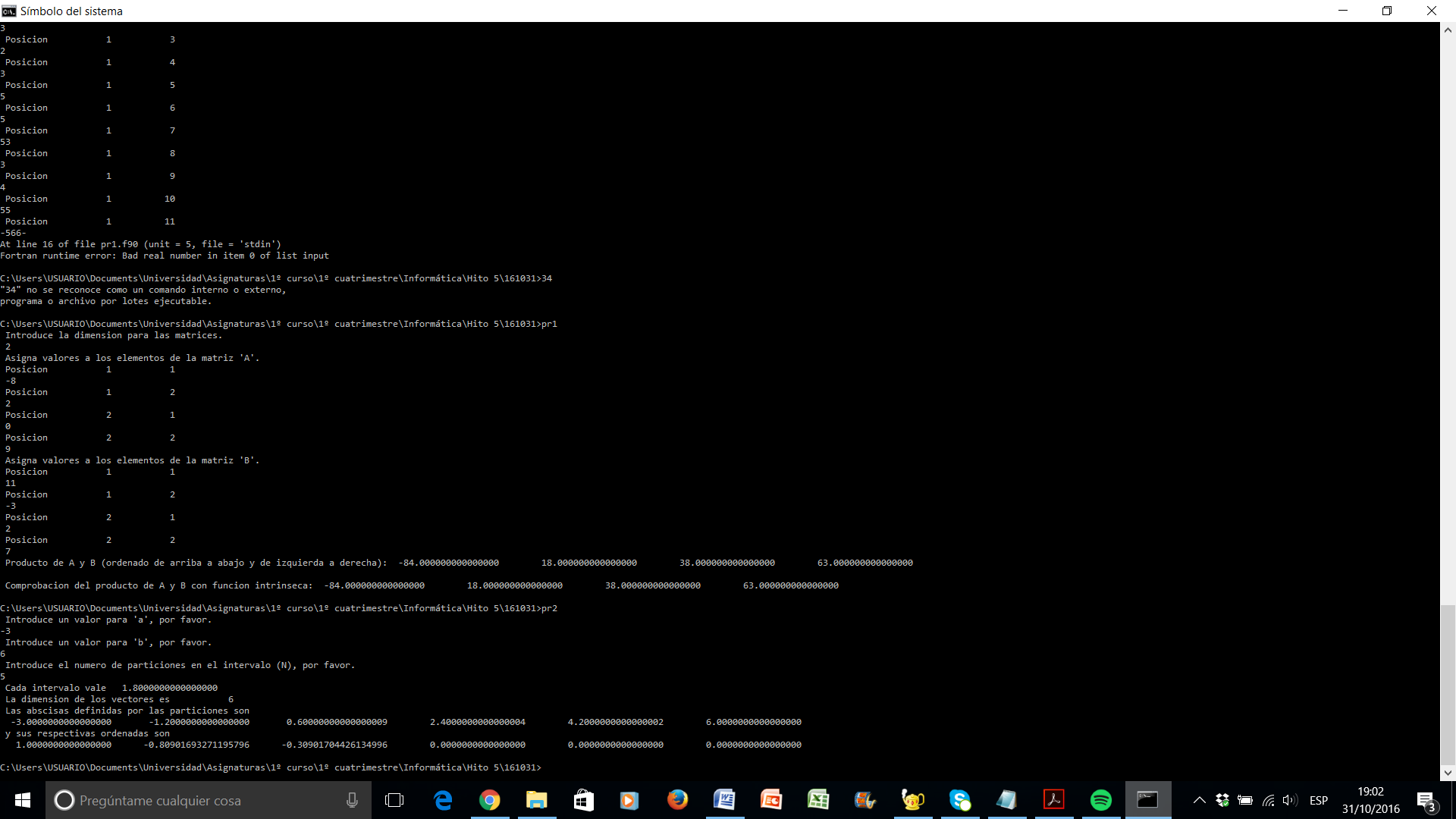
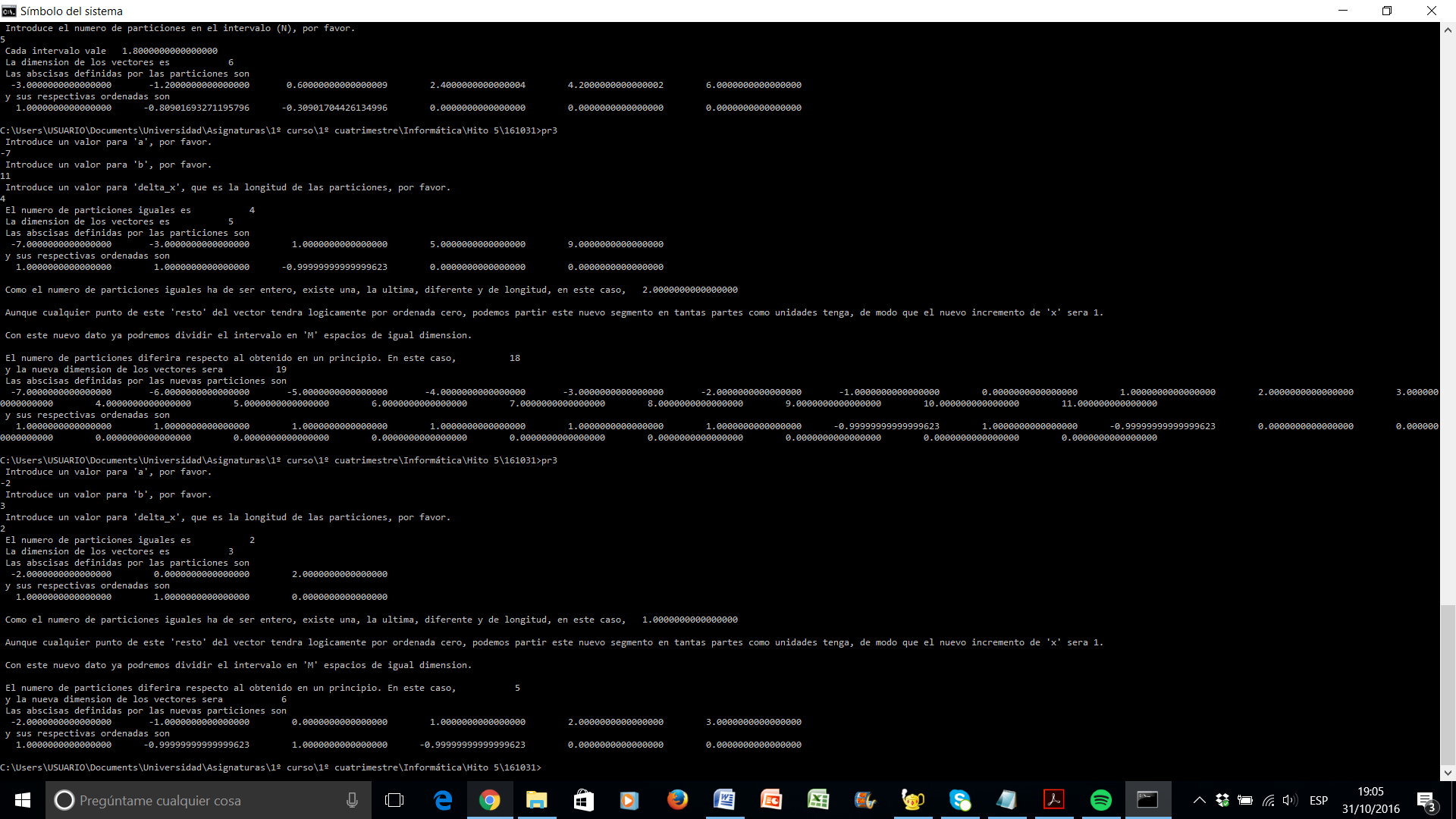
Compilación y ejecución del segundo programa

Compilación y ejecución del tercer programa

****

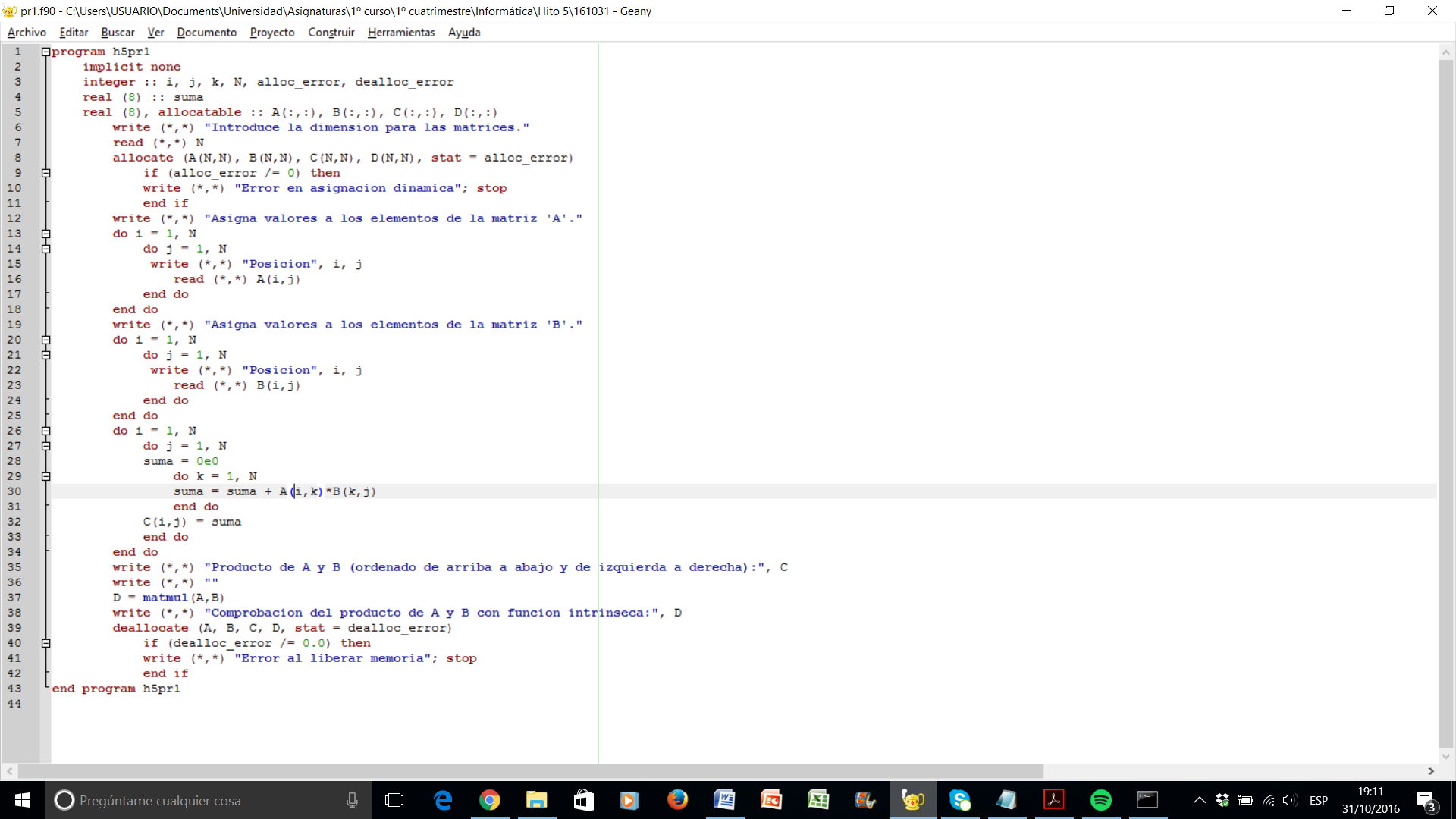
Prueba del primer programa en la consola

Prueba del primer programa en la consola

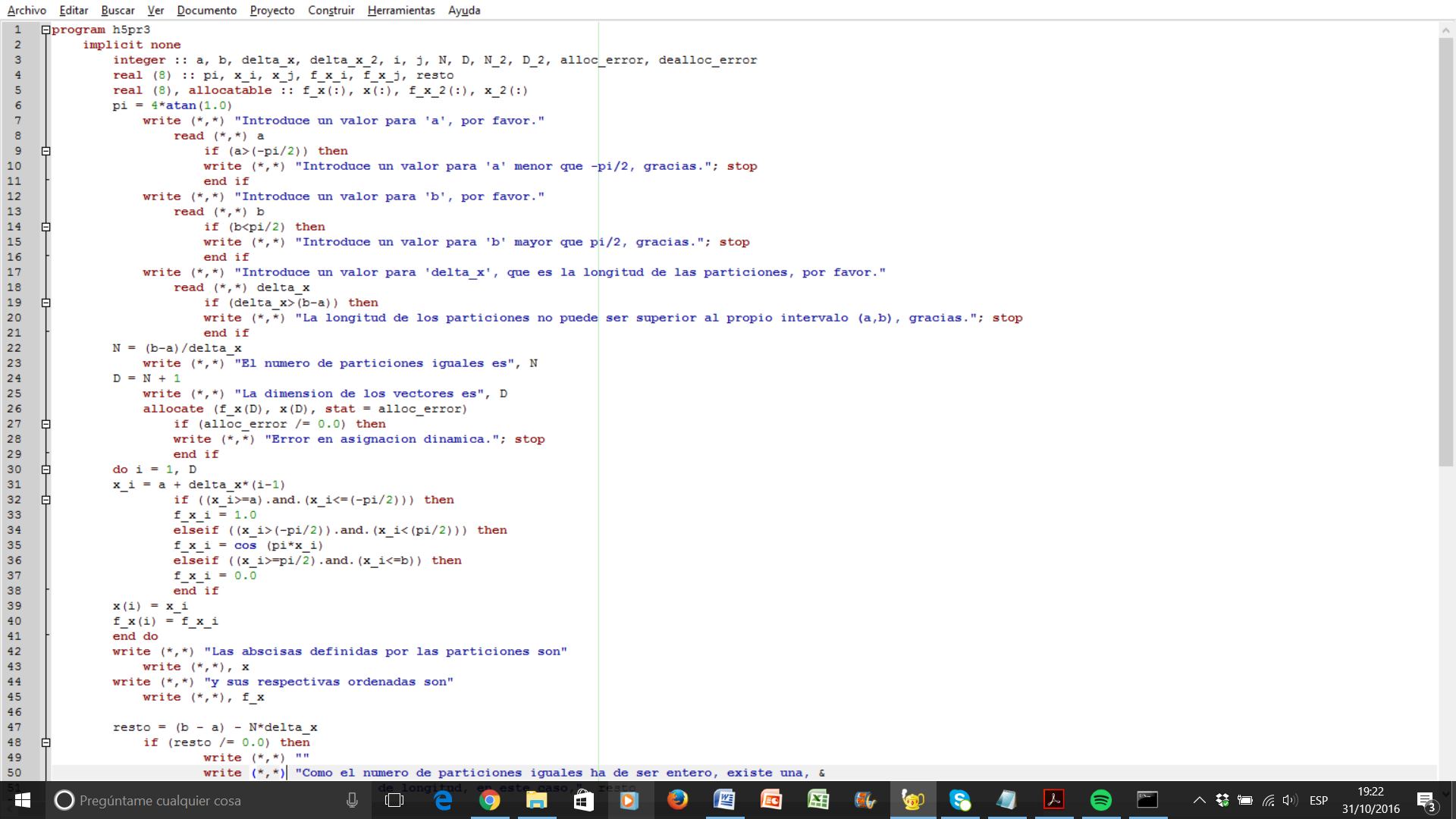
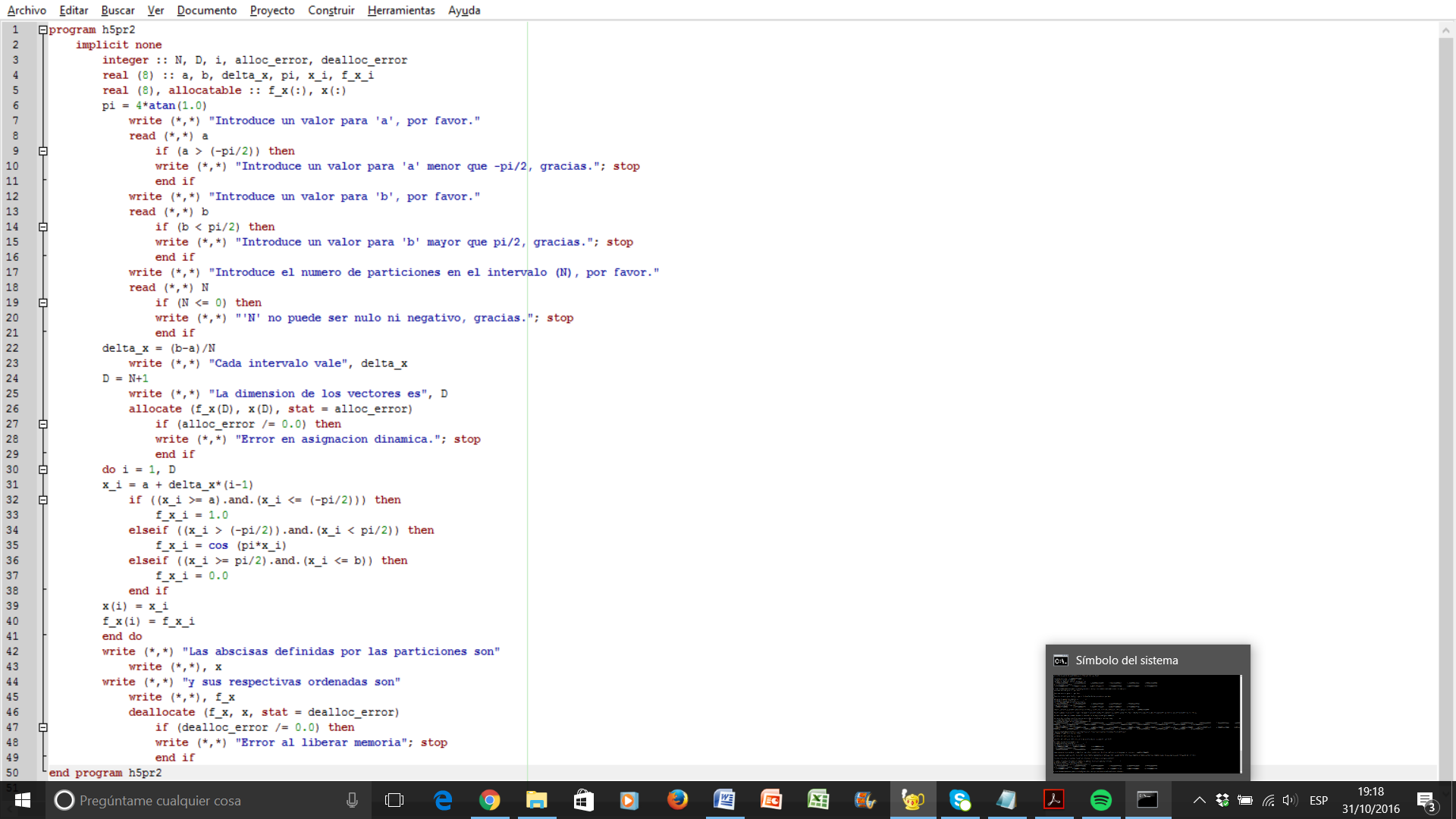
****

Prueba del segundo programa en la consola

Prueba del tercer programa en la consola

****

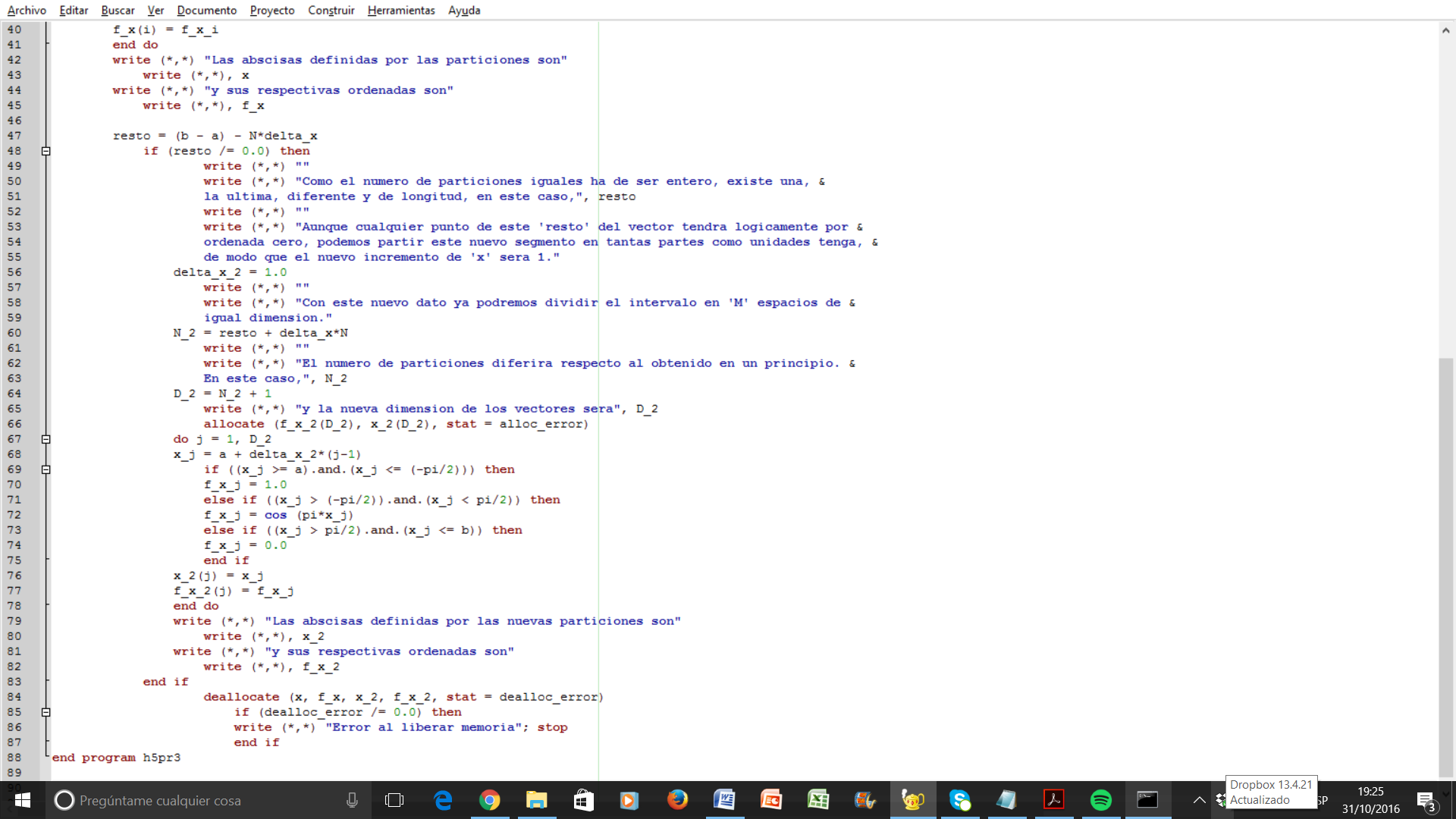
Escritura del primer programa en el editor “Geany”

****

Escritura del tercer programa en el editor “Geany” (I)

Escritura del segundo programa en el editor “Geany”

Escritura del segundo programa en el editor “Geany”

****

Escritura del tercer programa en el editor “Geany” (II)

**CONCLUSIÓN**

Con este hito hemos entendido las posibilidades que ofrece la asignación dinámica de memoria, es decir, podemos dar la dimensión que se desee a los vectores o matrices en cualquier parte del programa y, después, liberar esa memoria en cualquier otro punto.

Aunque nos gusta y tenemos ese afán por el conocimiento, el progreso y la meta (estando un día hasta las dos y media de la mañana: no se podía ir uno a dormir sin haber hecho bien el tercer programa), notamos que cada semana cuesta más terminar el trabajo a tiempo; se junta con otras tareas y requiere gran parte del día.

Desde otro punto de vista, y bastante más importante, este trabajo continuo y semanal nos está ayudando a potenciar nuestro aprendizaje y mecanizar cada una de las sentencias y funciones que vamos añadiendo.

Que lo que hace una o dos semanas te daba un tremendo dolor de cabeza te lleve ahora unos cinco, diez minutos es un buen motivo para seguir trabajando y aprendiendo.

¡Ahí seguimos!

*Evaristo de Vega Galindo*

*Yago Pego Martínez*