Metodología de desarrollo de programas I

La labor de programación puede considerarse como un caso particular de la resolución de problemas, lo cual, consiste en encontrar una estrategia para conseguir una solución. Dicha estrategia se expresará como una colección de reglas que si se siguen conducirán a la solución.

En el **modelo de programación imperativa** un programa se expresa como una serie de instrucciones u ordenes que gobiernan el funcionamiento de una máquina. Un programa puede considerarse como la solución de un problema determinado.

1. **Descomposición de un programa en Sub-programas.**

Los problemas que se pueden resolver de una manera directa se denominan problemas **simples o triviales.**

Cualquier programa de cierta complejidad necesitara una estrategia mas compleja para encontrar la solución. El método mas general de resolver este tipo de problemas complejos, es descomponerlo en subprogramas mas sencillos.

Por ejemplo:

Tenemos que construir una casa. Primero nos centramos en construir las paredes, el suelo, el techo. Luego en construir las ventanas y las puertas. Luego en pintar y decorar con muebles…etc.

1. **Desarrollo de un programa por refinamientos sucesivos.**

Consiste en descomponer un programa en subprogramas más simples. Es lo que se denomina **programación estructurada.** Dicha técnica consiste en expresar inicialmente el programa a desarrollar como una acción global y luego descomponerla en acciones mas sencillas hasta llegar a acciones simples que puedan ser expresadas directamente como sentencias de lenguaje de programación. La forma en que varias acciones simples se combinan formando una acción más global o compleja constituye el esquema de acción compuesta. El primer esquema que veremos será es el **esquema secuencial.**

**Un ejemplo claro de refinar sucesivamente:**

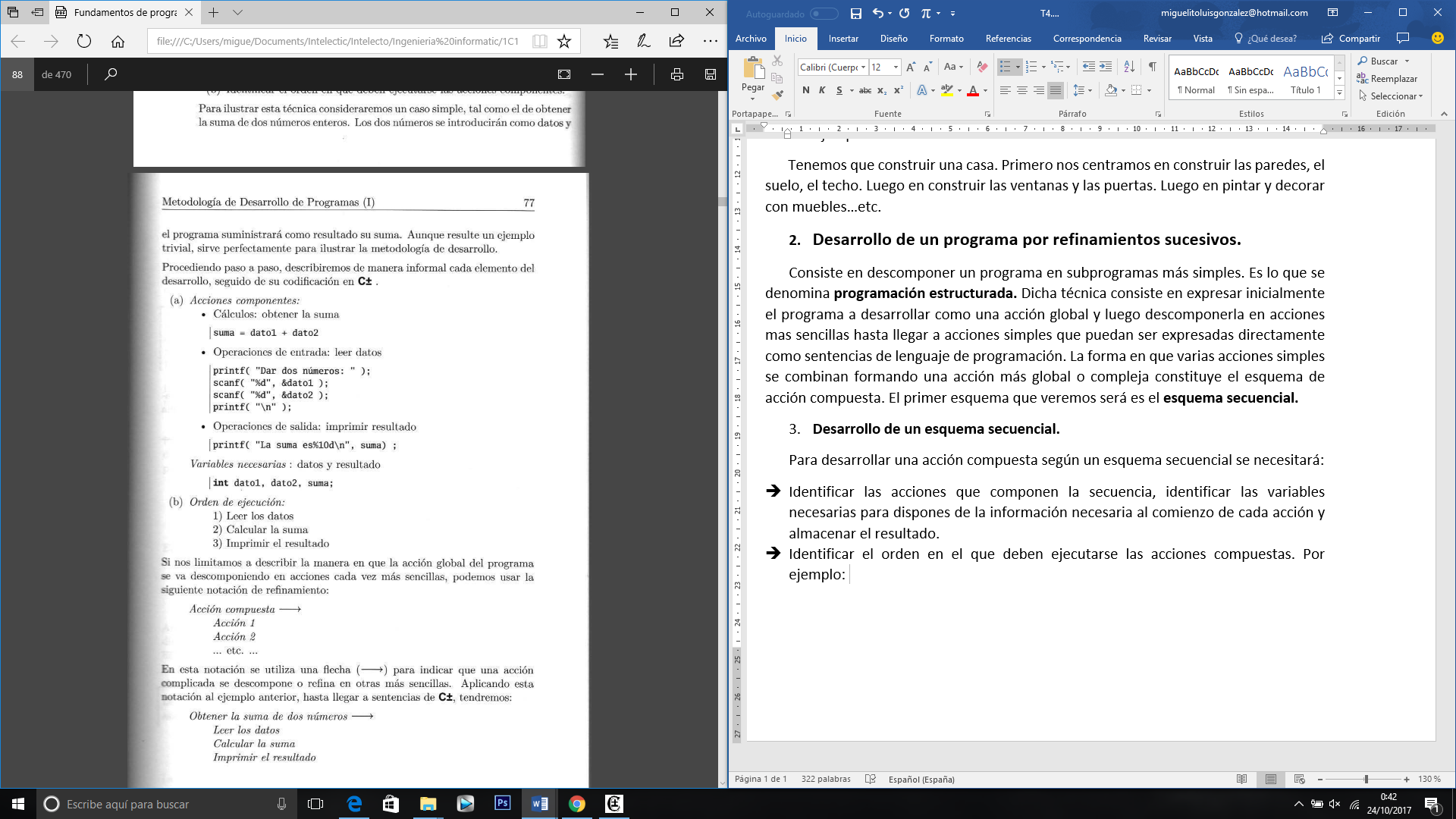
**Quiero obtener el grado en ingeniería informática🡪 REFINAMOS 🡪 Quiero aprobar el primer año de ingeniería informática 🡪 REFINAMOS 🡪 Quiero aprobar el primer cuatrimestre 🡪 REFINAMOS 🡪 Quiero aprobar x cantida de asignaturas.**

**Es decir PROBLEMA COMPLEJO 🡪 SUB-PROBLEMAS SIMPLES.**

1. **Desarrollo de un esquema secuencial.**

Para desarrollar una acción compuesta según un esquema secuencial se necesitará:

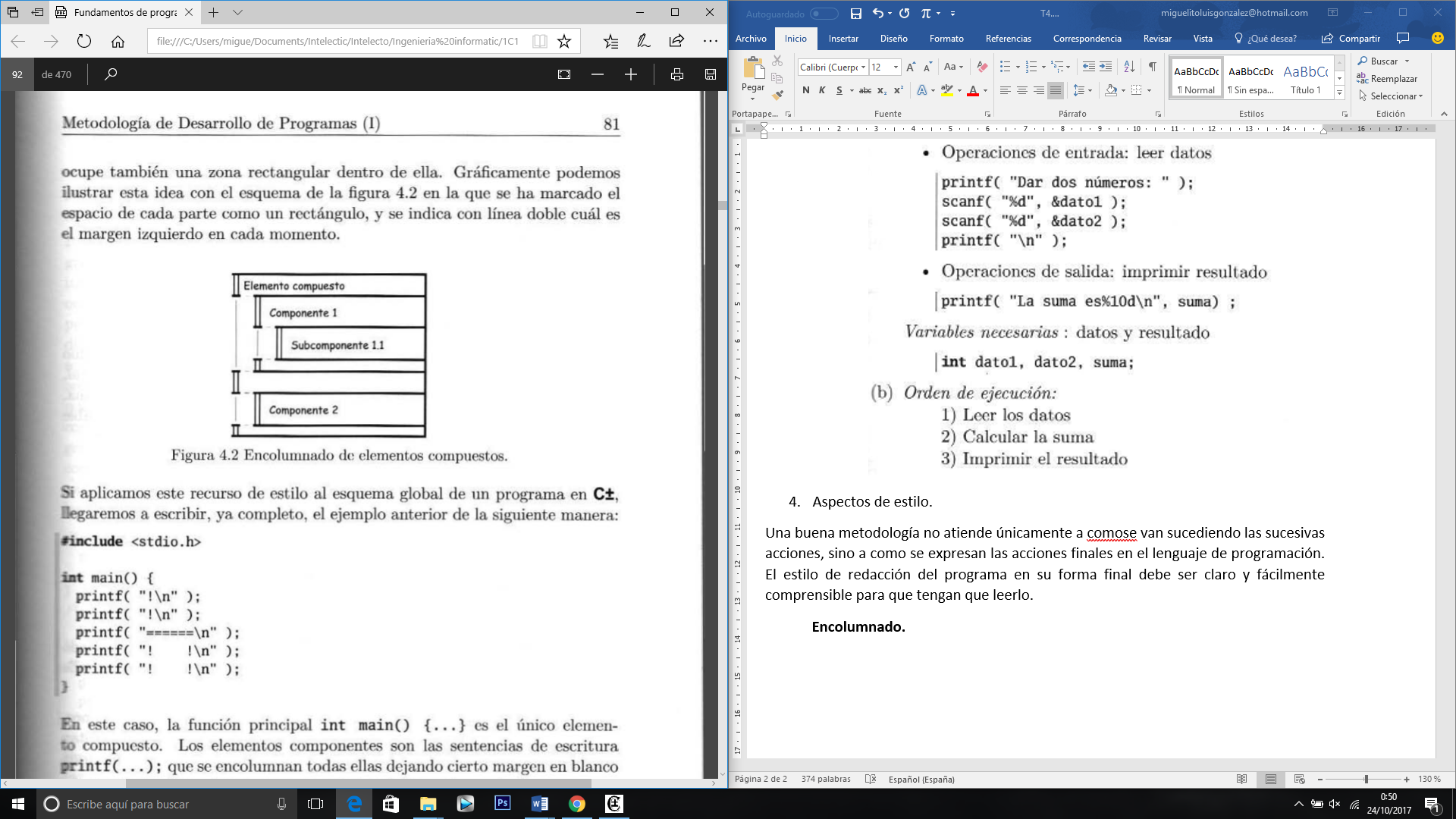
* Identificar las acciones que componen la secuencia, identificar las variables necesarias para dispones de la información necesaria al comienzo de cada acción y almacenar el resultado.
* Identificar el orden en el que deben ejecutarse las acciones compuestas. Por ejemplo:



1. Aspectos de estilo.

Una buena metodología no atiende únicamente a como se van sucediendo las sucesivas acciones, sino a como se expresan las acciones finales en el lenguaje de programación. El estilo de redacción del programa en su forma final debe ser claro y fácilmente comprensible para que tengan que leerlo.

**Encolumnado.**



## Comentarios

Otro recurso es el empleo de comentarios, ya se ha indicado como el lenguaje C+- permite intercalar comentario en el texto de un programa escribiéndolo entre la pareja de simbolos /\* comentario \*/

Los comentarios son libres, pero se recomienda usar estos tipos de comentarios:

* + Cabecera del programa
  + Cabecera de sección
  + Comentarios-orden
  + Comentarios al margen.

La **cabecera del programa** tiene como finalidad documentar el programa como un todo. Puede incluir datos de identificación, finalidad, descripción general..etc. Suele presentarse como una caja al comienzo del texto del programa, ocupando todo el ancho del listado.

***/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

***\*Programa: Imprimir Silla gráficamente.***

***\****

***\*Autor: Michael***

***\****

***\*Descripción: Este programa imprime de forma esquemática la silueta de una silla usando caracteres normales de la impresora.***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/***

Las **cabeceras de sección** sirven para documentar partes importantes de un programa relativamente largo. Al igual que la cabecera del programa, se presenta en forma de caja al comienzo de la sección correspondiente.

***/\*=====================================================================***

***PARTE EJECUTABLE DEL PROGRAMA***

***=====================================================================\*/***

Los **comentarios-orden** son un elemento metodológico fundamental, y sirven para documentar los refinamientos empleados en el desarrollo del programa.

***/\*——Imprimir el respaldo——\*/{***

***Printf(“!\n”);***

***Printf(“!\n”);***

***}***

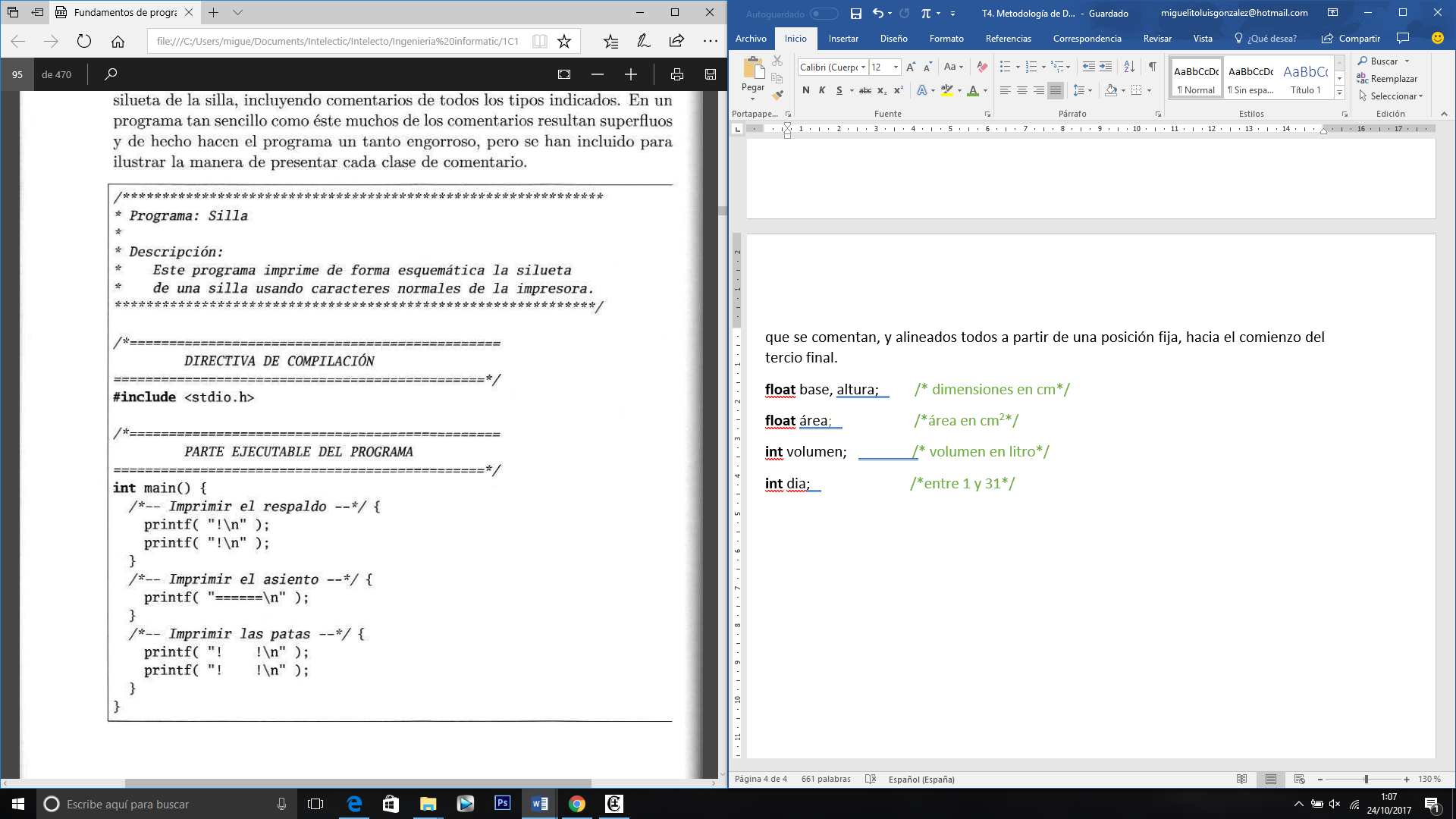
**Los comentarios al margen** sirven para aclarar el significado de ciertas sentencias del programa que pueden ser difíciles de interpretar al leerlas tal como aparecen escritas en el lenguaje de programación empleado. Una recomendación de estilo es situar estos comentarios hacia la parte derecha del listado, en las mismas líneas que las sentencias que se comentan, y alineados todos a partir de una posición fija, hacia el comienzo del tercio final.

**float** base, altura; /\* dimensiones en cm\*/

**float** área; /\*área en cm2\*/

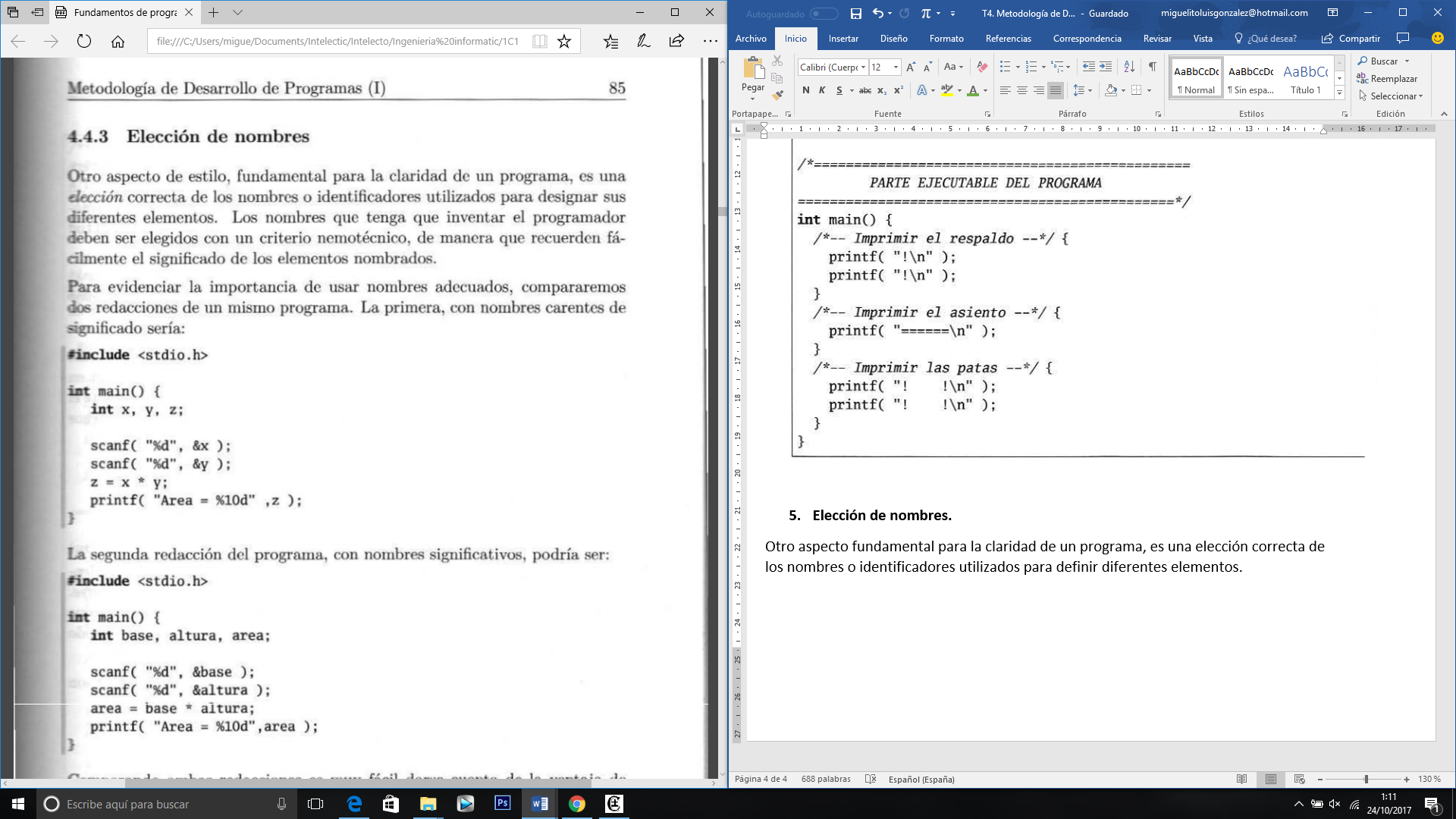
**int** volumen; /\* volumen en litro\*/

**int** dia; /\*entre 1 y 31\*/



1. **Elección de nombres.**

Otro aspecto fundamental para la claridad de un programa, es una elección correcta de los nombres o identificadores utilizados para definir diferentes elementos.



* Los valores (constantes, variables, etc.) deben ser designados mediante sustantivos.
* Las acciones (procedimientos, etc.) deben ser designadas con verbos.
* Los tipos deben ser designados mediante nombres genéricos.

1. **Uso de letras mayúsculas y minúsculas.**

• Los nombres de tipos, procedimientos y funciones empieílan por mayúscula:

**CalcularDias LeerFecha EscribirFecha**

**DiasEntre Hoy**

• Los nombres de variables y constantes empiezan por minúscula:

**fechaCumple fechaHoy dias**

• Los nombres que son palabras compuestas usan mayúsculas intercaladas al comienzo de cada siguiente palabra componente:

**TipoLongitud TipoFecha fechaHoy**

**CalcularDias LeerFecha EscribirFecha**

**DiasEntre fecha Cumple**

1. **Constantes con nombre.**

* Utilizar palabras que tengan significado nemotécnico, es decir, que viendo la palabra sepas para que se usa esa constante. Denominamos parámetros del programa a aquellas variables que consideramos fijas, pero que en el futuro por necesidades del programa vayan a ser modificadas.