|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Marco Antonio Martínez Quintana |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 3 |
| *No de Práctica(s):* | 7 |
| *Integrante(s):* | Flores Saavedra José Gerardo |
| *No. de Equipo de cómputo empleado:* | No Aplica |
| *No. de Lista o Brigada:* | 13 |
| *Semestre:* | 2021-1 |
| *Fecha de entrega:* | 16/11/2020 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Fundamentos de Lenguaje C**

**Objetivo:**

Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

**Introducción:**

La codificación se puede realizar en cualquier lenguaje de programación estructurada, como lo son Pascal, Python, Fortran o PHP. En este curso se aprenderá el uso del lenguaje de programación C.

**Lenguaje de programación C**

El proceso de desarrollo del lenguaje C se origina con la creación de un lenguaje llamado BCPL, que fue desarrollado por Martin Richards. BCPL tuvo influencia en un lenguaje llamado B, el cual se usó en 1970 y fue inventado por Ken Thompson, esto permitió el desarrollo de C en 1971, el cual lo inventó e implementó Dennis Ritchie.

C es un lenguaje de programación de propósito general que ofrece como ventajas economía de expresión, control de flujo y estructuras de datos y un conjunto de operadores. C es un lenguaje de propósito general basado en el paradigma estructurado. El teorema del programa estructurado, demostrado por Böhm-Jacopini, dicta que todo programa puede desarrollarse utilizando únicamente 3 instrucciones de control:

* Seciencia
* Selección
* Iteración

Por otro lado, C es un lenguaje compilado, es decir, existe un programa que, a partir de un código en lenguaje C, genera un código objeto. Para crear un programa en C se siguen tres etapas principales: edición, compilación y ejecución.

* Edición: Se escribe el código fuente en lenguaje C desde algún editor de textos.
* Compilación: A partir del código fuente (lenguaje C) se genera el archivo en lenguaje máquina (se crea el programa objeto o ejecutable).
* Ejecución: El archivo en lenguaje máquina se puede ejecutar en la arquitectura correspondiente.

Al momento de ejecutar un programa objeto (código binario), se ejecutarán únicamente las instrucciones que estén definidas dentro de la función principal. La función principal puede contener sentencias, estructuras de control y comentarios. Dentro de las sentencias se encuentran la declaración y/o asignación de variables, la realización de operaciones básicas, y las llamadas a funciones.

Al iniciar el programa se deben agregar todas las bibliotecas que se van a utilizar en el mismo, es decir, funciones externas necesarias para ejecutar el programa. En lenguaje C la biblioteca estándar de entrada y salida está definida en 'stdio.h' (standard in out) y provee, entre otras, funciones para lectura y escritura de datos que se verán a continuación.

**Comentarios**

Es una buena práctica en cualquier lenguaje de programación realizar comentarios para documentar el programa. En C existen dos tipos de comentarios: el comentario por línea y el comentario por bloque.

El comentario por línea inicia cuando se insertan los símbolos '//' y termina con el salto de línea (hasta donde termine el renglón)

El comentario por bloque inicia cuando se insertan los símbolos '/\*' y termina cuando se encuentran los símbolos '\*/'. Cabe resaltar que el comentario por bloque puede abarcar varios renglones.

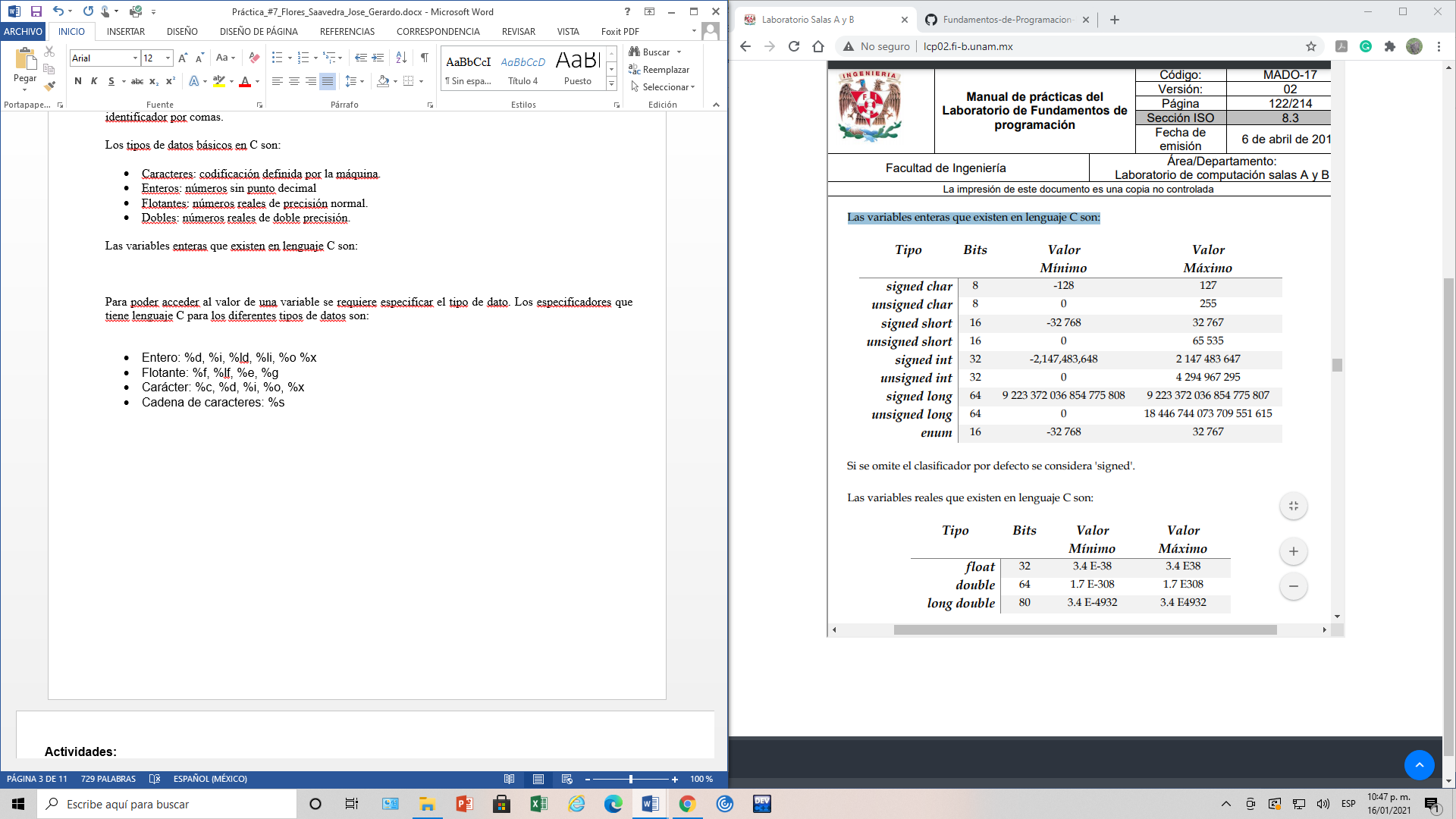
**Declaración de variables**

Por lo tanto, una variable puede tener modificadores, debe declarar el tipo de dato que puede contener la variable, debe declarar el identificador (nombre o etiqueta) con el que se va a manejar el valor y se puede asignar un valor inicial a la variable (opcional). También es posible declarar varios identificadores de un mismo tipo de dato e inicializarlos en el mismo renglón, lo único que se tiene que hacer es separar cada identificador por comas.

Los tipos de datos básicos en C son:

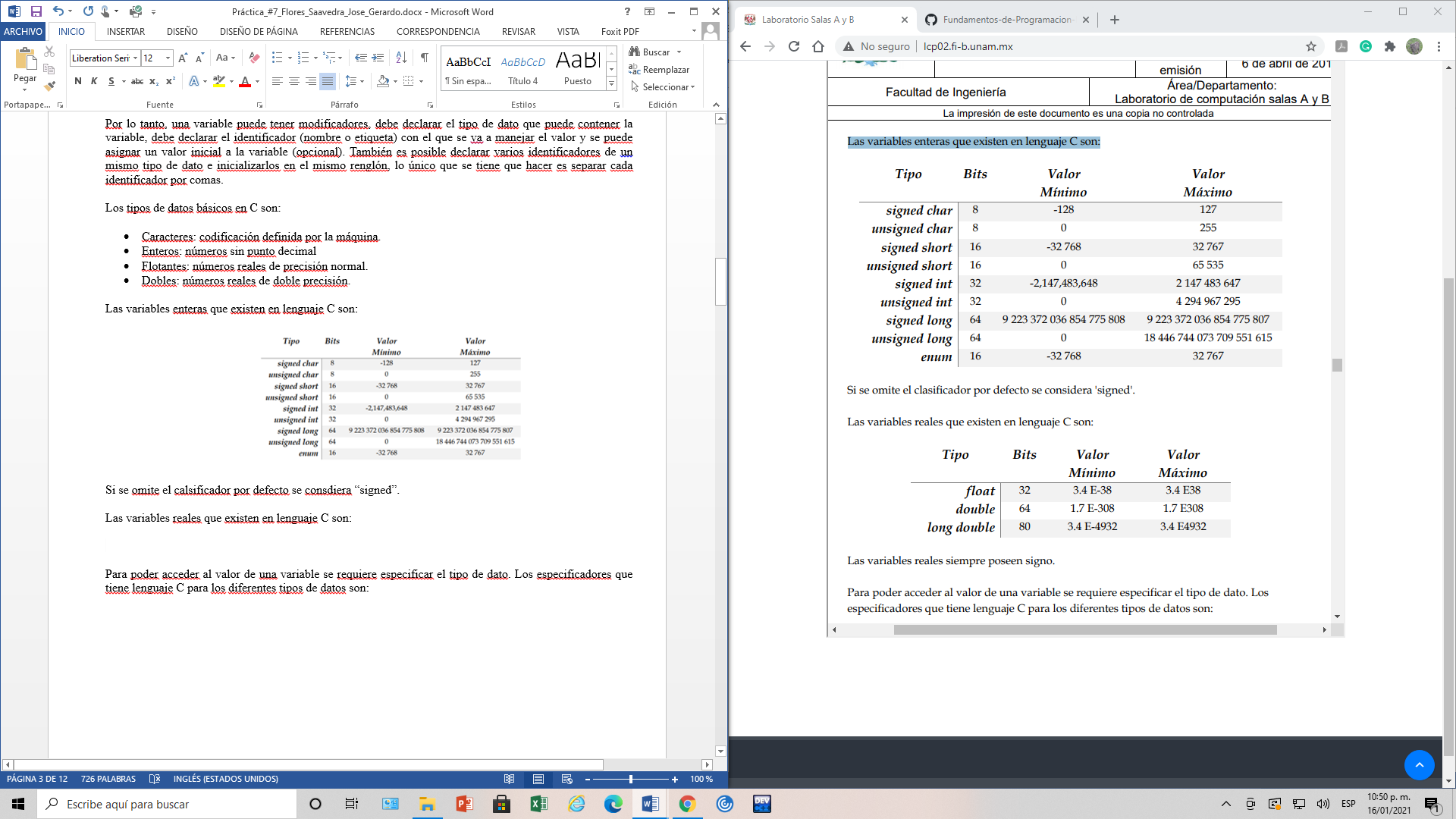
* Caracteres: codificación definida por la máquina.
* Enteros: números sin punto decimal
* Flotantes: números reales de precisión normal.
* Dobles: números reales de doble precisión.

Las variables enteras que existen en lenguaje C son:



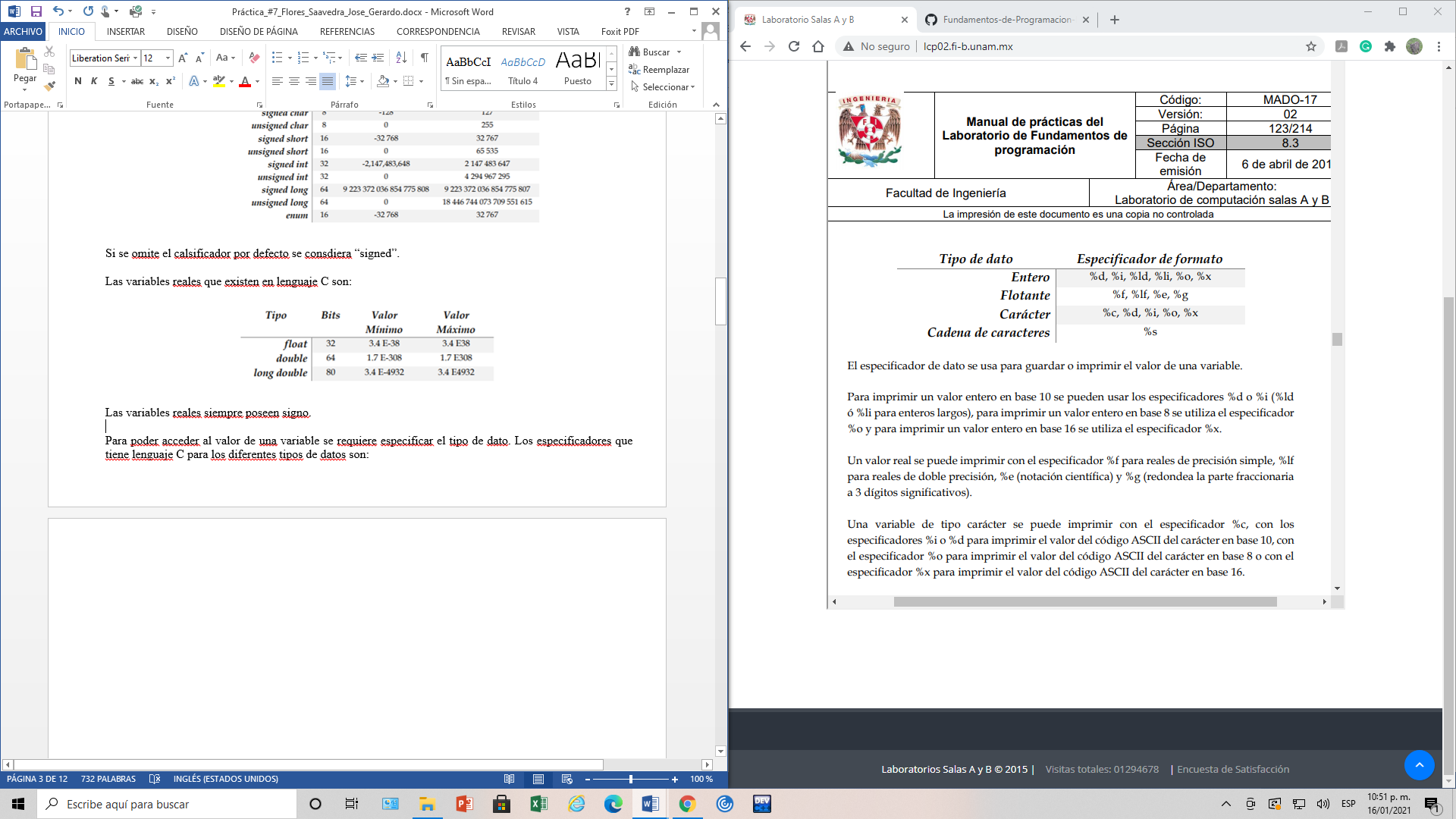
Si se omite el clasificador por defecto se considera “signed”.

Las variables reales que existen en lenguaje C son:



Las variables reales siempre poseen signo.

Para poder acceder al valor de una variable se requiere especificar el tipo de dato. Los especificadores que tiene lenguaje C para los diferentes tipos de datos son:



El especificador de dato se usa para guardar o imprimir el valor de una variable.

**Identificador**

Un identificador es el nombre con el que se va a almacenar en memoria un tipo de dato. Los identificadores siguen las siguientes reglas:

* Debe iniciar con una letra [a-z].
* Puede contener letras [A-Z, a-z], números [0-9] y el carácter guión bajo (\_).

La biblioteca 'stdio.h' contiene diversas funciones tanto para imprimir en la salida estándar (monitor) como para leer de la entrada estándar (teclado).

printf es una función para imprimir con formato, es decir, se tiene que especificar entre comillas el tipo de dato que se desea imprimir, también se puede combinar la impresión de un texto predeterminado.

scanf es una función que sirve para leer datos de la entrada estándar (teclado), para ello únicamente se especifica el tipo de dato que se desea leer entre comillas y en qué variable se quiere almacenar. Al nombre de la variable le antecede un ampersand (&), esto indica que el dato recibido se guardará en la localidad de memoria asignada a esa variable.

Para imprimir con formato también se utilizan algunas secuencias de caracteres de escape, C maneja los siguientes:

* \a carácter de alarma
* \b retroceso
* \f avance de hoja
* \n salto de línea
* \r regreso de carro
* \t tabulador horizontal
* \v tabulador vertical
* ‘\0’ carácter nulo

**Modificadores de alcance**

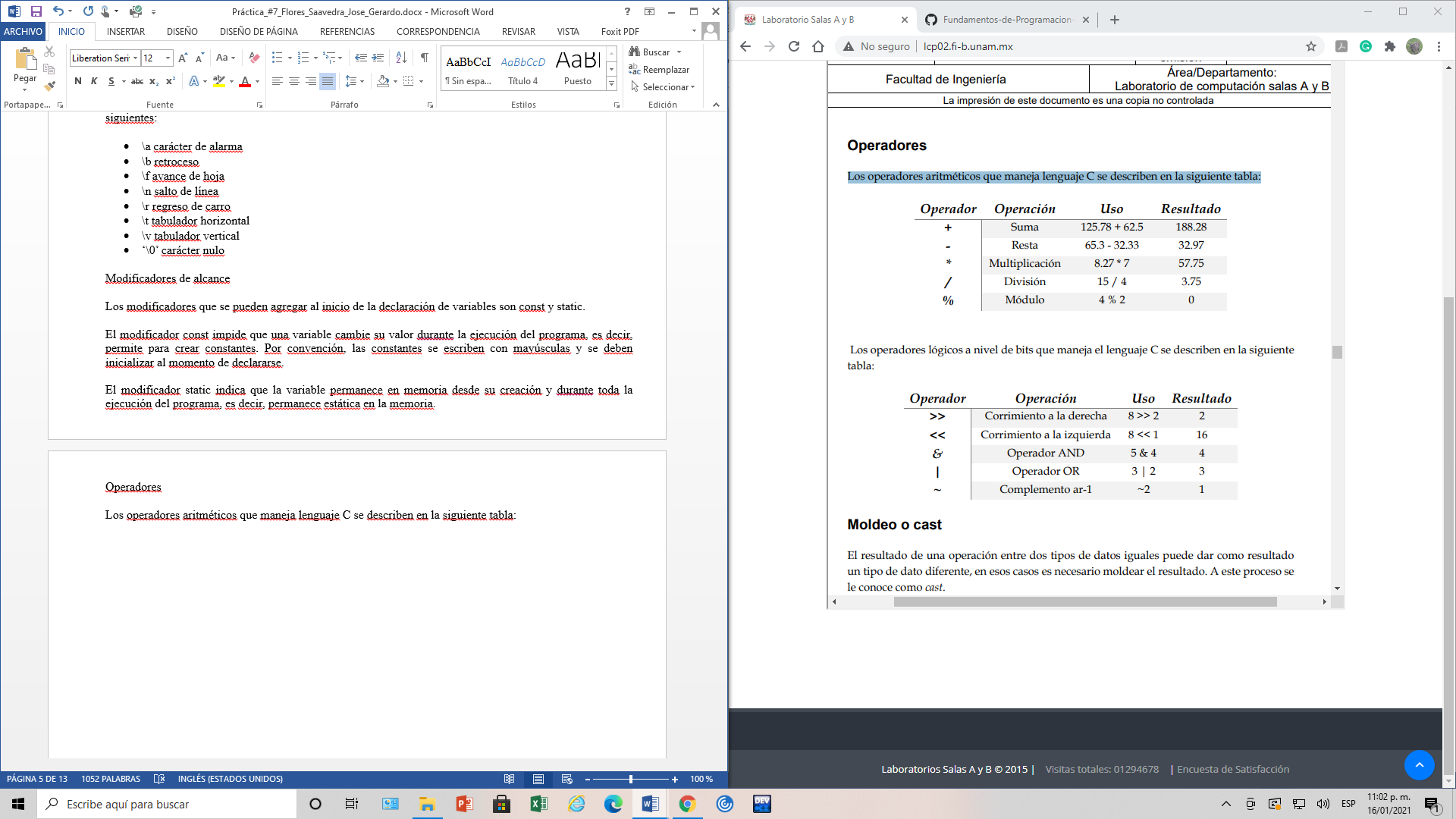
Los modificadores que se pueden agregar al inicio de la declaración de variables son const y static.

El modificador const impide que una variable cambie su valor durante la ejecución del programa, es decir, permite para crear constantes. Por convención, las constantes se escriben con mayúsculas y se deben inicializar al momento de declararse.

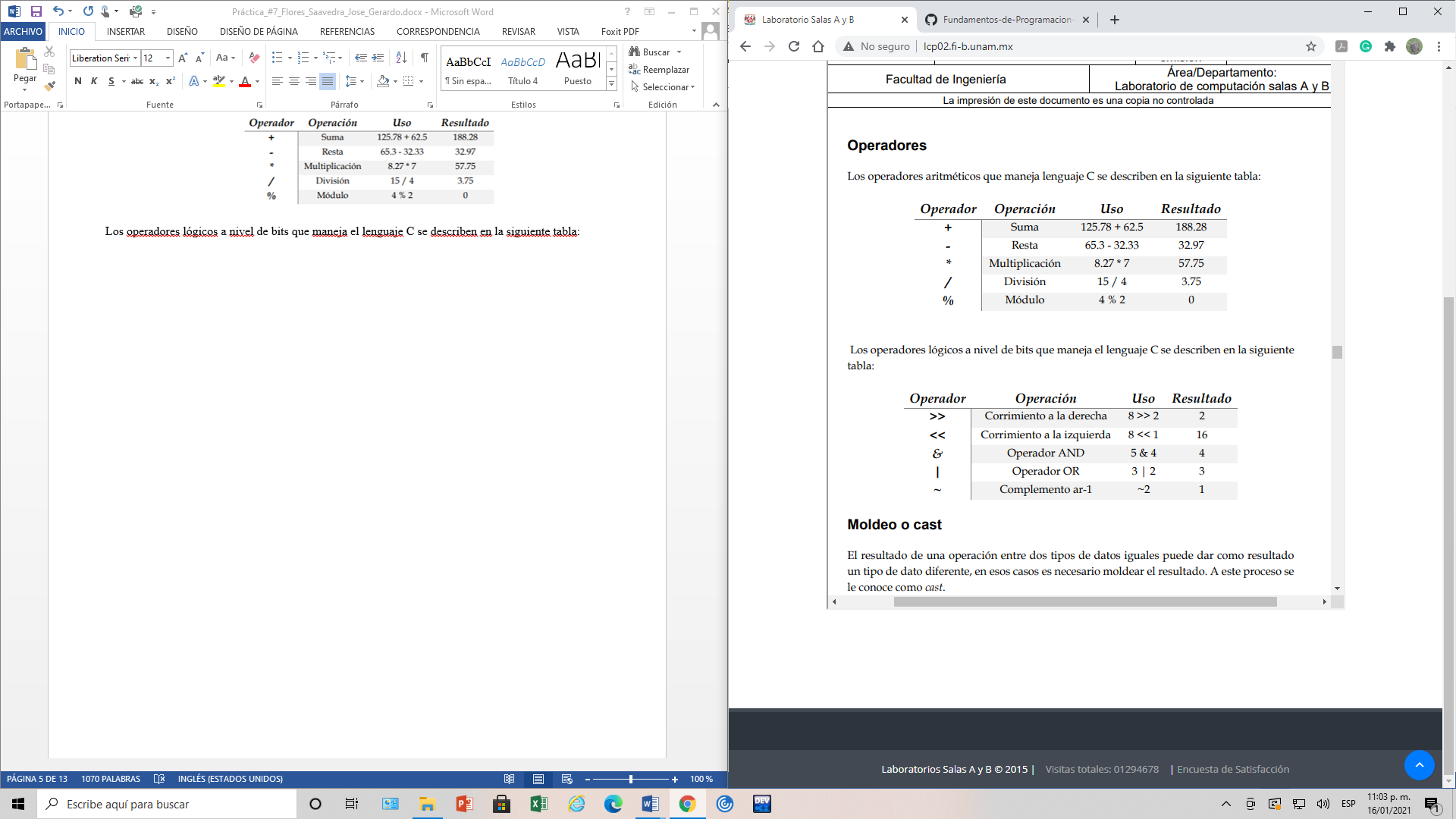
El modificador static indica que la variable permanece en memoria desde su creación y durante toda la ejecución del programa, es decir, permanece estática en la memoria.

**Operadores**

Los operadores aritméticos que maneja lenguaje C se describen en la siguiente tabla:



Los operadores lógicos a nivel de bits que maneja el lenguaje C se describen en la siguiente tabla:



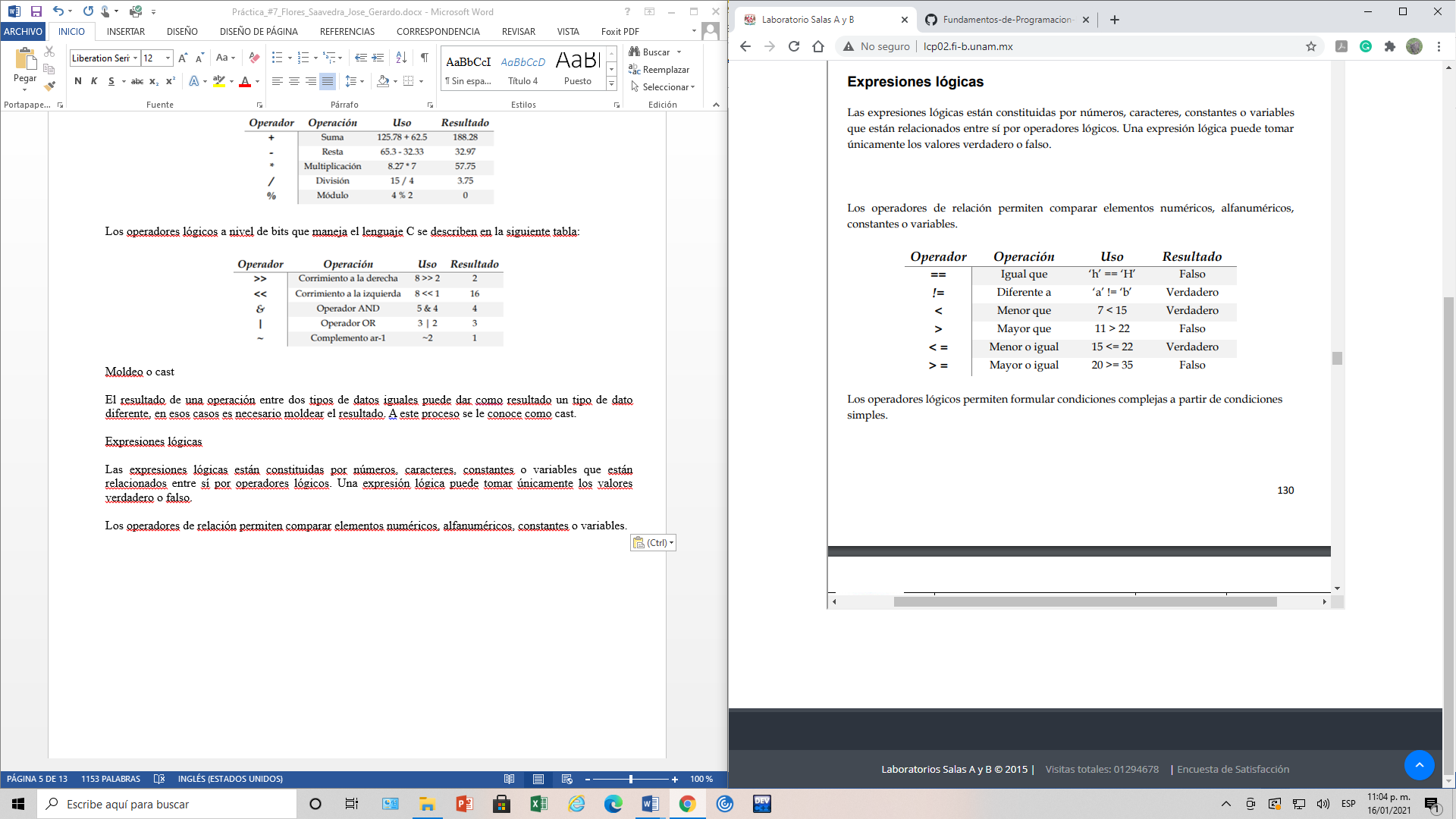
**Moldeo o cast**

El resultado de una operación entre dos tipos de datos iguales puede dar como resultado un tipo de dato diferente, en esos casos es necesario moldear el resultado. A este proceso se le conoce como cast.

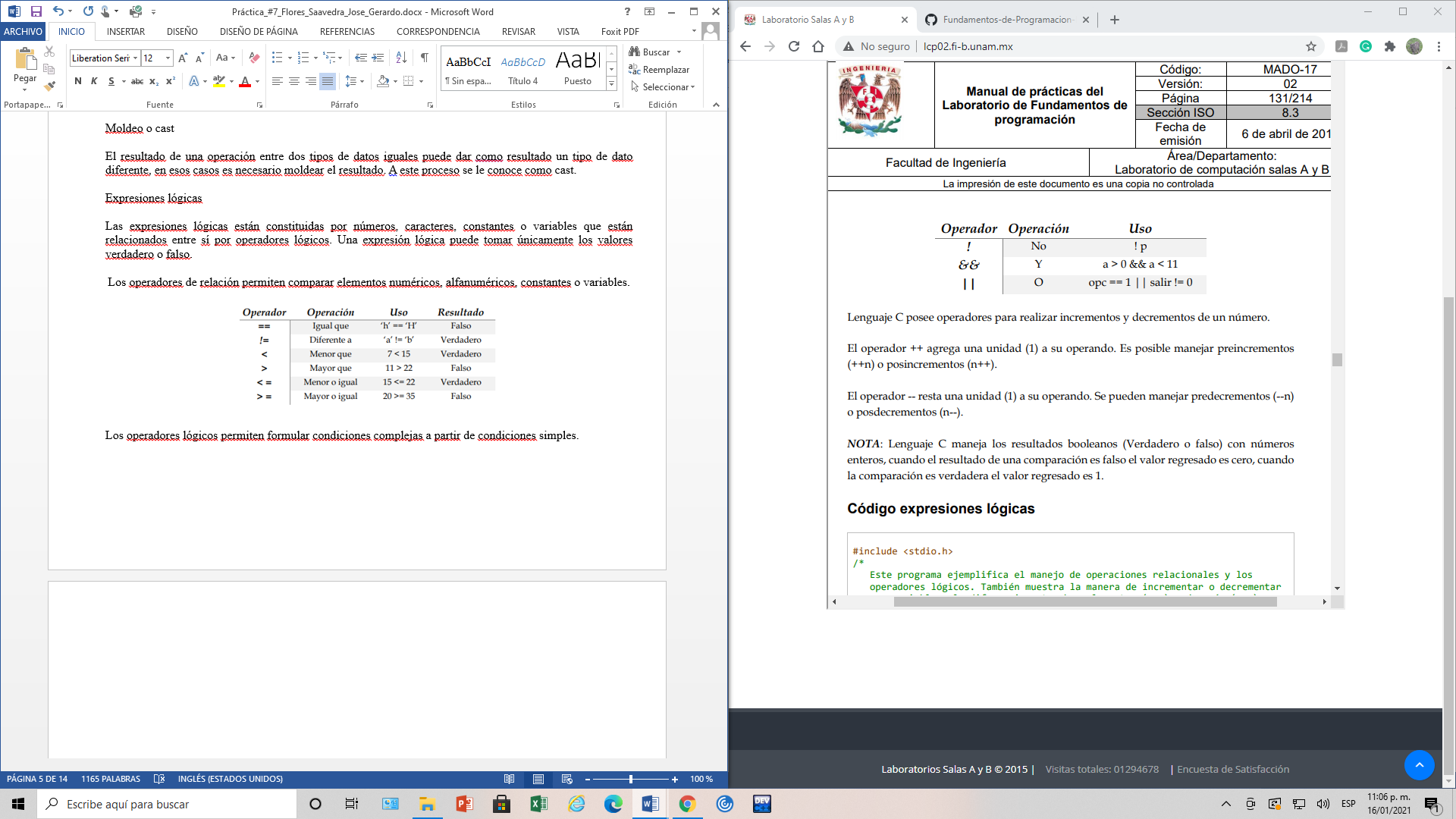
**Expresiones lógicas**

Las expresiones lógicas están constituidas por números, caracteres, constantes o variables que están relacionados entre sí por operadores lógicos. Una expresión lógica puede tomar únicamente los valores verdadero o falso.

Los operadores de relación permiten comparar elementos numéricos, alfanuméricos, constantes o variables.



Los operadores lógicos permiten formular condiciones complejas a partir de condiciones simples.



Lenguaje C posee operadores para realizar incrementos y decrementos de un número. El operador ++ agrega una unidad (1) a su operando. Es posible manejar preincrementos (++n) o posincrementos (n++). El operador -- resta una unidad (1) a su operando. Se pueden manejar predecrementos (--n) o posdecrementos (n--).

Lenguaje C maneja los resultados booleanos (Verdadero o falso) con números enteros, cuando el resultado de una comparación es falso el valor regresado es cero, cuando la comparación es verdadera el valor regresado es 1.

**Actividades:**

