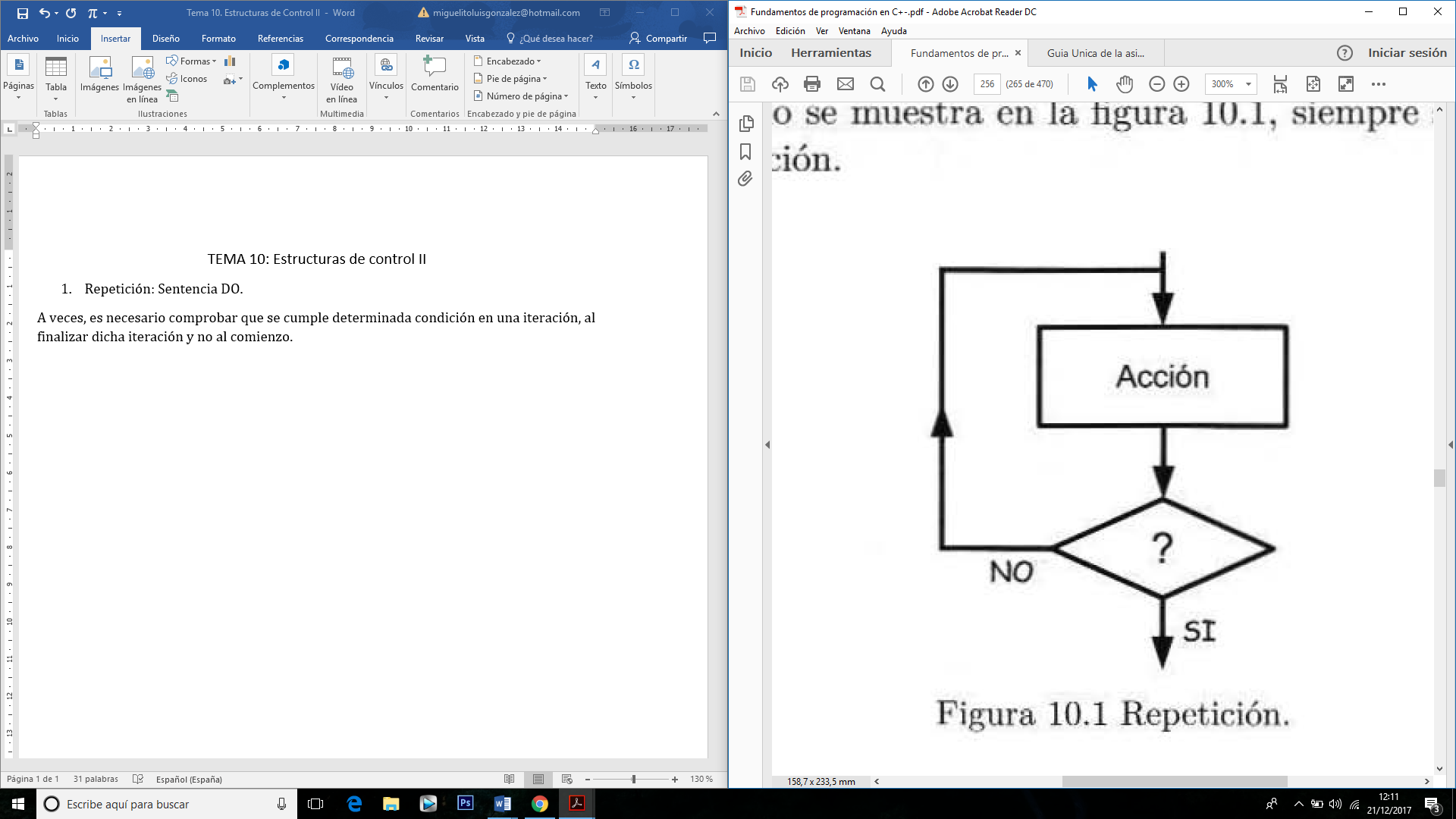
TEMA 10: Estructuras de control II

1. **Repetición: Sentencia DO.**

A veces, es necesario comprobar que se cumple determinada condición en una iteración, al finalizar dicha iteración y no al comienzo.



El formato de la estructura de repetición **DO** es la siguiente.

do {

Acción;

} while( Condición);🡪 Si la condición es verdadera, vuelve a ejecutar la acción, en caso contrario continua.

Un ejemplo de su uso (preguntar si repetir determinadas acciones):

do {

...

Operaciones;

...

Printf(“¿otra operación (S/N)?”);

Scanf(“%c”,&tecla);

} while (tecla == ‘S’);

}

Otro ejemplo de su uso (repetición has obtener una respuesta esperada)

Do {

printf(“¿mes Actual?”);

scanf(“%d”, &mes);

}while((mes<1)||(mes > 12));

1. **Sentencia Continue.**

La sentencia **continue** dentro de cualquier clase de bucle (while, for o do) finaliza la iteración en curso e inicia la siguiente iteración.

For (int i = 0; i < N; i++){

...

if (Divisor == 0){

**continue; 🡪 Esto hace que se reinicie el bucle con la “i” incrementada.**

}

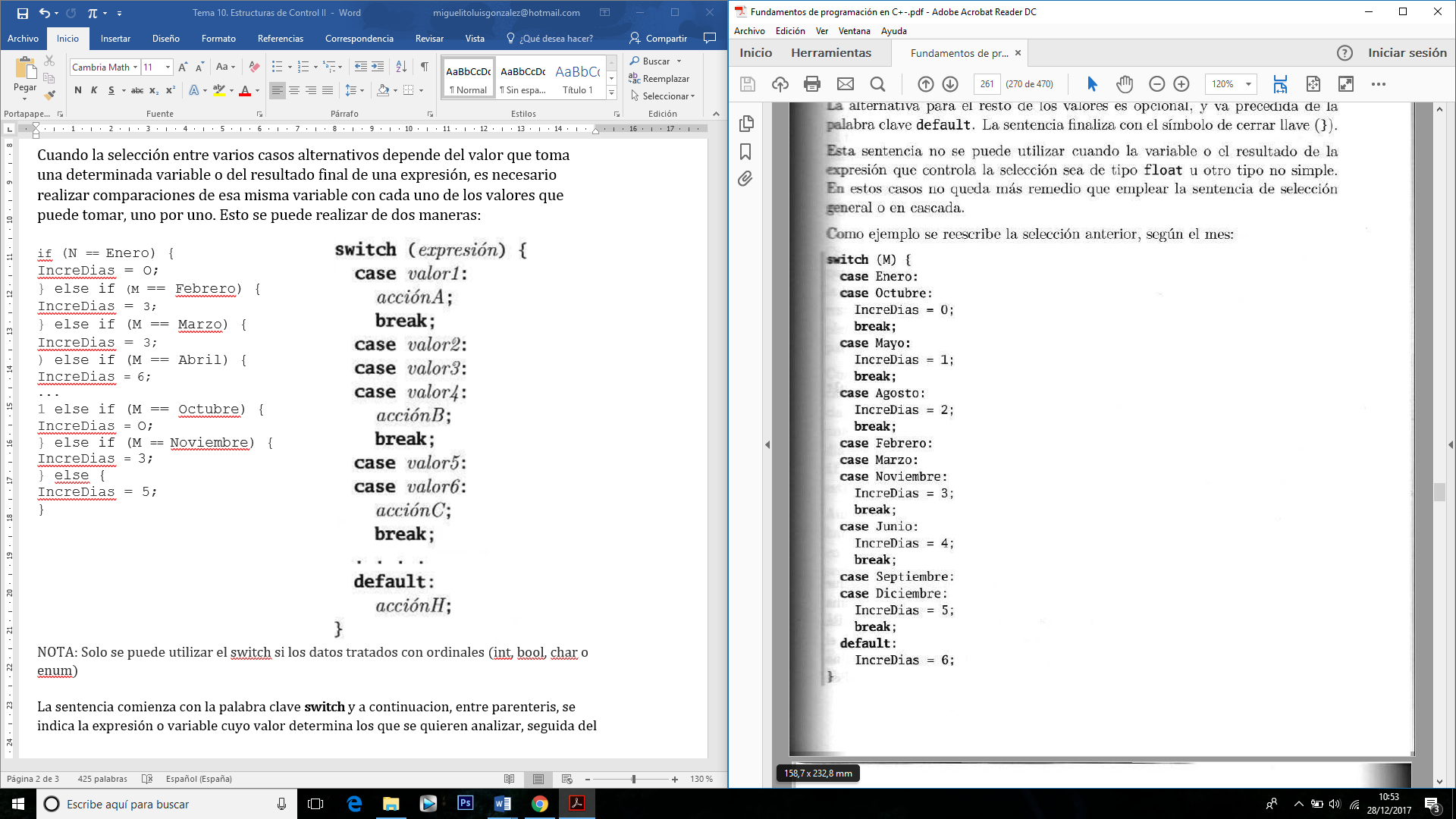
Dividendo / Divisor == Resultado;

}

1. **Estructuras complementarias de selección.**

Existen lenguajes de programación en los que la única sentencia para la selección entre varias opciones es IF. Hay otras sentencias de selección por caso, denomina SWITCH.

Cuando la selección entre varios casos alternativos depende del valor que toma una determinada variable o del resultado final de una expresión, es necesario realizar comparaciones de esa misma variable con cada uno de los valores que puede tomar, uno por uno. Esto se puede realizar de dos maneras:



if (N == Enero) {

IncreDias = O;

} else if (M == Febrero) {

IncreDias = 3;

} else if (M == Marzo) {

IncreDias = 3;

) else if (M == Abril) {

IncreDias = 6;

...

1 else if (M == Octubre) {

IncreDias = O;

} else if (M == Noviembre) {

IncreDias = 3;

} else {

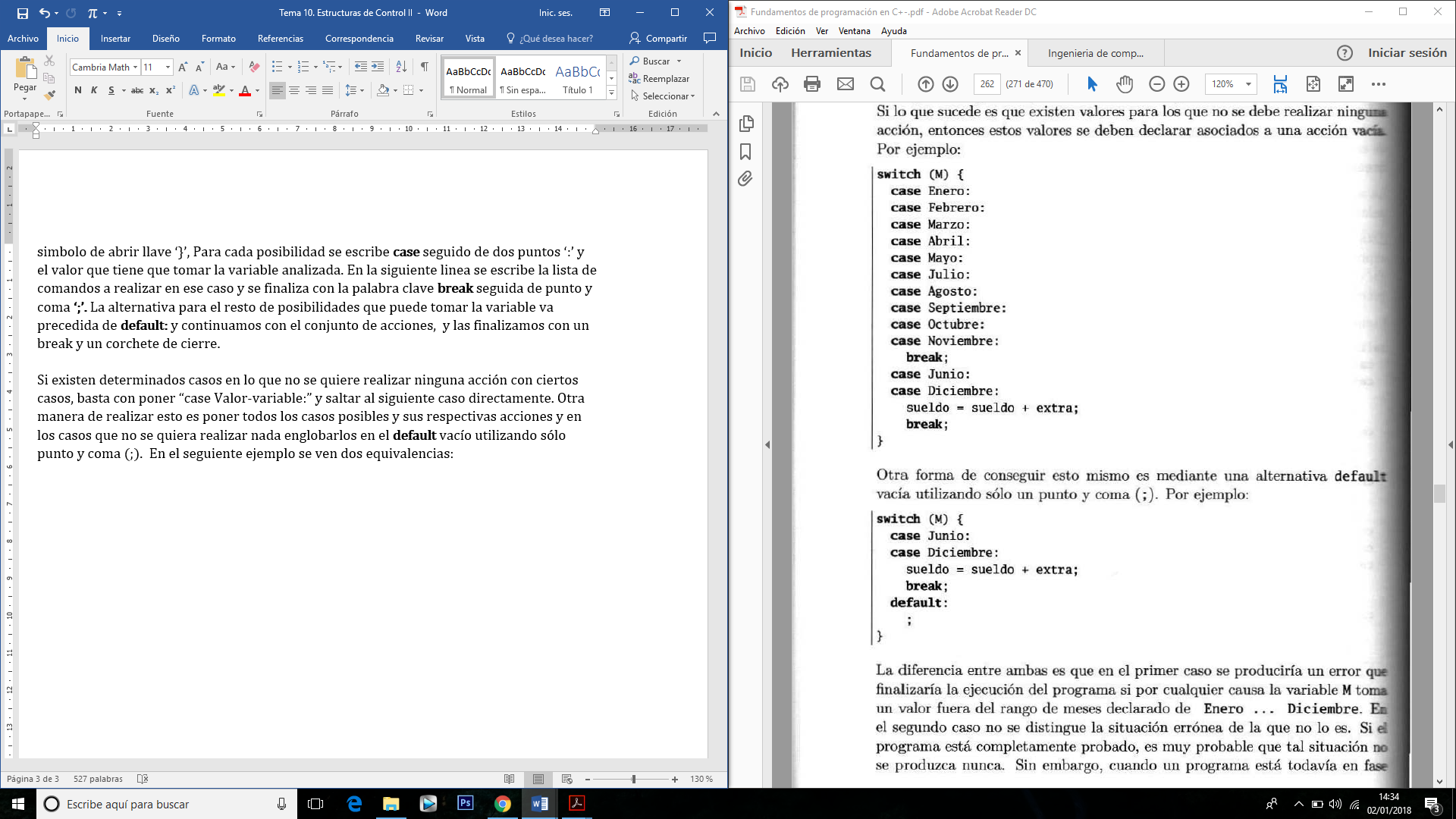
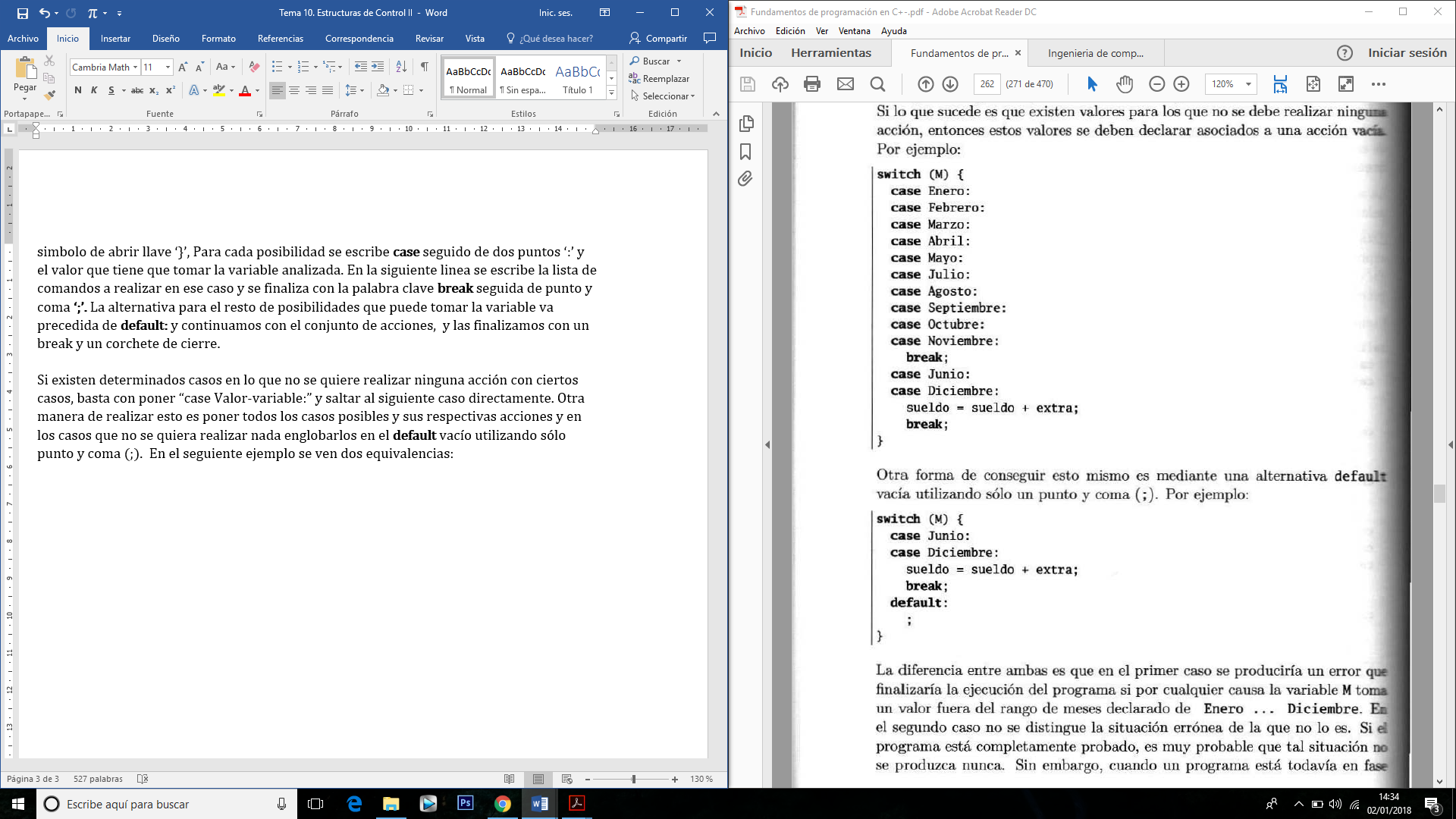
IncreDias = 5;

}

NOTA: Solo se puede utilizar el switch si los datos tratados son ordinales (int, bool, char o enum) no se puede utilizar con FLOAT

La sentencia comienza con la palabra clave **switch** y a continuacion, entre parenteris, se indica la expresión o variable cuyo valor determina los que se quieren analizar, seguida del simbolo de abrir llave ‘}’, Para cada posibilidad se escribe **case** seguido de dos puntos ‘:’ y el valor que tiene que tomar la variable analizada. En la siguiente linea se escribe la lista de comandos a realizar en ese caso y se finaliza con la palabra clave **break** seguida de punto y coma **‘;’.** La alternativa para el resto de posibilidades que puede tomar la variable va precedida de **default:** y continuamos con el conjunto de acciones, y las finalizamos con un break y un corchete de cierre.

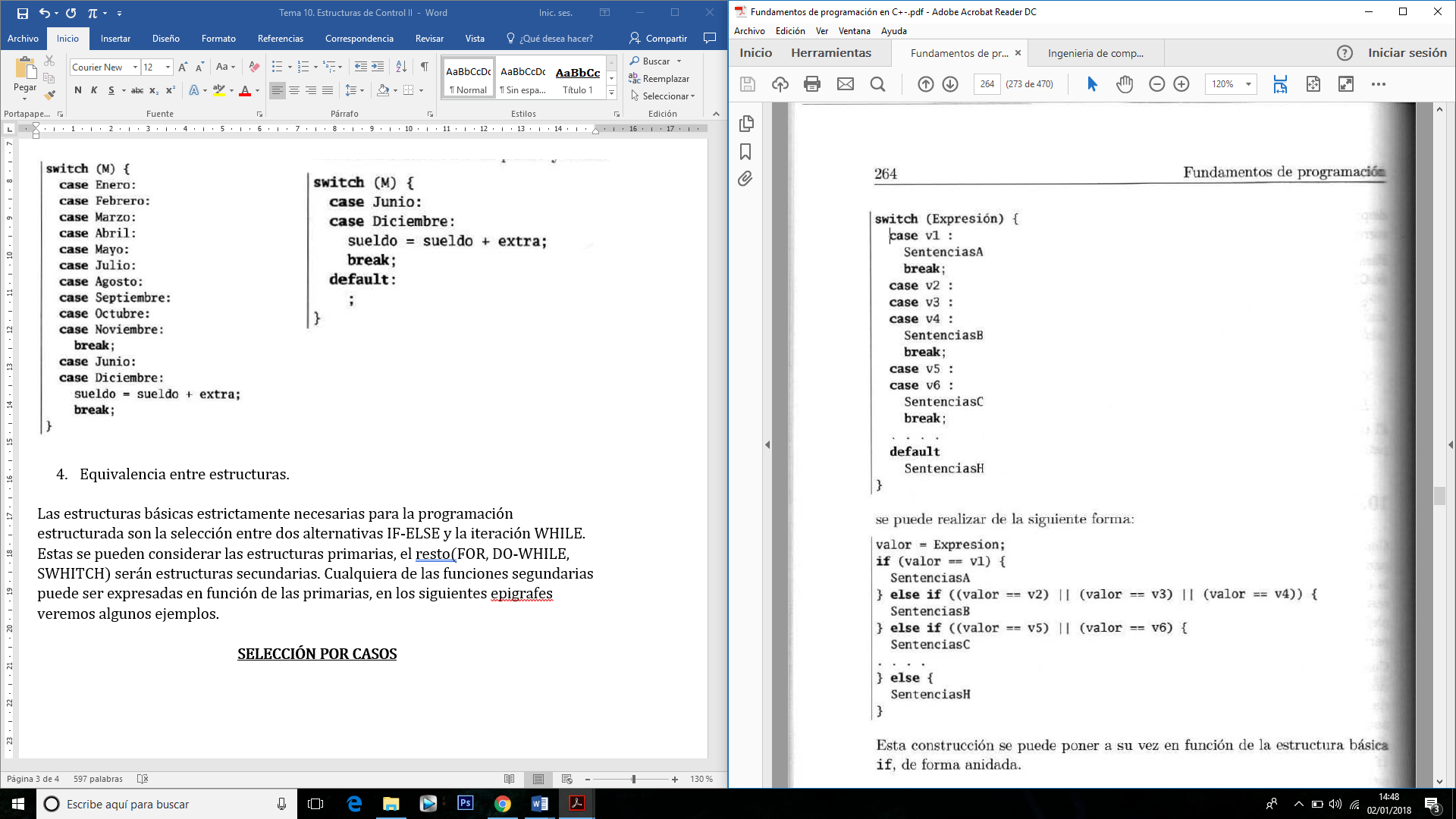
Si existen determinados casos en lo que no se quiere realizar ninguna acción con ciertos casos, basta con poner “case Valor-variable:” y saltar al siguiente caso directamente. Otra manera de realizar esto es poner todos los casos posibles y sus respectivas acciones y en los casos que no se quiera realizar nada englobarlos en el **default** vacío utilizando sólo punto y coma (;). En el seguiente ejemplo se aprecian dos codigos que realizan lo mismo pero implementado de diferente forma:



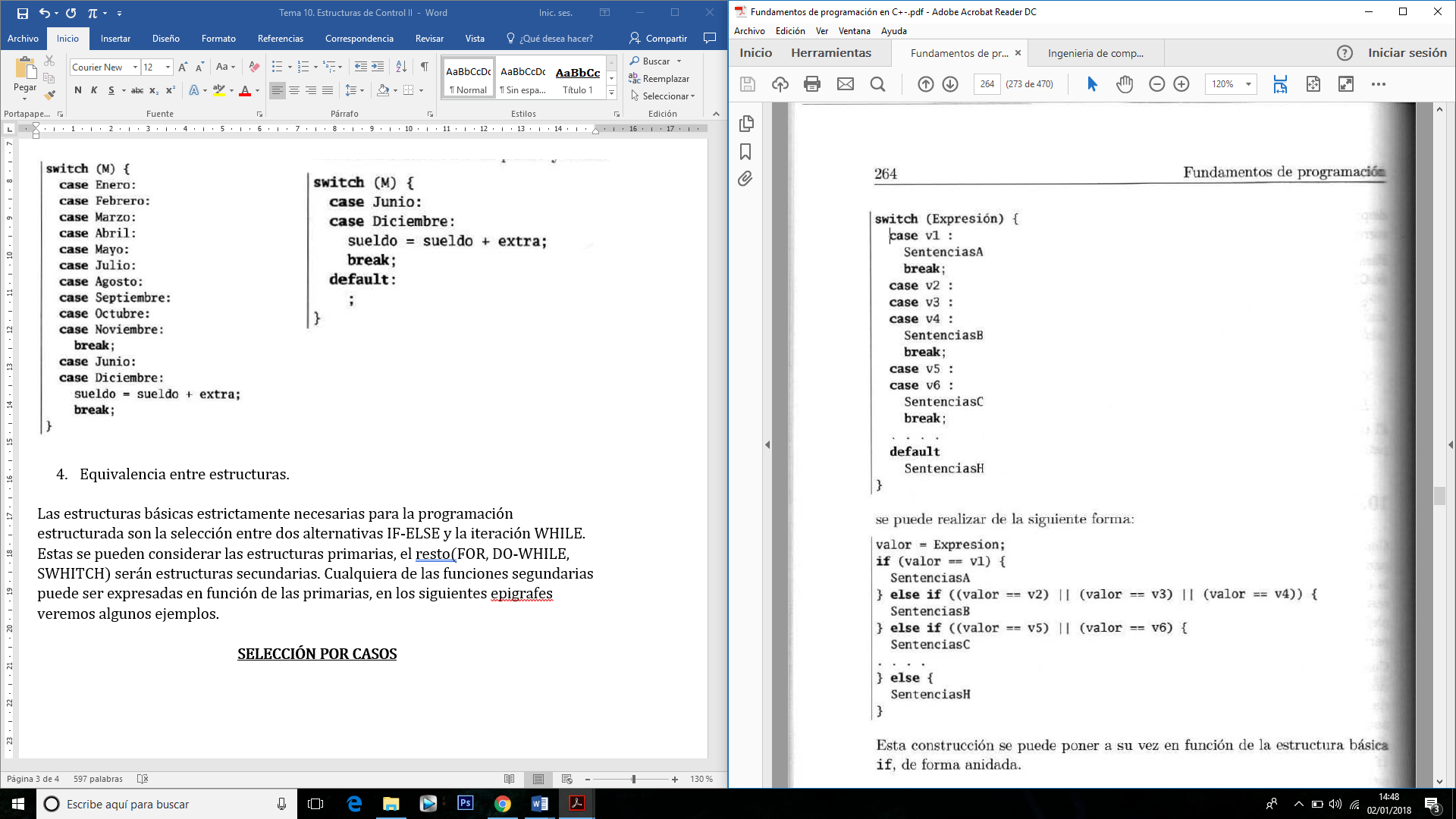
1. Equivalencia entre estructuras.

Las estructuras básicas estrictamente necesarias para la programación estructurada son la selección entre dos alternativas IF-ELSE y la iteración WHILE. Estas se pueden considerar las estructuras primarias, el resto(FOR, DO-WHILE, SWHITCH) serán estructuras secundarias. Cualquiera de las funciones segundarias puede ser expresadas en función de las primarias, en los siguientes epigrafes veremos algunos ejemplos.

# SELECCIÓN POR CASOS



Este codigo es equivalente a poner:



## BUCLE CONTADOR

for(**int** índice = inicial; índice <= Final; índice++){

SECUENCIA DE ACCIONES;

}

Este código es equivalente a:

**int** índice = inicial;

**while**(índice <= Final){

SECUENCIA DE ACCIONES;

Índice++;

}

## REPETICIÓN

La sentencia do se puede transformar en while forzando la ejecución incondicional de la primera iteración. La estructura:

### Do {

**Sentencias;**

**}while(Condicion);**

es equivalente a:

Sentencias

While(condición){

sentencias

}

O bien:

Seguir = true;

**while**(Seguir){

Sentencias;

Seguir = condición;

}