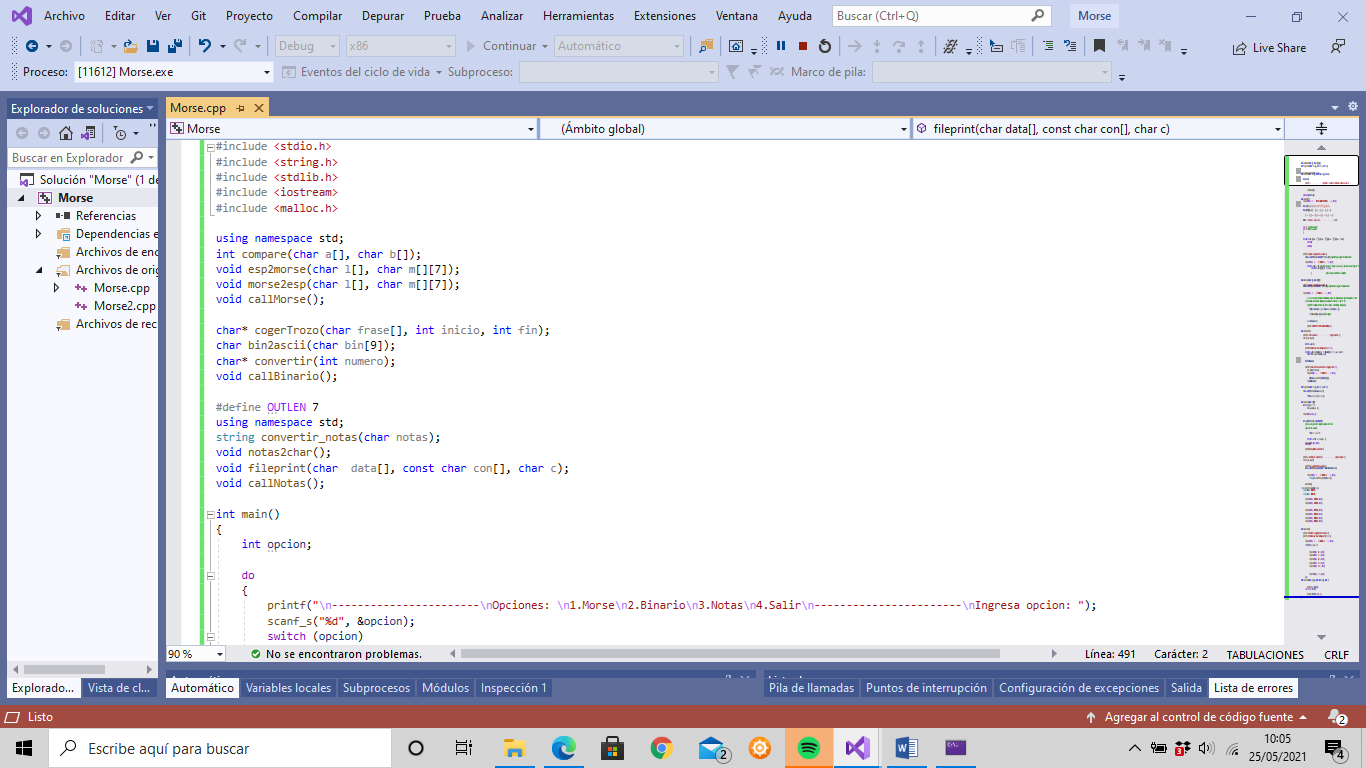
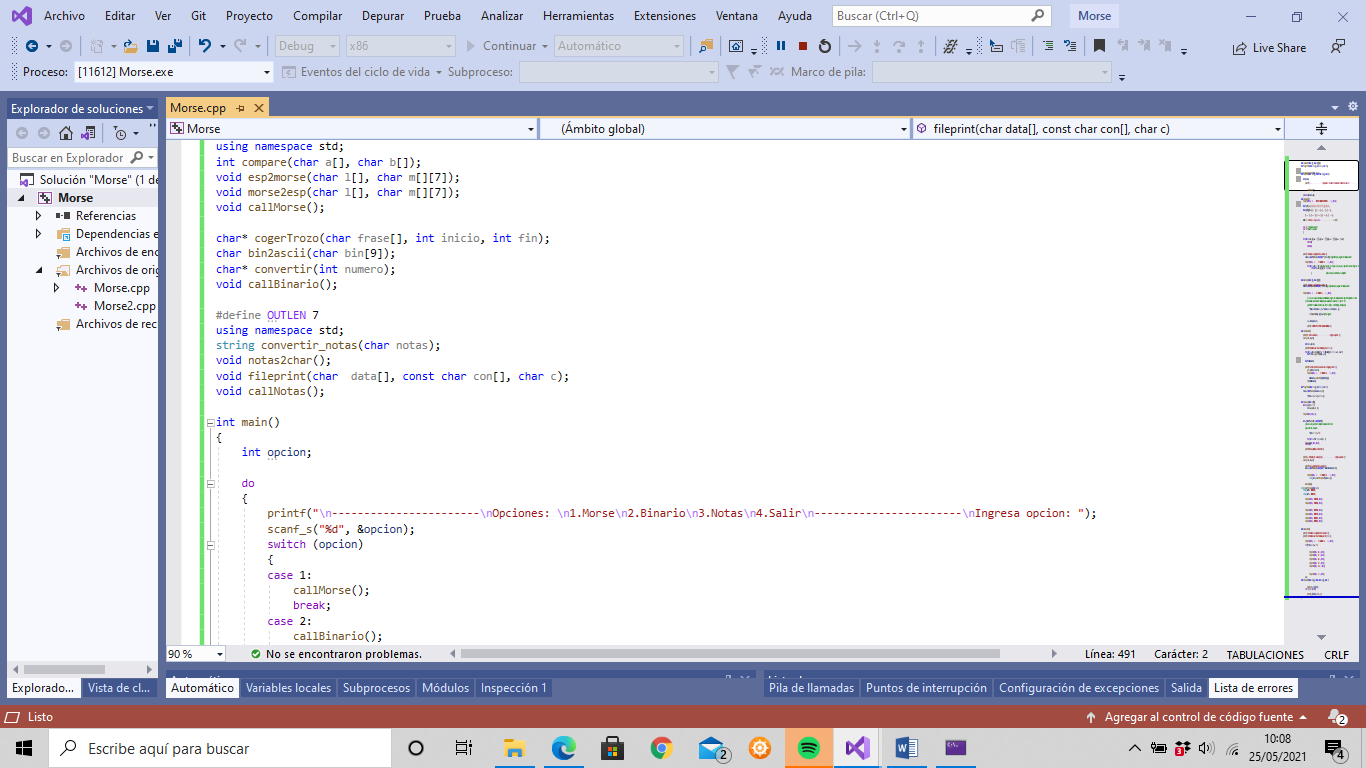
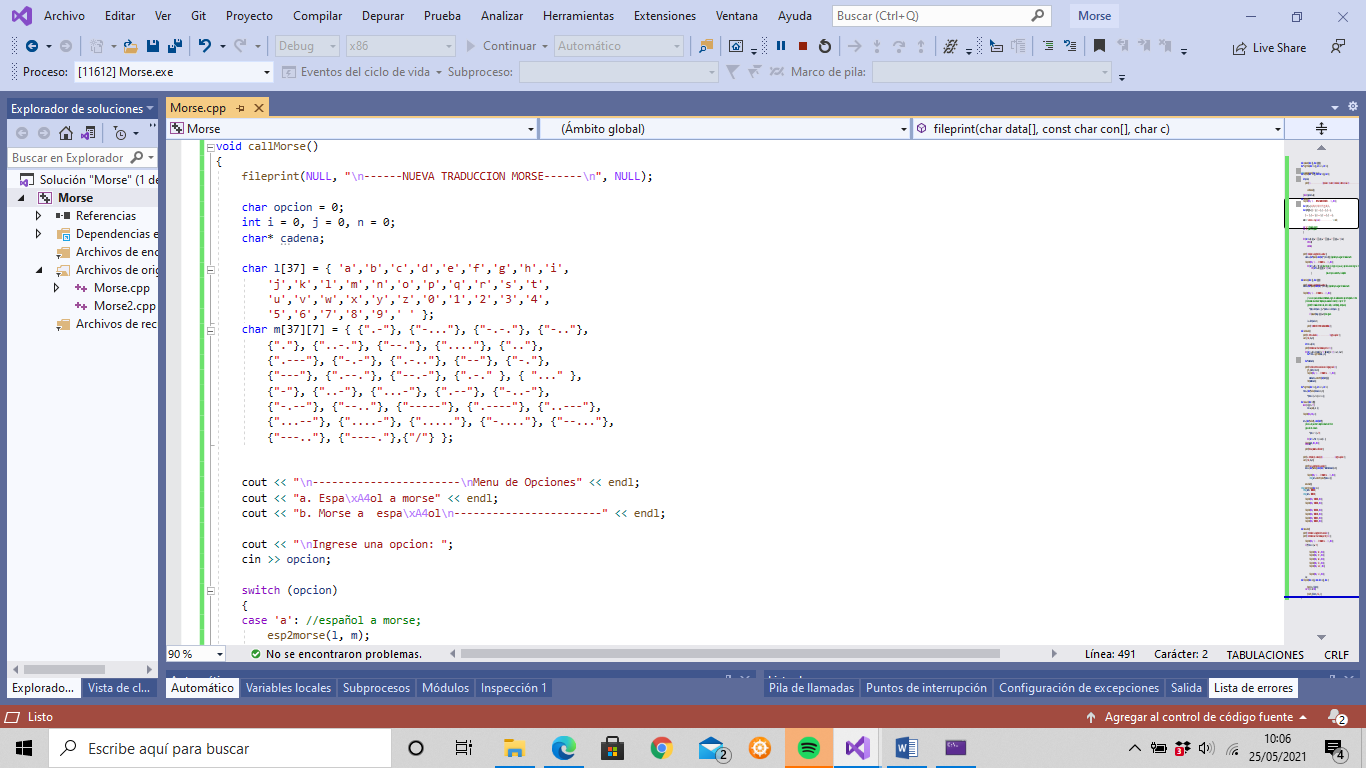
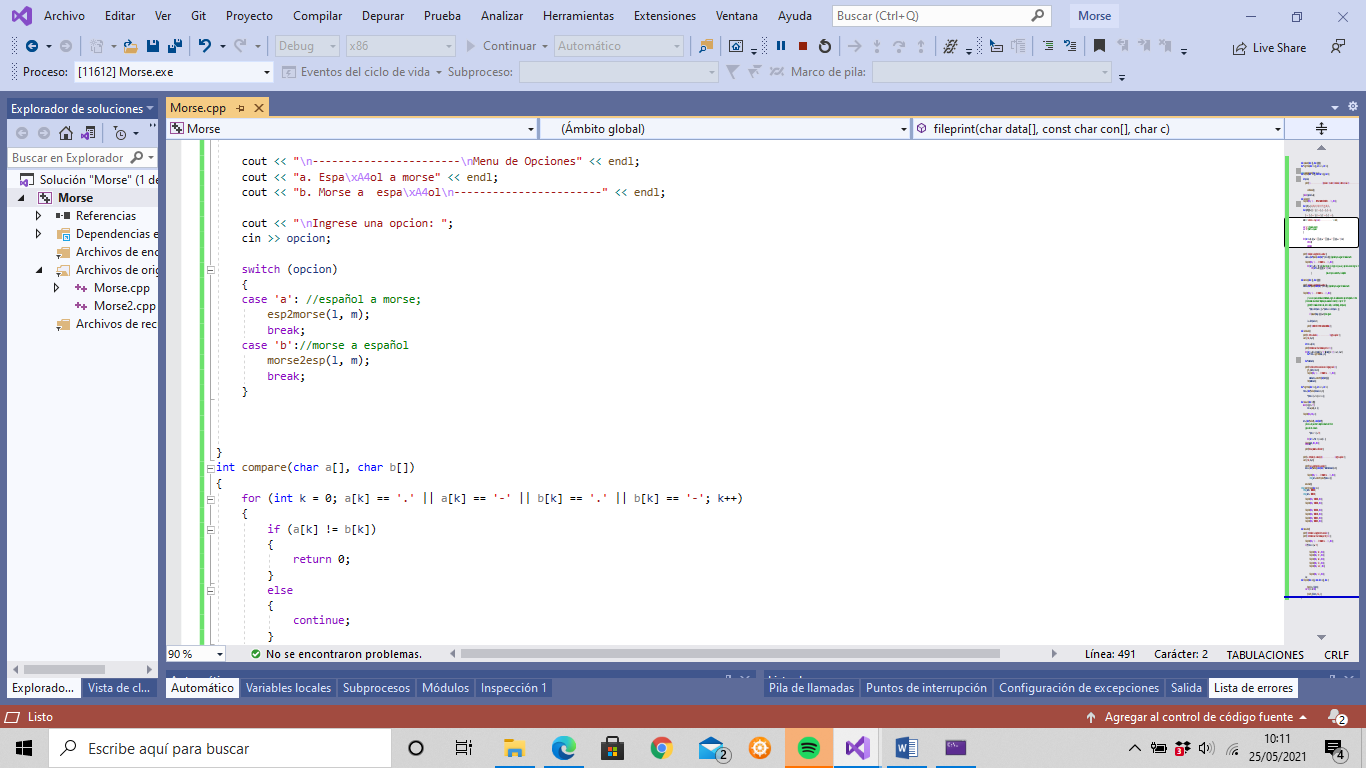
**Explicación del código**

Este código está creado para codificar y descodificar de texto a morse y viceversa. Para lograr esto se ha creado un código en el cual, se han empleado las siguientes librerías:

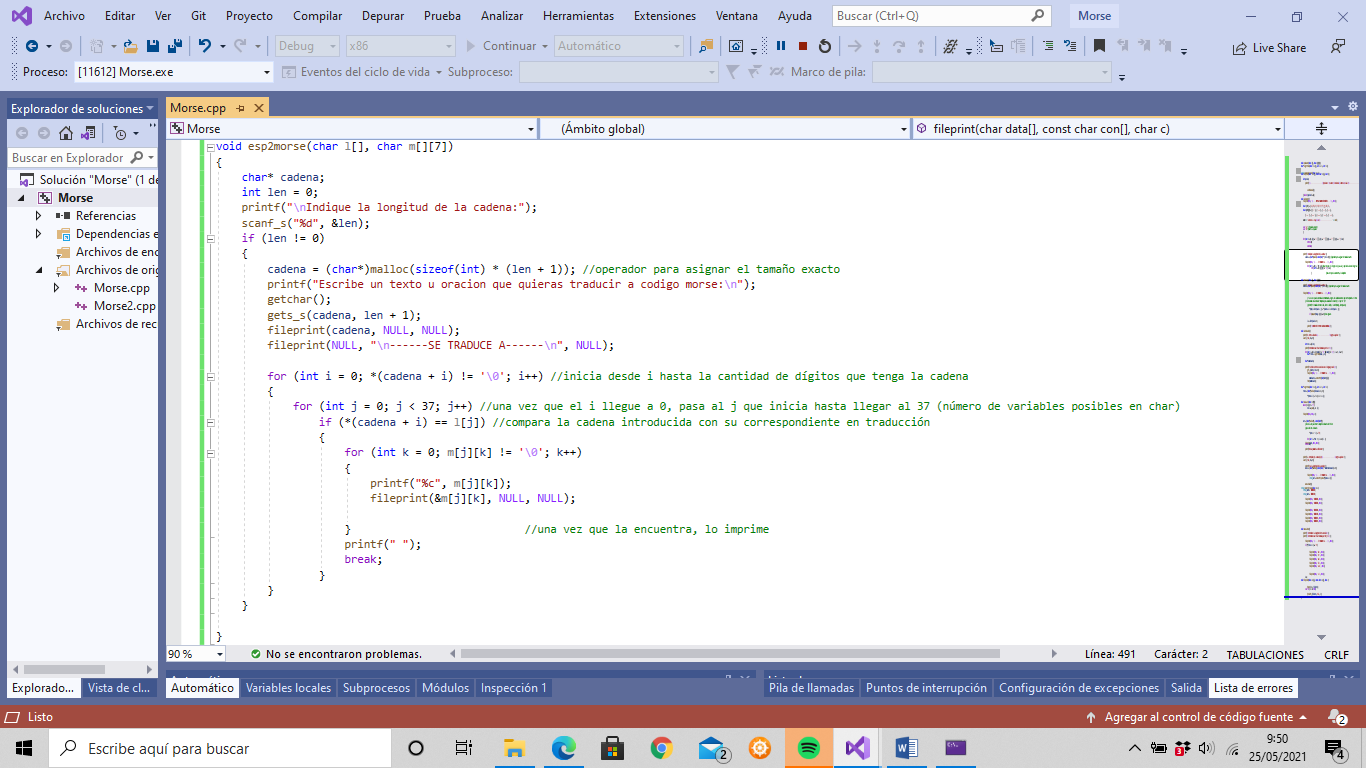
* #include <stdio.h>: contiene las funciones (macros) y tipos para manipular dato de entrada y de salida.
* #include <string.h>: para facilitar el uso de las cadenas de caracteres, ya que elimina dificultades que presentan los char.
* #include <stdlib.h>: contiene las funciones (macros) y tipos para uso general.
* #include <iostream>: contiene algoritmos estándar.
* #include <malloc.h>: necesaria para poder utilizar la función malloc que adjudica un espacio cuyo tamaño es especificado.
* using namespace std; : da acceso al espacio de nombres.

Por otro lado, se han creado tres funciones; la principal, de español a morse, de morse a español y la necesaria para incluir ficheros como archivos en los que poder escribir o añadir para su lectura.

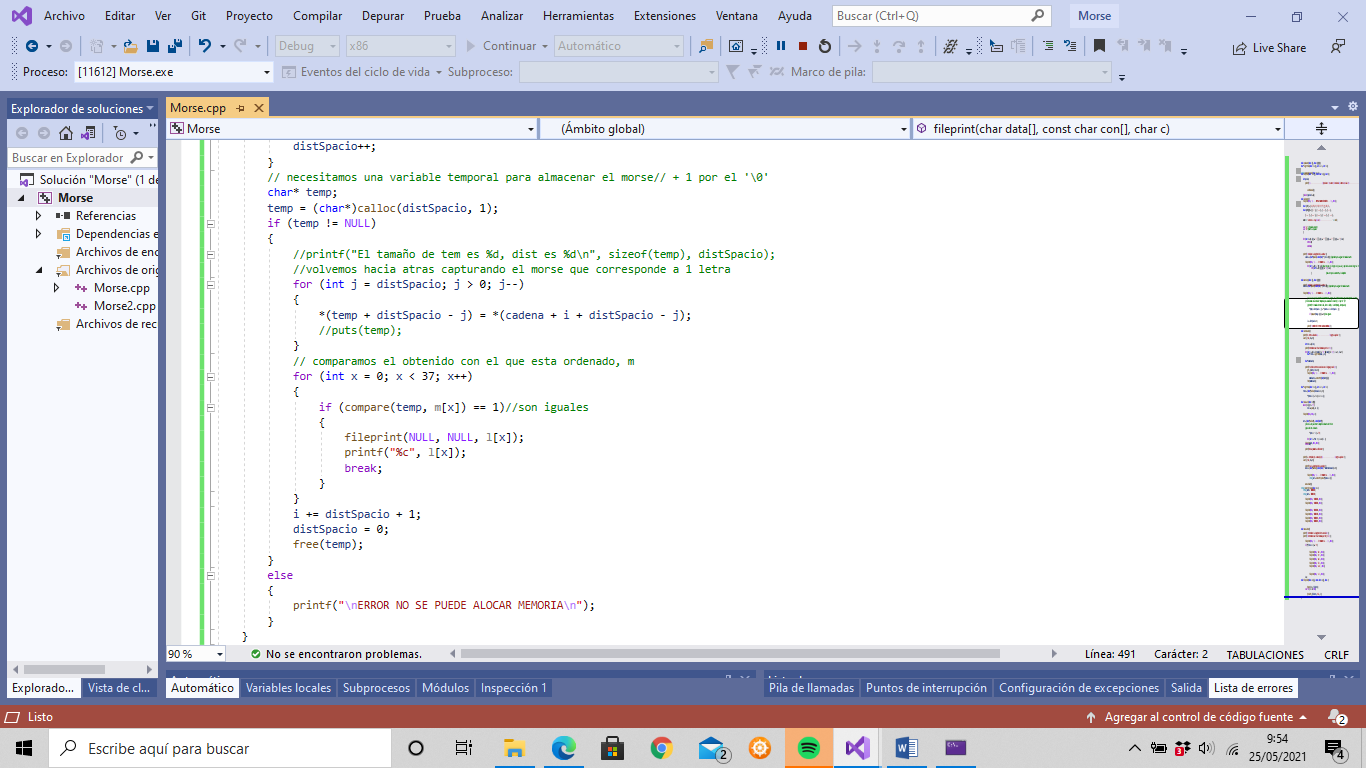
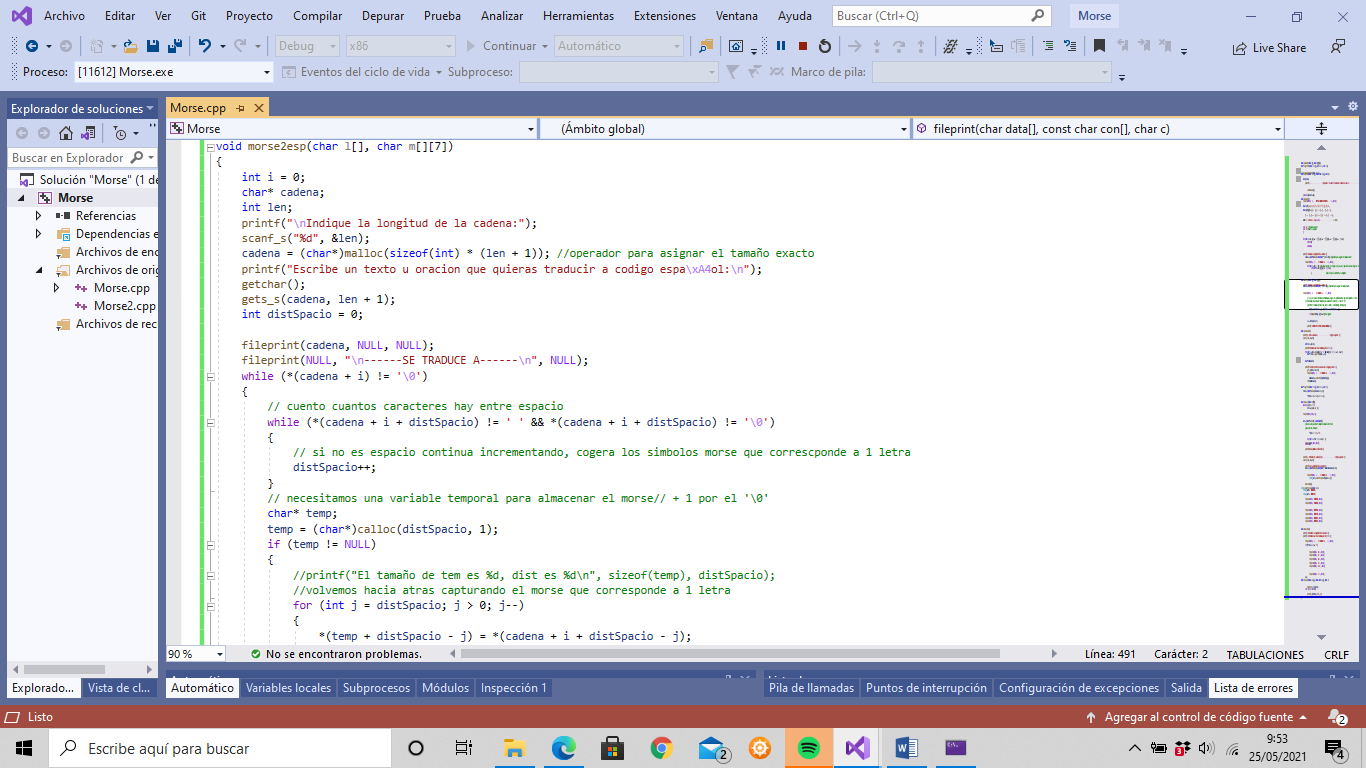
En la función principal, se han declarado las variables como char para las letras, números y sus correspondientes representaciones en morse. También, las necesarias para ejecutar el código como la char opción o char cadena (como puntero).

Una vez declaradas todas las variables se ejecuta un menú, con las opciones del código.

Para hacer que el usuario pueda elegir entre las opciones, se crea un “switch” con los casos en los que se introduce la función que les pertenece. Pues para distinguir de texto a morse y de morse a texto, se han creado dos funciones; esp2morse y morse2esp.

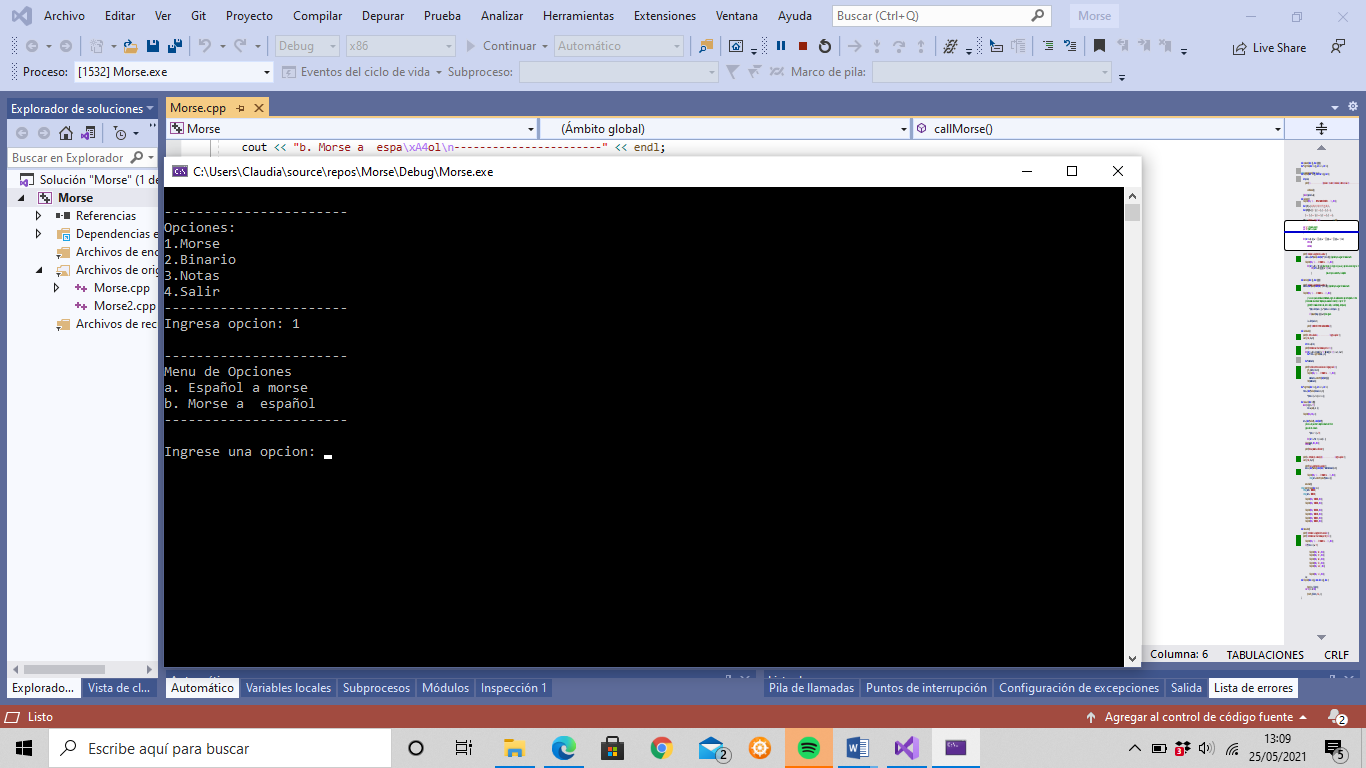
La primera de español a morse, en la cual, se inicia desde “i” hasta la cantidad de dígitos que haya seleccionado el usuario para la traducción de la cadena y una vez que el “i” llegue a 0, pasa al “j” que inicia hasta llegar al 37 (número de variables posibles en char). Hace esto, de tal modo que una vez que llega a un espacio que representará a un único caracter en ASCCI, lo selecciona para compararlo entre todos los posibles de morse para encontrar su correspondiente y coger la posición equivalente en la lista ASCCI.

La segunda de morse a español, en esta función, en primer lugar cuenta cuántos caracteres hay entre cada espacio. Las señales de morse no tienen longitud fija por ello, hay que capturar todos los caracteres entre espacio según incrementa que juntos representaran 1 señal morse y por lo tanto, una letra o número al ser traducido. Se compara el contenido con el que está ordenado hasta que sean iguales. Es decir, que el algoritmo recorre las líneas según el input, caracter por caracter haciendo así que cada uno corresponda con su equivalente en morse, tanto letras como números.

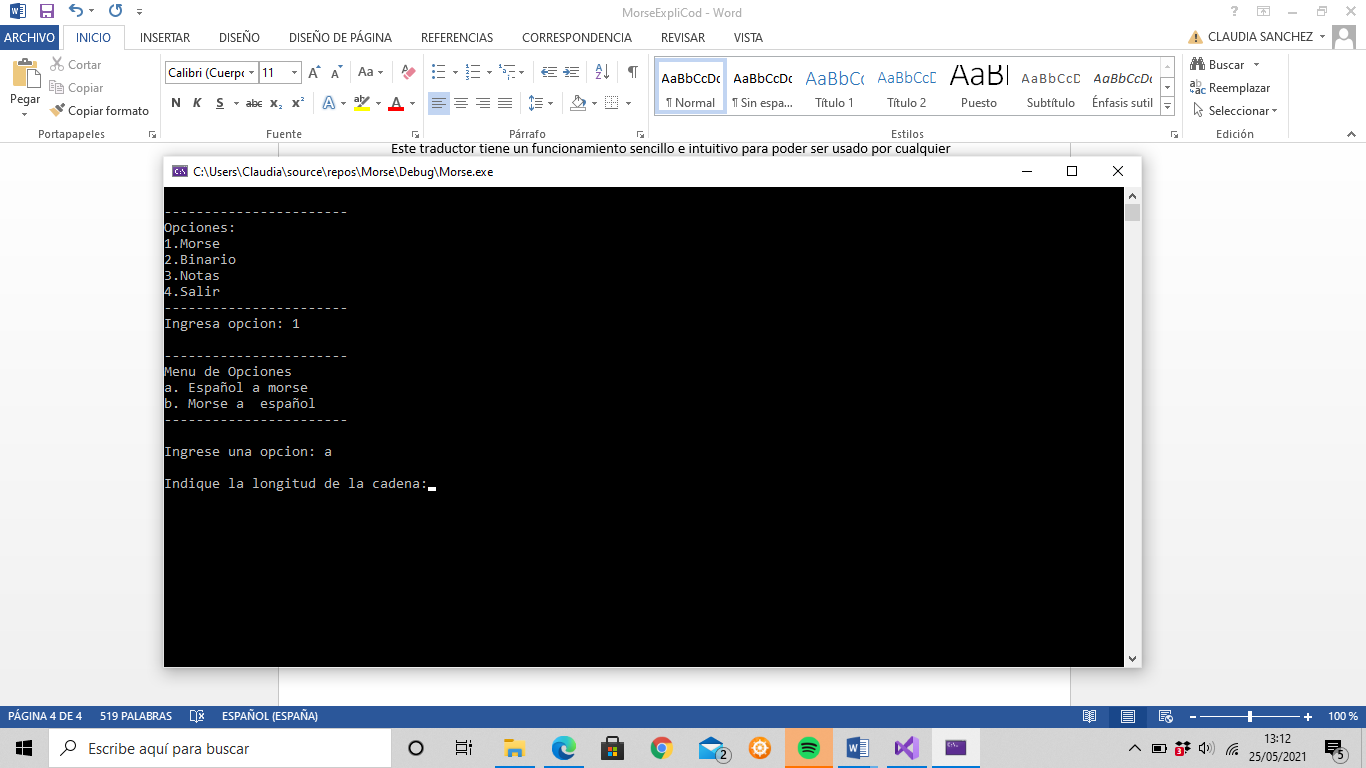


**¿Cómo utilizar el traductor?**

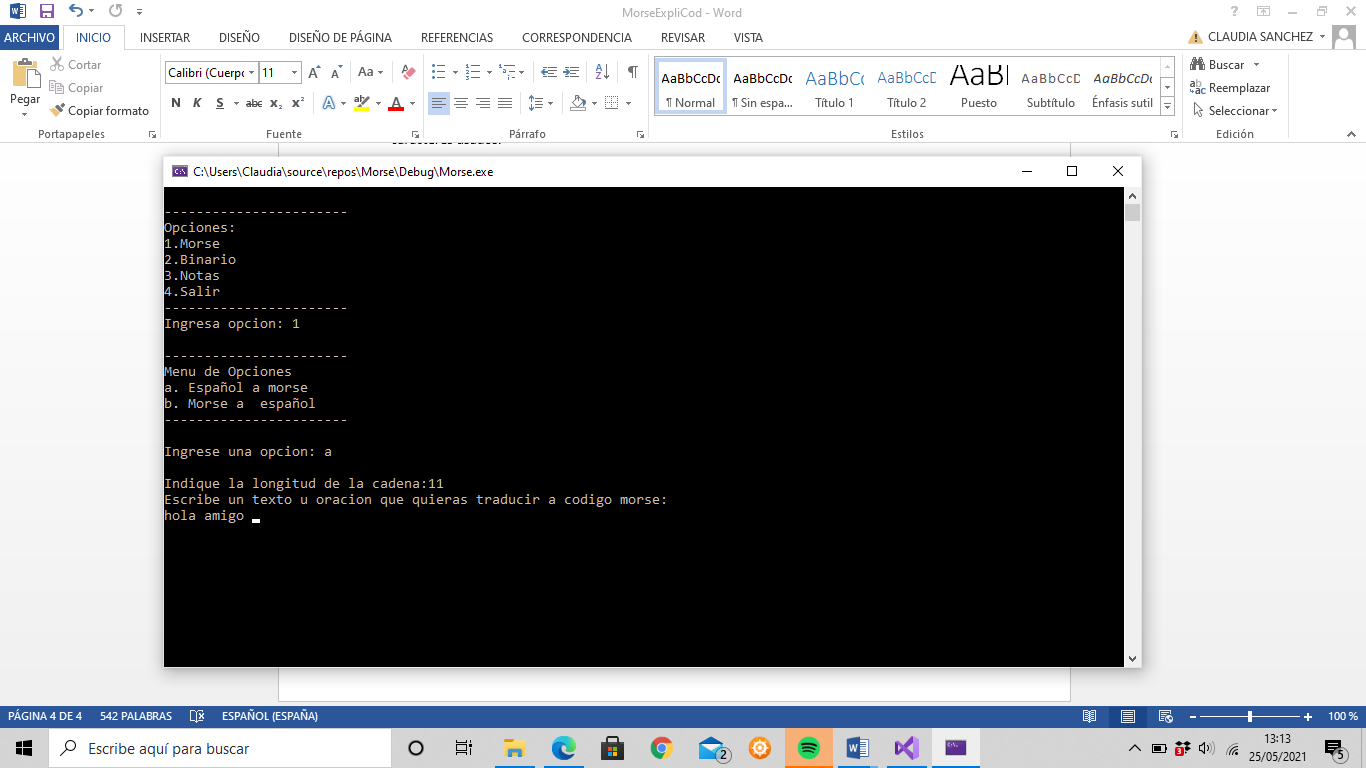
Este traductor tiene un funcionamiento sencillo e intuitivo para poder ser usado por cualquier usuario que desea traducir de morse a texto y/o viceversa.

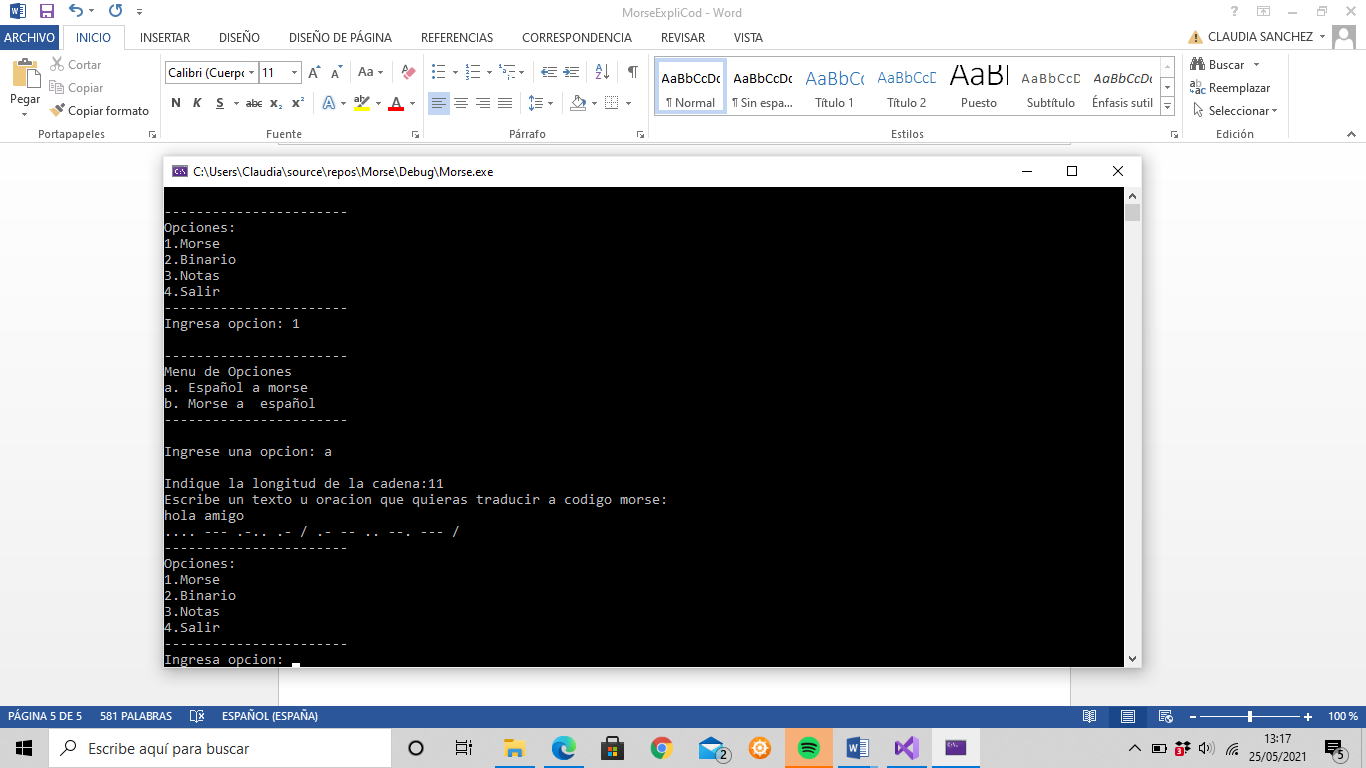
Una vez compilado el código y seleccionado la opción “Morse”, nos imprime en pantalla;

La opción seleccionada se escribirá con su correspondiente letra en el aparatado que pone “Ingrese una opción”.

Cuando se elige una opción, por ejemplo la “a”, el programa pedirá la longitud de la cadena, en la que el usuario deberá tener en cuenta los espacios empleados y el número de caracteres usados.

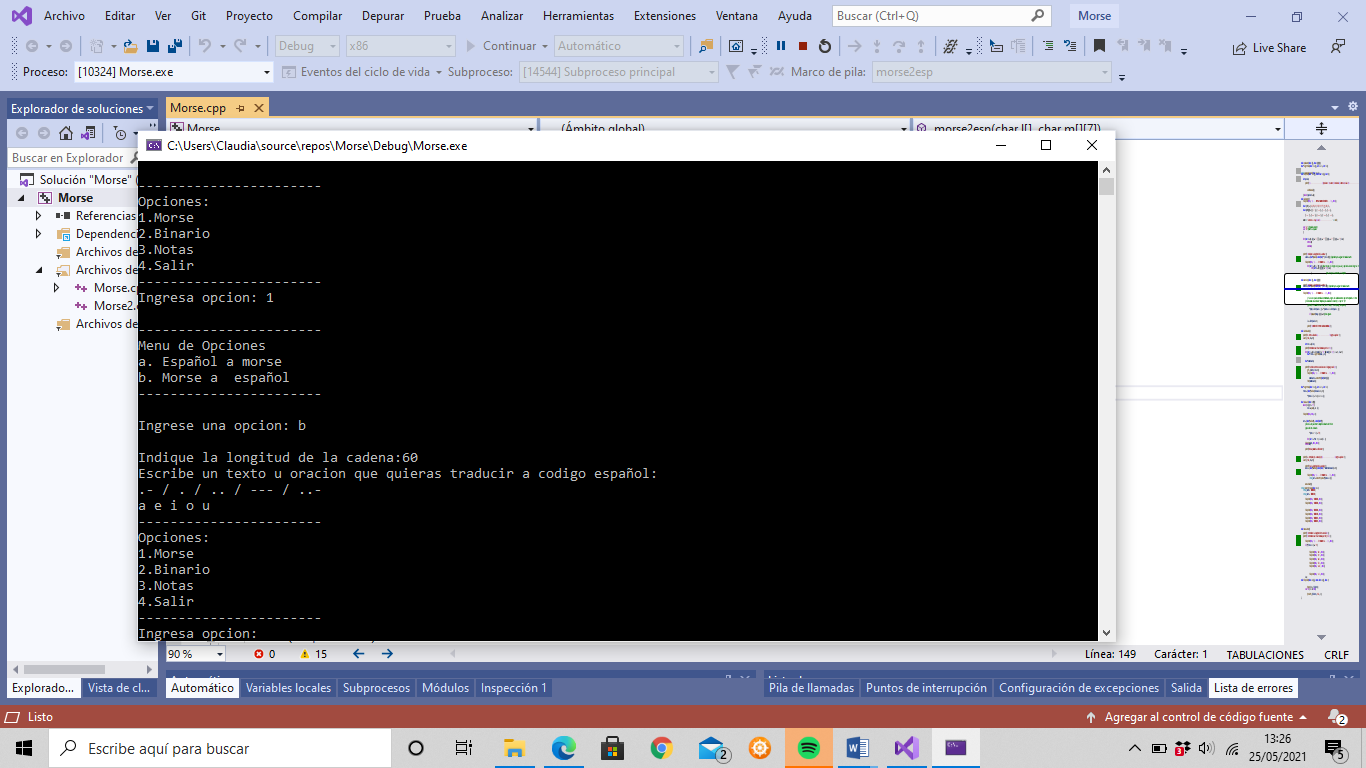
Por consiguiente, ya se escribirá el texto u oración que de desee traducir.

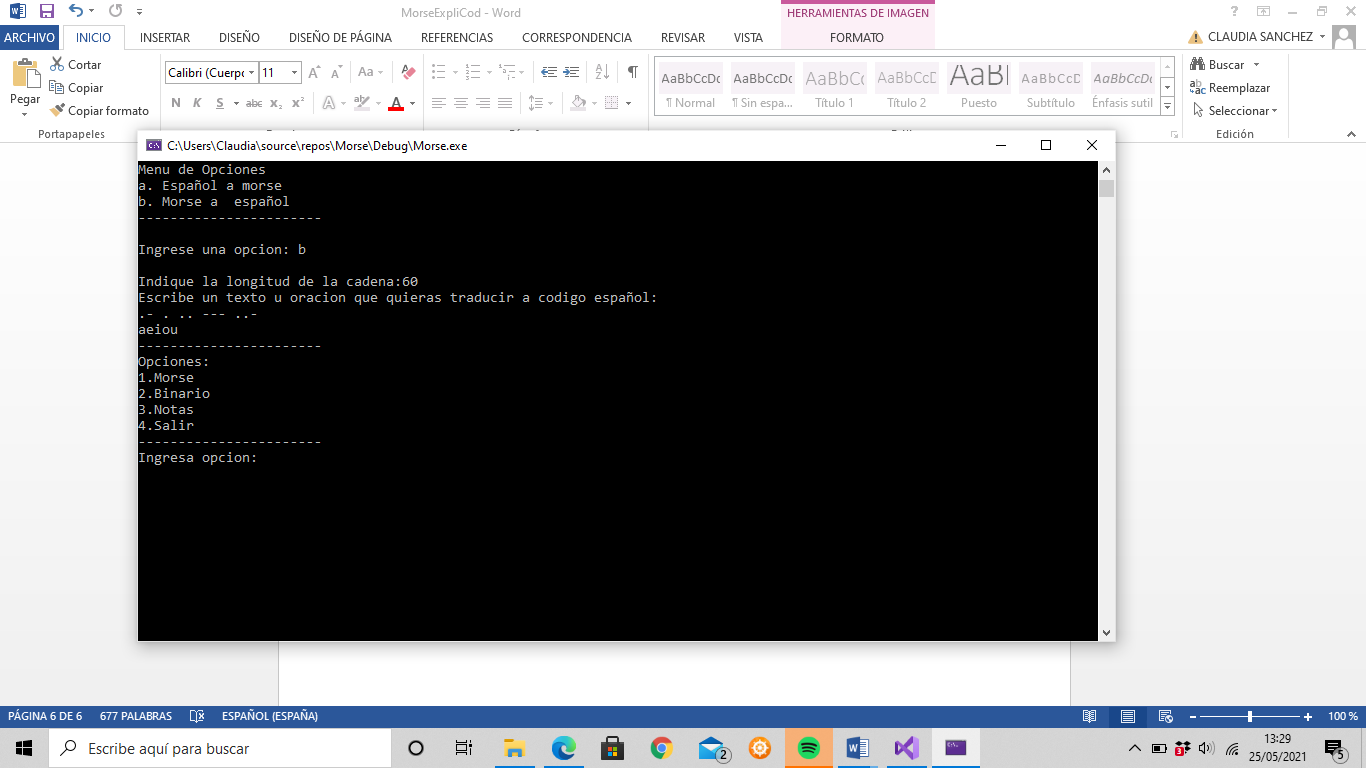


Para completar la traducción se presiona la tecla “ENTER” del teclado y éste, instantáneamente, traducirá a morse teniendo en cuenta que los espacios equivalen a “/”.

Como se ve en el ejemplo, el programa continuará en bucle hasta que se seleccione la opción de “Salir”. Por otro lado, si se seleccionase la opción b (“Morse a español”), el procedimiento del programa sería el mismo, tan sólo variando que el usuario deberá escribir en morse. En este caso, es importante tener en cuenta que los espacios corresponderán con presionar: espacio, barra, espacio, en este orden.

En el ejemplo siguiente se muestra.

Ejemplo 1. En este ejemplo, el usuario ha querido escribir las letras por separado. Para ello, ha empleado barras con espacios delante y detrás.



Ejemplo 2. En este ejemplo, el usuario ha querido escribir las letras juntas. Por ello, no se emplean barras, sino que se escriben las letras con espacios.