Carrera:

**Ingeniería en Tecnologías de la Información y Comunicación**

Materia:

**Programación de aplicaciones**

Nombre del alumno(a):

**Sergio Salvador Tafoya Ramírez**

**Alejandra Cisneros García**

Nombre del Profesor(a):

**MI. Ariana Gómez Contreras**

Grado y grupo:

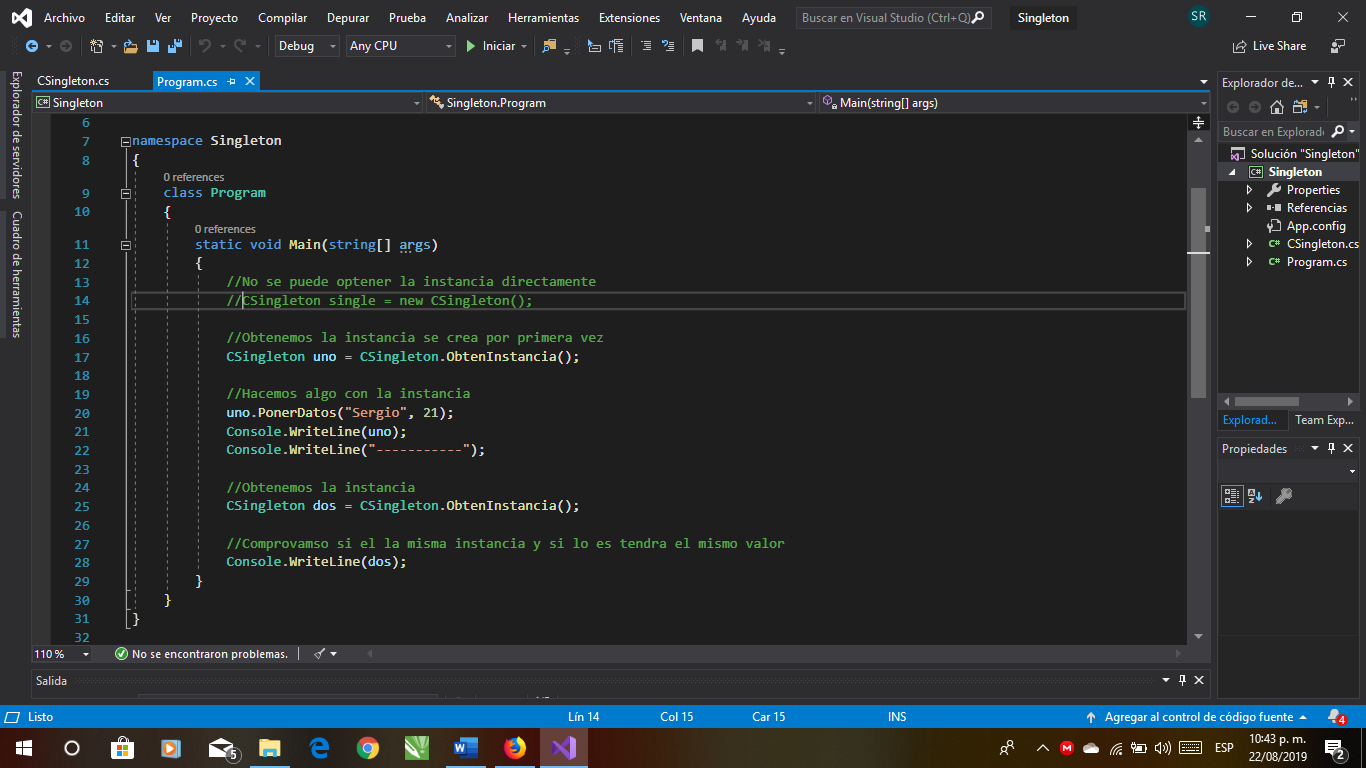
**9° “B”**

**Jueves 22 de agosto de 2019.**

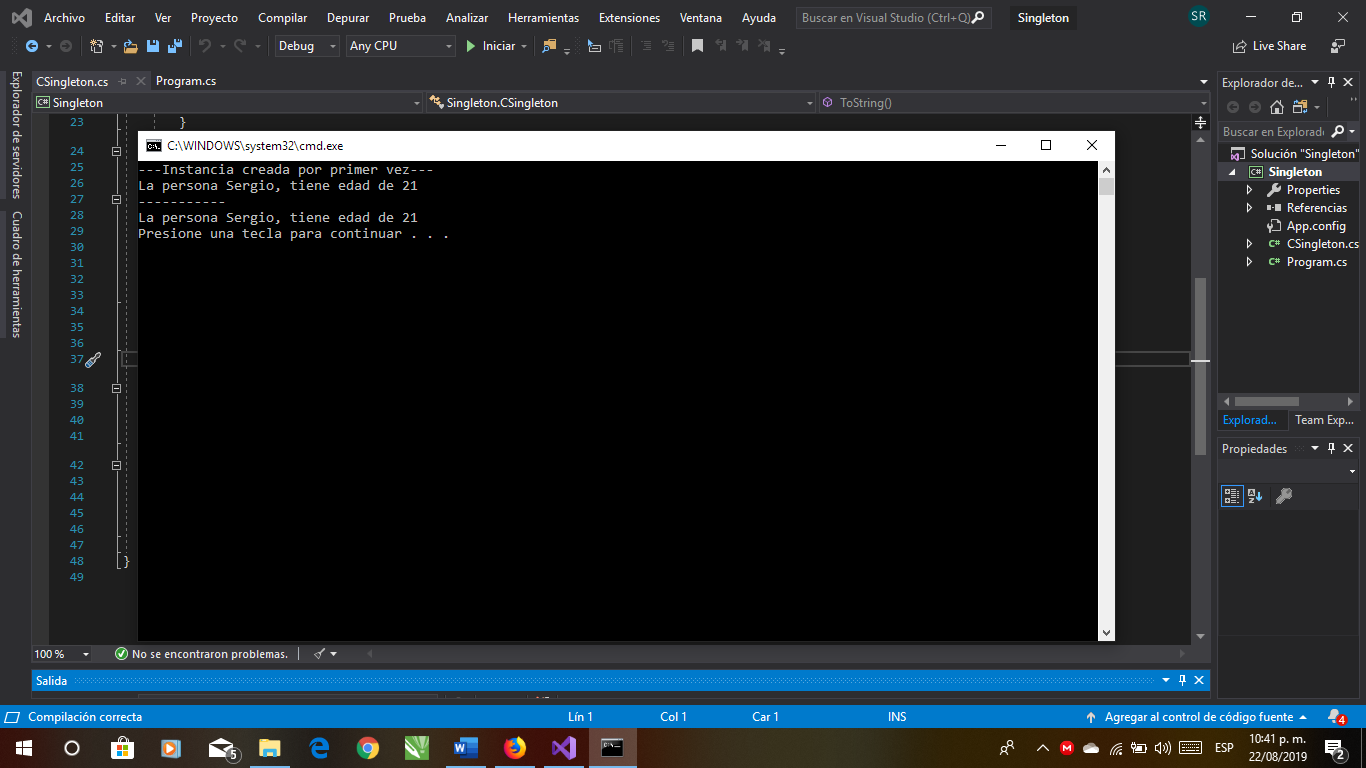
***Patrón de diseño Singleton.***

Este patrón de diseño busca garantizar que en una clase solo pueda crearse una sola instancia la cual tiene que proporcionar un punto de acceso global a ella.

Al momento de ejecutarse la aplicación intentaremos crear dos instancias (Ilustración 1), y como solo se puede crear una ves el segundo valor será el mismo que el primero como se muestra en la *Ilustración 2*.



Ilustración



Ilustración

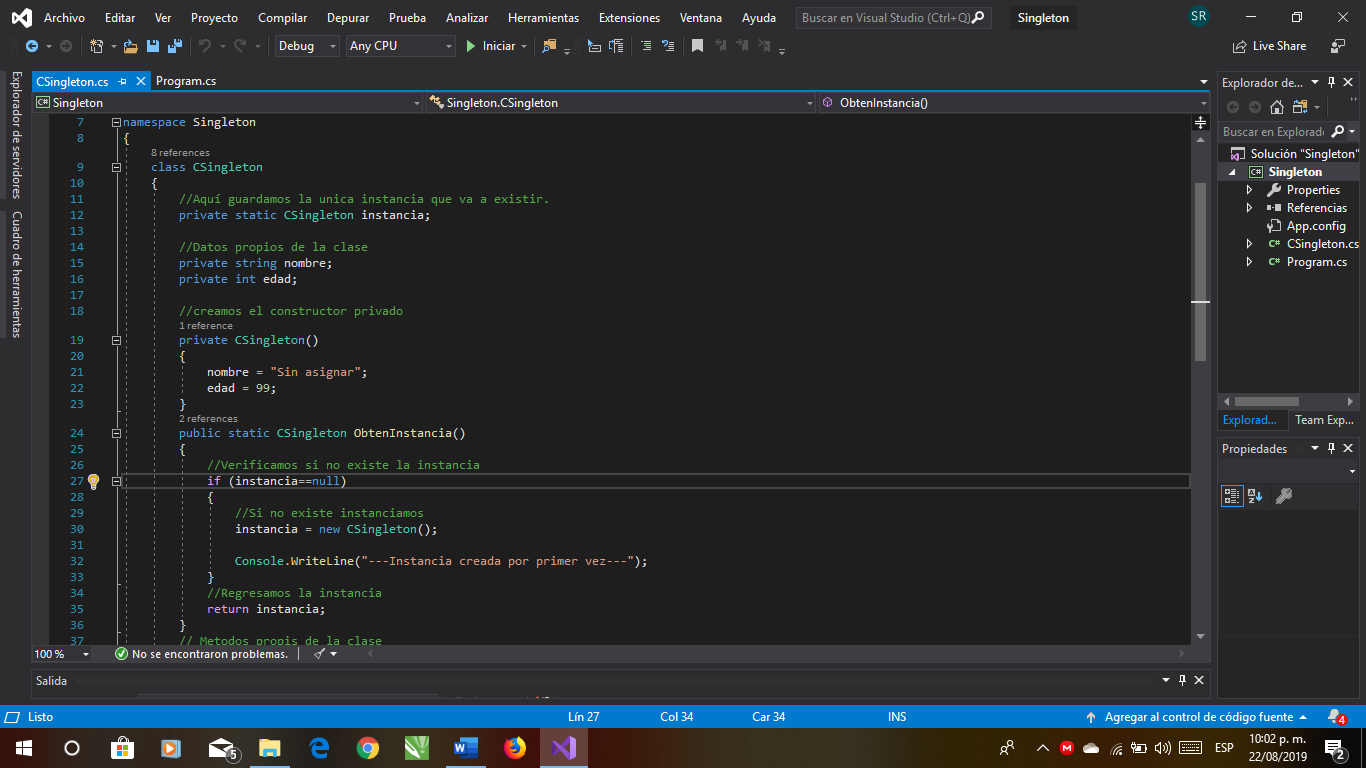
***Buenas prácticas implementadas (ilustración 3 y 4).***

1.- Las variables que utilizamos no contenían nombres claves o reservados.

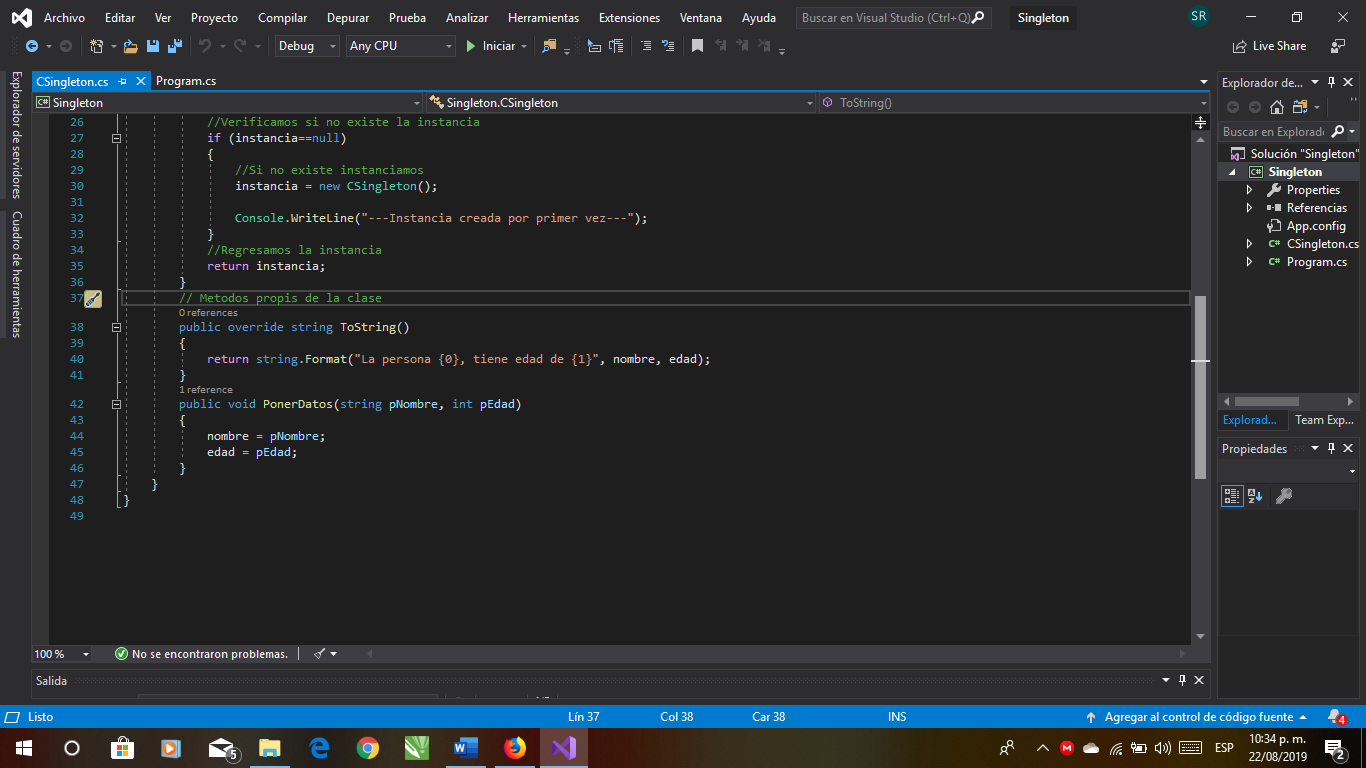
2.- Las variables o identificadores que utilizamos empezaban con una letra minúscula y se la componían dos palabras la segunda empezaba con mayúscula.

3.- En cada bloque de código colocamos comentarios para un mayor entendimiento.

4.- Utilizamos sangría para identificar de una mejor manera donde empieza y termina un bloque de código.



Ilustración



Ilustración

***Seguridad implementada.***

Lo que realizamos para que todo funcionara de la manera correcta fue crear un constructor privado para que no pudiera ser accedido desde el exterior (Ilustración 5).

Otra cosa muy importante que realizamos fue la creación de un método público para que pueda ser accedido desde el exterior el cual contenga una condicional que verifique que exista la instancia y si ya existe solo mandarla llamar y ya no créala (Ilustración 6).

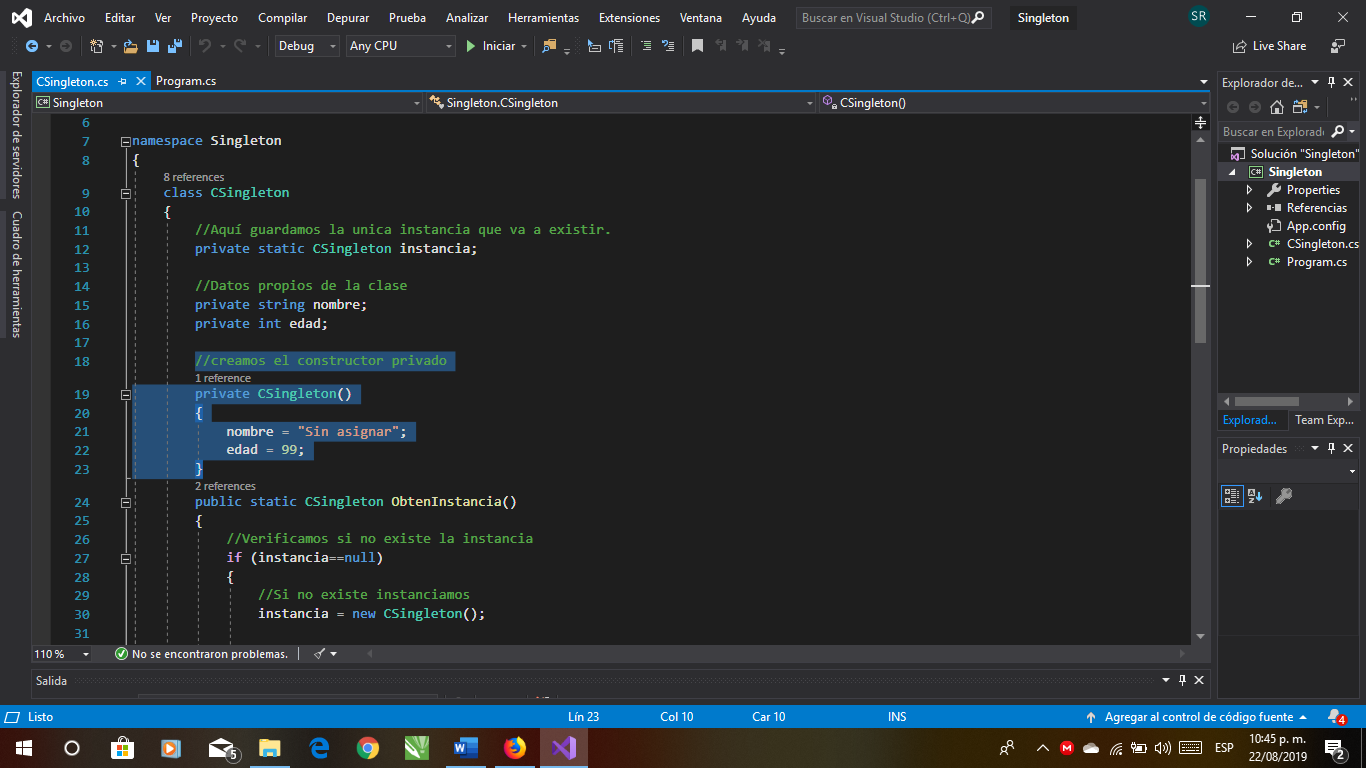


Ilustración 5

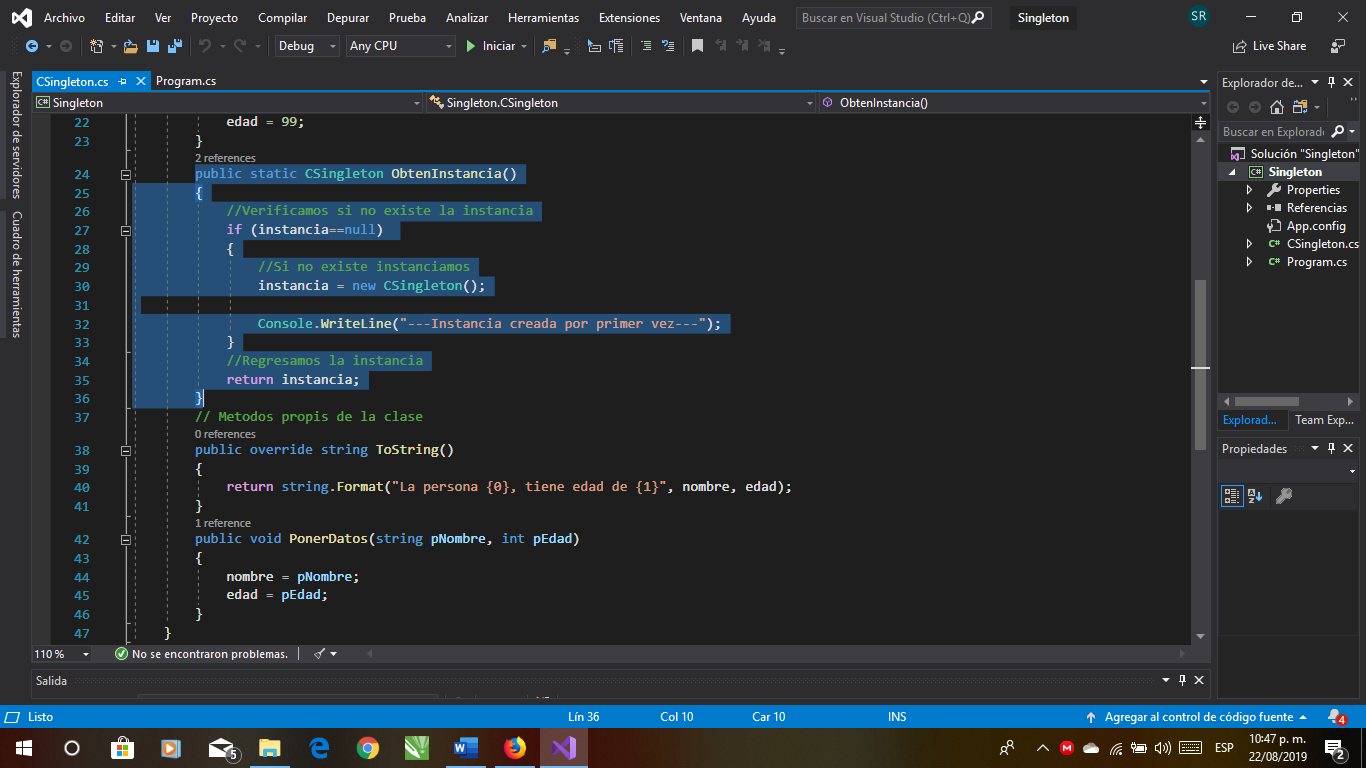


Ilustración 6