Laura Carratalá Castillo

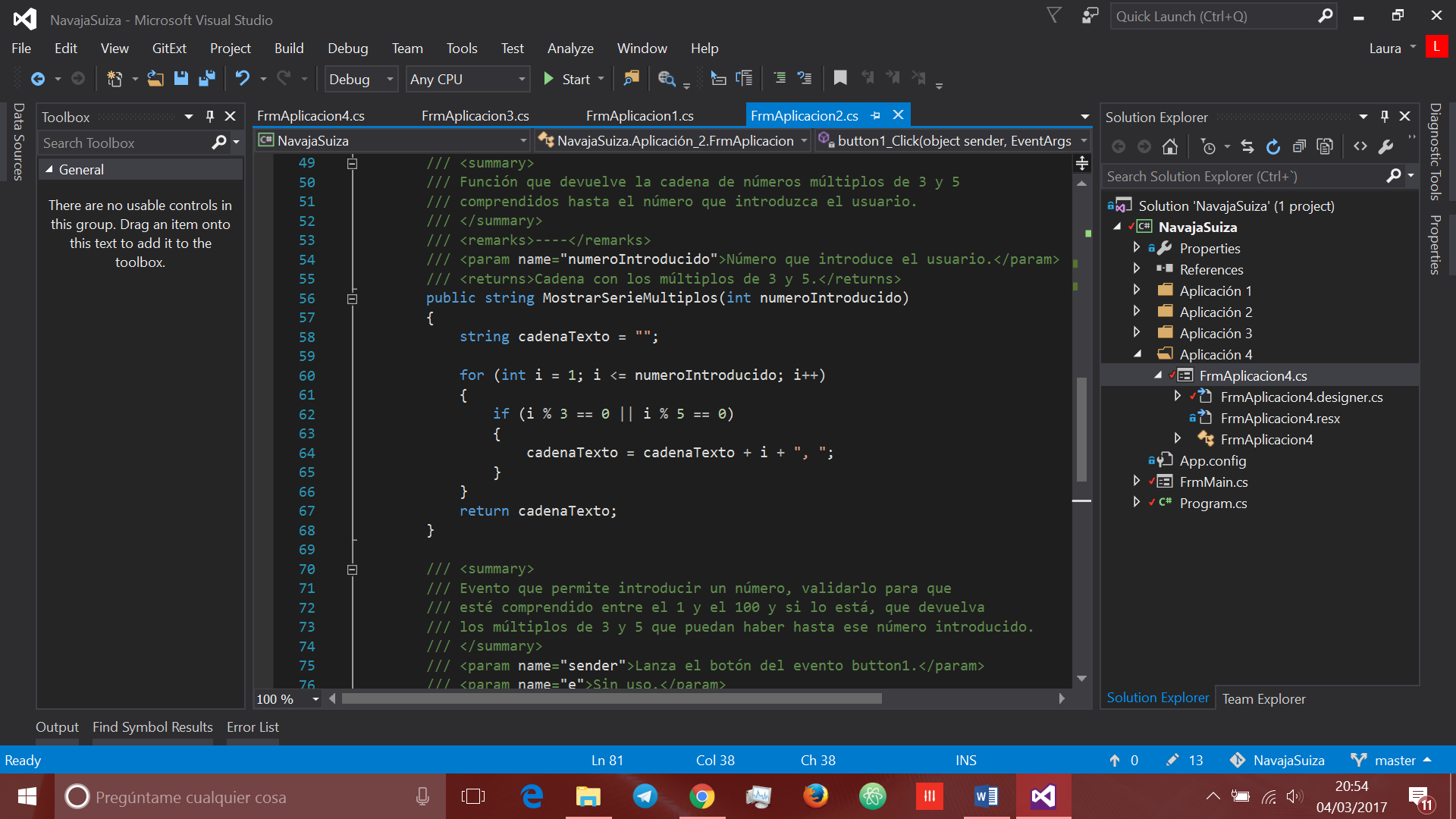
**Política de estilos para el desarrollo del proyecto Navaja Suiza**

A continuación se expondrá una breve explicación de los estilos que se han seguido para desarrollar las aplicaciones que forman parte del proyecto Navaja Suiza.

1. **Declaración de variables:**

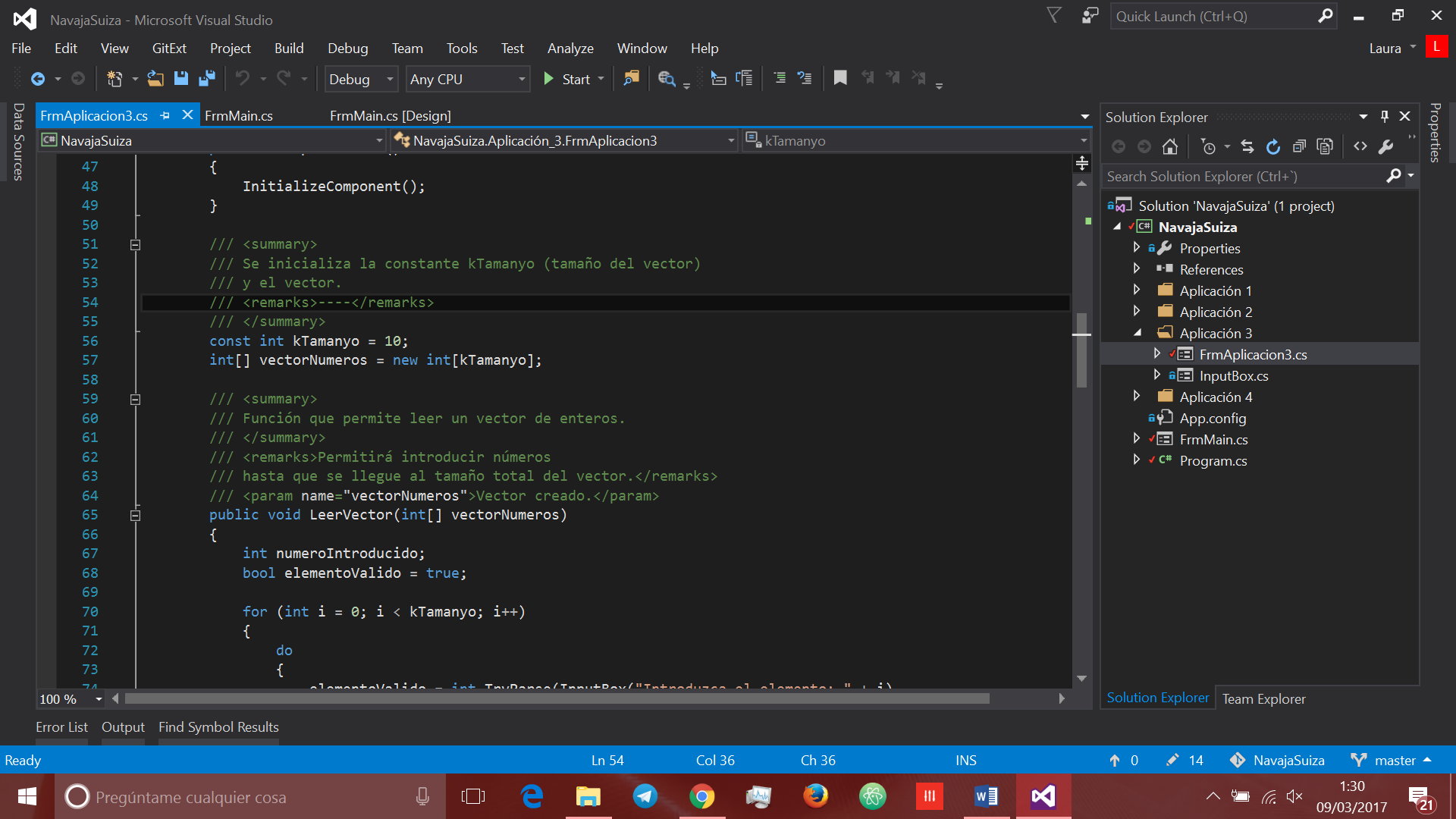
Se ha realizado una declaración por línea, inicializando en la medida de lo posible durante la declaración. Se han empleado nombres autoexplicativos y fáciles de entender y siguiendo el convenio de nombres de estilo **caMel** (la primera letra de cada palabra es minúscula y las iniciales siguientes son en mayúscula)**.**

Para los bucles, se usan contadores locales de tipo i, j, etc, y se declaran e inicializan en el mismo bucle.



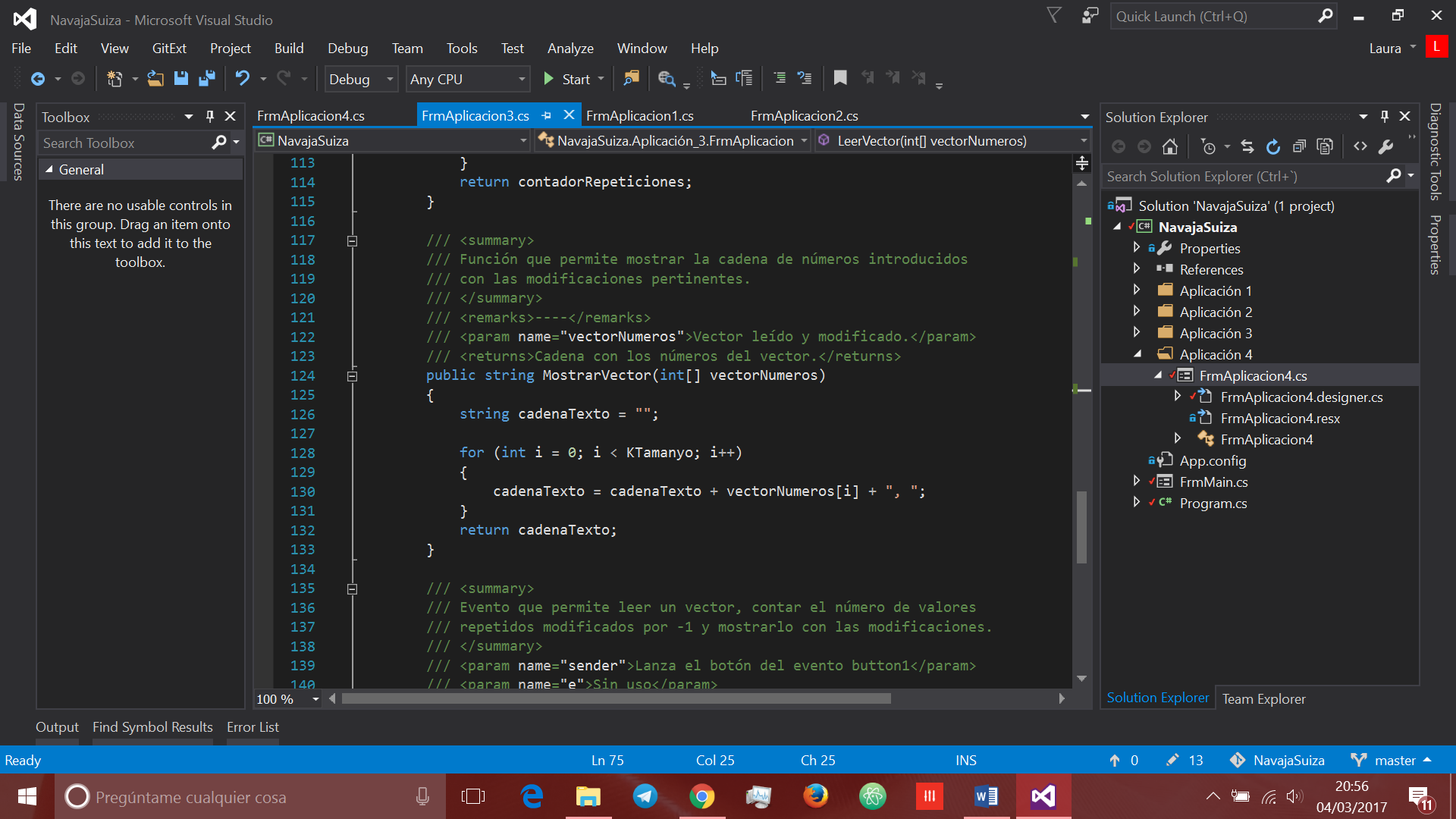
1. **Declaración de constantes:**

La declaración de la única constante empleada (kTamanyo) se ha hecho siguiendo el estilo **caMel** y no utilizando caracteres problemáticos como la ñ.



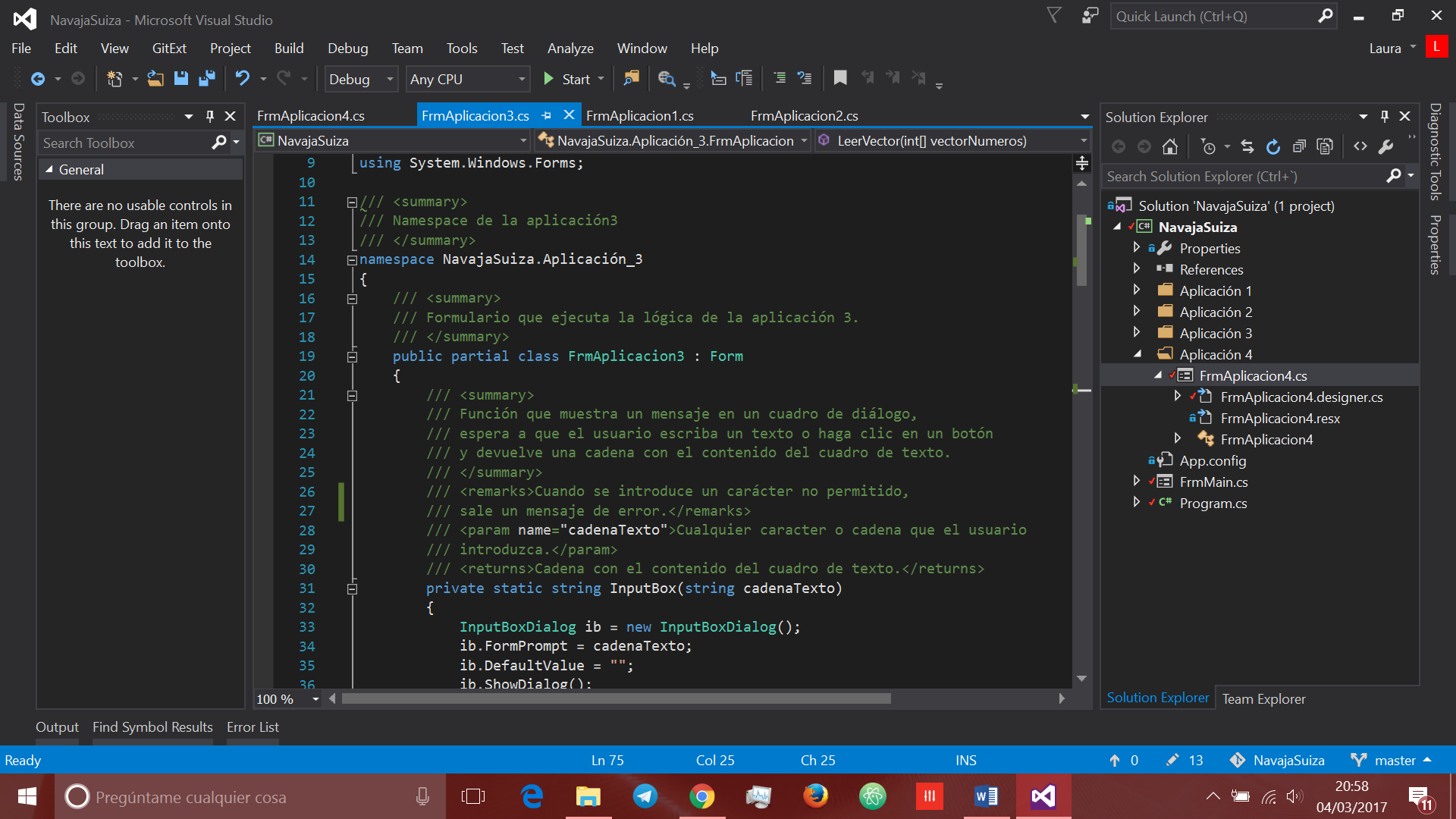
1. **Nombres de métodos:**

Los métodos se nombran al estilo **PasCal** (la primera letra de cada palabra es mayúscula y las iniciales siguientes son todas mayúsculas). Para nombrarlos se usan verbos indicativos de la función que se realiza en el método.



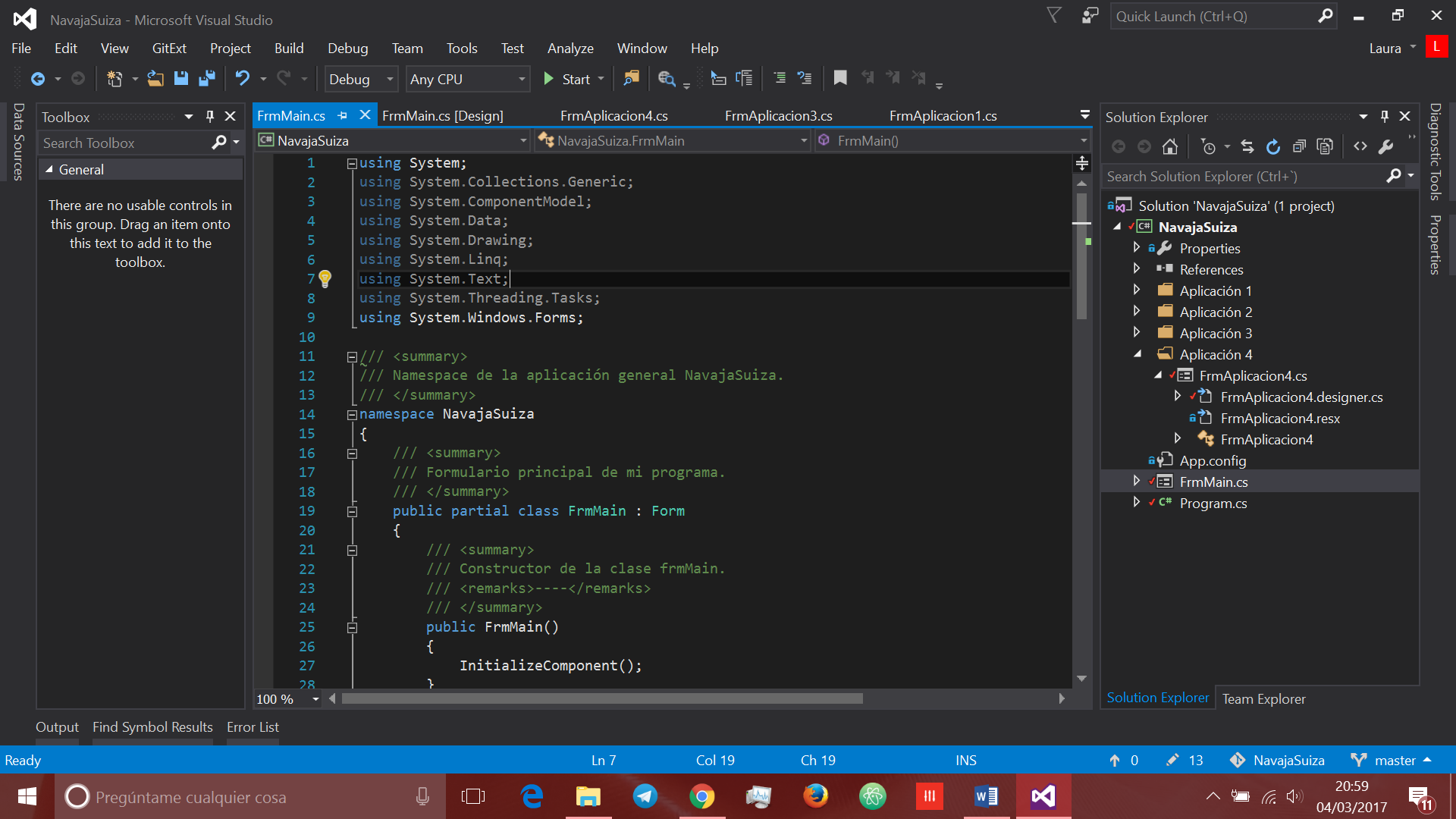
1. **Nombres de clases:**

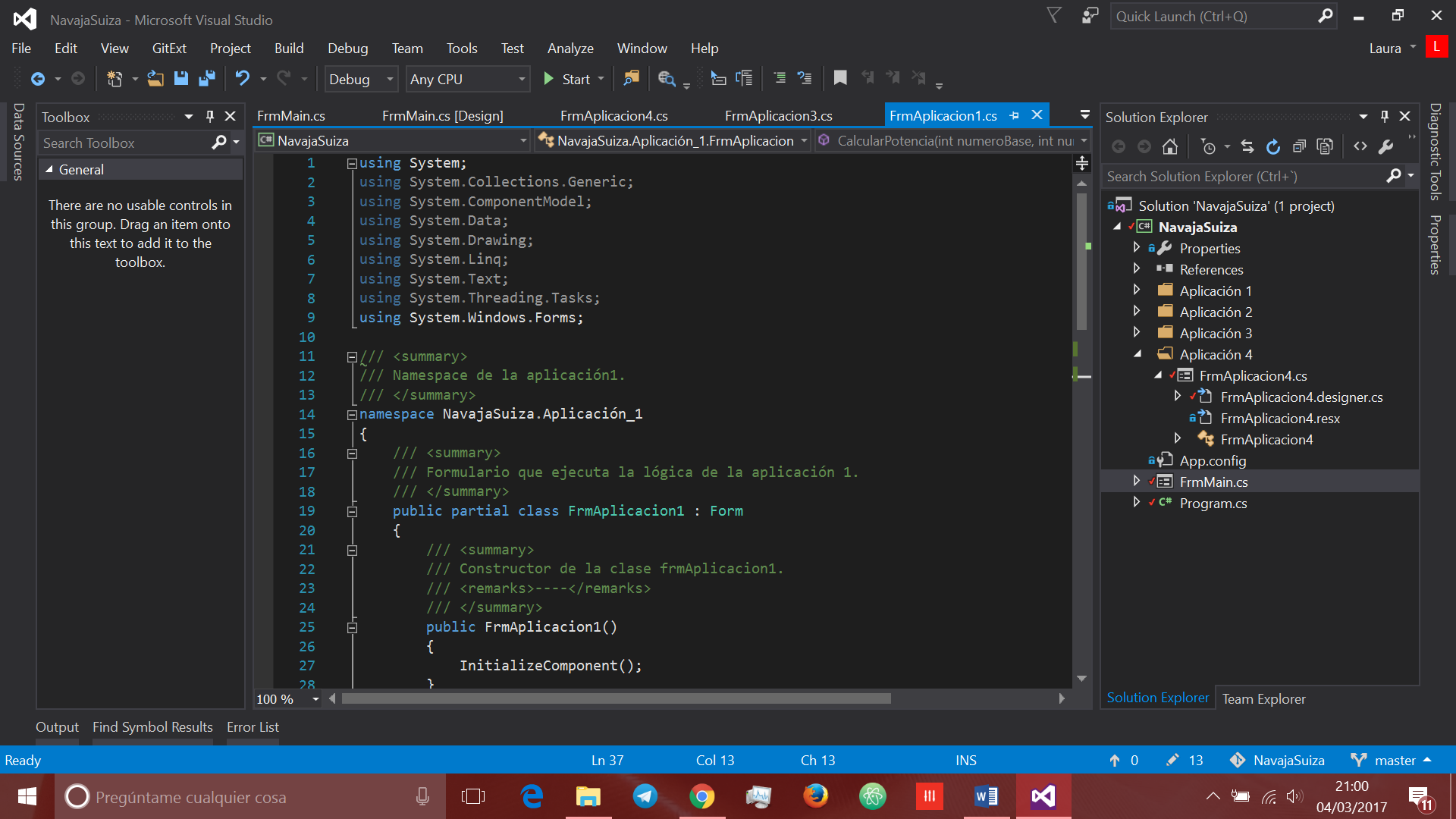
Las clases de los formularios se han renombrado al estilo **PasCal**, usando sustantivos entendibles e identificativos de cada aplicación (FrmMain, FrmAplicacion1, etc).



1. **Namespaces:**

Los espacios de nombres no están duplicados. Tenemos un namespace general que abarca todas las aplicaciones llamado “NavajaSuiza” y después cada una de las aplicaciones tiene su propio namespace identificativo. Así, la aplicación 1 tiene como namespace “NavajaSuiza.Aplicación\_1”, la aplicación 2 “NavajaSuiza.Aplicación\_2”, la aplicación 3 “NavajaSuiza.Aplicación\_3” y la aplicación 4 “NavajaSuiza.Aplicación\_4”.





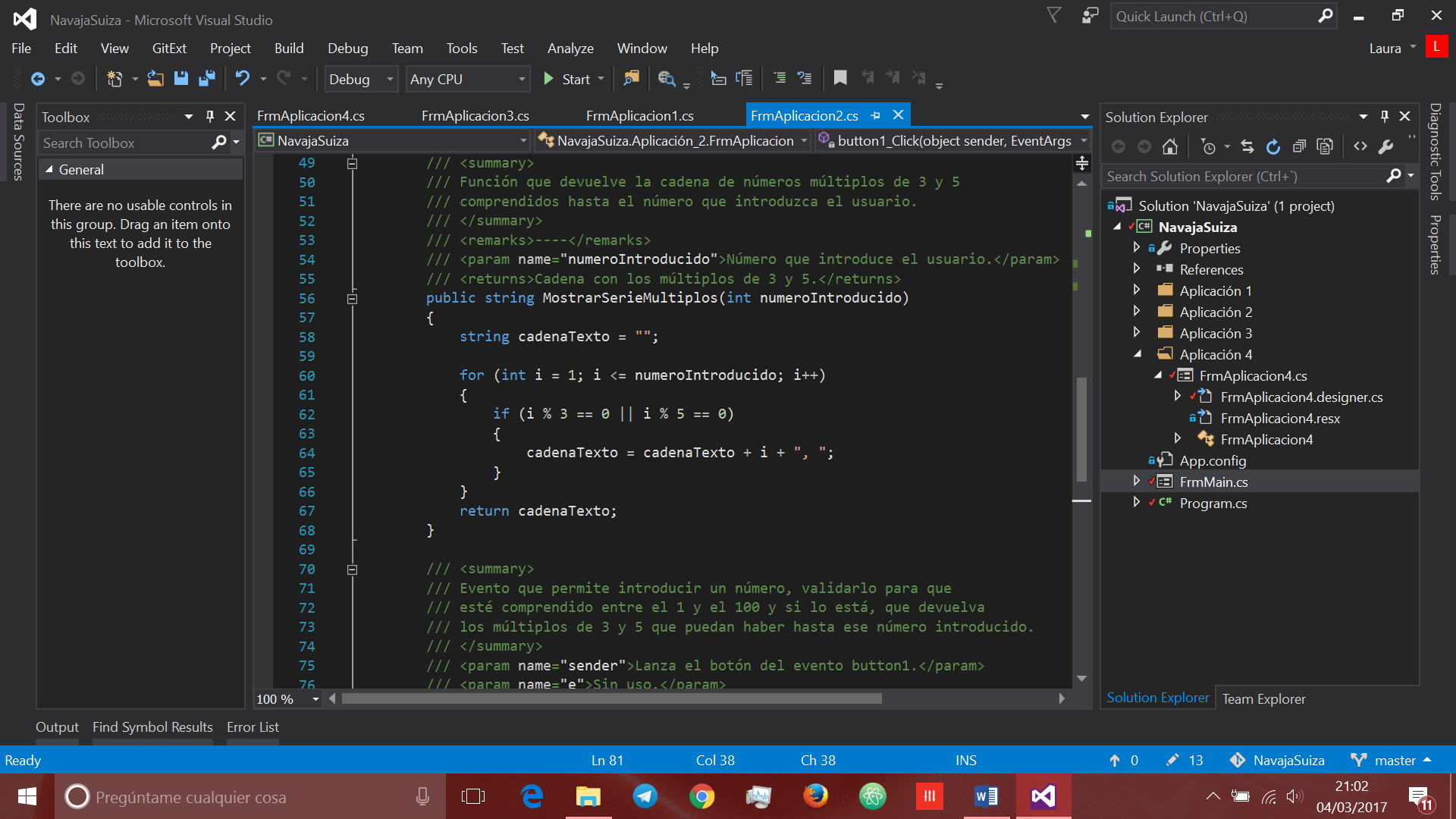
1. **Indentación, llaves, espacios y líneas en blanco:**

Se ha cuidado la indentación del código, no usando espacios y dividiendo líneas de código que sean demasiado largas después de comas.

También se han respetado las llaves, no dejando ningún bucle sin ellas y sin incluir ningún comentario o instrucción en ninguna de ellas.

Se han dejado también espacios en blanco después de coma o punto y coma y alrededor de operadores.

Se han usado líneas en blanco para separar declaraciones de variables del resto del código y dentro de los métodos para separar las diferentes secciones lógicas.



1. **Comentarios:**

Se han usado comentarios en todo el proyecto para explicar lo que hacen los métodos, clases, etc. y así realizar la documentación, utilizando todas las etiquetas pertinentes.

