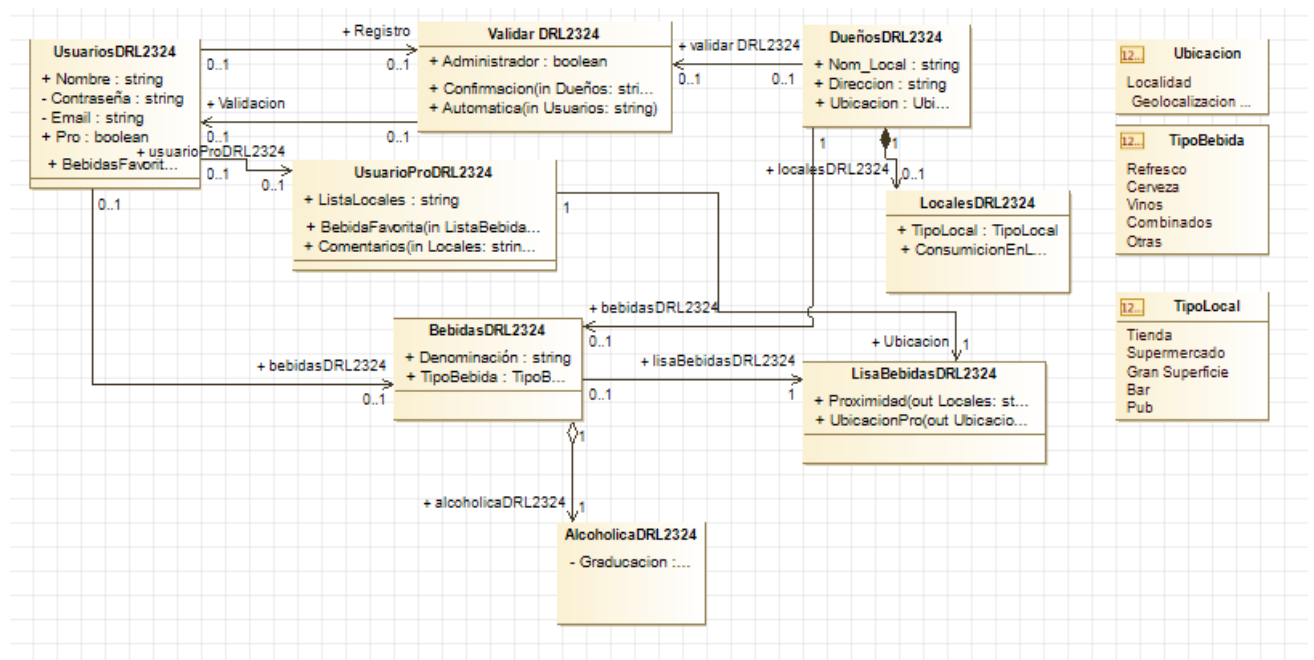


Pregunta 1

Dibujas el Diagramad de Calse



Pregunta 2

Apartado C:

Primero probamos el programa para ver en que puede fallar, nos salta el error que nos ha dicho José Ramón, lo solucionamos y volvemos a probar el programa, vemos que la primera frase si que la secciona bien, pero la segunda no

Colocamos un punto de interrupción cuando se realiza la primera separación de frase para ver que sucede

```

31 public int analizar(String f1, String f2)
32 {
33     String word;
34     words.Clear();
35     wordcount=0;
36     // primer paso, analizar la primera cadena
37     if (f1==null || f1.Length==0) return -1;
38
39     int posini=0;
40     int posfin=f1.IndexOf(' '); // encontramos el primer espacio
41     while (posfin!=-1) {
42         word=f1.Substring(posini,posfin-posini);
43         if (word.Length>0 && !words.Contains(word)) {
44             words.Add(word);
45             wordcount++;
46         }
47         posini = posfin + 1;
48         posfin = f1.IndexOf(' ', posini);
49     }
50     // añadimos la última palabra

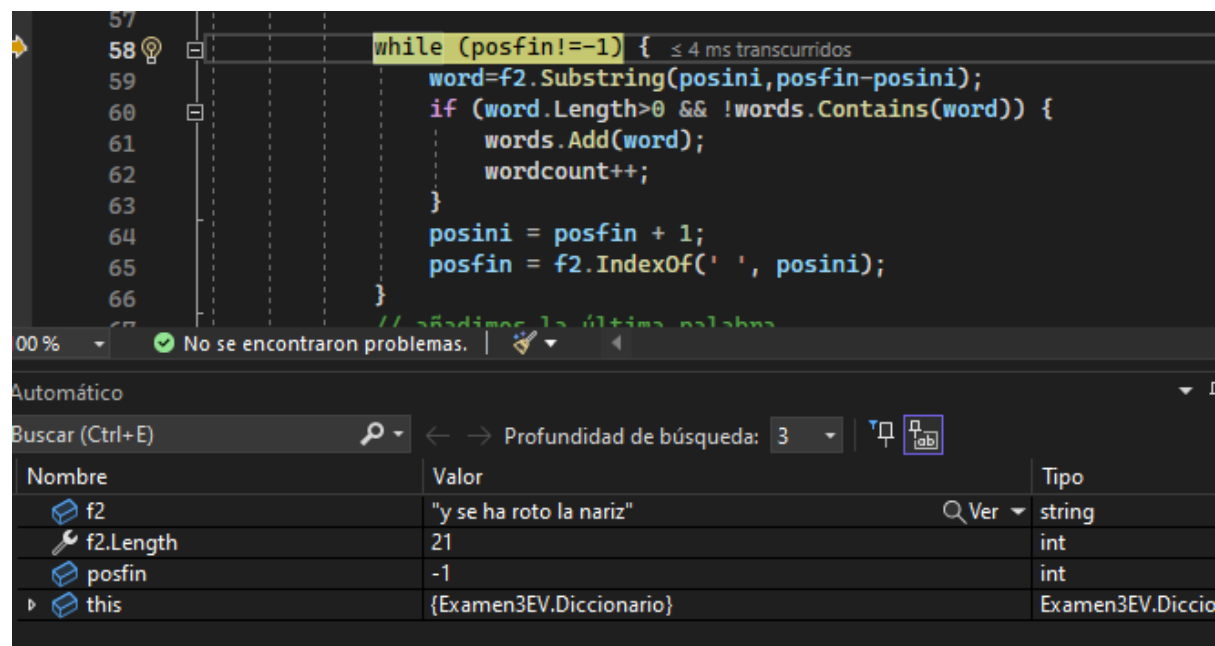
```

Y vemos que las variables mantienen el contenido correcto por lo que el fallo tiene que estar en las ejecuciones

Nombre	Valor	Tipo
f1	"Pelayo se cayó"	string
f2	"y se ha roto la nariz"	string
this	{Examen3EV.Diccionario}	Examen3EV.Diccionario

En el primer caso todo está correcto, así que observamos que sucede con la segunda frase.

La lectura de la variable es correcta y mantiene toda la frase



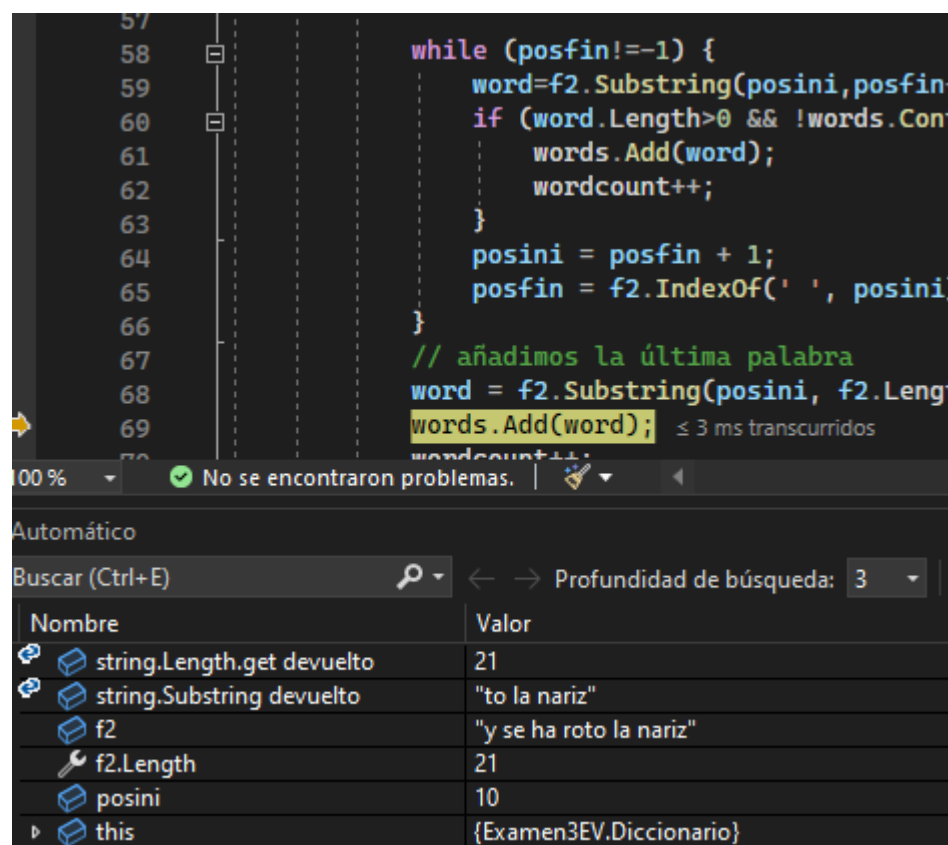
The screenshot shows a Visual Studio code editor with a C# program. The code is as follows:

```
57
58 while (posfin!=-1) { ≤ 4 ms transcurridos
59     word=f2.Substring(posini,posfin-posini);
60     if (word.Length>0 && !words.Contains(word)) {
61         words.Add(word);
62         wordcount++;
63     }
64     posini = posfin + 1;
65     posfin = f2.IndexOf(' ', posini);
66 }
```

Below the code, the Variable Watch window is open, showing the following variables:

Nombre	Valor	Tipo
f2	"y se ha roto la nariz"	string
f2.Length	21	int
posfin	-1	int
this	{Examen3EV.Diccionario}	Examen3EV.Diccionario

Comprobamos que no entra a realizar la separación de palabras porque no está indicada la función en la que la posini para F2 comience después del espacio, entonces trae el valor de la primera separación de palabra que es después del carácter 10



The screenshot shows the same Visual Studio code editor with the same C# program. The code is as follows:

```
57
58 while (posfin!=-1) {
59     word=f2.Substring(posini,posfin-posini);
60     if (word.Length>0 && !words.Contains(word)) {
61         words.Add(word);
62         wordcount++;
63     }
64     posini = posfin + 1;
65     posfin = f2.IndexOf(' ', posini);
66 }
67 // añadimos la última palabra
68 word = f2.Substring(posini, f2.Length);
69 words.Add(word); ≤ 3 ms transcurridos
70 wordcount++;
```

Below the code, the Variable Watch window is open, showing the following variables:

Nombre	Valor
string.Length.get devuelto	21
string.Substring devuelto	"to la nariz"
f2	"y se ha roto la nariz"
f2.Length	21
posini	10
this	{Examen3EV.Diccionario}

Para solucionarlo tenemos que volver a iniciar la variable posIni a 0 y indicarle para el F2 que empiece a contar despues un espacio en blanco

```
posini = 0;
posfin = f2.IndexOf(' '); // encontramos el primer espacio

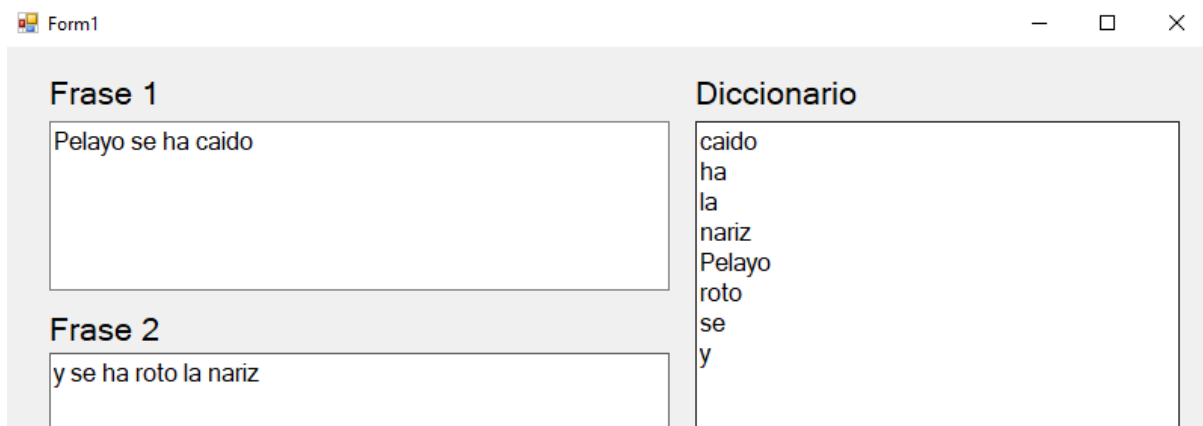
// segundo paso, analizar la segunda cadena
if (f2==null || f2.Length==0) return -2;

while (posfin!=-1) {
    word=f2.Substring(posini,posfin-posini);
    if (word.Length>0 && !words.Contains(word)) {
        words.Add(word);
        wordcount++;
    }
    posini = posfin + 1;
    posfin = f2.IndexOf(' ', posini);
}

// añadimos la última palabra
word = f2.Substring(posini, f2.Length - posini);
words.Add(word);
wordcount++;

// tercer paso, Ordenar las palabras
```

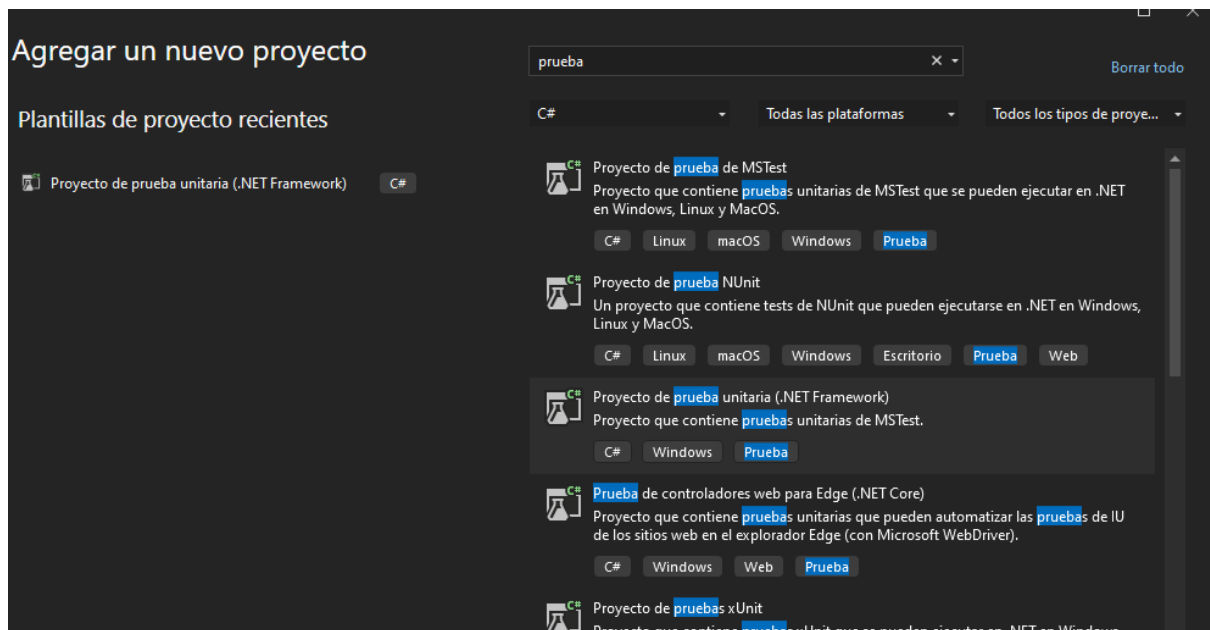
Y ya nos funciona correctamente el programa



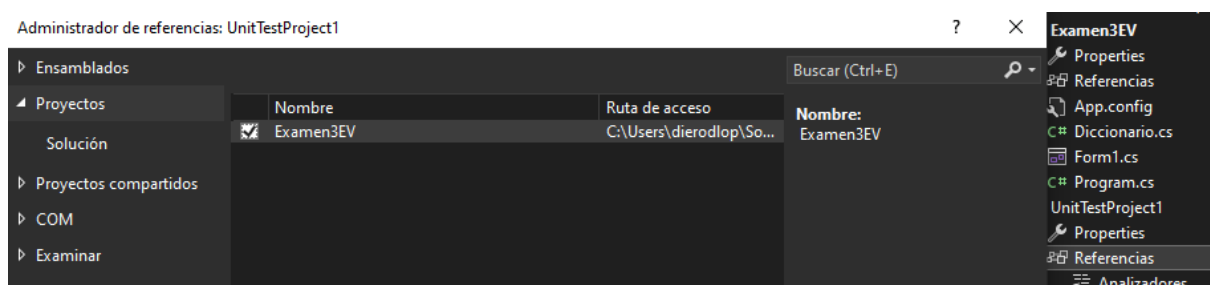
Frase 1	Frase 2	Diccionario
Pelayo se ha caido	y se ha roto la nariz	caido ha la nariz Pelayo roto se y

Apartado D

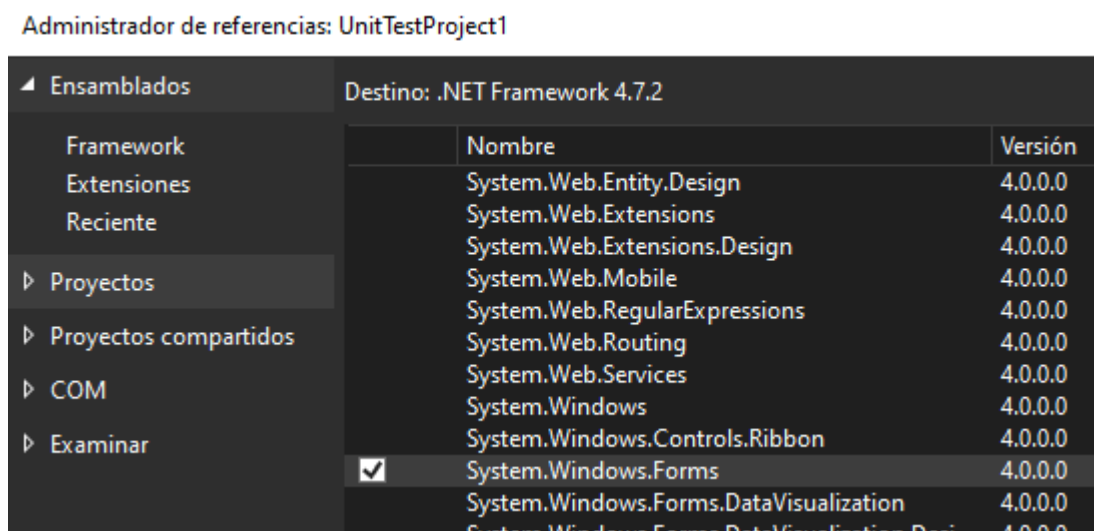
En primer lugar creamos el formulario de pruebas Unitarias llenando en visual studio al menu superior Archivo > Agregar > Nuevo Proyecto, seleccionamos la plantilla Proyecto de prueba Unitaria (.Net Framework) y lo creamos



Ahora en el Explorador de soluciones de visual studio nos vamos a las preferencias del proyecto de pruebas unitarias y modificamos en la pantalla que nos aparece el apartado proyecto para incluirlo en las pruebas



Ahora también tenemos que añadirle el tipo de ensamblado para windows Form



Y ahora ya podemos añadir nuestras pruebas unitarias

Primero añadimos los string con los textos y después las devoluciones que tiene que hacer, una con el resultado correcto y otro con el fallo

```
7 using System.Windows.Forms;
8
9 [TestClass]
10 public class UnitTest1
11 {
12     [TestMethod]
13     [DataRow(0)]
14     [DataRow(-1)]
15     public void IncluirFrases(int valor)
16     {
17         string Frase1, Frase2;
18         int devuelto;
19
20         Frase1 = "Pelayo se ha caido";
21         Frase2 = "y se ha roto la nariz";
22
23         Diccionario miApp = new Diccionario();
24
25         devuelto = miApp.analizar(Frase1, Frase2);
26
27         Assert.AreEqual(devuelto, valor);
28     }
29 }
30
31
```

Serie de pruebas finalizada: 2 prue

Prueba

- ✗ UnitTestProject1 (2)
 - ✗ UnitTestProject1 (2)
 - ✗ UnitTest1 (2)
 - ✗ IncluirFrases (2)
 - ✓ IncluirFrases (0)
 - ✗ IncluirFrases (-1)

Apartado B

Metodos demasiado largos, desde la linea 33 a la linea 77, se precisa estracion de codigo

Extraccion de codigo, linea desde la 37 a la 49, se puede utilizar visual studio en el menu superior clicamos Editar>Refactorizar>Extraer Codigo

```

wordcount = 0;
// primer paso, analizar la primera cadena
if (f1 == null || f1.Length == 0) return -1;

int posini, posfin;
ProcesadoPrimeraFrase(f1, out word, out posini, out posfin);
words.Add(word);
wordcount++;

```

Duplicidad en el código, eliminar código duplicado. Líneas de 41 a 42 y repetido en la línea 63 y 64

```

39      int posini, posfin;
40      ProcesadoPrimeraFrase(f1, out word, out posini, out posfin);
41
42      posini = 0;
43      posfin = f2.IndexOf(' '); // encontramos el primer espacio
44
45      // segundo paso, analizar la segunda cadena
46      if (f2 == null || f2.Length == 0) return -2;
47
48      while (posfin != -1)
49      {

```

Separación de las operaciones con los resultados, añadir {}. Línea 46

```

44
45      // segundo paso, analizar la segunda cadena
46      if (f2 == null || f2.Length == 0)
47      {
48          return -2;
49      }
50

```

Apartado A

Nombre de los métodos en estilo Pascal, Línea 31

```

30      //
31      0 referencias
32      public int Analizar(String f1, String f2)
33      {
34          String word;
35          words.Clear();

```

Espacios en blanco entre declaraciones, Línea 33, 34 y 35

```
public void Analizar(String  
{  
    String word;  
    words.Clear();  
    wordcount = 0;
```

Separación en la simbología, líneas 19 y 20

```
17 public Diccionario()  
18 {  
19     words = new List<String>();  
20     wordcount = 0;  
21 }  
22  
23
```