



Las Americas Institute of Technology

Matricula:

2023-0224

Nombre:

Jaime Antonio Casanova

Materia:

Estructura de Datos

Profesor:

Francis Ramirez

Fecha:

01/07/2025

Practica sobre recursividad

- 1) Dado un número natural n , se requiere determinar la suma de los dígitos que lo componen. Para ello, se solicita el desarrollo de dos enfoques algorítmicos: uno que utilice técnicas recursivas y otro que emplee una solución de tipo iterativa.

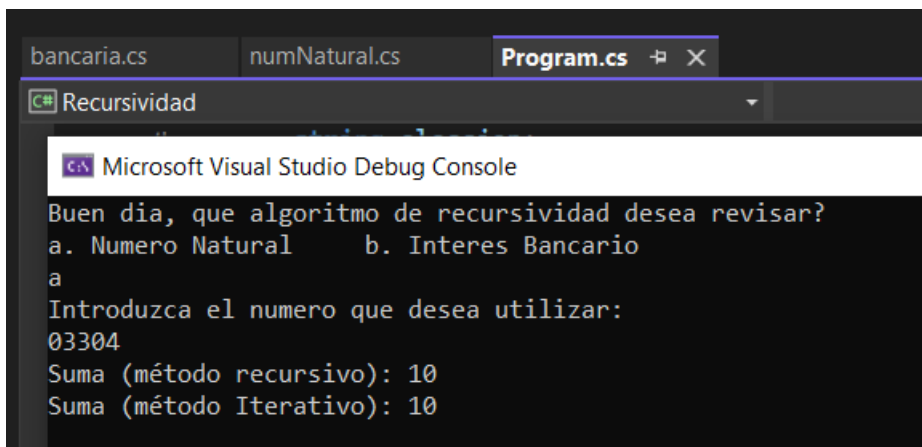
El ejercicio plantea el desafío de calcular la suma de todos los dígitos que componen un número natural mediante dos enfoques algorítmicos distintos: recursivo e iterativo. La solución recursiva se fundamenta en el principio de dividir el problema en subproblemas más pequeños, estableciendo como caso base cuando el número tiene un solo dígito ($n < 10$), momento en el cual se retorna el número mismo.

En el caso recursivo, se extrae el último dígito mediante la operación módulo 10 ($n \% 10$) y se suma recursivamente con el resultado de aplicar la misma función al resto del número obtenido por división entera entre 10 ($n / 10$), creando así una cadena de llamadas que se resuelve desde el caso base hacia arriba.

Por otro lado, la solución iterativa emplea un bucle while que continúa mientras el número sea mayor que cero, extrayendo en cada iteración el último dígito con la operación módulo 10, acumulándolo en una variable suma, y reduciendo el número mediante división entera entre 10 hasta procesarlo completamente.

Ambos enfoques requieren el uso de tipos de datos enteros (int) para manejar correctamente los números naturales y evitar los errores de precisión que ocurren con tipos decimales.

Ejecución:



```
bancaria.cs  numNatural.cs  Program.cs  ↗ ✕
C# Recursividad
Microsoft Visual Studio Debug Console
Buen día, que algoritmo de recursividad desea revisar?
a. Numero Natural    b. Interes Bancario
a
Introduzca el numero que desea utilizar:
03304
Suma (método recursivo): 10
Suma (método Iterativo): 10
```

- 2) Se ha depositado en una institución bancaria un monto de capital m , el cual genera un interés anual del $X\%$. El problema consiste en determinar cuál será el capital acumulado al cabo de n años, considerando la capitalización anual de los intereses. Se solicita desarrollar un subprograma recursivo que resuelva este problema, identificando claramente los estados básico y recursivo.

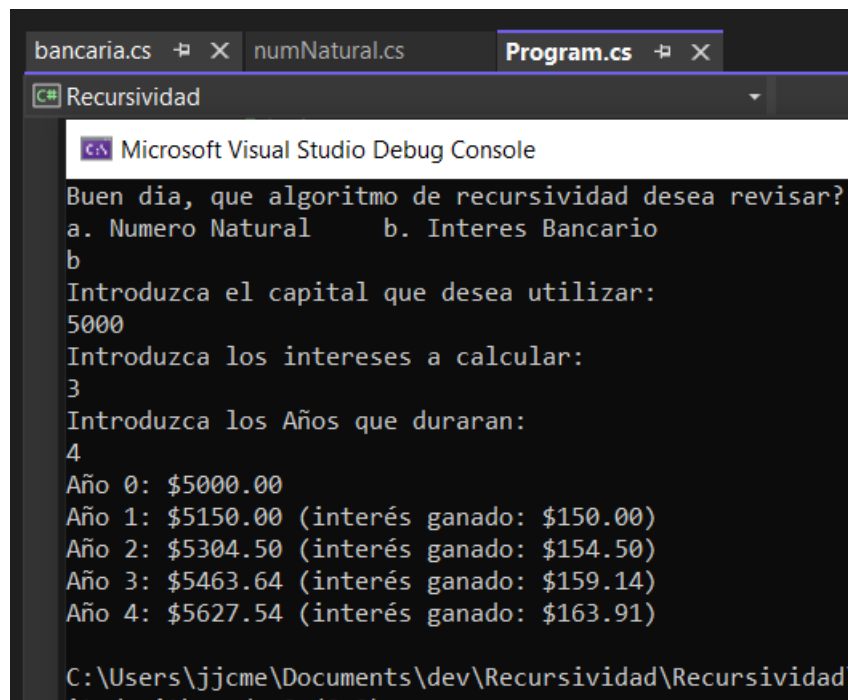
El ejercicio plantea calcular el capital acumulado después de depositar un monto inicial M en una institución bancaria que ofrece un interés anual del $X\%$, con capitalización anual durante N años.

Para resolverlo, se debe desarrollar un subprograma recursivo que determine el monto final considerando la acumulación de intereses año tras año. En la solución, el estado básico de la recursión corresponde al caso cuando $n=0$, es decir, no han pasado años y el capital acumulado es simplemente el monto inicial m .

El estado recursivo se define para $n>0$, donde el capital acumulado después de N años es igual al capital acumulado después de $n-1$ años multiplicado por $(1+X/100)$, reflejando la capitalización anual de los intereses. De esta forma, la función recursiva calcula progresivamente el valor final del capital con interés compuesto sobre el período indicado.

Además, se añade una función aparte, la cual se utilizara para iterar cada uso de la recursividad y poder generar un desglose de como evoluciono el capital.

Ejecución:



```
bancaria.cs  numNatural.cs  Program.cs
C# Recursividad
Microsoft Visual Studio Debug Console
Buen día, que algoritmo de recursividad desea revisar?
a. Numero Natural      b. Interes Bancario
b
Introduzca el capital que desea utilizar:
5000
Introduzca los intereses a calcular:
3
Introduzca los Años que duraran:
4
Año 0: $5000.00
Año 1: $5150.00 (interés ganado: $150.00)
Año 2: $5304.50 (interés ganado: $154.50)
Año 3: $5463.64 (interés ganado: $159.14)
Año 4: $5627.54 (interés ganado: $163.91)

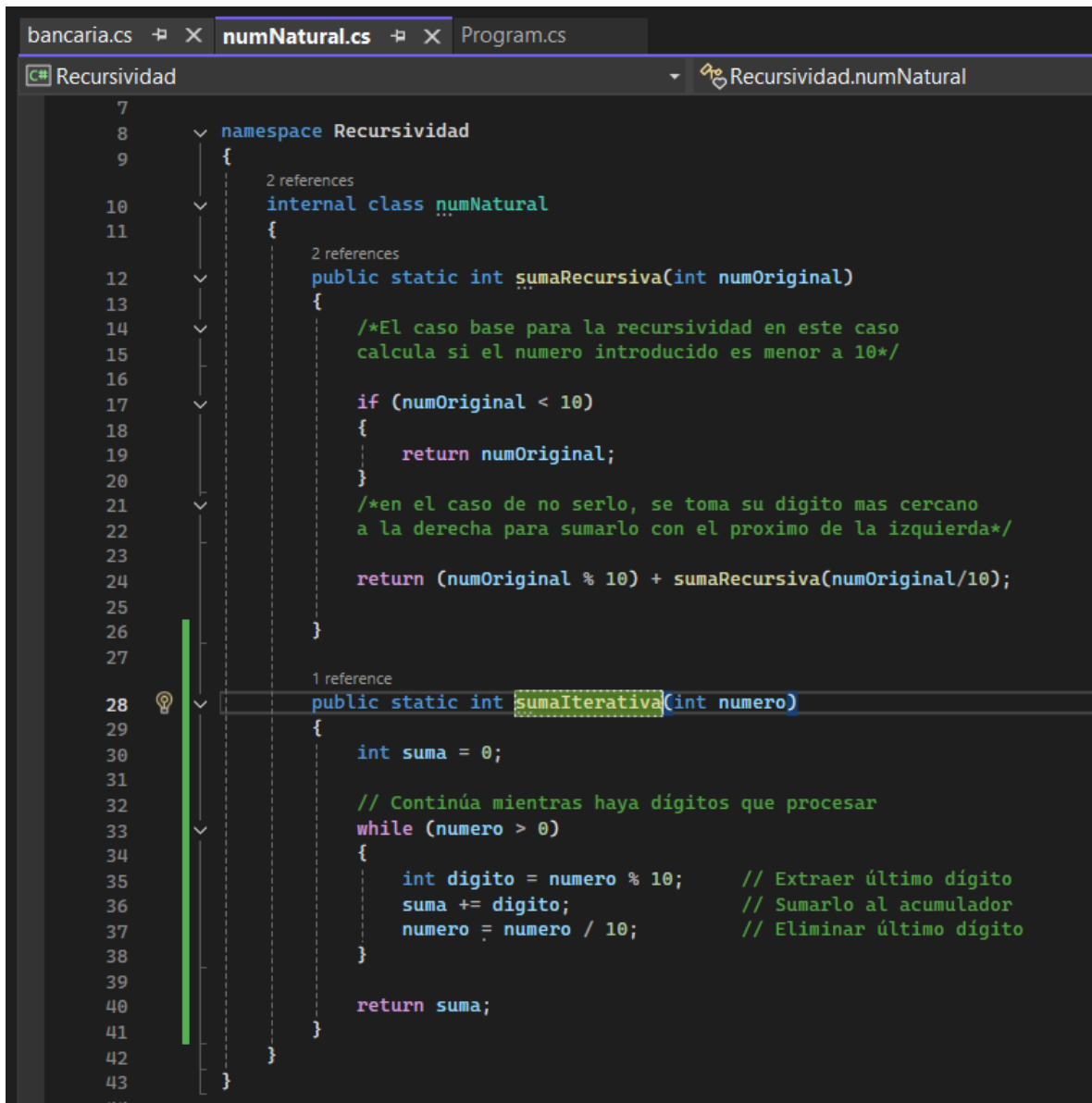
C:\Users\jjcme\Documents\dev\Recursividad\Recursividad\
it...ith...nd...0...0)
```

Código Fuente:

Selección de ejercicio:

```
bancaria.cs  numNatural.cs  Program.cs  X
Recursividad
1  using Recursividad;
2
3  //Inicializar string para la eleccion de el usuario y proceder a tomar la informacion
4  string eleccion;
5
6  Console.WriteLine("Buen dia, que algoritmo de recursividad desea revisar?");
7  Console.WriteLine("a. Numero Natural      b. Interes Bancario");
8  eleccion = Convert.ToString(Console.ReadLine());
9
10
11 //Condicional if para elegir entre uno de los dos algoritmos de recursividad
12 if (eleccion == "a")
13 {
14     //Inicializando la variable para el input de el usuario y convertirlo a int como natural entero
15     int numOriginal;
16     Console.WriteLine("Introduzca el numero que desea utilizar:");
17     numOriginal = Convert.ToInt32(Console.ReadLine());
18
19     //Confirmar que el numero utilizado por el usuario sea mayor que 0,
20     //siendo que los numeros naturales son siempre positivos enteros
21     if (numOriginal > 0)
22     {
23         int resultadoRecursivo = numNatural.sumaRecursiva(numOriginal);
24         Console.WriteLine($"Suma (método recursivo): {resultadoRecursivo}");
25     }
26     else //Arrojar error en caso de que el input no sea mayor a cero
27     {
28         Console.WriteLine("Error: Ingrese un número natural válido (entero positivo)");
29     }
30 }
31 else if (eleccion == "b")
32 {
33     //Inicializando las variables que seran utilizadas en la funcion
34     double capital;
35     double intereses;
36     int cantAños;
37
38     //Obteniendo el input de el usuario
39     Console.WriteLine("Introduzca el capital que desea utilizar:");
40     capital = Convert.ToDouble(Console.ReadLine());
41
42     Console.WriteLine("Introduzca los intereses a calcular:");
43     intereses = Convert.ToDouble(Console.ReadLine());
44
45     Console.WriteLine("Introduzca los Años que duraran:");
46     cantAños = Convert.ToInt32(Console.ReadLine());
47
48     //llamando a la funcion de calculo y luego a la funcion de desglose
49     bancaria.interesAnual(capital, intereses, cantAños);
50     bancaria.desglose(capital, intereses, cantAños);
51
52 } else //En caso de que el usuario no introduzca 'a' o 'b' al inicio se ejecuta error
53 {
54     Console.WriteLine("Error. Ingrese una opcion valida.");
55 }
56
```

Clase Ejercicio1:



```
7
8 namespace Recursividad
9 {
10     2 references
11     internal class numNatural
12     {
13         2 references
14         public static int sumaRecursiva(int numOriginal)
15         {
16             /*El caso base para la recursividad en este caso
17             calcula si el numero introducido es menor a 10*/
18
19             if (numOriginal < 10)
20             {
21                 return numOriginal;
22             }
23             /*en el caso de no serlo, se toma su digito mas cercano
24             a la derecha para sumarlo con el proximo de la izquierda*/
25
26             return (numOriginal % 10) + sumaRecursiva(numOriginal/10);
27         }
28
29         1 reference
30         public static int sumaIterativa(int numero)
31         {
32             int suma = 0;
33
34             // Continúa mientras haya dígitos que procesar
35             while (numero > 0)
36             {
37                 int digito = numero % 10; // Extraer último dígito
38                 suma += digito;           // Sumarlo al acumulador
39                 numero = numero / 10;     // Eliminar último dígito
40             }
41
42             return suma;
43         }
44     }
45 }
```

Clase Ejercicio2:

```
10 internal class bancaria
11 {
12     2 references
13     public static double interesAnual(double capital, double intereses, int cantAño)
14     {
15         4 references
16         /*Nuestro caso base para este algoritmo es si la cantidad de años es igual a 0,
17         de no serlo, procedemos con el calculo anual*/
18         if (cantAño == 0)
19         {
20             return capital;
21         }
22
23         /*Cada vez que esta linea se ejecuta, reduce la cantidad de años en la funcion,
24         una vez esta llega a 0, el caso base hace que la funcion vuelva a correr calculando
25         los intereses multiplicandolos con el capital, hasta llegar de nuevo a los parametros del caso inicial*/
26         return interesAnual(capital, intereses, cantAño - 1) * (1 + intereses / 100);
27     }
28
29     1 reference
30     public static void desglose(double capital, double intereses, int cantAño)
31     {
32         //Escribimos los valores iniciales para muestra
33         Console.WriteLine($"Año 0: ${capital:F2}");
34
35         //Iteramos cada vez que pasa un año para calcular el desglose por año
36         for (int i = 1; i <= cantAño; i++)
37         {
38             /*Utilizamos el algoritmo recursivo para poder editar estas variables iterables,
39             esto nos permite escribir una linea cada vez que se utiliza el algoritmo
40             recursivo, para ser capaces de ver como cambia en detalle*/
41             double capitalAnterior = interesAnual(capital, intereses, i - 1);
42             double capitalActual = interesAnual(capital, intereses, i);
43             double interesGanado = capitalActual - capitalAnterior;
44
45             //El :F2 marca el formato de decimales para que no se puedan exceder de 2 digitos
46             Console.WriteLine($"Año {i}: ${capitalActual:F2} (interés ganado: ${interesGanado:F2})");
47         }
48     }
49 }
```