Seminari

SeT8-1 Proves

(Grup 3A)

Tema 8.
Proves **(Camí Bàsic)**

(Grup 3A)

Enginyeria del Programari
ETS Enginyeria Informàtica
DSIC – UPV

Curs 2024-2025

Camí Bàsic

1. **Aplicar la tècnica del Camí Bàsic per a construir els casos de prova.**

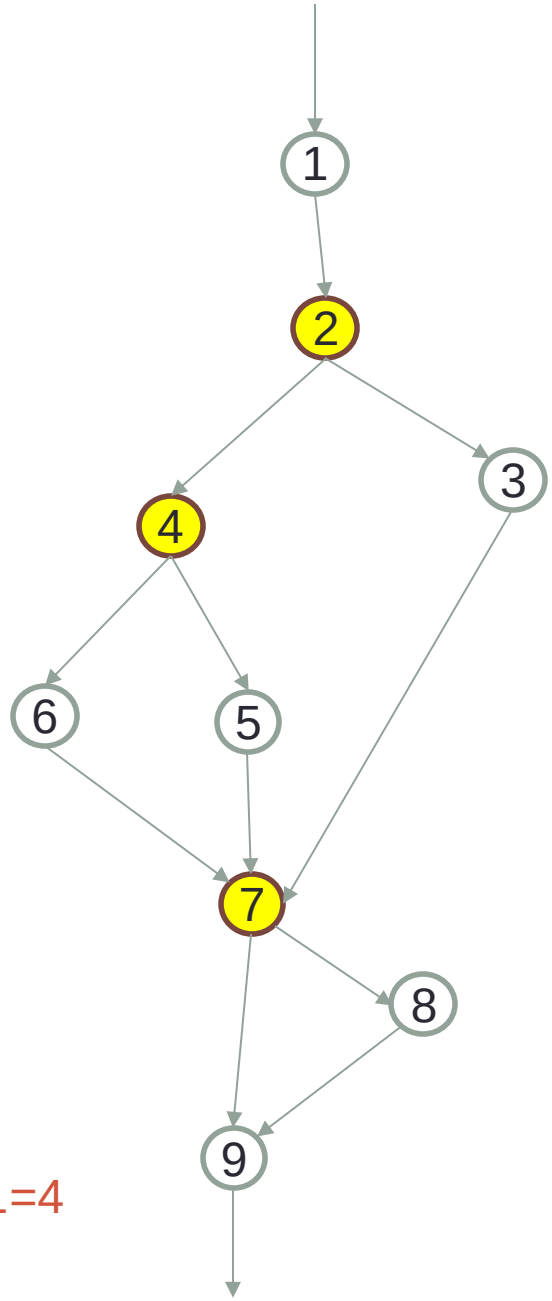
```
INICIO
Leer_Nota_de_Teoría (NT);
Leer_Nota_de_Prácticas (NP);
Leer_Nota_de_Trabajos (NTR);
SI NP = No Apto
  ENTONCES NF = 4;
SINO
  SI NT<4.5
ENTONCES NF=NT
  SINO NF = NT + NTR;
     FINSI
  SI NF>10
   ENTONCES NF = Matricula de Honor;
  FINSI
FINSI
FIN
```

1

```
INICIO
Leer_Nota_de_Teoría (NT);
Leer_Nota_de_Prácticas (NP);      1
Leer_Nota_de_Trabajos (NTR);
SI NP = No Apto   2
  ENTONCES NF = 4;    3
SINO
  SI NT<4.5   4
ENTONCES NF=NT      5
  SINO NF = NT + NTR;   6
     FINSI
  SI NF>10   7
    ENTONCES NF = Matricula de Honor;   8
  FINSI
FINSI
FIN   9
```

1

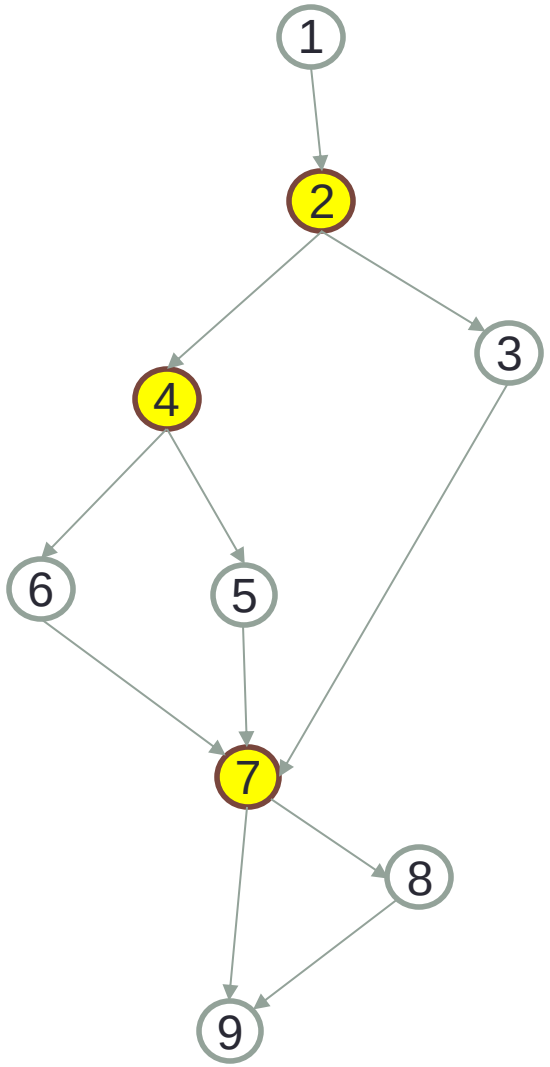
```
INICIO
Leer_Nota_de_Teoría (NT);
Leer_Nota_de_Prácticas (NP);      1
Leer_Nota_de_Trabajos (NTR);
SI NP = No Apto                   2
  ENTONCES NF = 4;                3
SINO
  SI NT<4.5                       4
ENTONCES NF=NT                    5
  SINO NF = NT + NTR;             6
     FINSI
  SI NF>10                        7
    ENTONCES NF = Matricula de Honor;   8
  FINSI
FINSI
FIN                               9
```



V(G)= 4
Áreas = 4
Nodos Predicado = 3 → 3+1=4
Nodos = 9 → 11-9+2 =4
Aristas = 11

1 Casos de Prova

Cami	Entrada	Eixida
{1,2,3, 7, 9}	NT= 5 NP= No Apto NTR=5	NF= 4
{1,2,3,7, 8, 9}	Cami impossible	
{1,2,4, 5, 7, 9}	NT= 4 NP= Apto NTR=6	NF=4
{1,2,4,6, 7, 9}	NT=5 NP= Apto NTR=4	NF= 9



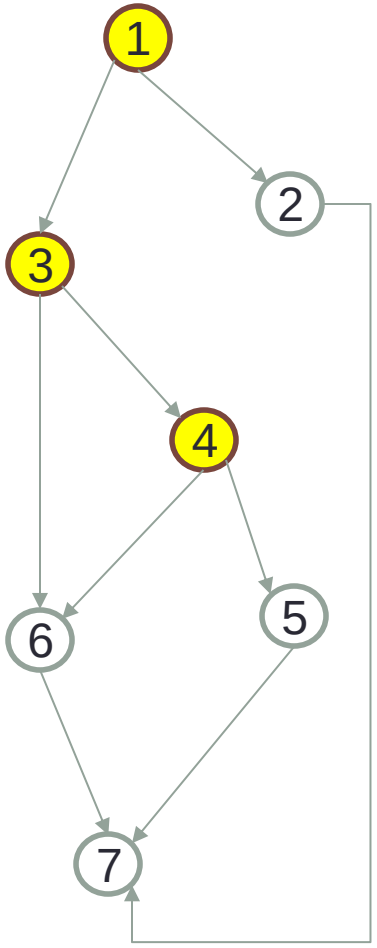
Camí Bàsic

2. Aplicar la tècnica del Camí Bàsic per a construir els casos de prova.

```
typedef struct {int dia,mes} Fecha;

int validar_no_caducada (Fecha t, Fecha h)
/* Entrada: t, fecha de la tarjeta;
            h, fecha del día de hoy
   Salida: 1 si la fecha de la tarjeta es mayor o igual que hoy
           0 en otro caso
*/
{
  if (t.mes<h.mes) return 0;
  else if (t.mes==h.mes && t.dia<h.dia) return 0;
       else return 1;
}
```

2

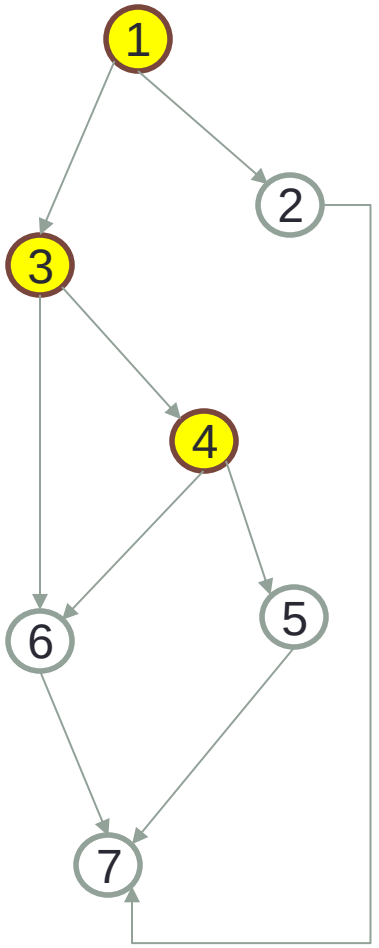
```
typedef struct {int dia,mes} Fecha;

int validar_no_caducada (Fecha t, Fecha h)
/* Entrada: t, fecha de la tarjeta;
            h, fecha del día de hoy
   Salida: 1 si la fecha de la tarjeta
           es mayor o igual que hoy
           0 en otro caso
*/
{
  if (t.mes<h.mes) return 0;
else if (t.mes==h.mes && t.dia<h.dia) return 0;

   else return 1;
}
```



V(G)= 4
Áreas = 4
Nodos Predicado = 3 → 3+1=4
Nodos = 7 → 9-7+2 =4
Aristas =9

2 Casos de Prova

Cami	Entrada	Eixida
{1,2,7}	Fecha tarjeta=01/08 Fecha hoy=05/11	0
{1,3,4,5,7}	Fecha tarjeta=04/11 Fecha hoy=05/11	0
{1,3,4,6,7}	Fecha tarjeta=05/11 Fecha hoy=05/11	1
{1,3,6,7}	Fecha tarjeta=01/12 Fecha hoy=05/11	1



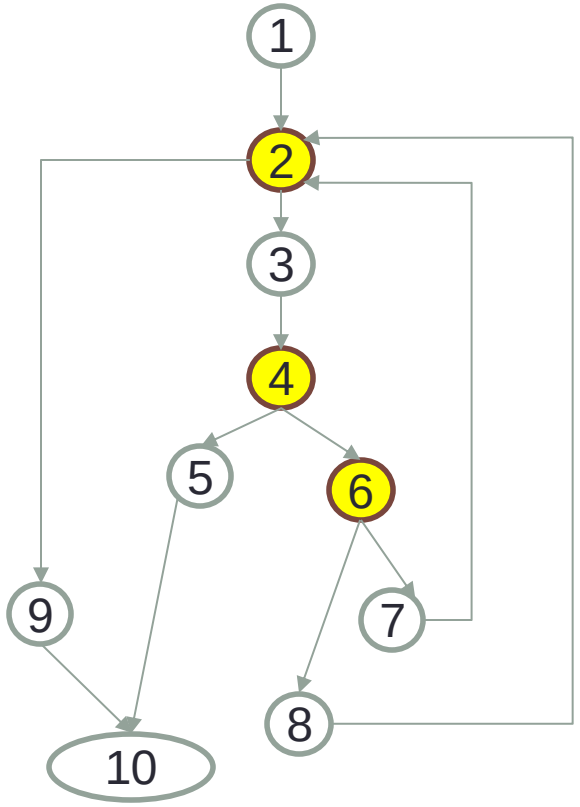
Camí bàsic

- **3. Aplicar la tècnica del Camí Bàsic per a construir els casos de prova.**

```
static public int search(char c, char []v)
        {
            int a, z, m;
            a = 0;
            z = v.Length - 1;
            while (a <= z)
            {
                m = (a + z) / 2;
                if (v[m] == c) {
                    return 1;
                }
                else if(v[m] < c)
                {
                    a = m + 1;
                }
                else
                {
                    z = m - 1;
                }
            }
            return 0;
        }
```

3

```
static public int search(char c, char []v)
        {
            int a, z, m;
1           a = 0;
            z = v.Length - 1;
2           while (a <= z)
            {
                m = (a + z) / 2;        3
4               if (v[m] == c) {
                    return 1;           5
                }
6               else if(v[m] < c)
                {
                    a = m + 1;          7
                }
                else
                {
                    z = m - 1;          8
                }
            }
            return 0;                   9
10      }
```



V(G)= 4

Áreas = 4

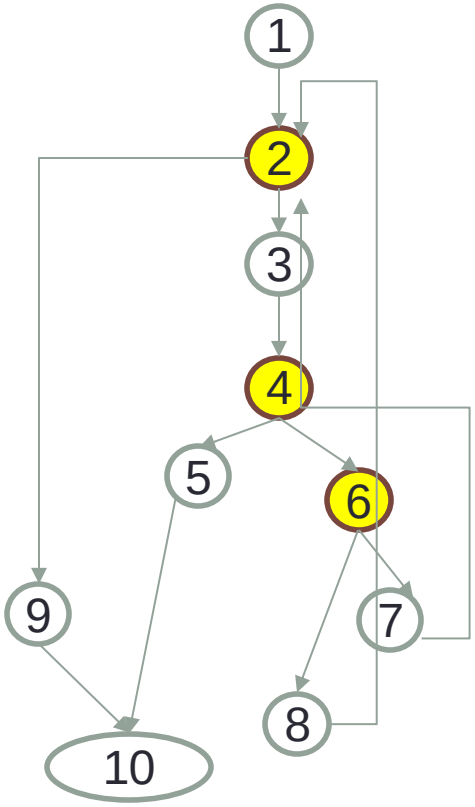
Nodos Predicado = 3 → 3+1=4

Nodos = 10 → 12-10+2 =4

Aristas = 12

3 Casos de Prova

Path	Input	Output
{1,2,9,10}	V="" c='a'	0
{1,2,3,4,5,10}	V="a" c='a'	1
{1,2,3,4,6,7,2,9,10}	V="a" c='b'	0
{1,2,3,4,6,8,2,9,10}	V="b" c='a'	0

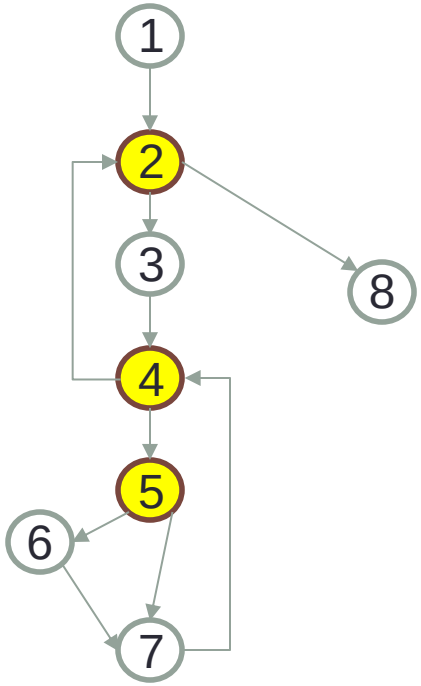


Camí Bàsic

```
static public void sort(int[] testArray)
        {
            int tempValue;
            int i = 0;
            bool isSwapped = true;
            while (isSwapped)
            {   isSwapped = false;
                i++;
                Console.Out.WriteLine("Before "+i+" iteration :");
                Console.Out.WriteLine("");
                for (int j = 0; j < testArray.Length - i; j++)
                {
                    if (testArray[j] > testArray[j + 1])
                    {
                        tempValue = testArray[j];
                        testArray[j] = testArray[j + 1];
                        testArray[j + 1] = tempValue;
                        isSwapped = true;
                    }
                }
            }

        }
```

4

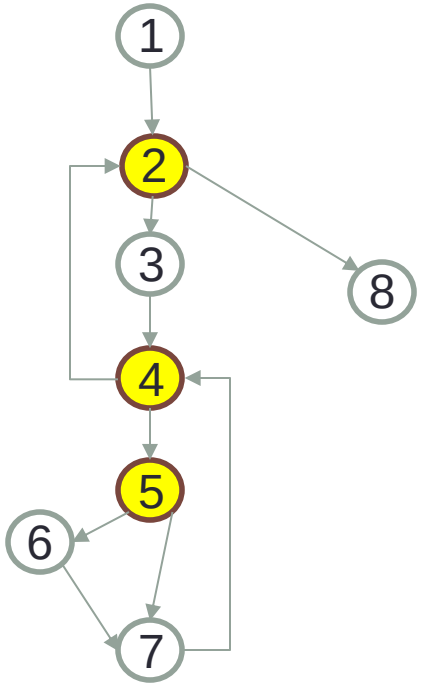
```
static public void sort(int[] testArray)
{
   int tempValue;
1  int i = 0;
   bool isSwapped = true;
   while (isSwapped) 2
   {   isSwapped = false;
       i++;
    3  Console.Out.WriteLine("Before "+i+" iteration :");
       Console.Out.WriteLine("");
       for (int j = 0; j < testArr 4 Length - i; j++) 7
       {
             if (testArray[j] > testArray[j + 1]) 5
             {
                  tempValue = testArray[j];
               6  testArray[j] = testArray[j + 1];
                  testArray[j + 1] = tempValue;
                  isSwapped = true;
              }
        }
    }

8 }
```



V(G)= 4
Áreas = 4
Nodos Predicado = 3 → 3+1=4
Nodos = 8 → 10-8+2 =4
Aristas = 10

4 Casos de Prova

Path	Input	Output
{1,2,8} Cami impossible	No possible	No possible
{1,2,3,4,2,8}	[ ]	[]
{1,2,3,4,5,6,7,4,2,8}	[1,2]	[1,2]
{1,2,3,4,6,8,2,9,10}	[2,1]	[1,2]



Camí Bàsic

ValidateId és un mètode que retorna true si el NIF proporcionat és vàlid o false en cas contrari. ***ValidateId*** fa ús del mètode ***NIFLetter***, que retorna la lletra associada als dígits d'un NIF. Si no se li proporcionen 9 dígits, ***NIFLetter*** llança una excepció del tipus ArgumentException

```
public static bool ValidateId(string id)
{
 int validSize = 9;
 if (id.Length < validSize || id.Length > validSize)
   return false;
 //checking if the last digit is the correct letter
 string digitsId = id.Substring(0, id.Length - 1);
 try
 {
   char letter = NIFLetter(digitsId);
  if (id.EndsWith("" + letter))
   return true;
  else
   return false;

 }
 catch (ArgumentException)
  {
  return false;
    }
}
```

Camí Bàsic

ValidateId és un mètode que retorna true si el NIF proporcionat és vàlid o false en cas contrari. ***ValidateId*** fa ús del mètode ***NIFLetter***, que retorna la lletra associada als dígits d'un NIF. Si no se li proporcionen 9 dígits ***NIFLetter*** llança una excepció del tipus ArgumentException

```
public static bool ValidateId(string id)
{
 int validSize = 9;
 if (id.Length < validSize || id.Length > validSize)
   return false;
 //checking if the last digit is the correct letter
 string digitsId = id.Substring(0, id.Length - 1);
 try
 {
   char letter = NIFLetter(digitsId);
  if (id.EndsWith("" + letter))
   return true;
  else
   return false;

 }
 catch (ArgumentException)
 {
  return false;
   }
}
```

V(G) = 5 + 1 = 6