



UNIVERSIDADE DA CORUÑA

FACULDADE DE INFORMÁTICA
Departamento de Computación
Estructura de Datos e da Información

Práctica 0b: Recursividad

1. Comparar el rendimiento entre las implementaciones recursivas e iterativas del problema de la serie de Fibonacci.

```
program fibonacci;  
{Solucion recursiva}  
uses sysutils;  
var  
    pos: longint;  
  
function Fibo (pos:integer): longint;  
begin  
    if pos = 1  
    then Fibo := 1  
    else if pos = 2  
    then Fibo := 1  
    else Fibo := Fibo(pos -1) + Fibo (pos -2);  
end;  
  
begin  
    writeln ('Serie de Fibonacci: que posicion?');  
    readln (pos);  
    writeln('El tiempo actual es ', FormatDateTime('hh:nn:ss',Time));  
    writeln ('El resultado es ', Fibo(pos));  
    Writeln ('The current time is : ', FormatDateTime('hh:nn:ss',Time));  
end.
```

```
program fibonacci;  
{Solucion iterativa}  
uses sysutils;  
var  
    pos: integer;  
  
function Fibo (pos:integer): longint;  
var  
    sig, previo, actual: longint;  
    i: integer;  
begin  
    if (pos <=2)  
    then Fibo := 1  
    else begin  
        previo:= 1;  
        actual:= 1;  
        for i:=3 to pos do begin  
            sig:= previo+actual;  
            previo:= actual;
```

```

                actual:= sig;
            end;
        Fibo := actual;
    end;
end;

begin
    writeln ('Serie de Fibonacci: que posicion?');
    readln (pos);
    writeln('El tiempo actual es ', FormatDateTime('hh:nn:ss',Time));
    writeln ('El resultado es ', Fibo(pos));
    writeln('El tiempo actual es ', FormatDateTime('hh:nn:ss',Time));
end.

```

2. Obtener la suma de los dígitos de un número.

```

function sumarecursiva (n: integer): integer;
begin
    if n<=9
    then sumarecursiva := n
    else sumarecursiva := sumarecursiva (n div 10) + (n mod 10);
end;

function sumaiterativa (n: integer): integer;
var
    suma: integer;
begin
    suma := 0;
    while n>9 do begin
        suma := suma + (n mod 10);
        n := n div 10;
    end;
    suma := suma + n;
end;

```

3. Recursividad infinita.

```

program overflow;
type
    TArray = array[1..255] of string;
var
    p: ^TArray;
    i: longint;

procedure reserva;
begin
    i:= i + 1;
    writeln (i);
    new(p);
    {dispose(p);}
    reserva;
end;

begin
    i:= 1;
    reserva;
end.

```

4. Determinar si una palabra es o no es un palíndromo.

```
program palindromo;
var
  palabra: string;

function Espalindromo (pal: string;i,j: integer): boolean; begin
  if i>=j then Espalindromo := true
  else if pal[i] <> pal[j]
  then Espalindromo := false
  else Espalindromo := Espalindromo (pal, i+1, j-1);
end;

begin
  write('Intro palabra '); readln(palabra);
  if Espalindromo (palabra, 1, length(palabra))
  then writeln ('Es palidromo')
  else writeln ('No es palindromo');
end.
```