

Introducción a la Programación



**CODE
SPACE**
A C A D E M Y

2. Algoritmos, diagramas de flujos y programas

2.12 Recursividad

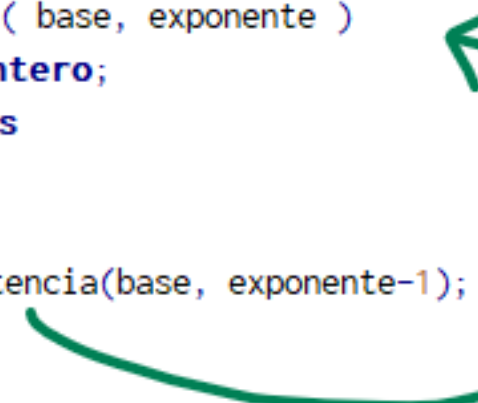
Definición

La recursividad o **recursión** consiste, en esencia, en que una función se llame a sí misma. Debemos controlar su última llamada para no caer en una recursión infinita, normalmente lo haremos devolviendo un valor fijo.

Ejemplos:

- Calcular el factorial de un número.
- Calcular la potencia de número.

```
1  Funcion resultado <- potencia ( base, exponente )
2      Definir resultado Como Entero;
3      Si exponente == 0 Entonces
4          resultado = 1;
5      SiNo
6          resultado = base * potencia(base, exponente-1);
7      FinSi
8  Fin Funcion
```



2. Algoritmos, diagramas de flujos y programas

2.13 Excepciones

2.13.1 Definición

Errores ocasionados durante la ejecución de un programa informático. Cuando ocurre cierto tipo de error, el sistema reacciona ejecutando un fragmento de código que resuelve la situación, por ejemplo retornando un mensaje de error o devolviendo un valor por defecto.

Su función es separar el código del manejo de errores de la lógica de aplicación del programa. En aquellos lenguajes que incluyen soporte para el manejo de excepciones.

2. Algoritmos, diagramas de flujos y programas

2.13 Excepciones

2.13.2 Capturar excepciones

try ... catch corresponde a un tipo de estructura de control con la que distribuir el flujo de un programa en función a posibles errores.

La primera parte, ***try***, ejecutará el código susceptible de contener errores. Si el error a controlar efectivamente ocurre, el flujo del programa parará en esa línea y seguirá en el bloque de ***catch***.

```
try {  
    // Código que puede contener errores  
} catch( error ) {  
    // Controla el error y actúa  
}
```

2. Algoritmos, diagramas de flujos y programas

2.13 Excepciones

2.13.3 Lanzar excepciones

También podemos lanzar una excepción cuando detectamos un comportamiento programado. Por ejemplo, si en una acción es necesario que el usuario esté logueado y no lo está, puedo lanzar una excepción.

La forma más sencilla para lanzar errores es utilizando `throw`. Este comando permite enviar al navegador un evento similar al que se produce cuando ocurre algún imprevisto o nos encontramos ante un tipo inesperado de datos. El lenguaje permite enviar todo tipo de elementos, incluyendo texto, números, valores booleanos o incluso objetos. Sin embargo, la opción más usual es enviar el objeto nativo `Error`:

```
throw new Error( "Something bad happened." );
```

Introducción a la Programación



**CODE
SPACE**
A C A D E M Y