



aprenderaprogramar.com

# Manejo de módulos y recursión en programación. Pseudocódigo. (CU00204A)

**Sección:** Cursos

**Categoría:** Curso Bases de la programación Nivel II

**Fecha revisión:** 2024

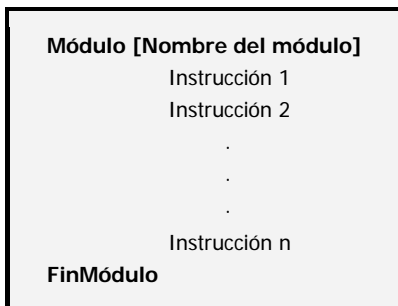
**Autor:** Mario R. Rancel

**Resumen:** Entrega n°3 del Curso Bases de la programación Nivel I

24

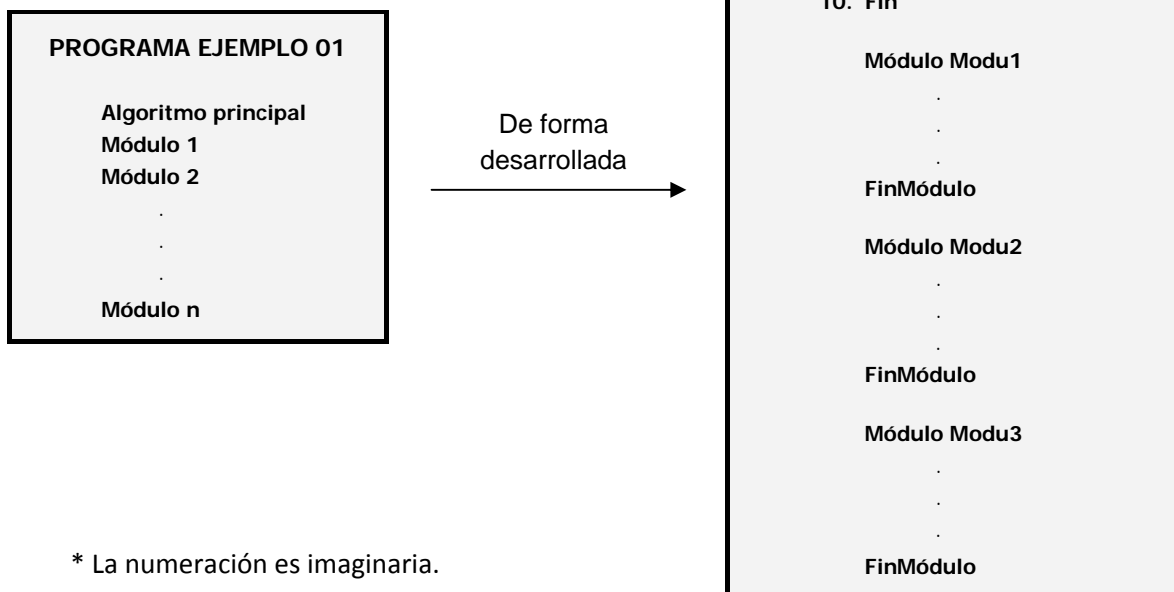
## INSERCIÓN DE UN MÓDULO EN UN PROGRAMA. DECLARACIÓN. INSTRUCCIÓN LLAMAR. RECURSIÓN

Para poder insertar un módulo en un programa o algoritmo hemos de declarar su existencia y contenido. Para ello escribiremos:



Para nombrar a un módulo se seguirán las mismas pautas que con las variables. Se buscará que los nombres sean interpretables y claros. Se evitará que sean similares entre sí o con las variables que existan en el programa.

La numeración de las líneas del módulo es independiente de la del algoritmo principal, pudiendo repetirse si se quiere. Se entiende que el módulo se encuentra dentro del programa pero no se ejecuta hasta que es llamado desde el algoritmo principal de acuerdo con esta estructura.



\* La numeración es imaginaria.

Hemos dicho que un módulo no se ejecuta hasta que es llamado a ejecutarse desde el algoritmo principal. Para ello utilizaremos la instrucción Llamar de acuerdo con esta sintaxis:

*Llamar [Nombre del Módulo]*

Cuando se llama a un módulo la ejecución del algoritmo principal se detiene y el flujo se desvía al módulo. Una vez se llega al fin del módulo, la ejecución en el algoritmo principal continúa por la sentencia inmediatamente posterior a la instrucción Llamar. Un módulo puede ser llamado cuantas veces se quiera. Es posible llamar a un módulo desde otro módulo: el efecto es el mismo al comentado anteriormente. La ejecución del módulo en curso se detiene prosiguiendo el flujo por el módulo que ha sido llamado. Una vez se termina de procesar el módulo llamado, se continúa en la instrucción inmediatamente posterior a la de llamada.

Si en un programa llamamos en distintos puntos a un módulo, éste realiza una serie de procesos sin necesidad de repetir el código. Nos supondrá por tanto, un ahorro considerable.

La llamada a un módulo desde sí mismo también es posible, dando lugar a un anidamiento o módulo recursivo. Como en el caso de los bucles, la repetición normalmente irá sujeta a la evaluación de una condición que evoluciona a medida que se producen recursiones hasta dar lugar a una salida, que va devolviendo el control al punto desde el cual se autollamó el módulo. El sistema equivale a la generación sucesiva de copias del módulo. Las autollamadas se deben evitar siempre que sea posible, pues dificultan el seguimiento del flujo de los programas y es habitual que si no se controlan bien den lugar a situaciones de bloqueo o bucles infinitos. No haremos un desarrollo teórico de la recursión aunque el usuario que lo desee puede dirigirse a otros cursos o artículos de [aprenderaprogramar.com](http://aprenderaprogramar.com) donde se aborda la recursión.

Veamos un ejemplo de aproximación al concepto de módulo:

**1. Inicio [PROGRAMA COMUNICADO ejemplo aprenderaprogramar.com]**

2. Mostrar "Comunicado de la empresa"
3. Llamar Saludo
4. Llamar Comunicado
5. Llamar Despedida

**6. Fin**

**Módulo Saludo**

1. Mostrar "Con motivo de la celebración el próximo día 5 del Día Mundial del Medioambiente la empresa saluda a todos los empleados y les agradece el compromiso con el cuidado de la naturaleza"

**FinMódulo**

**Módulo Comunicado**

1. Mostrar "Con motivo de dicha conmemoración está previsto realizar un acto de plantación de árboles en los jardines del edificio central el próximo día 5 a las 12 del mediodía al que están todos invitados"

**FinMódulo**

**Módulo Despedida**

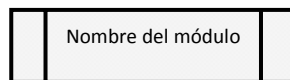
1. Mostrar "La empresa agradece su participación y les invita a sumarse al programa <<Empleados por una ciudad sostenible>>.  
Atentamente, El Director General"

**FinMódulo**

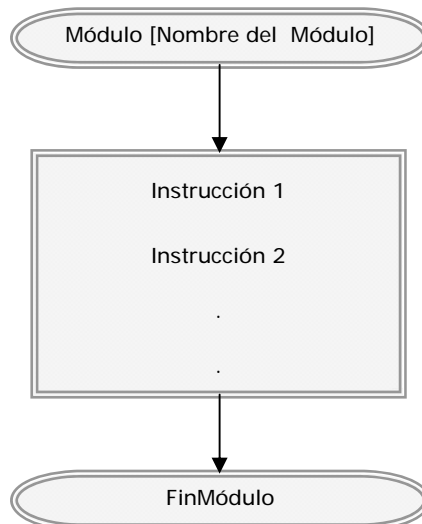
**Comentarios:** La ejecución es controlada por el algoritmo principal. Si existiera un módulo que no es llamado desde el algoritmo principal (ni a través de un módulo sí llamado) no se ejecutaría nunca.

Los módulos pueden situarse antes o después del algoritmo principal y seguir el orden de llamada o no. En este caso se han ordenado los módulos con Saludo → Comunicado → Despedida buscando un orden lógico.

La ordenación de módulos y algoritmo principal debe obedecer a la intención de facilitar la lectura y comprensión del programa, al igual que los nombres que se usen. La representación gráfica de un módulo en el diagrama de flujo la haremos con el símbolo:

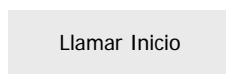


Si existen llamadas a módulos en un diagrama de flujo, habrá de existir, a continuación del diagrama del algoritmo principal, otros diagramas explicativos de cada módulo (un diagrama por módulo). La representación será:



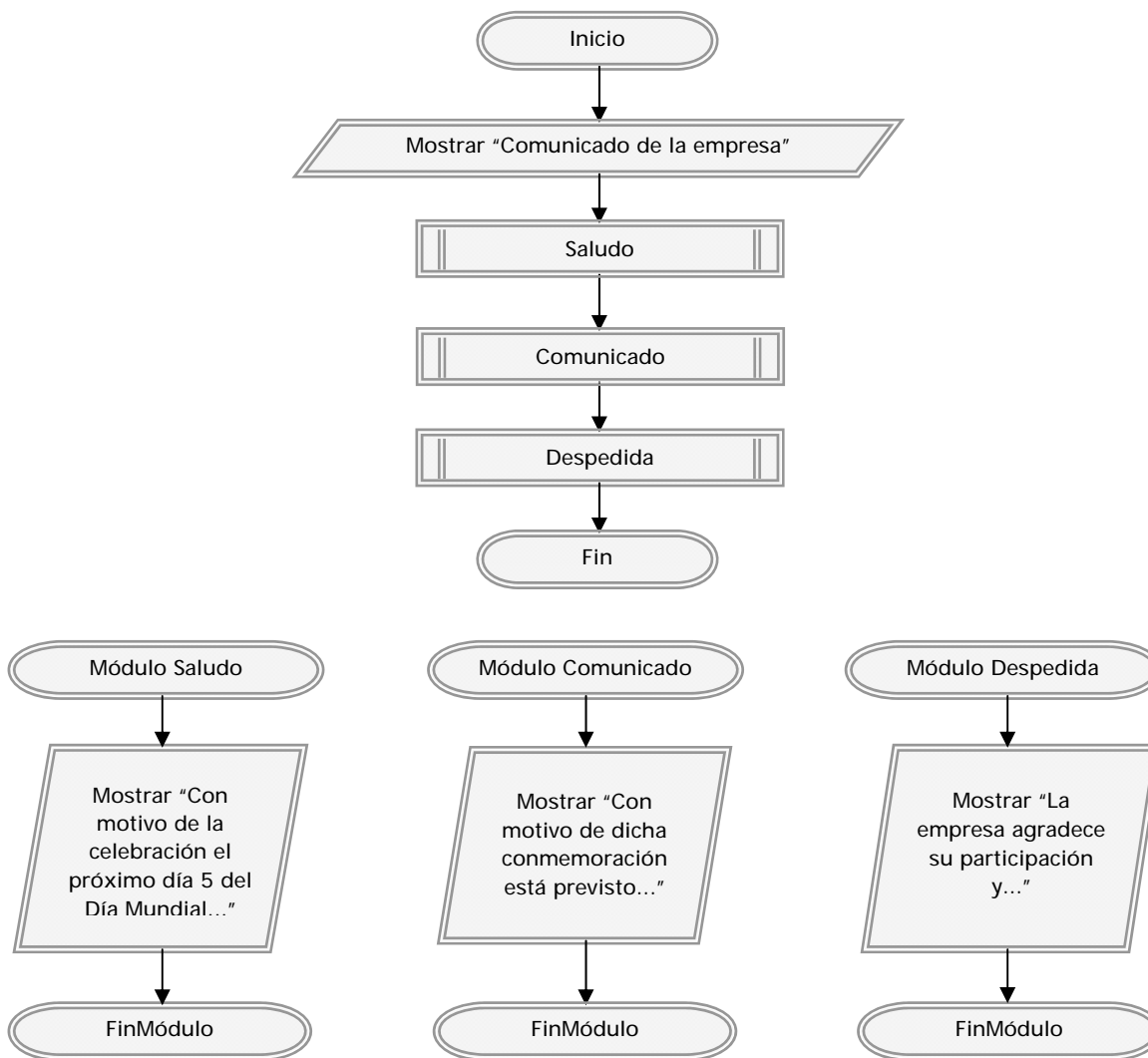
Obsérvese que un módulo es tal por cuanto se identifica así y se subordina al algoritmo principal, pero su estructura es la de un programa y se le podría hacer funcionar como tal.

El algoritmo principal puede llamarse a sí mismo o ser llamado desde otro módulo. Para ello escribiremos:



Llamar Inicio constituye en cualquier caso una autollamada debiendo tenerse en cuenta lo ya comentado al respecto.

La representación gráfica del ejemplo anterior sería la siguiente:



Aunque este ejemplo por su sencillez no da demasiado juego, sí podemos intuir que supone una potente herramienta de cara a conseguir una mejor organización de los programas largos y complejos y facilitar su comprensión. Procesos muy complejos pueden quedar resumidos en el nombre de un módulo.

**Próxima entrega: CU00205A**

**Acceso al curso completo** en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente: [http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=36&Itemid=60](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=36&Itemid=60)