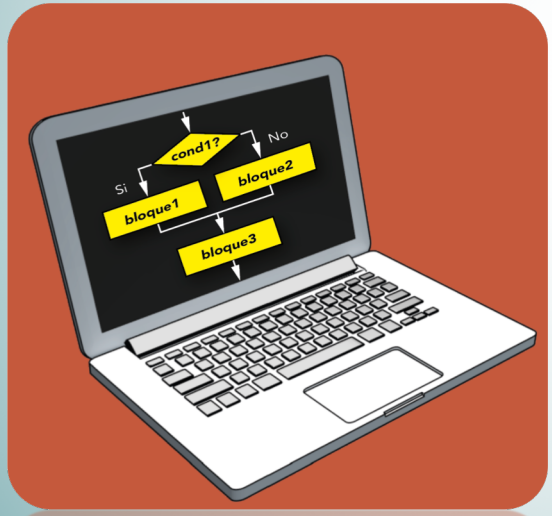
Fundamentos de Computación

Bloque I.

Introducción a la Informática. Sistemas Operativos. Hardware y Software

Lenguajes de programación. Representación de la Información



Sixto Herrera García

Departamento de Matemática Aplicada y Ciencias de la Computación

Fundamentos de Computación

Introducción a la Informática.
Sistemas Operativos. Hardware y Software

Conceptos Básicos

Informática: Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la **información** por medio de **ordenadores**.

Información: Conjunto de símbolos que representan hechos, objetos o ideas.

Ordenador: Es una máquina capaz de aceptar unos **datos de entrada**, efectuar con ellos operaciones lógicas y aritméticas y proporcionar la información resultante a través de un medio de salida; todo ello sin intervención de un operador humano y bajo el control de un conjunto de **instrucciones** previamente almacenado en el ordenador.

Datos: Conjuntos de símbolos utilizados para expresar o representar un valor numérico, un hecho, un objeto o una idea; en la forma adecuada para ser objeto de tratamiento por el ordenador.

Conceptos Básicos

Instrucción: Mandato elemental que se da a un ordenador a fin de que efectúe una determinada operación. Las instrucciones son indicaciones sencillas y no ambiguas.

Algoritmo: Conjunto de instrucciones combinadas de forma adecuada para resolver un determinado problema en una cantidad finita de tiempo.

Programa: es un algoritmo expresado en un lenguaje comprensible para un ordenador.

Conceptos Básicos

Instrucción: Mandato elemental que se da a un ordenador a fin de que efectúe una determinada operación. Las instrucciones son indicaciones sencillas y no ambiguas.

Algoritmo: Conjunto de instrucciones combinadas de forma adecuada para resolver un determinado problema en una cantidad finita de tiempo.

Programa: es un algoritmo expresado en un lenguaje comprensible para un ordenador.

Programación: es el proceso de comunicar al ordenador una secuencia de instrucciones que señalan las acciones que ejecuta éste. Para ello se utilizará un **Lenguaje de Programación**.

Componentes de un Ordenador (Hardware)

CPU: ejecución de programas. Unidad de control (producción de órdenes) y aritmético-lógica (ejecución de órdenes).

Memoria Principal: unidad de almacenamiento de órdenes, instrucciones, datos, resultados, etc.

Periféricos: unidades de entrada y salida. Intercambio de información entre el ordenador y el usuario/exterior.

Todas las componentes se encuentran interconectadas.

Componentes de un Ordenador (Hardware)

CPU: ejecución de programas. Unidad de control (producción de órdenes) y aritmético-lógica (ejecución de órdenes).

Memoria Principal: unidad de almacenamiento de órdenes, instrucciones, datos, resultados, etc.

Periféricos: unidades de entrada y salida. Intercambio de información entre el ordenador y el usuario/exterior.

Todas las componentes se encuentran interconectadas.

Software

Software: está compuesto por todos los programas necesarios para realizar con el ordenador el tratamiento de la información.

¿Qué software hay disponible en la Universidad de Cantabria?

Software

Software: está compuesto por todos los programas necesarios para realizar con el ordenador el tratamiento de la información.

¿Qué software hay disponible en la Universidad de Cantabria?

Sistema Operativo (S.O): es el conjunto de programas encargados de coordinar las tareas a ejecutar por el ordenador (gestión del espacio y acceso al sistema de ficheros, de los periféricos, etc.), optimizando su rendimiento.

Lenguajes de Programación: es un código o conjunto de códigos que permiten la comunicación con el ordenador y la descripción de algoritmos y/o funciones que será ejecutadas por éste.

Aplicaciones: son conjuntos de programas que realizan diferentes tareas comunes a diferentes usuarios. Por ejemplo, procesadores de texto, hojas de cálculo, bases de datos, multimedia, etc.

¿Cómo clasificarías el software instalado en la UC?

Tipos de ficheros vs Aplicaciones

Toma un fichero con cualquier extensión (pdf, doc, docx, txt, etc...) ¿Qué tipo de fichero es? ¿Con qué aplicación se explora?

Cambia la extensión del fichero, ¿qué ocurre?¿Cambia la aplicación con la que se explora?¿Qué componente controla ese cambio?

Realiza un par de cambios y trata de responder a las preguntas anteriores de cara a sacar una conclusión razonada.

Tipos de ficheros vs Aplicaciones

Toma un fichero con cualquier extensión (pdf, doc, docx, txt, etc...) ¿Qué tipo de fichero es? ¿Con qué aplicación se explora?

Cambia la extensión del fichero, ¿qué ocurre?¿Cambia la aplicación con la que se explora?¿Qué componente controla ese cambio?

Realiza un par de cambios y trata de responder a las preguntas anteriores de cara a sacar una conclusión razonada.

Se debe distinguir entre la naturaleza del fichero/documento/archivo y la aplicación o aplicaciones que el sistema operativo identifica o asigna para explorar éstos.

Programación

Programación: es el proceso de comunicar al ordenador una secuencia de instrucciones que señalan las acciones que ejecuta éste. Para ello se utilizará un **Lenguaje de Programación**.

Programar: El diseño de un programa presenta dos fases, una primera de **análisis** del problema a resolver/tarea a realizar y el desarrollo de un algoritmo, y una segunda fase de **realización de un programa** a partir del algoritmo desarrollado en el lenguaje de programa elegido.

Fase de depuración: En la mayoría de lenguajes actuales de medio o alto nivel se han desarrollado entornos de ***debugging*** que permite la ejecución de partes del código y la depuración de errores en el programa.

Metodologías de Programación

Programación: es el proceso de comunicar al ordenador una secuencia de instrucciones que señalan las acciones que ejecuta éste. Para ello se utilizará un **Lenguaje de Programación**.

Podemos distinguir las siguientes metodologías de programación:

Programación estructurada: Programación basada en las estructuras básicas: secuencial, condicional y cíclica.

Programación modular: Dividir el problema en sub-tareas simples y abordables en forma de: funciones, subrutinas, ...

Programación orientada a objetos: Definición de “objetos” dentro del programa, los cuales pertenecen a “clases” que definen el tipo de manipulaciones que se puede aplicar a estos “objetos”. Programación aun mas intuitiva que la modular. No entraremos en mas detalles ya que Octave no es un lenguaje orientado a objetos.

Metodologías de Programación

Programación: es el proceso de comunicar al ordenador una secuencia de instrucciones que señalan las acciones que ejecuta éste. Para ello se utilizará un **Lenguaje de Programación**.

Podemos distinguir las siguientes metodologías de programación:

Programación estructurada: Programación basada en las estructuras básicas: secuencial, condicional y cíclica.

Programación modular: Dividir el problema en sub-tareas simples y abordables en forma de: funciones, subrutinas, ...

Desarrollo de Algoritmos

“Top-down”: Se ataca el problema a grandes rasgos al principio, planteando las tareas que va a ser necesarias y despedazándolas hasta llegar a tareas simples.

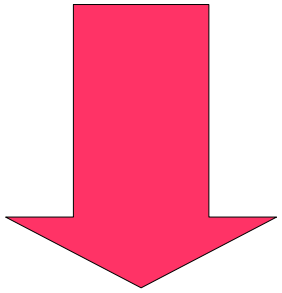
Desventaja: El programa no “funciona” hasta una fase muy avanzada.

“Bottom-up”: Se van haciendo pequeñas tareas que se pueden depurar independientemente y al final componen el programa completo.

Desventaja: Perdida de la perspectiva global del programa. Quizá se programen tareas que finalmente no se adecuan bien y tienen que ser reprogramadas.

Desarrollo de Algoritmos

“Top-down”: Se ataca el problema a grandes rasgos al principio, planteando las tareas que va a ser necesarias y despedazándolas hasta llegar a tareas simples.



“Bottom-up”: Se van haciendo pequeñas tareas que se pueden depurar independientemente y al final componen el programa completo.

Programación: Fase de Análisis

En la fase de Análisis debemos entender el problema, para poder esquematizar los paso necesarios para resolverlo. Para ello tenemos dos herramientas básicas:

- Diagramas de Flujo.
- Pseudo-Código.

Diagramas de Flujo

En la fase de Análisis debemos entender el problema, para poder esquematizar los paso necesarios para resolverlo. Para ello tenemos dos herramientas básicas:

- **Diagramas de Flujo.**
- Pseudo-Código.

Son una representación gráfica de la secuencia de operaciones necesarias para que un sistema informático resuelva un problema dado.

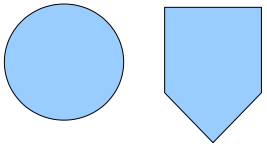
Si lo que se pretende está claro, escribir el programa se reduce a traducir el diagrama de flujo del castellano a Octave.

Se utilizan normalmente para describir con diferente grado de detalle la lógica de un problema.

En problemas complejos, describir todo detalladamente puede ser tedioso (e inútil, en el caso de programadores con experiencia). Un diagrama con los pasos más importantes sí resulta de utilidad para clarificar los pasos a seguir en un programa complejo.

Diagramas de Flujo: Elementos (ISO 5807)

Comienzo/Fin



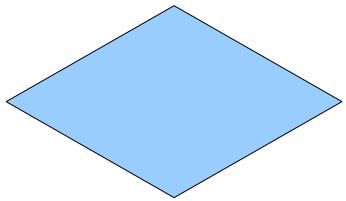
Enlaces

Calculo basico

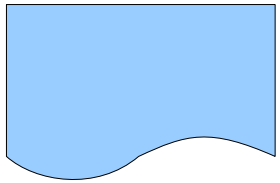
Calculo predefinido

Preparacion

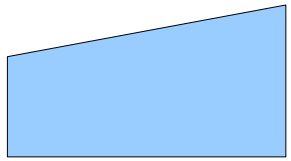
Entrada/Salida (IO) generica



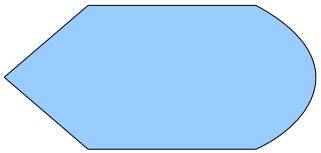
Decision (2 salidas!)



IO Fichero



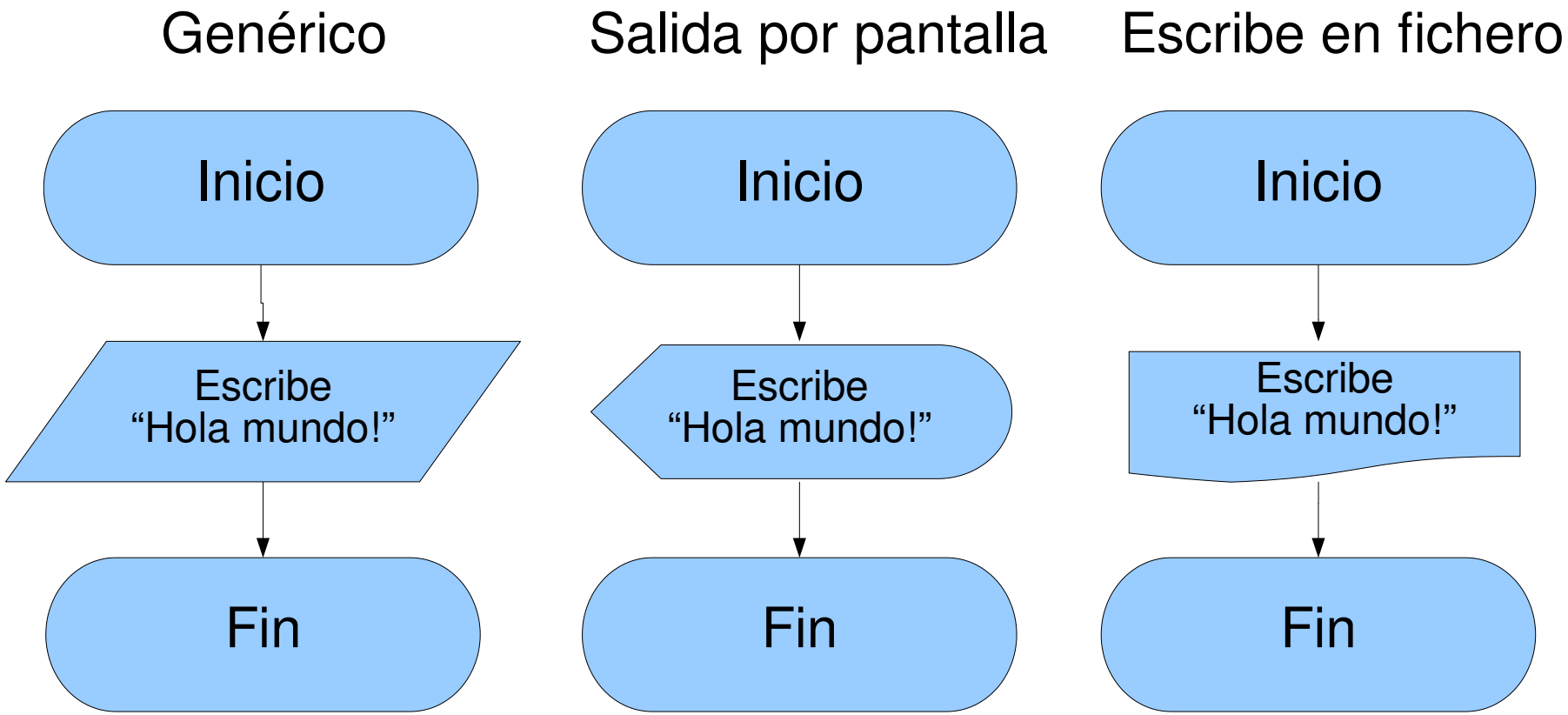
Entrada manual



Salida a pantalla

Diagramas de Flujo

Hacer un programa que escriba “Hola Mundo”:

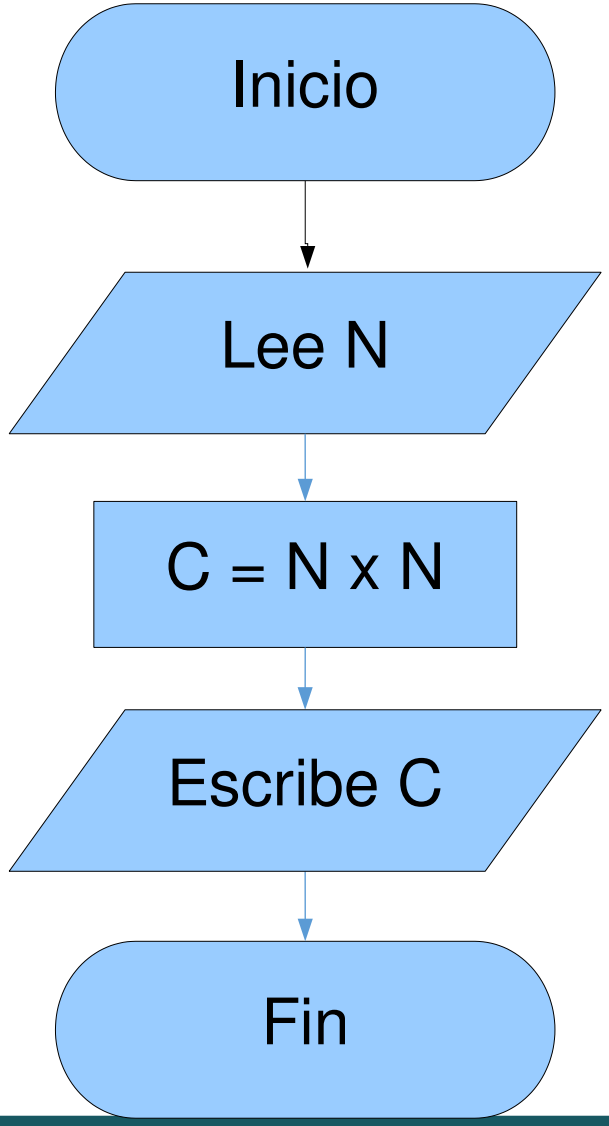


Calcula el Cuadrado

Hacer un programa que solicite un número y calcule su cuadrado:

Calcula el Cuadrado

Hacer un programa que solicite un número y calcule su cuadrado:

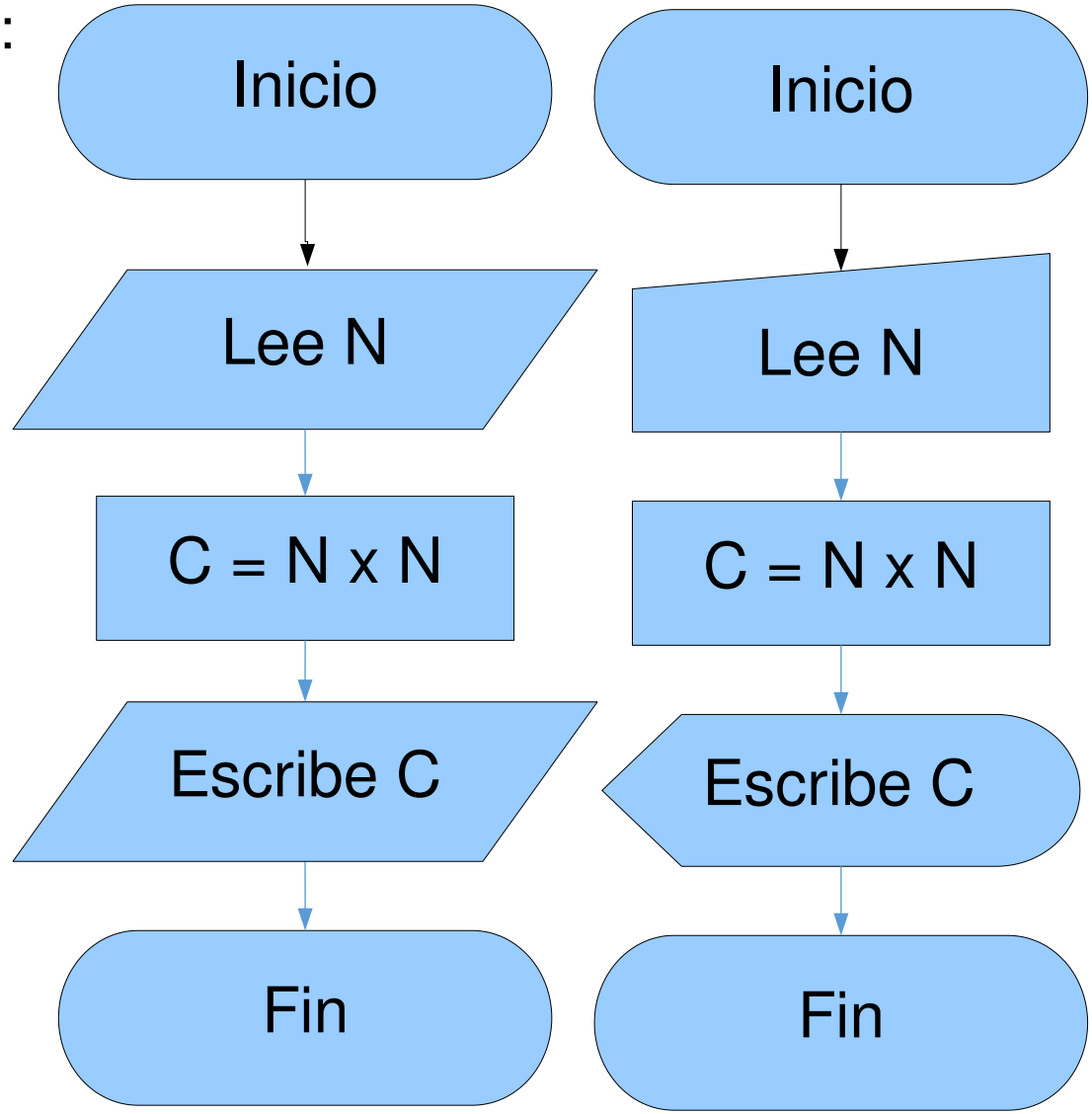


Calcula el Cuadrado

Hacer un programa que solicite un número por teclado y muestre su cuadrado por pantalla:

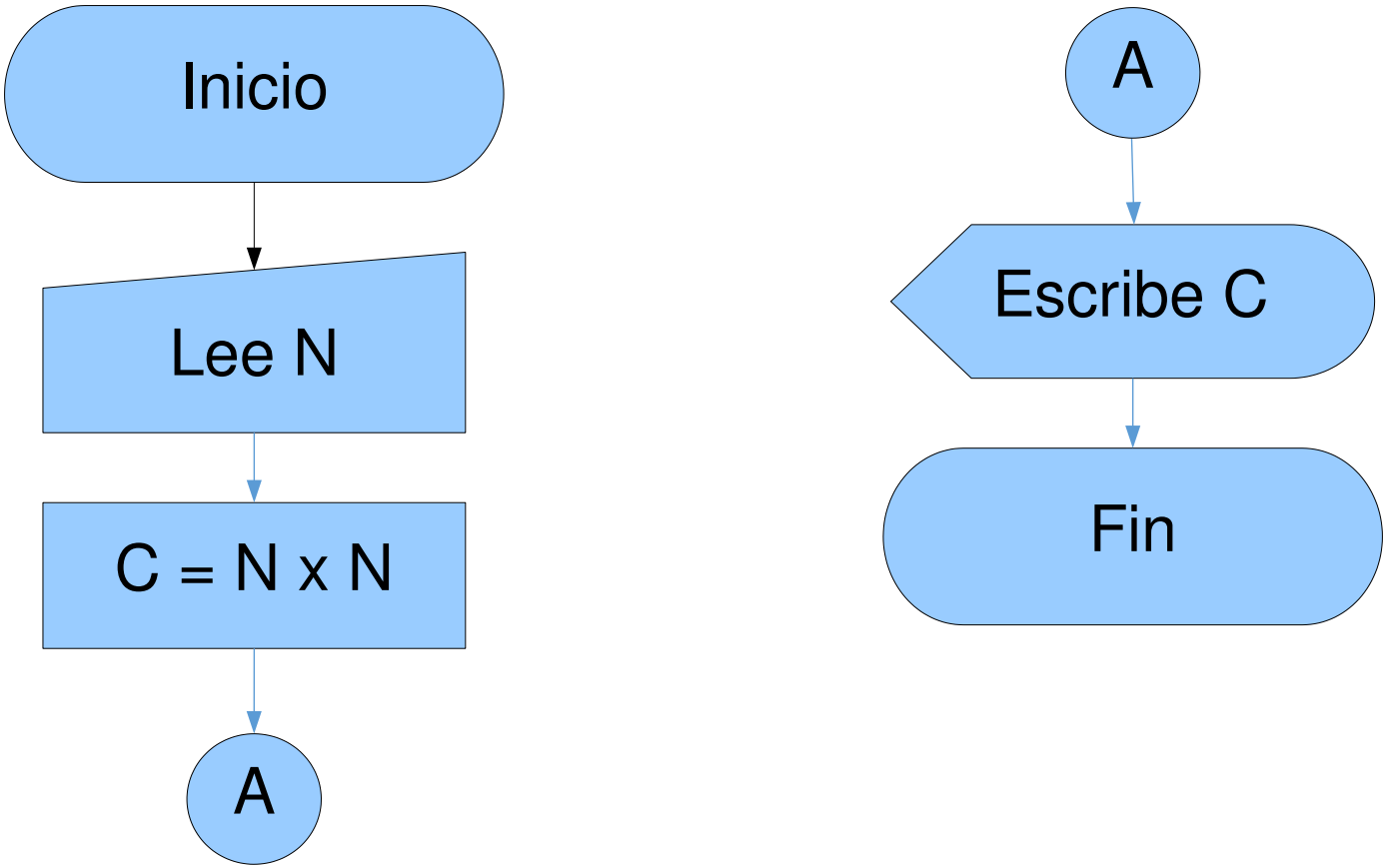
Calcula el Cuadrado

Hacer un programa que solicite un número por teclado y muestre su cuadrado por pantalla:



Calcula el Cuadrado

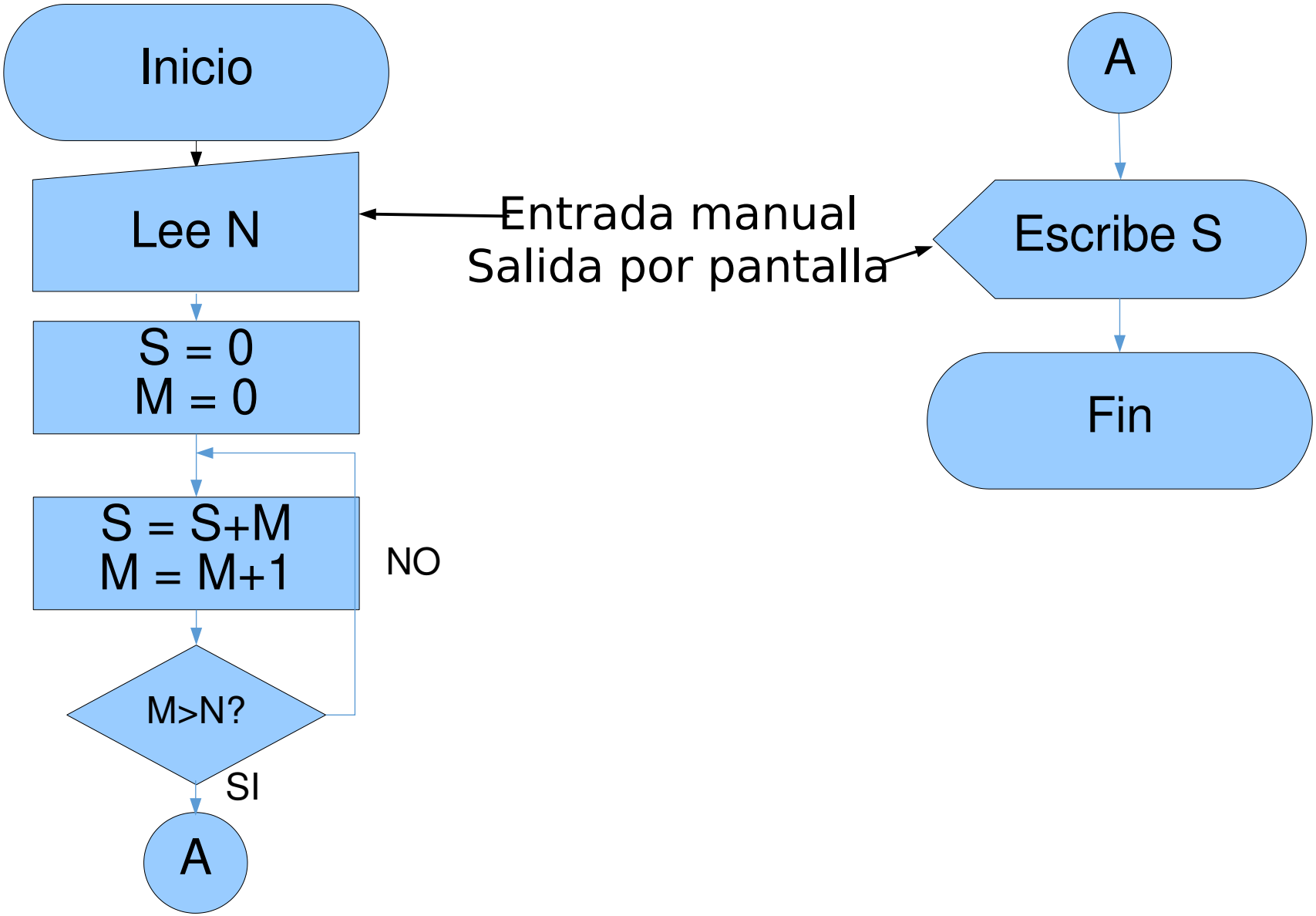
El uso de enlaces puede clarificar y ordenar el diagrama:



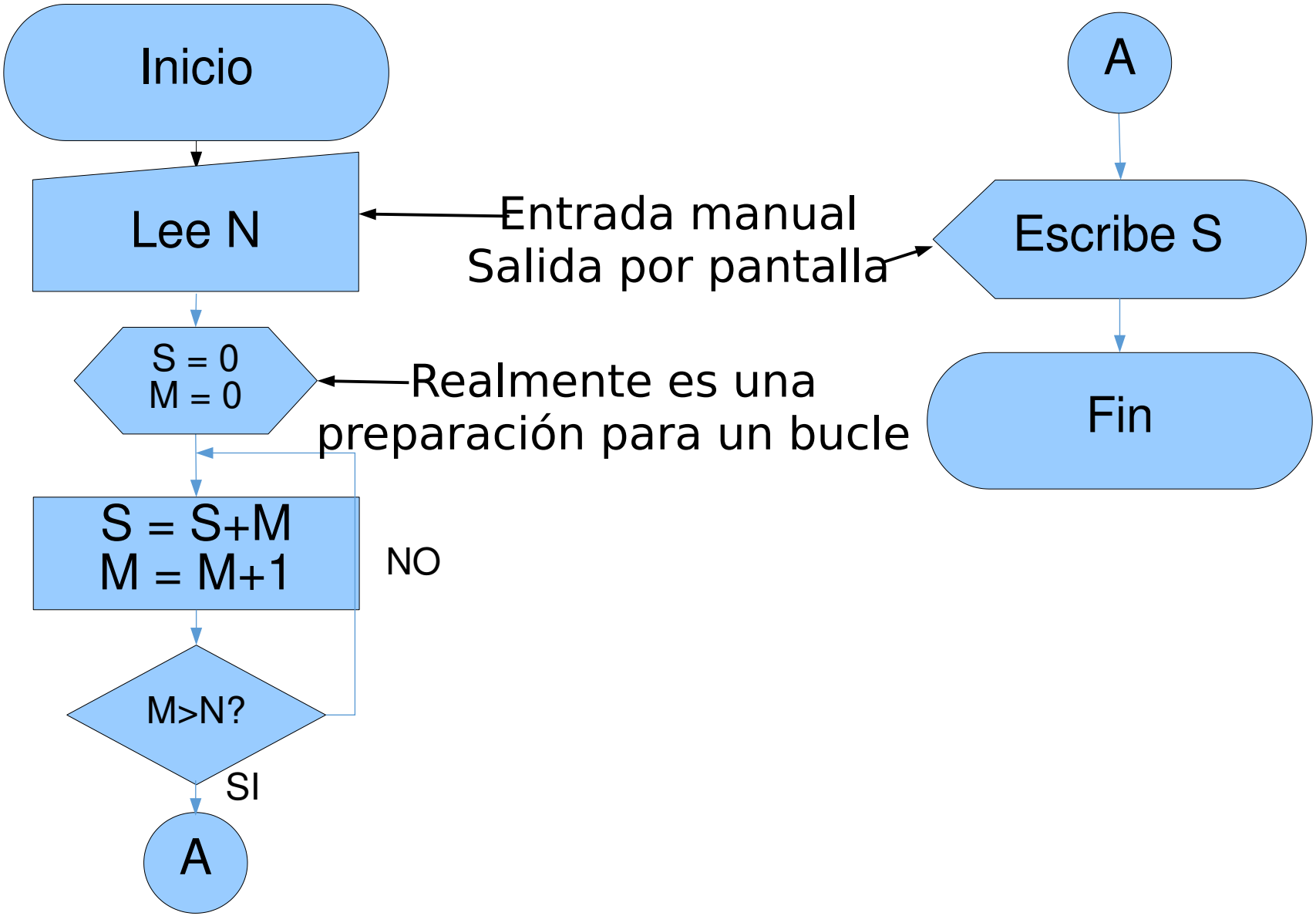
Suma de los Enteros de 1 a N

Hacer un programa que escriba por pantalla la suma de los enteros positivos de 1 a **N**, donde **N** es un número que introduce el usuario manualmente.

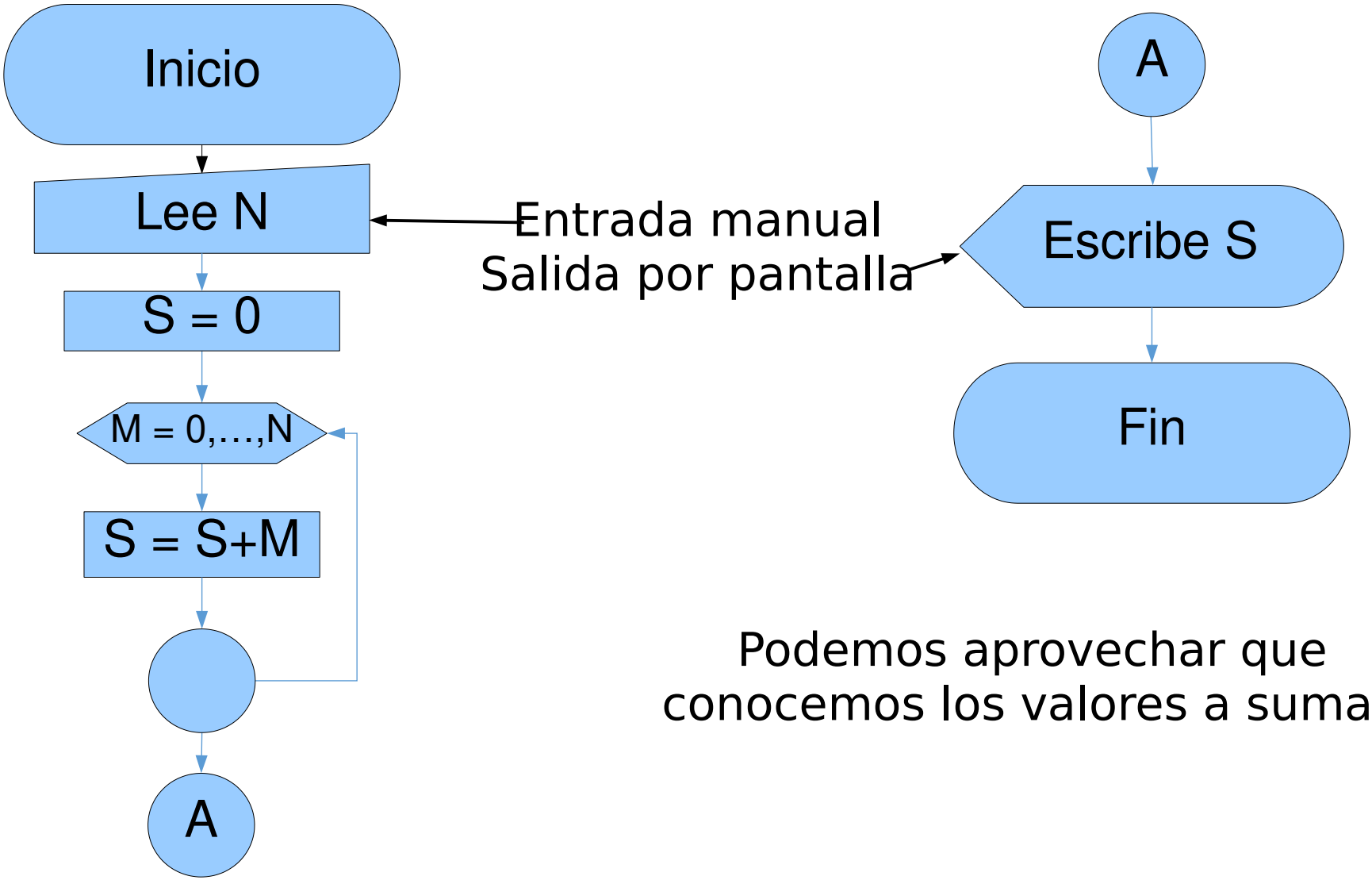
Suma de los Enteros de 1 a N



Suma de los Enteros de 1 a N



Suma de los Enteros de 1 a N

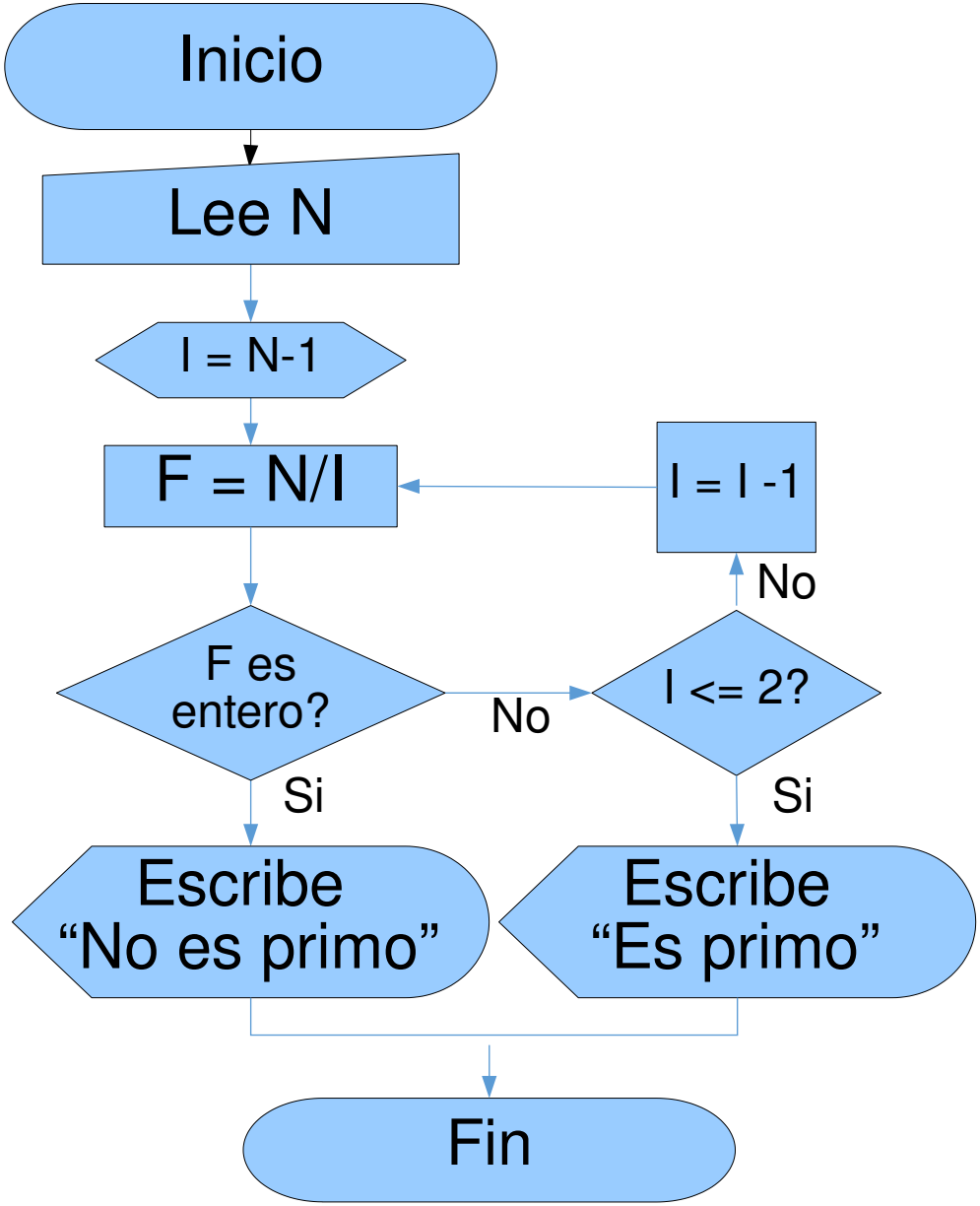


Podemos aprovechar que conocemos los valores a sumar.

Números Primos

Hacer un programa que escriba por pantalla si un número **N** leído de un fichero es o no un número primo.

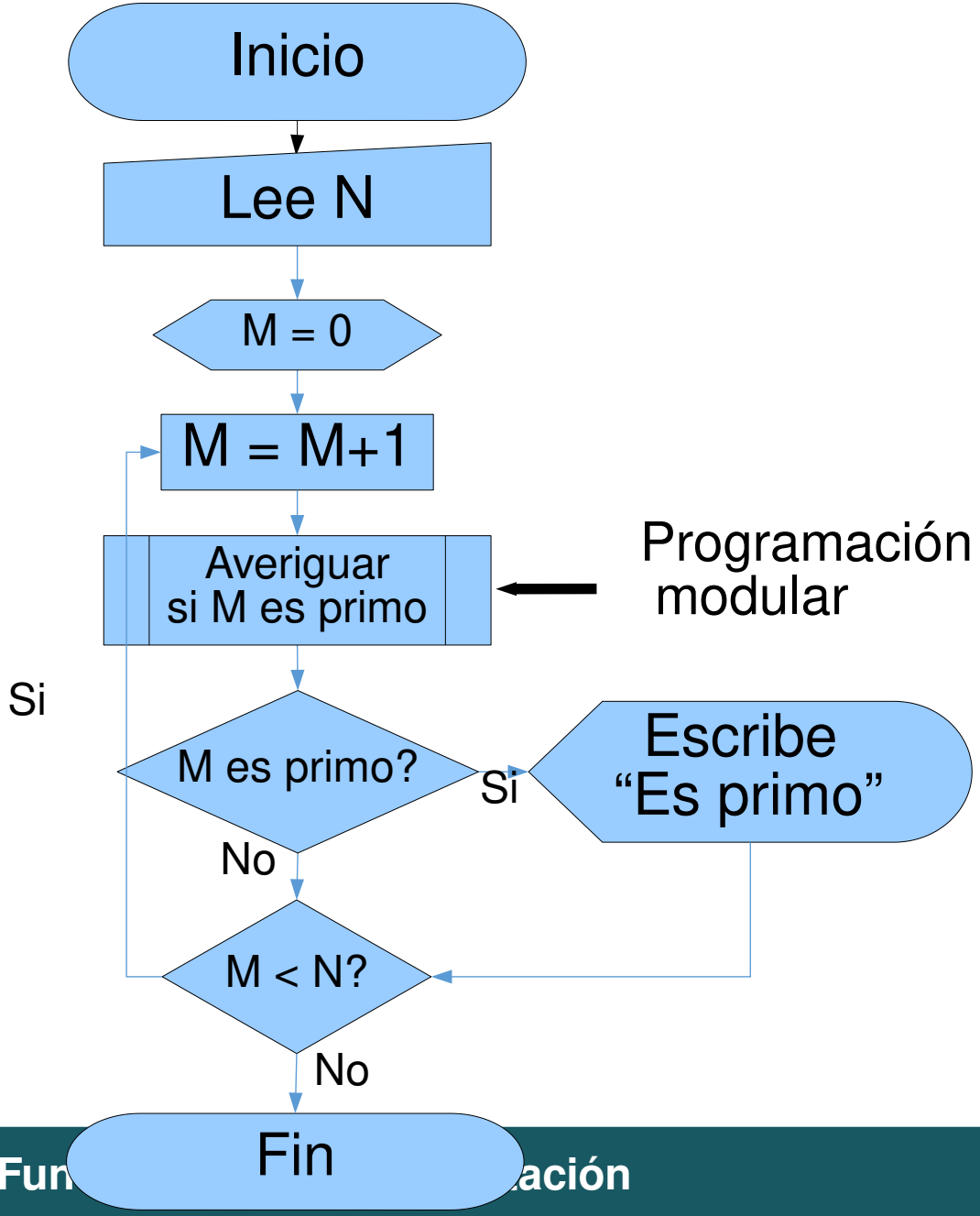
Números Primos



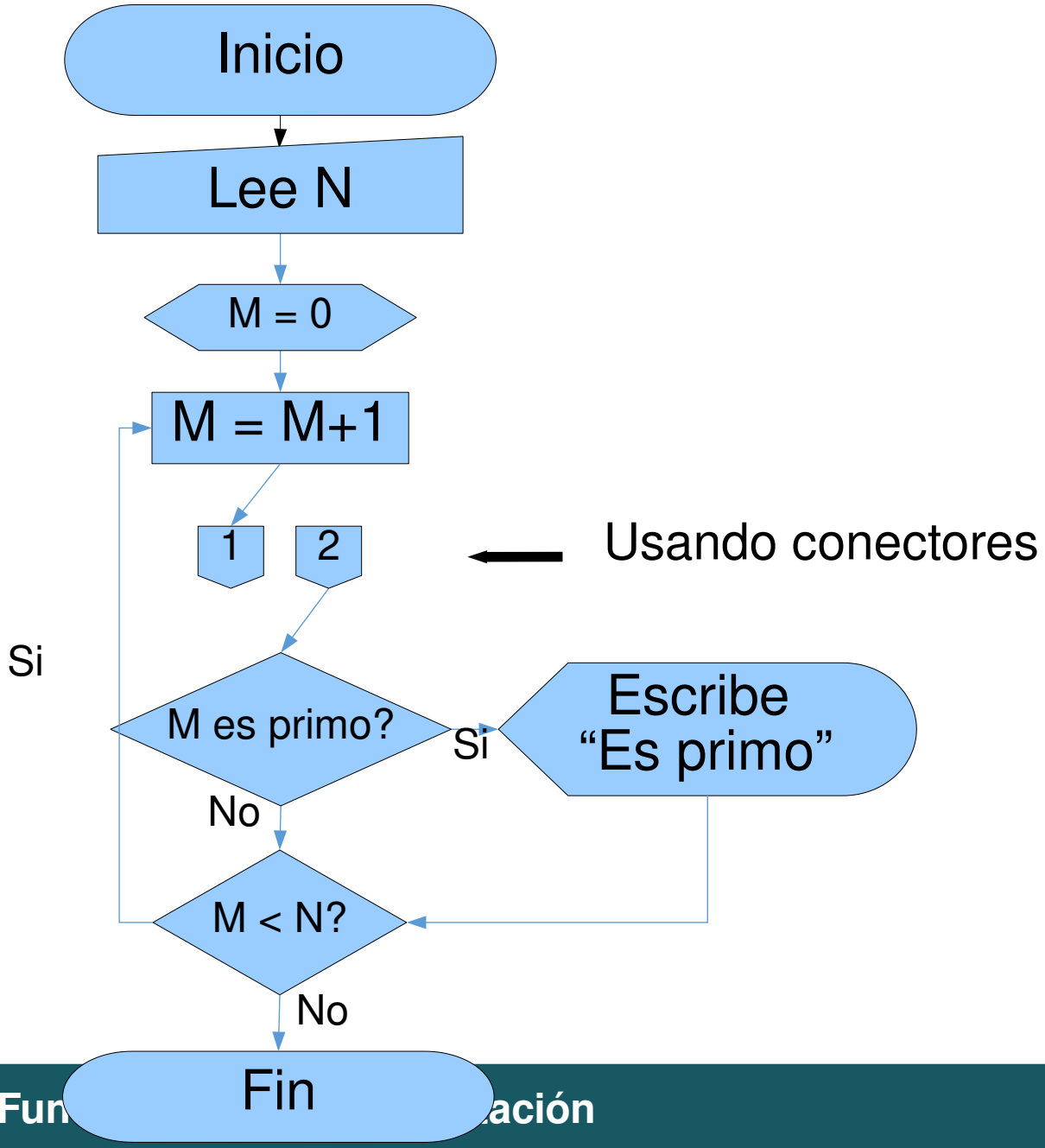
Mostrar los primos menores que N

Hacer un programa que escriba por pantalla todos los números primos menores que un número **N** introducido por el usuario.

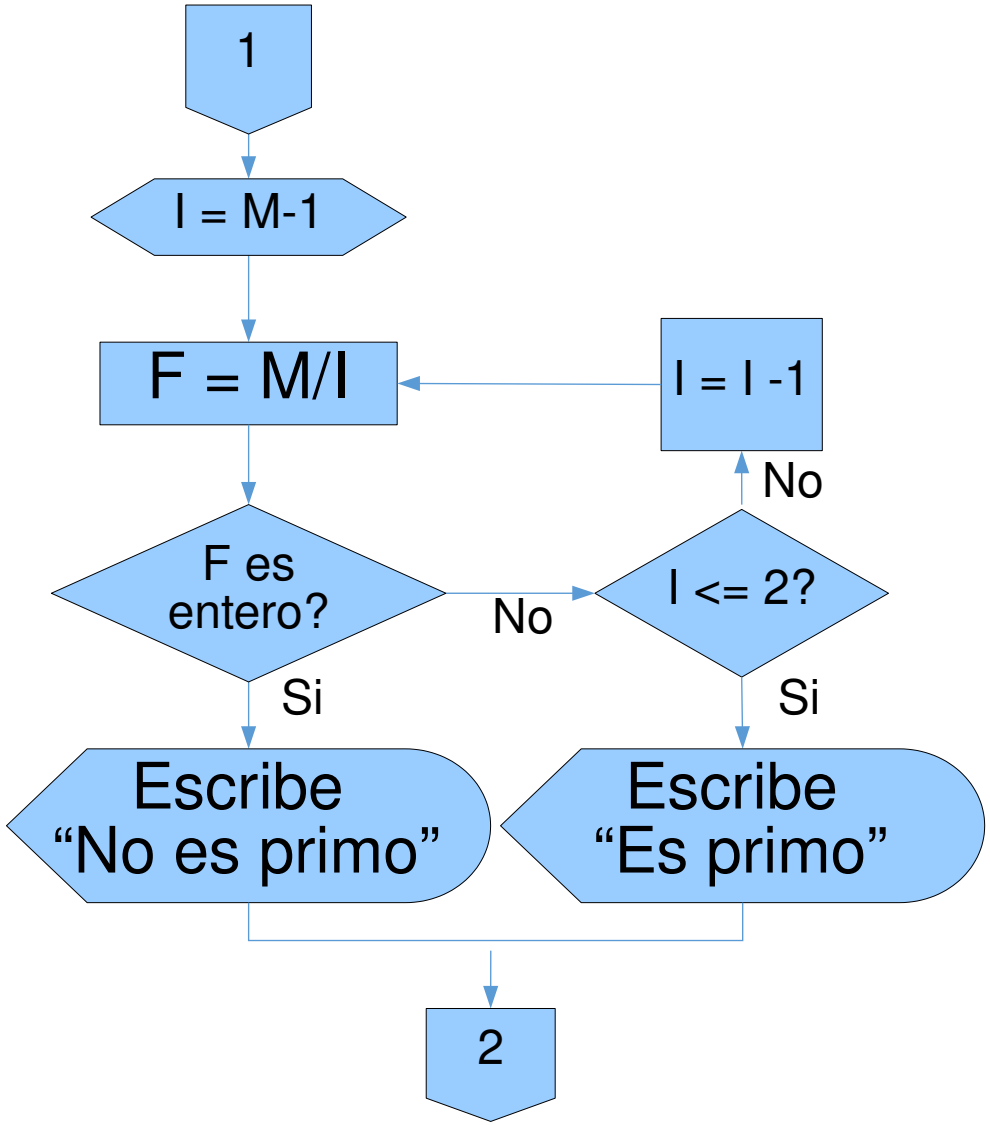
Mostrar los primos menores que N



Mostrar los primos menores que N



Mostrar los primos menores que N



Ejercicios

Modificar los diagramas anteriores, cuando sea posible, utilizando los elementos del diagrama asociados a iteraciones.

Hacer un diagrama que solicite el radio de una circunferencia y muestre por pantalla el diámetro, el perímetro y el área.

Hacer un diagrama para convertir una velocidad en m/s a km/h.

Hacer un diagrama que muestre por pantalla la temperatura en grados Kelvin y en grados Fahrenheit a partir de la temperatura en grados Celsius introducida manualmente por el usuario.

Hacer un diagrama que te pida un número y te diga si es par o es non

Hacer un diagrama para imprimir los primeros N términos de la sucesión de Fibonacci, donde el número será introducido manualmente por el usuario.

Hacer un diagrama que pida 3 números y diga cual es el mayor.

Ejercicios

Hacer un diagrama que pida la hora y muestre si es por la mañana (<12), por la tarde (<20) o de noche(>=20).

Hacer un diagrama para calcular el factorial de un número.

Hacer un diagrama que solicite 4 calificaciones y diga si está aprobado o no.

Hacer un diagrama que pida un número N y despliegue todas las combinaciones de dos números que sumados den N.

Hacer un diagrama que despliegue la tabla de multiplicar de un número X.

Fundamentos de Computación

Lenguajes de Programación
Representación de la Información.

Conceptos Básicos

Informática: Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la **información** por medio de **ordenadores**.

Información: Conjunto de símbolos que representan hechos, objetos o ideas.

Ordenador: Es una máquina capaz de aceptar unos **datos de entrada**, efectuar con ellos operaciones lógicas y aritméticas y proporcionar la información resultante a través de un medio de salida; todo ello sin intervención de un operador humano y bajo el control de un conjunto de **instrucciones** previamente almacenado en el ordenador.

Datos: Conjuntos de símbolos utilizados para expresar o representar un valor numérico, un hecho, un objeto o una idea; en la forma adecuada para ser objeto de tratamiento por el ordenador.

Conceptos Básicos

Instrucción: Mandato elemental que se da a un ordenador a fin de que efectúe una determinada operación. Las instrucciones son indicaciones sencillas y no ambiguas.

Algoritmo: Conjunto de instrucciones combinadas de forma adecuada para resolver un determinado problema en una cantidad finita de tiempo.

Programa: es un algoritmo expresado en un lenguaje comprensible para un ordenador.

Conceptos Básicos

Instrucción: Mandato elemental que se da a un ordenador a fin de que efectúe una determinada operación. Las instrucciones son indicaciones sencillas y no ambiguas.

Algoritmo: Conjunto de instrucciones combinadas de forma adecuada para resolver un determinado problema en una cantidad finita de tiempo.

Programa: es un algoritmo expresado en un lenguaje comprensible para un ordenador.

Programación: es el proceso de comunicar al ordenador una secuencia de instrucciones que señalan las acciones que ejecuta éste. Para ello se utilizará un **Lenguaje de Programación**.

Representación de la Información

Bit (Binary digiT): es la unidad básica de representación de la información en un dispositivo electrónico. Toma dos únicos valores, normalmente representados por 0 y 1.

Sistema Binario: sistema de numeración basado en potencias del número 2. Por ejemplo, el 1111 = 1*2^3+1*2^2+1*2^1+1*2^0, sería el número 15, representado por 1*10^1+5*10^0 en el sistema decimal.

byte: el byte (octeto) es la agrupación de 8 bits y suele ser el grupo básico utilizado. De este modo, se definen el Kilobyte (KB=1024=2^10 bytes), el Megabyte (MB=2^20 bytes), el Gigabyte (GB=2^30 bytes), etc.

Código ASCII: es la representación en código binario de los diferentes tipos de caracteres (alfabéticos, numéricos, especiales, de control y expandidos) utilizados para escribir o representar la información. Cada carácter tiene asociado un byte (8 bits).

Representación de la Información

Números enteros: los enteros sin signo se representan directamente en base dos. Dependiendo del número de bits asignado a la representación de números enteros se podrán tener número mayores o menores (p.e. **intmax**). Si consideramos el signo, uno de los bits debe representar el signo (0: positivo; 1: negativo), de modo que el resto representará el valor absoluto del número representado. **¿Cuántos bits utiliza Octave para representar los enteros con signo?**

Números reales: en este caso, la representación se basa en el formato exponencial de los números reales: *mantisa*2^exponente*. De este modo, tanto la mantisa como el exponente se guardarán en formato binario, incluyendo el signo del número en la mantisa. **¿Cuántos bits utiliza Octave para representar los números reales? ¿Cuál es el real máximo?, ¿y el mínimo? (p.e realmax/realmin).**

NOTA: dado que la magnitud y la precisión de los números está limitada por el número de bits dedicado a su representación, y éste es finito, es claro que existe un error de representación (p.e. **eps**) dado por la precisión.

Representación de la Información

Caracteres Lógicos: los caracteres lógicos se reducen a falso (false) y verdadero (true), y suelen representarse por un 0 y un 1, respectivamente. Dado que son intrínsecamente binarios, bastaría con dedicar un bit a su representación, si bien, como veremos durante el curso, algunos lenguajes guardan los caracteres lógicos como enteros.

Por ejemplo, teclea lo siguiente en Octave:

```
B=1;
A=(B==1);
C=int8(B);
whos
```

¿Cuántos bytes dedica Octave a la representación de cada variable?

Representación de la Información

Bit (Binary digiT): es la unidad básica de representación de la información en un dispositivo electrónico. Toma dos únicos valores, normalmente representados por 0 y 1.

Código ASCII: es la representación en código binario de los diferentes tipos de caracteres (alfabéticos, numéricos, especiales, de control y expandidos) utilizados para escribir o representar la información. Cada carácter tiene asociado un byte (8 bits).

Como hemos visto los ordenadores sólo entienden instrucciones codificadas en secuencias de 0s y 1s (**lenguaje máquina** ó de **bajo nivel**), las cuales son ininteligibles y pueden depender de las características del ordenador (hardware, software, etc.).

Lenguajes de Programación

Lenguaje de Programación: Es una notación formal para describir algoritmos o funciones que serán ejecutadas por un ordenador.

Existen cientos de lenguajes de programación que pueden clasificarse de formas muy diversas. Las más habituales son:

- Según su grado de independencia de la máquina: **Bajo** (máquina, ensamblador), **Medio** (C, etc.) o **Alto Nivel** (Python, ***Octave/Matlab***, Java, etc.).
- Según esté (Java, Python, etc.) o no (***Octave/Matlab***, C, etc.) **Orientado a Objetos**.
- Según sea **Compilado** (C, Fortran, etc.) o **Interpretado** (***Octave/Matlab***, etc.).
- Lenguajes orientados a problemas concretos (SQL, SPSS, etc.).

Por lo tanto, Octave es un lenguaje interpretado, no orientado a objetos y de alto nivel.

Ventana de Comandos

La ventana de comandos nos permite tanto navegar y gestionar el sistema de archivos de nuestro ordenador, como consultar el uso de las funciones de Octave a través de los comandos de ayuda.

Por ejemplo:

```
Terminal
octave:2> help ls
'ls' is a function from the file /usr/share/octave/3.8.1/m/miscellaneous/ls.m

 -- Command: ls
 -- Command: ls filenames
 -- Command: ls options
 -- Command: ls options filenames
     List directory contents.  For example:

          ls -l
               -| total 12
               -| -rw-r--r--   1 jwe  users  4488 Aug 19 04:02 foo.m
               -| -rw-r--r--   1 jwe  users  1315 Aug 17 23:14 bar.m

     The 'dir' and 'ls' commands are implemented by calling your
     system's directory listing command, so the available options will
     vary from system to system.

     Filenames are subject to shell expansion if they contain any
     wildcard characters '*', '?', '[]'.  If you want to find a literal
     example of a wildcard character you must escape it using the
     backslash operator '\'.

     See also: dir, readdir, glob, what, stat, filesep, ls_command.


Additional help for built-in functions and operators is
available in the online version of the manual.  Use the command
'doc <topic>' to search the manual index.

Help and information about Octave is also available on the WWW
at http://www.octave.org and via the help@octave.org
mailing list.
octave:3>
```

Ventana de Comandos: Sistema de Archivos

El sistema de ficheros nos permite organizar y gestionar nuestros directorios y ficheros de datos.

- ¿En qué directorio estoy trabajando?
- ¿Qué ficheros y directorios hay en el directorio?
- ¿Cómo me muevo por el sistema de ficheros/directorios?
- ¿Cómo creo directorios nuevos?
- Atajos (~, -, ., ..)

Ventana de Comandos: Sistema de Archivos

El sistema de ficheros nos permite organizar y gestionar nuestros directorios y ficheros de datos.

- ¿En qué directorio estoy trabajando? → ***pwd***
- ¿Qué ficheros y directorios hay en el directorio? → ***ls***
- ¿Cómo me muevo por el sistema de ficheros/directorios? → ***cd***
- ¿Cómo creo directorios nuevos? → ***mkdir***
- Atajos (., .., <TAB>): *directorio actual, directorio superior, autocompletado de directorios con el tabulado, etc...*

Normas de estilo y recomendaciones:

No usar espacio en blanco en los nombres.

No comenzar los nombres con un -(se usa para los flags)

No usar caracteres extraños en los nombres (ñ, à, etc...).

No usar otros caracteres reservados (|, ., _, etc)

Ventana de Comandos: Sistema de Archivos

El sistema de ficheros nos permite organizar y gestionar nuestros directorios y ficheros de datos.

- ¿En qué directorio estoy trabajando? → ***pwd***
- ¿Qué ficheros y directorios hay en el directorio? → ***ls***
- ¿Cómo me muevo por el sistema de ficheros/directorios? → ***cd***
- ¿Cómo creo directorios nuevos? → ***mkdir***
- Atajos (., .., <TAB>): *directorio actual, directorio superior, autocompletado de directorios con el tabulado, etc...*
- Gestionando ficheros:
  - Copiando ficheros: ***copyfile***
  - Eliminando ficheros: ***delete***
  - Moviendo ficheros: ***movefile***
  - Renombrando ficheros: ***movefile***
  - Generando ficheros: ***edit nameFile***

Ventana de Comandos: Sistema de Archivos

El sistema de ficheros nos permite organizar y gestionar nuestros directorios y ficheros de datos.

- ¿En qué directorio estoy trabajando? → ***pwd***
- ¿Qué ficheros y directorios hay en el directorio? → ***ls***
- ¿Cómo me muevo por el sistema de ficheros/directorios? → ***cd***
- ¿Cómo creo directorios nuevos? → ***mkdir***
- Atajos (., .., <TAB>): *directorio actual, directorio superior, autocompletado de directorios con el tabulado, etc...*
- Interacción con el sistema: la función ***system*** (***help system***) permite ejecutar órdenes del sistema (e.g. shell, UNIX, etc...) directamente desde la ventana de comandos de Octave.