

Formacion Integral

Carrera Programador full-stack

Resolución de problemas

¿Qué es un programa?

Formalmente, es una secuencia de instrucciones, escritas para realizar una tarea específica en un computador.

El programa tiene un formato ejecutable que la computadora puede utilizar directamente para ejecutar las instrucciones.

Un software está compuesto por programas.

Tiene un código fuente, escrito por un
Desarrollador.

Resuelven problemas concretos.

Tienen inputs, procesos, outputs.



¿Qué es un algoritmo?

Formalmente, es un conjunto de instrucciones o **reglas definidas** y **no-ambiguas**, **ordenadas** y **finitas** que permite, típicamente, solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades.

Dados un **estado inicial** y una **entrada**, siguiendo los **pasos** sucesivos se llega a un **estado final** y se obtiene una solución.

¿Y por qué: definidas, no ambiguas, ordenadas, finitas, solución, problema...?



¿Qué es un algoritmo?

Un algoritmo debe ser **Preciso** e indicar el orden de realización de cada paso.

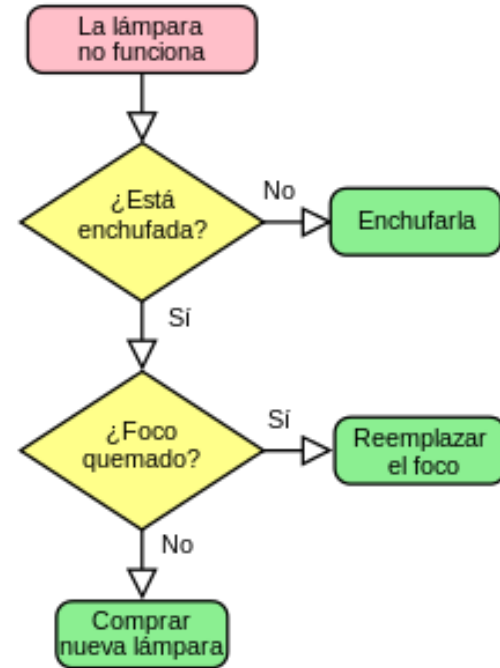
Un algoritmo debe ser **Definido**, es decir, si se sigue un algoritmo dos veces, se debe obtener el mismo resultado.

Un algoritmo debe ser **Finito**, es decir, si se sigue el algoritmo se debe terminar en algún momento.

Una secuencia de instrucciones claras y finitas.

Debe ser correcto y debe resolver el problema planteado en todas sus facetas.

Debe ser legible



¿Qué es un algoritmo?

Qué pasa con las siguientes expresiones en un algoritmo:

Gire, Gire a derecha, Gire un poco a la derecha.

Gane a la quiniela.

Agregue sal, Bata un huevo, casque un huevo.

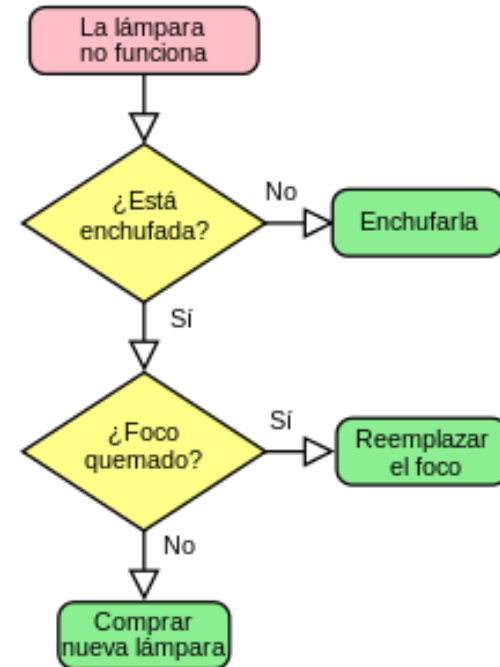
Enumere el conjunto de los números naturales.

Sea feliz.

Verifique la temperatura.

Agregue una condimentos varios.

Dibuje la chiforinfla del chavo.

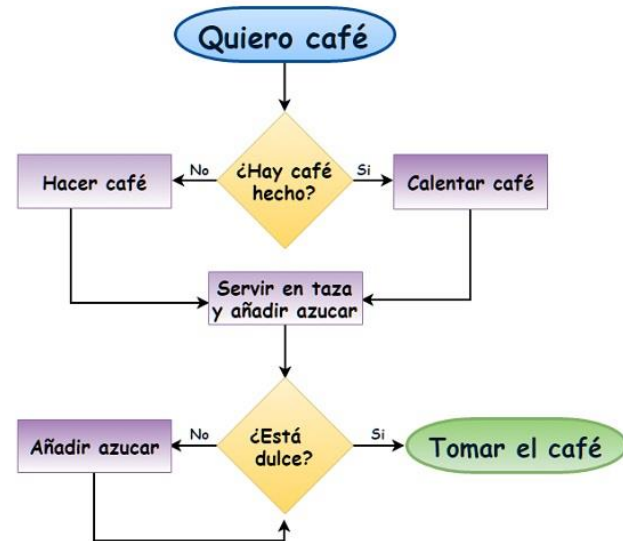


¿Qué es un algoritmo?

Intuitivamente las personas efectuamos cotidianamente una serie de pasos, procedimientos o acciones que nos permitan alcanzar algún resultado o resolver un problema (bañarnos, desayunar). Todo el tiempo estamos aplicando algoritmos para resolver problemas.



¿Cómo hacer un café?



¿Qué es un pseudocódigo?

Formalmente: es una descripción de **alto nivel** compacta e informal del principio operativo de un programa informático u otro algoritmo.

Es superficial.

Es interpretable por humanos, no máquinas.

Es más flexible.

Es independiente de un lenguaje de programación

Es interpretable, pero se espera que no ambiguo.

¿Qué es un pseudocódigo?

Cómo sería un pseudocódigo para el algoritmo de preparar un café.

Preguntarse si hay cafe.

Si hay: calentarlo

Si no hay: prepararlo

Servirlo en una taza

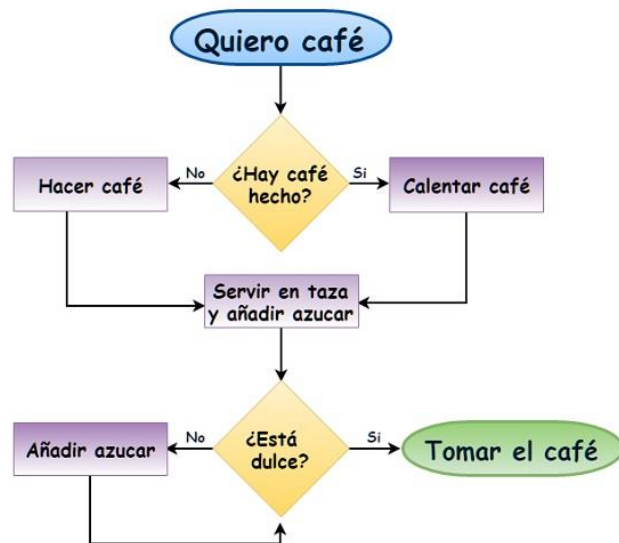
Agregar azucar

Probar si está dulce:

Si no esta dulce: agregar

más.

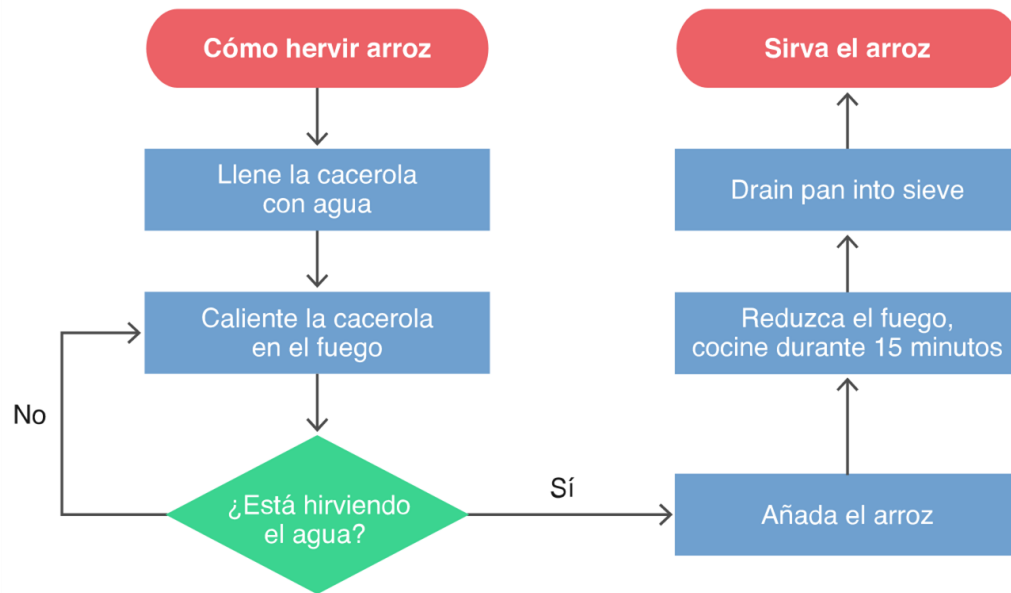
Si está dulce: beberlo



Diagramas de Flujo

Es la representación gráfica de un algoritmo o proceso. Se utiliza en disciplinas como programación, economía, procesos industriales y psicología cognitiva.

Estos diagramas utilizan **símbolos** con significados definidos que representan los pasos del algoritmo, y representan el flujo de ejecución mediante flechas que conectan los puntos de inicio y de fin del proceso.





Inicio/Final

Se utiliza para indicar el inicio y el final de un diagrama; del Inicio sólo puede salir una línea de flujo y al Final sólo debe llegar una línea.



Entrada General

Entrada/Salida de datos en General (en esta guía, solo la usaremos para la Entrada).



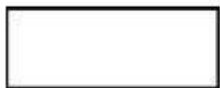
Entrada por teclado

Instrucción de entrada de datos por teclado. Indica que el computador debe esperar a que el usuario teclee un dato que se guardará en una variable o constante.



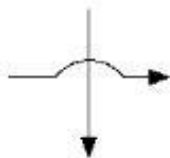
Llamada a subrutina

Indica la llamada a una subrutina o procedimiento determinado.



Acción/Proceso General

Indica una acción o instrucción general que debe realizar el computador (cambios de valores de variables, asignaciones, operaciones aritméticas, etc).



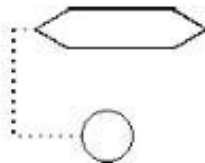
Flujo

Indica el seguimiento lógico del diagrama. También indica el sentido de ejecución de las operaciones.



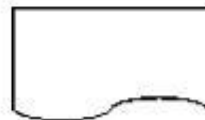
Decisión

Indica la comparación de dos datos y dependiendo del resultado lógico (falso o verdadero) se toma la decisión de seguir un camino del diagrama u otro.



Iteración

Indica que una instrucción o grupo de instrucciones deben ejecutarse varias veces.



Salida Impresa

Indica la presentación de uno o varios resultados en forma impresa.



Salida en Pantalla

Instrucción de presentación de mensajes o resultados en pantalla.



Conector

Indica el enlace de dos partes de un diagrama dentro de la misma página.



Conector

Indica el enlace de dos partes de un diagrama en páginas diferentes.

¿Cuáles usamos para:
Proposiciones, operaciones matemáticas,
“ingrese su nombre”, ingresar un nombre?

¿Cómo plantear un problema?

Siempre a la hora de plantear un problema lo primero que debemos hacer es leer detenidamente lo que nos piden y comprenderlo, detectar si necesita datos de **entrada** o de **salida**, si no comprendo algo hacer una **investigación** mayor sobre eso y ya luego **separar** por **partes** para empezar a plantear la solución.



Plantear un problema (juntos)

Hacer un algoritmo de flujo que dado dos números muestre cuál es el mayor.

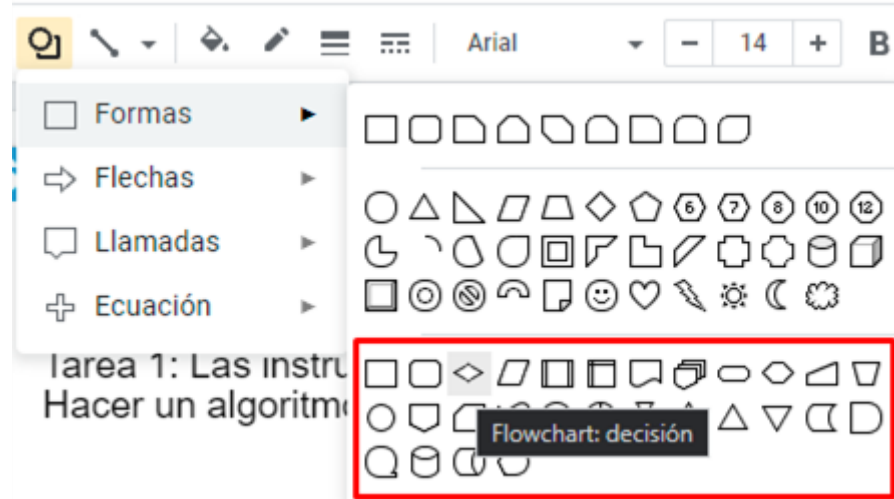
¿Tiene entradas? Si es así cuántas y cuáles

¿Tiene salidas? Si es así cuántas y cuáles

¿Tiene operaciones? Si es así cuántas y cuáles

¿Es necesario imprimir algo en pantalla? Si es así cuántas y cuáles

¿Cuáles símbolos necesitamos y dónde los encontramos?



Plantear un problema (juntos)

Hacer un algoritmo de flujo que dado dos números muestre cuál es el mayor. (escribamos primero un pseudocódigo)

#TODO: Escribamos acá el código

Inicio, Ingresar número_1, ingresar número_2, ¿número_1 = número_2?

Si imprimo “son iguales”.

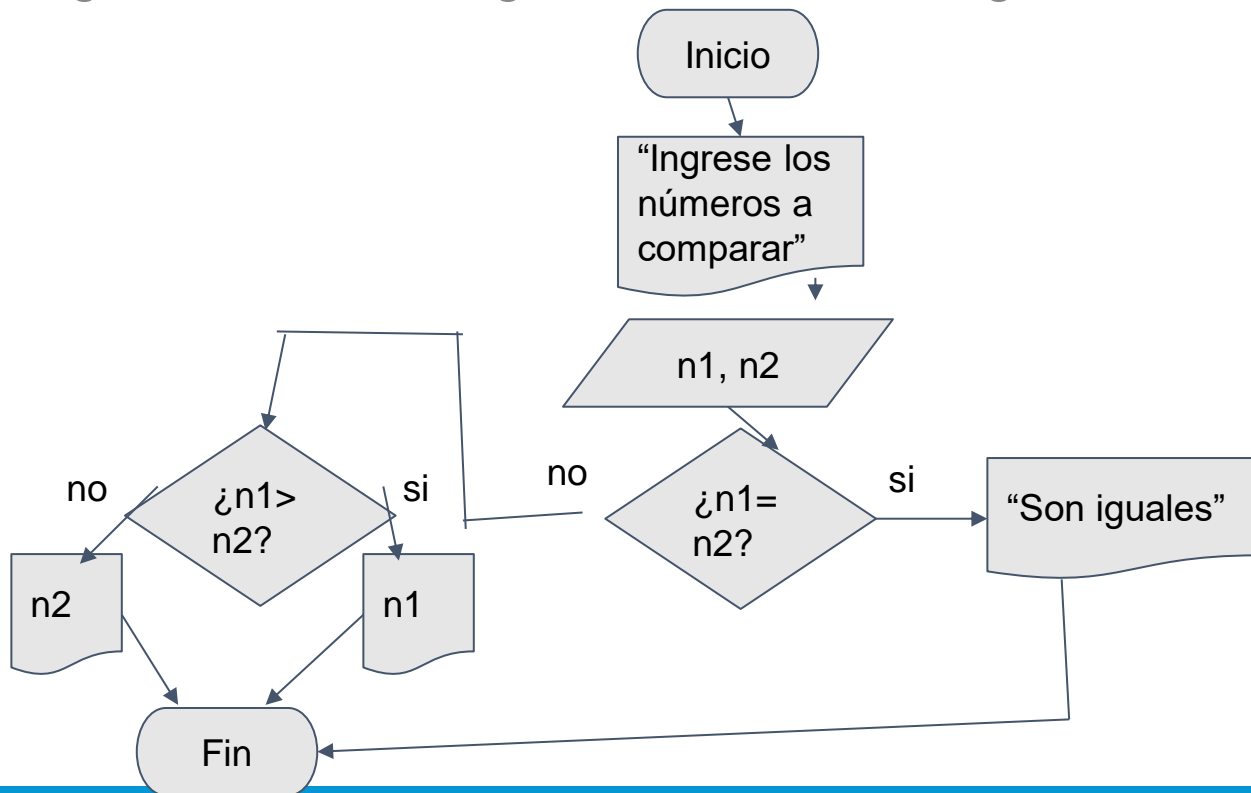
Si NO, ¿número_1 > número_2? Si: imprimo: número_1.

Si NO: impimo: número_2.

Fin

Plantear un problema (juntos)

Hagamos ahora el algoritmo como un Diagrama de flujo



Probando la solución

¿Cómo probamos nuestra solución?

Simulamos ingresar distintos valores y probamos que nos devuelve el algoritmo.

¿El output es el apropiado para cada input simulado?

¿Dónde podría haber algún error?

¿Qué pasa cuando el usuario ingresa un valor no esperado?

Casos de prueba

Los casos de prueba son una combinación de inputs simulados con outputs esperados.

Se crean varios y se corre una prueba.

Lo mismo se hace para probar los código que realizamos.

Los programadores lo hacen todo el tiempo.

De hecho hay paquetes, librerías, dependencias, etc que se usan para correr tests.

Tareas

El resto de la clase nos vamos a enfocar en resolver problemas.

Va a ser todo parte práctica.

Este es el [link](#) de las tareas

Help!

Es importante que cuando alguien quiera ayudar a un compañero, no le diga la respuesta. Sino que le haga alguna pregunta que le ayude a reflexionar o a encontrar dónde su razonamiento es equivocado.