



universidad
cenfotec_
La U de la informática

Principios de Programación.

Ejemplos de estructuras iterativas con contador.

Ejemplos resueltos de estructuras iterativas con contador

A continuación, se presentan algunos de los ejemplos de estructuras iterativas con contador.

Primer ejemplo de estructuras iterativas con contador

Ejemplo 1: Ejercicio 3 propuesto en semana 6

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Problema:

Se requiere calcular el factorial de un número entero dado por el usuario.

Definición:

Un número factorial se define de la siguiente manera:

$$n! = 1 * 2 * 3 * 4 * 5 \dots * n$$

$$1! = 1$$

$$2! = 1 * 2 = 2$$

$$3! = 1 * 2 * 3 = 6$$

$$4! = 1 * 2 * 3 * 4 = 24$$

$$n ! : \text{ se lee "n factorial".}$$

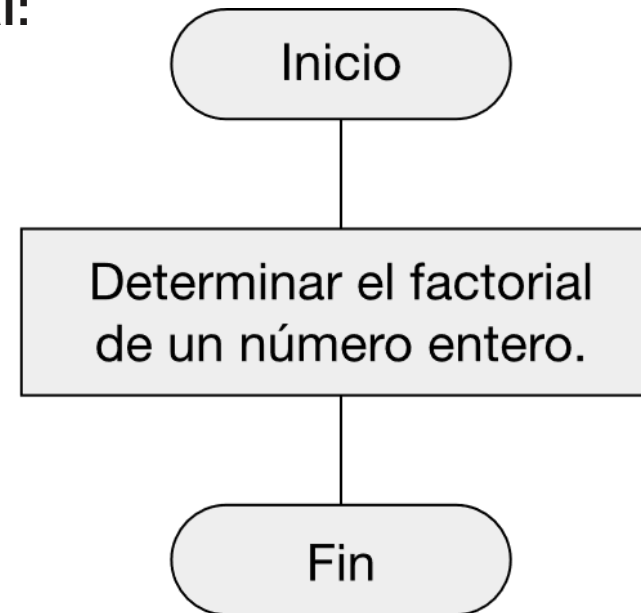
Ejemplo 1:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Problema:

Se requiere calcular el factorial de un número entero dado por el usuario.

Diagrama general:



Ejemplo 1:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Tabla de variables de entrada, intermedia y salida

Variables del programa			
Descripción	Notación		Ejemplo
	Nombre	Tipo	
Entrada			
Número entero dado por el usuario	numero	entero	5
Intermedia			
---	---	---	---
Salida			
Valor del factorial del número	factorial	entero	120

Ejemplo 1:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

¿Qué acciones se repiten? (Cuerpo del ciclo)

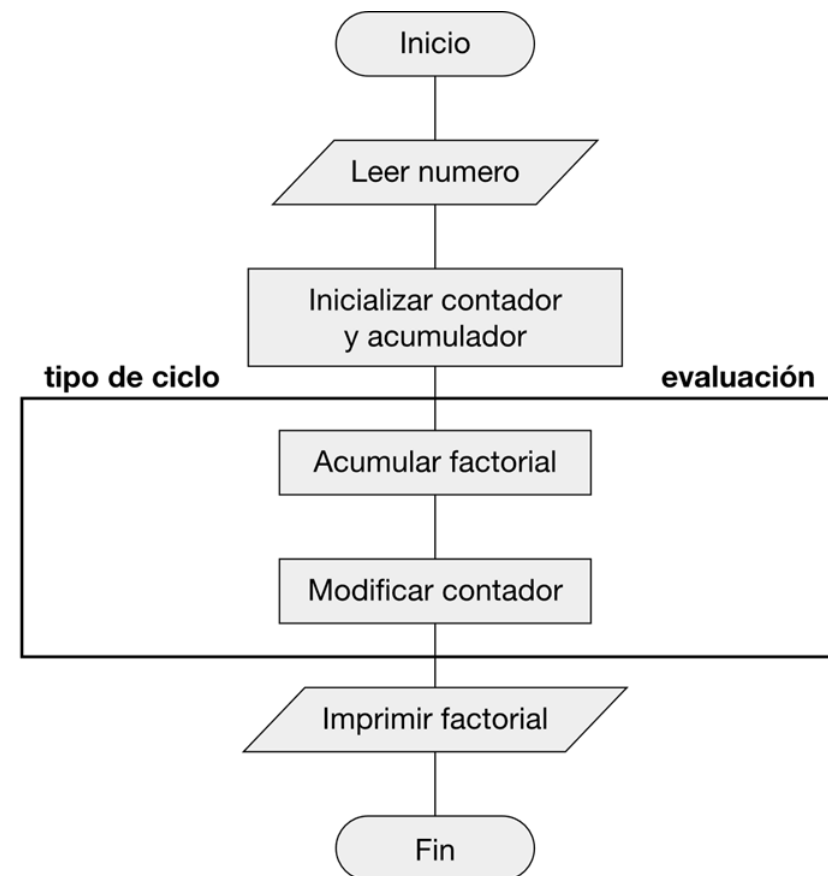
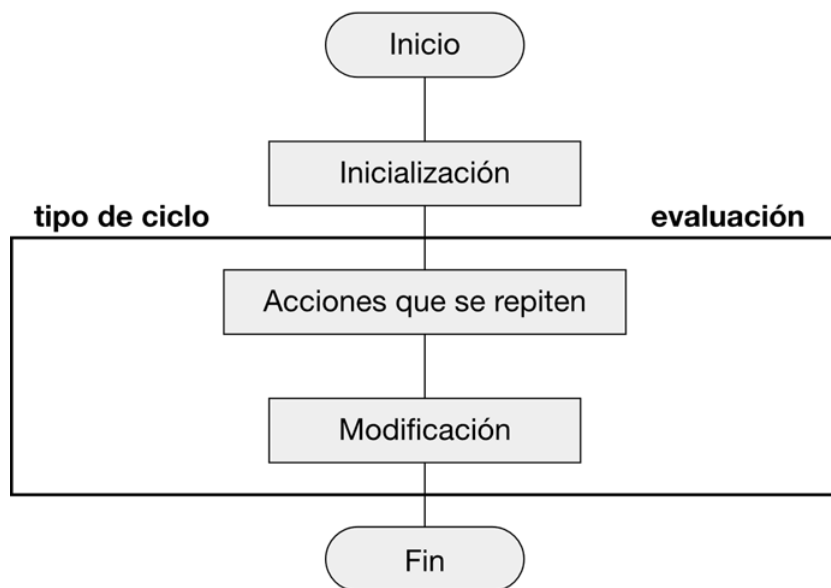
Se requiere calcular el factorial de un número entero dado por el usuario, este número solicitado será almacenado en una variable llamada *numero*.

Para calcular el factorial, se debe calcular la multiplicación de todos los números desde 1 hasta *numero*, por lo que se utilizará la misma variable *contador*, que toma esos mismos valores. Así, la variable *contador* debe ser **acumulada** en una variable llamada *factorial* donde vamos a ir guardando el valor de la multiplicación, de la siguiente forma:

`factorial = factorial * contador`

Ejemplo 1:

Diagrama explicativo:



Ejemplo 1:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

¿Cuántas veces se repetirá el ciclo? (evaluación)

El ciclo se repetirá mientras que el contador sea menor o igual a la variable *numero*. El **contador** durante la ejecución del ciclo va cambiando, de uno en uno, desde 1 hasta llegar al valor de la variable *numero*.

contador \leq numero

¿Cómo se modificará el contador? (modificación)

De uno en uno, **contando** desde 1 hasta llegar al valor de la variable *numero*.

contador = contador + 1

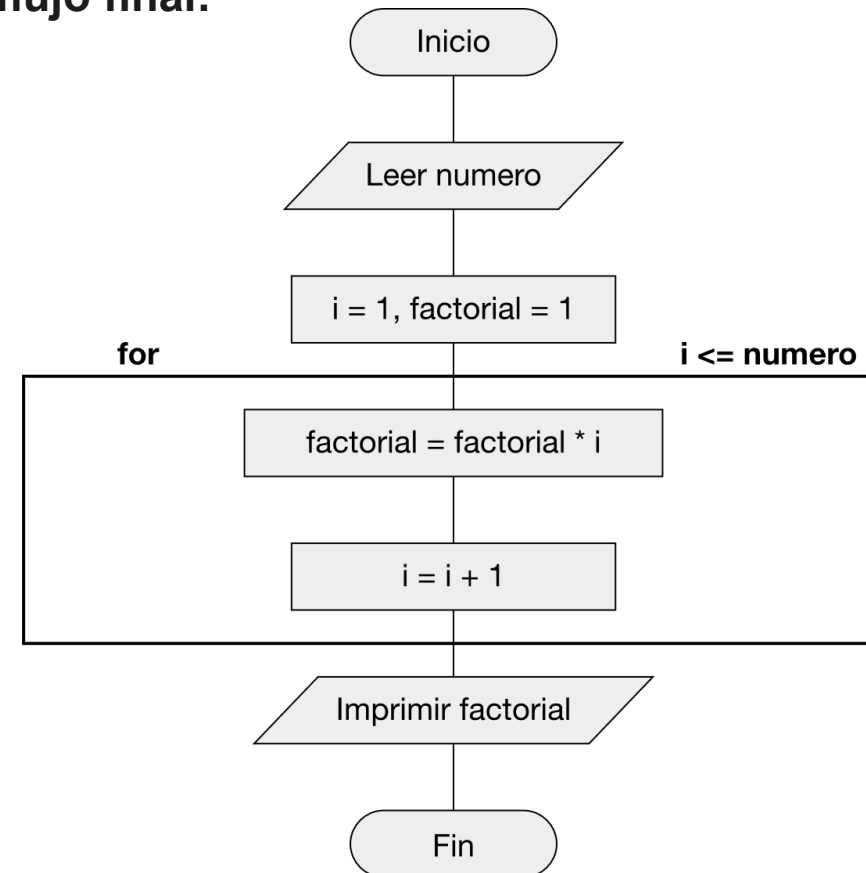
¿Con qué valor se inicializa el contador? (inicialización)

contador = 1

Ejemplo 1:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Diagrama de flujo final:



Ejemplo 1:

factorial.py

```
1  factorial = 1
2  numero = 0
3  i = 1
4  numero = int(input("Por favor teclee el número: "))
5
6  while (i <= numero):
7      factorial = factorial * i
8      i += 1
9  print(f"El valor del factorial es: {factorial}")
```

Segundo ejemplo de estructuras iterativas con contador

Ejemplo 2: Ejercicio 6 propuesto en semana 6

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

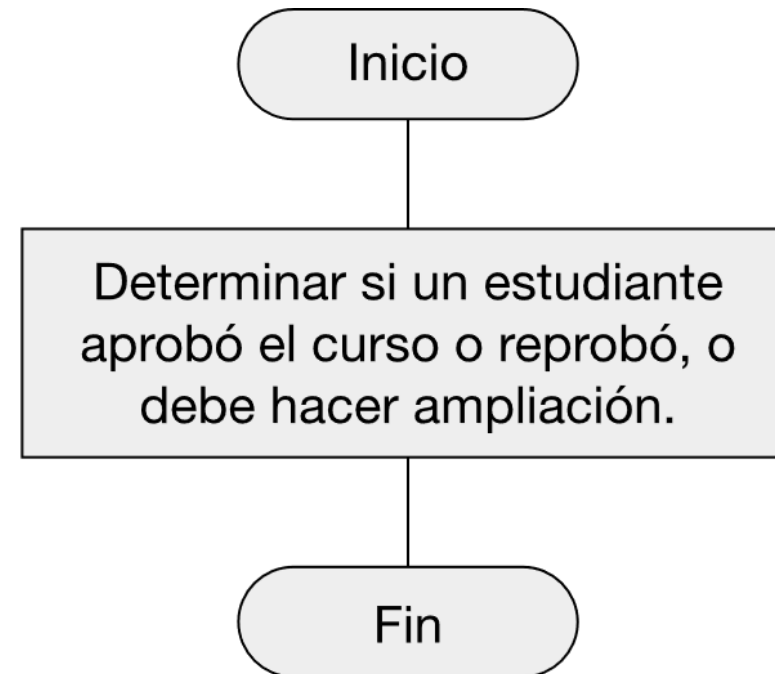
Problema:

Haga un programa que reciba como entrada la **cantidad de exámenes** realizados por un estudiante en un curso, y debe calcular la nota del curso que se obtiene del **promedio** de todos los exámenes. El programa deberá determinar además, si el estudiante aprobó, debe ir a ampliación o reprobó el curso. Un estudiante aprueba el curso si su nota final es mayor o igual a 70, debe hacer ampliación si su nota final es inferior a 70 pero superior o igual a 60, o reprueba el curso si la nota final es menor que 60.

Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Diagrama general:



Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Tabla de variables de entrada, intermedia y salida

Variables del programa			
Descripción	Notación		Ejemplo
	Nombre	Tipo	
Entrada			
Número de exámenes del curso	num_exámenes	entero	5
Nota de cada examen	nota_examen	float	78.8
Entrada constantes			
Nota mínima aprobatoria	APROBACION	float	70
Nota mínima para ampliación	AMPLIACION	float	60
Intermedia			
Almacena la suma de las notas	suma	float	256
Salida			
Se imprime un mensaje si el estudiante aprobó o no el curso, o tiene que hacer ampliación.	---	---	---

Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

¿Qué acciones se repiten? (Cuerpo del ciclo)

Se requiere obtener las notas de cada uno de los exámenes realizados por un estudiante. La cantidad de exámenes realizados por el estudiante se almacena en la variable *num_examenes*. Debido a que en cada iteración se obtendrá un nuevo valor para la variable *nota_examen*, esta deberá ser **acumulada** en la variable *suma*.

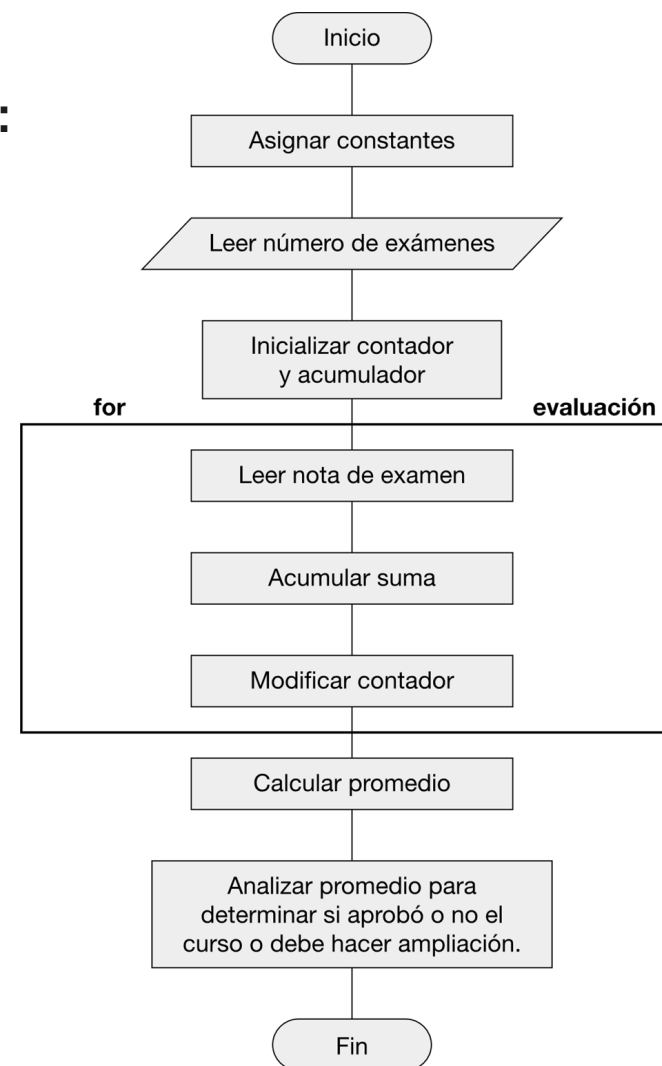
suma = suma + nota_examen

Una vez se finalicen todas las iteraciones, se utilizará el valor de la variable *suma*, para calcular el promedio de todas las notas obtenidas.

Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

Diagrama explicativo:



Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.

¿Cuántas veces se repetirá el ciclo? (evaluación)

El ciclo se repetirá mientras que el **contador** sea menor o igual a la variable *num_examenes*. El **contador** durante la ejecución del ciclo va cambiando, de uno en uno, desde 1 hasta llegar al valor de la variable *num_examenes*. Por lo que se utilizará un ciclo **for**.

contador <= *num_examenes*

¿Cómo se modificará el contador? (modificación)

De uno en uno, **contando** desde 1 hasta llegar al valor de la variable *num_examenes*.

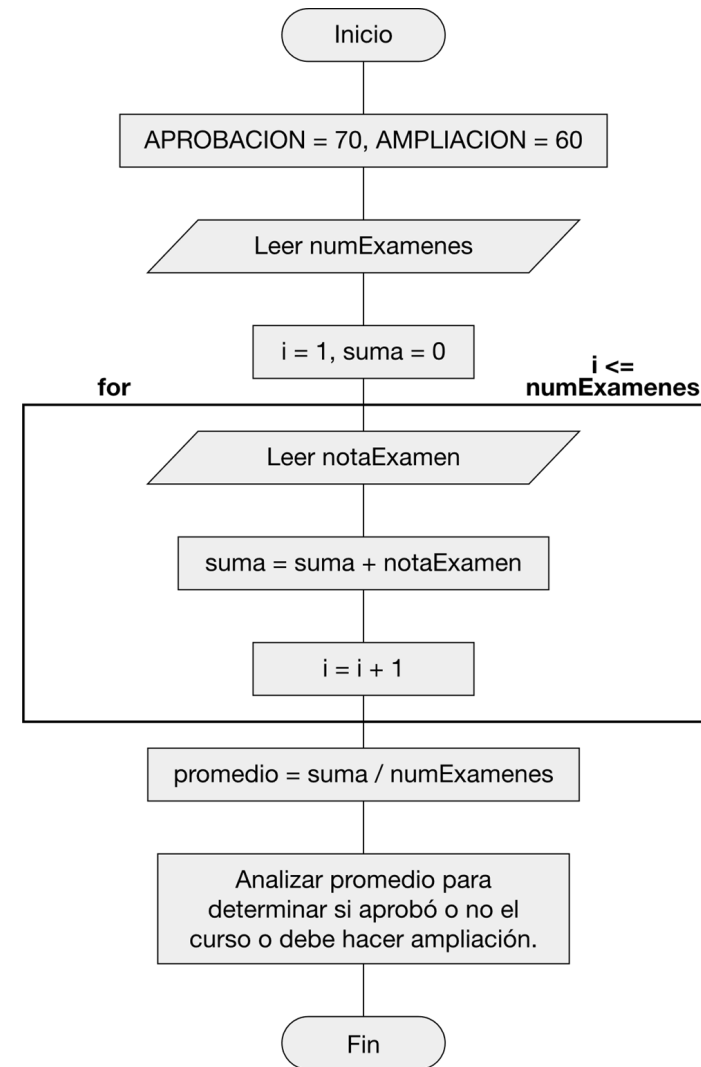
contador = **contador** + 1

¿Con qué valor se inicializa el contador? (inicialización)

contador = 1

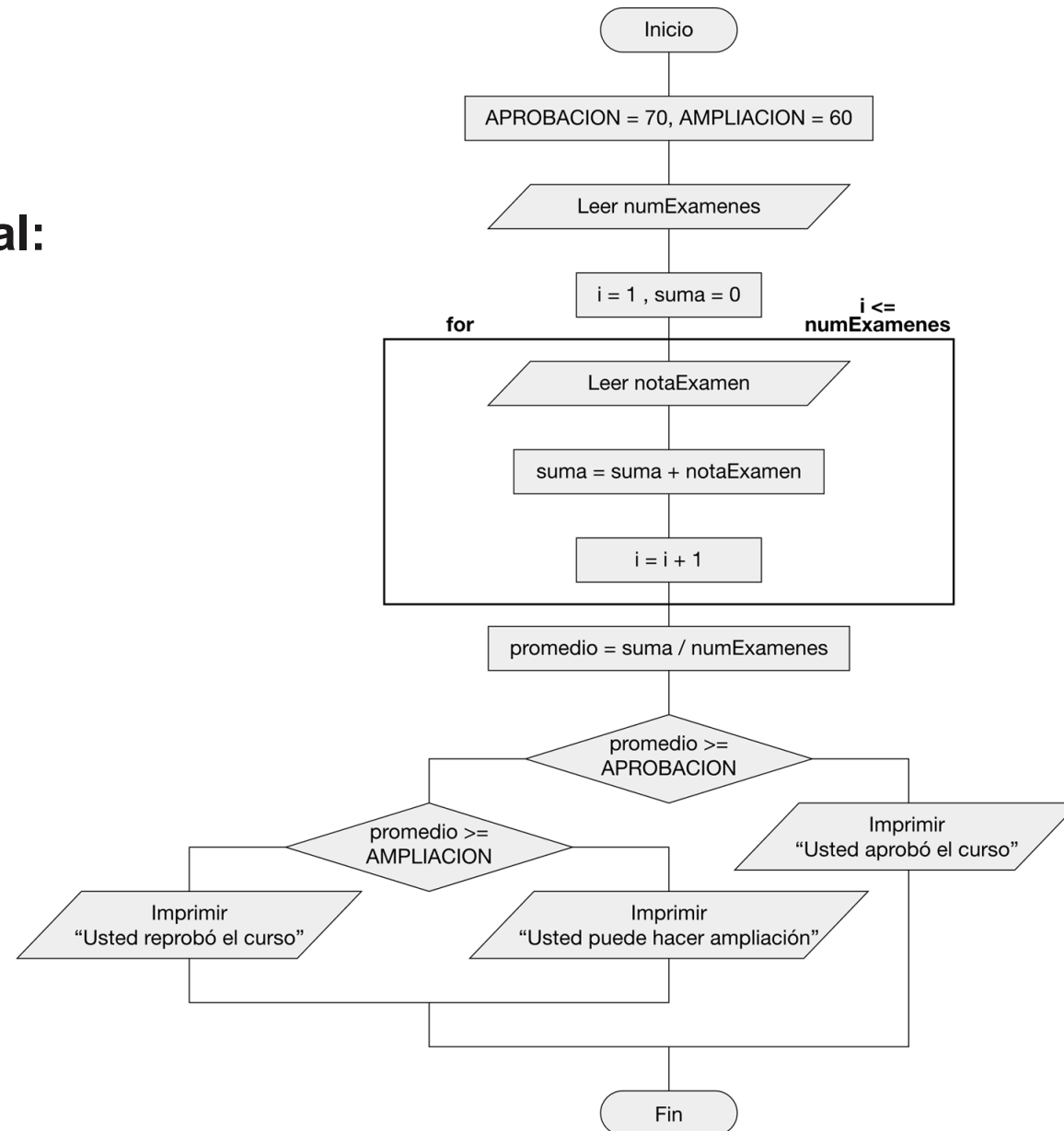
Ejemplo 2:

1. Comprender el problema y lo que se quiere averiguar.
2. Identificar las entradas y salidas.
3. Identificar las acciones que se repiten dentro del proceso de solución.
4. Definir la evaluación, la modificación y la inicialización.
5. Traducir el algoritmo a expresiones computacionales.



Ejemplo 2:

Diagrama de flujo final:



Ejemplo 2:

aprobacion_estudiante.py

```
1  APROBACION = 70.0
2  AMPLIACION = 60.0
3  i = 1
4  num_examenes = 0
5  nota_examen = 0.0
6  suma = 0.0
7  promedio = 0.0
8  num_examenes = int(input("Por favor indique el número de exámenes del curso: "))
9  while(i <= num_examenes):
10     nota_examen = float(input("Por favor indique la nota del examen " + str(i) + ": "))
11     suma += nota_examen #Es lo mismo que suma = suma + nota_examen
12     promedio = suma / num_examenes;
13     if (promedio >= APROBACION):
14         print("El estudiante aprobó el curso.")
15     else:
16         if (promedio >= AMPLIACION):
17             print("El estudiante tiene derecho a hacer ampliación.")
18         else:
19             print("El estudiante no aprobó el curso.")
20     i += 1
```



universidad
cenfotec_
La U de la informática