Escuela Profesional de Ciencia de la Computación
Faculdad de Ciencias
Universidad Nacional de Ingeniería

Examen Parcial

Curso: Introducción a la Ciencia de la Computación **Ciclo: 2017.1**
Versión: A

---

1. *a*) (2 ptos.) Siguiendo la definición formal, marcar cuáles no son algoritmos y por qué:

   (α) La preparación de una ensalada rusa. Sí corresponde a un algoritmo pues toda receta de cocina lo es.

   (β) La suma de todos los números primos. No corresponde a un algoritmo pues dicha tarea implicaría realizar una cantidad "infinita"de sumas y sería un proceso que NO terminaría.

   (γ) Una llamada telefónica al exterior del país. Sí corresponde a un algoritmo pues realizar una llamada implica: marcar el número al que se desea llamar, conversar y finalmente colgar.

   (δ) Juan orienta a Carlos por el teléfono cómo llegar de la UNI al Palacio de Gobierno del Perú: pasando el Óvalo Habich volteas dos cuadras a la derecha. No corresponde a un algoritmo pues Juan NO indica en qué dirección Carlos debe pasar el Óvalo Habich (¿de norte a sur o de sur a norte?); es decir, esta instrucción es ambigüa.

   b) (2 ptos.) Indicar la diferencia entre un algoritmo, proceso, programa y pseudocódigo. Un algoritmo es un concepto abstracto, mientras que un programa es la representación (física, concreta) de un algoritmo. El proceso es el desarrollo de un programa, mientras que un pseudocódigo es una descripción informal de un algoritmo.

   c) (1 pto.) Indicar verdadero (V) o falso (F) según corresponda:

   (F) Existe una sola forma de escribir pseudocódigos. Debido a su carácter informal, hay infinitas formas de hacer un pseudocódigo para un mismo algoritmo.

   (F) No es una buena práctica indentar las sentencias imperativas que están dentro de una estructura selectiva. Indentar dichas sentencias ayuda en

la legibilidad pues hacen notar que las mismas están dentro de dicha estructura selectiva.

(F) Los diagramas de flujo no ofrecen dificultad alguna en representar algoritmos largos y complejos. En algún momento los diagramas de flujo se convirtieron en una maraña entrelazada de flechas que se cruzaban, lo que hacía que comprender la estructura del algoritmo (largo y complejo) subyacente resultara difícil.

(F) Las variables de un programa son únicamente de uno de los siguientes tipos:

- entero
- real
- carácter

Además de estos tipos de datos, las variables en un programa son a veces asociadas con estructuras de datos que resultan de la combinación de estos, e.g. un arreglo numérico.

d) (1 pto.) ¿Cuáles son los elementos básicos de la programación estructurada? Los elementos son las estructuras de control, las funciones y los bloques.

e) (1 pto.) Mencionar un lenguaje de bajo nivel y otros tres de alto nivel. El lenguaje ensamblador es de bajo nivel; C++, Python, Java y Fortran son de alto nivel.

2. *a*) (2 ptos.) Expresar en pseudocódigo lo siguiente: se tienen 10 pacientes esperando en la sala y la doctora brindará atención a cada uno de sus pacientes hasta que no se tenga a nadie en la lista de espera.
Solución:

```
/*************** Sentencias declarativas *****************/
int i // declaramos que i es una variable entera


/*************** Sentencias imperativas *****************/
i = 10
while(i>0)
{
    atender paciente
    i = i - 1
}
```

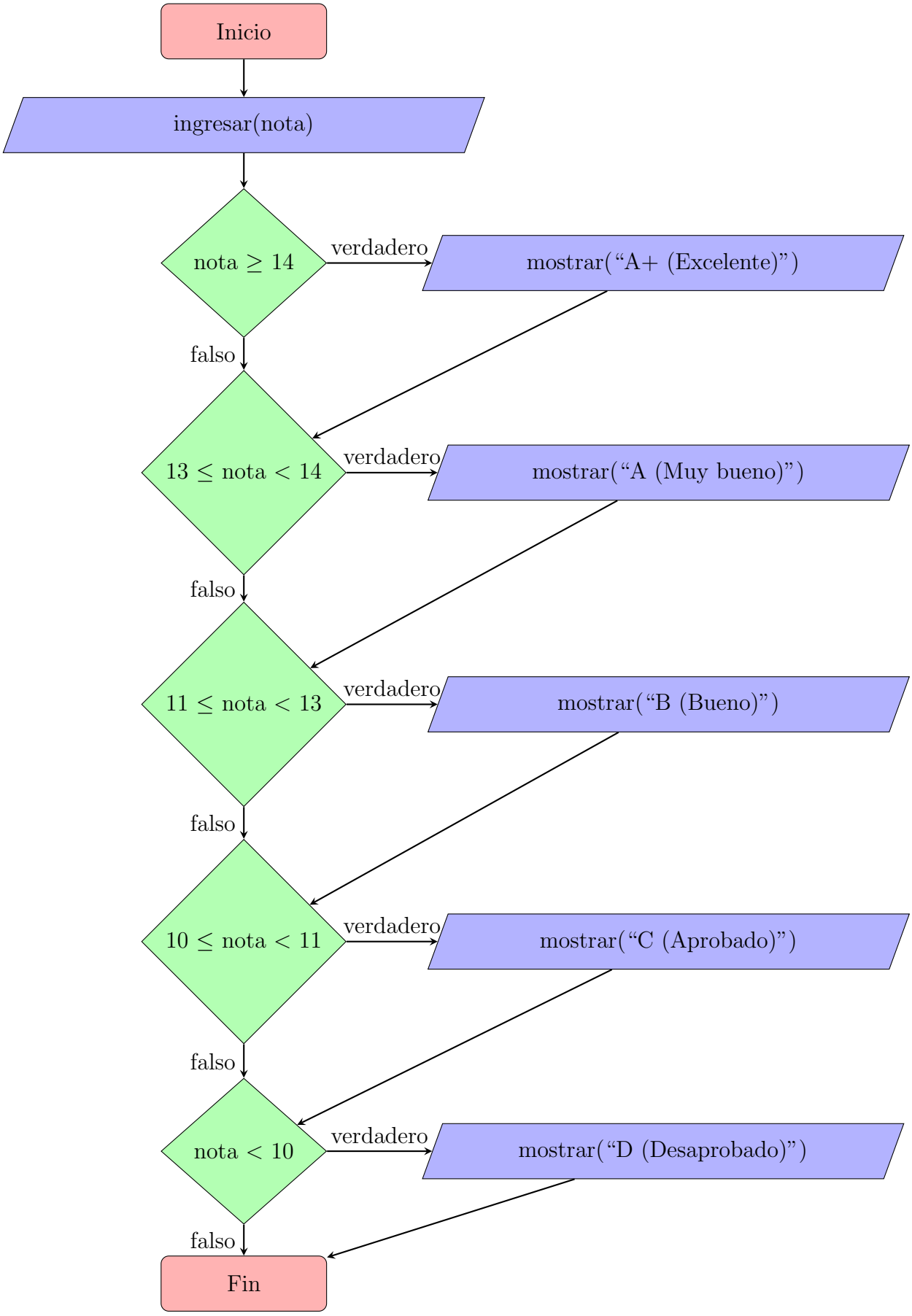
b) (2 ptos.) En el artículo número 23 del Reglamento de Evaluación para Estudiantes de Antegrado de la UNI (Resolución Rectoral Nro. 0116 con fecha 25 de enero de 2017) se establece que en los certificados de estudios la nota

final de los cursos, debe tener la equivalencia literal de acuerdo a la siguiente tabla:

Rango de calificación	Mención	Escala literal
De 14 a 20	Excelente	A+
De 13.0 a 13.9	Muy bueno	A
De 11 a 12.9	Bueno	B
De 10 a 10.9	Aprobado	C
Menor o igual a 9.9	Desaprobado	D

Desarrolle un diagrama de flujo en el que se ingrese la nota final (con un decimal) de un curso y muestre, según la tabla de arriba, la mención y escala literal correspondiente. Por ejemplo, si se ingresa 18.2 se debe mostrar A+ (excelente).

Solución: nota es nuestra única variable real.


Inicio
ingresar(nota)
nota ≥ 14
verdadero
mostrar(“A+ (Excelente)”)
falso
13 ≤ nota < 14
verdadero
mostrar(“A (Muy bueno)”)
falso
11 ≤ nota < 13
verdadero
mostrar(“B (Bueno)”)
falso
10 ≤ nota < 11
verdadero
mostrar(“C (Aprobado)”)
falso
nota < 10
verdadero
mostrar(“D (Desaprobado)”)
falso
Fin

c) (3 ptos.) Un estudiante de la UNI ha rendido 3 prácticas. Desarrolle un pseudocódigo que lea sus notas $n1$, $n2$ y $n3$, y calcule cuánto debe ser su cuarta nota $n4$, la cual debe cumplir dos condiciones:

- ser la mayor posible y
- hacer que la suma de las prácticas sea múltiplo de 4 (para que el promedio sea entero.)

Solución:

```
/*************** Sentencias declarativas *****************/
int n1, n2, n3, n4
// hemos declarado que n1, n2, n3 y n4 son variables enteras

/*************** Sentencias imperativas *****************/
ingresar(n1,n2,n3)
n4 = 20
while( ((n1+n2+n3+n4) MOD 4) > 0 )
{
    n4 = n4 - 1
}
mostrar(n4)
```

3. *a*) (2 ptos.) Sobre el siguiente conjunto de sentencias:

```
int i=1, suma=0;
while (i>0)
{
    suma = suma + i*i;
    i = i+2;
}
```

(α) ¿Qué es lo que finalmente hace?

(β) ¿Representa un algoritmo según la definición formal? Fundamente su respuesta.

Lo que hace es sumar los cuadrados de los númros impares desde 1 hasta que la capacidad de la computadora lo permita. NO representa un algoritmo según la definición formal pues dicho proceso no tiene un fin, es un bucle infinito.

b) (1 pto.) En la página 220 de [Brookshear and Brylow, 2015] dice: "el algoritmo para convertir medidas de temperatura de Celsius a Fahrenheit se representa normalmente mediante la fórmula algebraica:

$$F = (9/5)C + 32,$$

pero podría representarse mediante la sentencia:

> multiplicar la medida de temperatura en Celsius por 9/5 y luego sumar 32 al producto

o incluso puede representarse en forma de circuito electrónico."Sobre esta cita, ¿qué naturaleza de los algoritmos se está poniendo en evidencia y por qué? Se está poniendo en evidencia la NATURALEZA ABSTRACTA de los algoritmos porque como algortimo dicho objeto existe en el mundo de nuestras ideas; sin embargo, su representación (lo concreto) puede variar.

c) (3 ptos.) Utilizando únicamente las siguientes palabras:

- ASCII
- diagrama(s) de flujo
- lenguaje(s) de programación
- primitiva(s)
- pseudocódigo(s)

llenar las siguientes citas extraídas del [Brookshear and Brylow, 2015]:

(c.1) "En resumen, pueden surgir problemas de comunicación cuando el lenguaje utilizado para representar un algoritmo no está definido de forma precisa o cuando no se proporciona la información con el suficiente detalle. Las Ciencias de la computación tratan de resolver estos problemas estableciendo un conjunto bien definido de elementos fundamentales de construcción de software (building block) a partir de los cuales puedan construirse representaciones de algoritmos. Esos elementos se denominan primitivas. La asignación de definiciones precisas a estas primitivas elimina muchos problemas de ambigüedad y el exigir que los algoritmos se definan en términos de estas primitivas establece un nivel de detalle uniforme. Un conjunto de primitivas junto con una serie de reglas que indiquen cómo pueden combinarse esas primitivas para representar ideas más complejas constituye un lenguaje de programación."(página 221)

(c.2) "Durante las décadas de 1950 y 1960, los diagramas de flujo constituían la herramienta de diseño más avanzada. Sin embargo, los diagramas de flujo se convirtieron pronto en una maraña entrelazada de flechas que se cruzaban, lo que hacía que comprender la estructura del algoritmo subyacente resultara difícil. Por ello, el uso de diagramas de flujo como herramienta de diseño ha dejado paso a otras técnicas de representación. Un ejemplo es el pseudocódigo utilizado en este texto mediante el que se representan los algoritmos usando estructuras textuales bien definidas. Los diagramas de flujo siguen siendo útiles cuando el objetivo es la presentación del algoritmo más que su diseño."(página 224)

(c.3) “En la década de 1940 y 1950, se diseñaron y utilizaron muchos de esos códigos con diferentes tipos de equipos, lo que generó una lógica proliferación de problemas de comunicación. Para aliviar esta situación, el instituto ANSI adoptó el código ASCII. Este código utiliza patrones de siete bits para representar las letras mayúsculas y minúsculas del alfabeto inglés, además de los signos de puntuación, los dígitos 0 a 9 y cierta información de control, como por ejemplo las indicaciones de avance de línea, retorno de carro y tabulación.” (página 46)

Referencias

[Brookshear and Brylow, 2015] Brookshear, G. and Brylow, D. (2015). *Computer Science - An Overview*. Pearson Education Limited, 12th edition.

UNI, 11 de mayo de 2017.