



UANL



# Portafolio de Evidencias Oportunidades Extraordinarias

## Tecnología de la Información y Comunicación 2

Nombre del estudiante :

\_\_\_\_\_

Matricula: \_\_\_\_\_

Fecha: \_\_\_\_/\_\_\_\_/ 2017

Maestro: \_\_\_\_\_

**Señala la oportunidad correspondiente:**

3ª

4ª

5ª

6ª

El presente portafolio forma parte del 40% de tu calificación. Este valor se obtendrá siempre y cuando cumpla con los siguientes requisitos:

1. Escrito a mano con buena presentación (limpieza y ortografía)
2. Contestado en su totalidad y con las respuestas correctas.
3. Datos de identificación completos.
4. Se entregará únicamente al iniciar el examen al maestro que realice la guardia.
5. FAVOR DE CONTESTAR SOLAMENTE CON TINTA AZUL.

### **ADVERTENCIA**

El plagio y comercio de material académico contenido en este portafolio será sancionado en los términos de la Legislación Universitaria.

**NOMBRE:**

**GRUPO:**

**I.- DEFINIR CADA UNO DE LOS SIGUIENTES CONCEPTOS.**

- 1.- Programa
- 2.- Atributos de un programa de calidad
- 3.- Programación
- 4.- Programadores
- 5.- Lenguaje de programación
- 6.- Fases del proceso de programación y definición de cada una de ellas
- 7.- Clasificación de los tipos de datos y definir cada uno de ellos
- 8.- Símbolos que representan a los operadores aritméticos
- 9.- Símbolos que representan a los operadores de comparación
- 10.- Algoritmo
- 11.- Es la tarea que tiene el desarrollo de algoritmos
- 12.- Tipos de algoritmos
- 13.- Diagrama de Flujo
- 14.- Prueba de escritorio
- 15.- Elaborar tabla con el nombre, función y definición de cada símbolo utilizado en los diagramas de flujo.

## II.- DEFINE CADA UNO DE LOS SIGUIENTES CONCEPTOS

CONCEPTO	DEFINICIÓN	CONCEPTO	DEFINICIÓN
1) Calles 2) Avenidas 3) Manzanas 4) Esquina 5) Robot Karel 6) Cursor 7) Muros ó bardas 8) Trompos ó zumbadores 9) Mochila 10) Botón nuevo 11) Botón Abrir 12) Botón guardar 13) Botón guardar como... 14) Botón imprimir 15) Tipos de lenguaje para programar 16) Escribe el esqueleto de un programa 17) Pasos para elaborar un programa 18) Compilar 19) Botón de cortar 20) Botón de copiar 21) Botón de pegar 22) Pasos para ejecutar un programa 23) Botón adelante 24) Botón atrás 25) Botón pasos/min 26) Sensores 27) En qué instrucciones se pueden utilizar los sensores 28) frontIsClear() 29) frontIsBlocked() 30) anyBeepersInBeeperBag() 31) noBeepersInBeeperBeg() 32) rightIsClear() 33) rightIsBlocked() 34) leftIsClear() 35) leftIsBlocked() 36) nextToABeeper() 37) notnextToABeeper() 38) facingEast() 39) notFacingEast() 40) facingWest()		41) notFacingWest() 42) facingNorth() 43) notFacingNorth() 44) facingSouth() 45) notFacingSouth() 46) Bloqueado ó despejado 47) Sintaxis 48) Escribe ejemplos de comandos básicos 49) Escribe ejemplos de estatutos ó sentencias de control 50) Ejemplos de Funciones ó módulos 51) Comando move() 52) Comando turnleft() 53) Comando putbeeper() 54) Comando pickbeeper() 55) Comando turnoff() 56) Sentencia de control lterate() 57) Módulo ó función 58) Sentencia de control que se emplea en los programas para realizar una tarea <b>si</b> se cumple con una condición: 59) Sentencia de control que se emplea en los programas para realizar una tarea <b>si</b> se cumple con una condición y en caso de ser falsa cumple con otra tarea específica: 60) Sentencia de control que se emplea en los programas para realizar una tarea <b>mientras</b> se cumple la condición: 61) Objetivo de los módulos múltiples: 62) Parámetros 63) Recursividad 64) Función Succ() 65) Función Pred()	

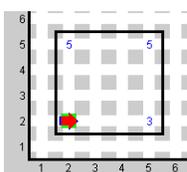
67.-Instrucción con la cual el Robot Karel deja de realizar las tareas.

68.-Código del programa cuya instrucción ó grupo de instrucciones te permite levantar dos trompos ó zumbadores en cada cuadra.

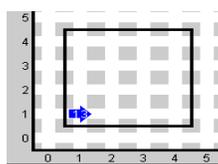
69.-¿Qué acción realiza putbeeper() en el código de un programa?

70.-Tacha el código del programa correcto para el mundo especificado:

MUNDO INICIAL



MUNDO FINAL



CÓDIGOS

DE  
PROGRAMA

```

class program {
2
3
4 void camina() {
5     while (frontIsClear()) {
6         move();
7     }
8
9 void levanta() {
10    while (nextToABeeper()) {
11        pickbeeper();
12    }
13
14
15 void dejar() {
16    while (anyBeepersInBeeperBag())
17        putbeeper();
18
19
20
21
22 program() {
23    iterate (4) {
24        camina();
25        levanta();
26        turnLeft();
27    }
28    dejar();
29
30 }

```

```

class program {
2
3
4 void camina() {
5     while (frontIsClear()) {
6         move();
7     }
8
9 void levanta() {
10    while (nextToABeeper()) {
11        pickbeeper();
12    }
13
14
15 program() {
16    iterate (4) {
17        camina();
18        levanta();
19        turnLeft();
20    }
21    turnoff();
22 }

```