

Evidencia examen primer departamental

Prof.: Miguel Ángel Rodríguez Castillo
Examen Primer Departamental

Grupo: 3CV2

No. de Boleta: 2020630675

Nombre completo: Colín Ramiro Joel

Instrucciones: Responde **a mano** y con tus **propias palabras** lo que se te pide en una hoja (Teoría y Ejercicios) una vez que hayas concluido, escanea o toma fotografía y sube tu examen a la Plataforma Microsoft Teams para su revisión.

Parte I. Teoría (4 pts.)

1. Explica con tus propias palabras qué es un Paradigma de Programación.

Es un estilo de programar, según las necesidades de los que se dedican a la programación y sus clientes

2. Explica algunas de las clasificaciones de lenguajes de programación.

Alto nivel: Son la más parecida al lenguaje humano por ende son más sencillos
Bajo nivel: Estos se concentran directo en el hardware, están condicionados a la estructura de la computadora.

3. Menciona y explica 3 criterios de los Lenguajes de programación:

Fácil de escribir: es la medida de que tan fácil resulta usar ese lenguaje
Confiable: verifica los programas para probar su corrección
Eficiente: que tantos recursos ocupa, el tiempo y espacio y el esfuerzo humano.

4. Con tus propias palabras explica ¿Qué es el paradigma imperativo? y que Lenguajes de Programación utilizan este paradigma.

Es el paradigma que se enfocan en el "cómo" y no en el "qué", los lenguajes que lo utilizan son: Fortran, Cobol, Basic, Pascal, C, ADA, etc.

5. Con tus propias palabras explica ¿Qué es el paradigma declarativo? y que Lenguajes de Programación utilizan este paradigma.

Es el paradigma, el contrario del imperativo que se enfocan en el "qué" los lenguajes que lo utilizan son: Prolog, Lisp, Haskell, Miranda, Erlang, etc.

6. Define la programación pura.

Es la programación cuya solución tiene sentido solamente si una o todas las variables son números enteros. Trabajan con funciones puras, las cuales con la misma entrada, devuelven la misma salida.

7. ¿Cómo se le denomina a las funciones que reciben otras funciones como parámetro?

funciones de orden superior

8. Escribe la sintaxis de la función Lambda.

(arguments) -> {cuerpo}

9. Escribe dos diferencias entre la programación lógica y la programación funcional.

La programación lógica se concentra en el predicado y la funcional se concentra en la función

Parte II. Ejercicios

1. Instrucciones: **Desarrolla el código** que solucione el siguiente programa (en lenguaje programación que desees) bajo el paradigma de programación funcional y realiza **dos pruebas de escritorio: (2 pts.)**

- Escribe una función recursiva que se comporte como el operador ++

Se evaluará:

- Código fuente.
- Dos pruebas de escritorio.

```
#include <iostream>
using namespace std;
int inc(int, int);
int main(){
    int n1,n2;
    cout << "Introduzca el primer numero: ";
    cin >> n1;
    cout << "Introduzca el segundo numero: ";
    cin >> n2;
    cout << "La suma es: " << inc(n1,n2) << endl;
    system("pause");
}
int inc(int a, int b)
{
    int inc;

    if(b==0)
        return a;
    else if(a==0)
        return b;
    else
    {
        return 1+inc(a,b-1);
    }
}
```

Pruebas de escritorio

1.-

inc	n
1	1+1
2	1+2
3	1+3
4	1+4
5	1+5
6	1+6

2.-

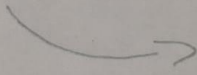
res	inc
2	1
3	2
4	3
5	4
6	5
7	6

2. Instrucciones: Evalúa con las técnicas de programación (eager y lazy) la expresión:
(1 pto.)

prim(multi(cuad 5, cuad 3), divide(1, prim(0, cuad 4)))

Supóngase que se definen las funciones

- o $\text{cuad } x = x * x$
- o $\text{prim}(x, y) = x$
- o $\text{multi}(x, y) = x * y$



2- eager

$$\begin{aligned} &= \text{prim}(\text{multi}(5*5, 3*3), \text{divide}(1, \text{prim}(0, \text{cuad } 4))) \\ &= \text{prim}(25*9, \text{divide}(1, 0)) \\ &= \text{prim}(225, \text{divide}(1, 0)) \\ &= 225 \end{aligned}$$

lazy

$$\begin{aligned} &= \text{prim}(\text{multi}(5*5, 3*3), \text{divide}(1, \text{prim}(0, \text{cuad } 4))) \\ &= \text{multi}(5*5, 3*3) \\ &= 25*9 \\ &= 225 \end{aligned}$$

3. Instrucciones: • Dados los siguientes Hechos en Prolog: (3 pts.)

```
protagonista(shaggy,masculino).  
protagonista(fred,masculino).  
protagonista(daphne,femenino).  
protagonista(velma,femenino).  
protagonista('scooby doo',masculino).  
protagonista('scrappy doo',masculino).  
humano(shaggy).  
humano(fred).  
humano(daphne).  
humano(velma).  
perro('scooby doo').  
perro('scrappy doo').
```

- Escribe una regla héroe(x) que identifique aquellos protagonistas que sean masculinos y humanos:

héroe(x):- protagonista(X,masculino), humano(X).

- Escribe una regla heroína(x) que identifique aquellos protagonistas que sean femeninos y humanos:

heroína(x):- protagonista(X,femenino), humano(X).

- Escribe una regla estrellas(x) que identifique aquellos protagonistas que sean masculinos y perros:

estrellas(x):- protagonista(X,masculino), perro(X).

- Realiza las siguientes consultas:

1. Muestra los protagonistas. `protagonista(X,Y).`
2. Muestra los héroes. `héroe(X).`
3. Muestra las heroínas. `heroína(X).`
4. Muestra las estrellas. `estrellas(X).`
5. Pregunta si hay protagonistas masculinos. `protagonista(X,masculino).`

3.-

1- x= shaggy
y= masculino;
x= fred
y= masculino;
x= daphne
y= femenino;
x= velma
y= femenino;
x= 'scooby doo'
y= masculino;
x= 'scrappy doo'
y= masculino;

2- x= shaggy;
x= fred;

3.- x= daphne
x= velma;

4- x= 'scooby doo'
x= 'scrappy doo'

5.- x= shaggy.
x= fred.
x= 'scooby doo'.
x= 'scrappy doo'.