

Paradigmas de Programación

PA

15 de Diciembre de 2014

Rut: _____
Profesor: _____

Instrucciones:

- Contestar cada parte de la prueba en hojas separadas (**No páginas/planas**). Cada hoja debe incluir su rut y nombre del profesor
- El incumplimiento del requisito anterior implicarán descuentos en su prueba de hasta 2 puntos.
- Para las preguntas de alternativas y de verdadero/falso, cada tres respuestas incorrectas se descuenta un punto.

Parte I: Imperativo (8 pts)

- 1) Se ha detectado que la invocación a la siguiente función en C es demasiado lenta. ¿Qué problema detecta y qué cambios le haría para mejorar esta situación?.

```
#define PASADAS 100
struct clavesCifrado {
    char chaveA[2048];
    char claveB[2048];
    char claveC[1024];
    char claveD[1024];
}
char *cifradoCompuesto(struct clavesCifrado claves, char *arrayDatosACifrar, int
length) {
    char *arrayCifrado = (char *)malloc(length*sizeof(char));
    int i;
    for (i = 0; i < length; i++) { arrayCifrado[i] = arrayDatosACifrar[i]; }
    for (i = 0; i < PASADAS; i++) { cifradoSimple(claves, arrayCifrado, length); }
    return arrayCifrado;
}
```

El problema es que al llamar la función se copia completamente la estructura al stack de llamadas (6kb de datos), lo cual es muy costoso, en comparación con sólo copiar una dirección de memoria (**4 pts**).

Como solución se debe utilizar paso de parámetros por referencia (**4 pts**).

Parte II: Funcional (15 pts)

- 2) **(5 pts)** Modifique el siguiente código para que se utilice recursión de cola y la función entregue el mismo resultado. Mantenga la misma cabecera de la función para encapsular el tipo de recursión empleada.

```
(define (algo x)
  (if (null? x)
      1
      (+ 1 (algo (cdr x)))))
)
```

Solución

```
(define (algo x)
  (define (algoAux x i)
    (if (null? x)
        1
        (algoAux (cdr x) (+ 1 i))))
  (algoAux x 1))
)
```

La función algoAux también puede ser definida fuera de la función no obstante de hacerlo de esta manera tiene un descuento de 1 punto, ya que según enunciado se busca encapsular.

- 3) **(10 pts)** Realice una función en Scheme que busque un elemento en una lista a partir de un criterio definido mediante una función booleana *f* y entregue la posición en la lista (comenzando con la posición 0 para el primer elemento) donde se encuentra la primera ocurrencia del elemento buscado de izquierda a derecha. La función requerida se debe invocar de la siguiente forma:

```
(busqueda '(5 2 88 "hola" -4) f "hola")
```

donde *f* en este caso es

```
(define (f p1 p2)
  (equal? p1 p2))
)
```

Luego, para el ejemplo anterior el retorno esperado es 3. En caso de no encontrarse el elemento a partir del criterio definido por la función f se debe retornar -1. De ser necesario puede crear más funciones además de la función 'busqueda'.

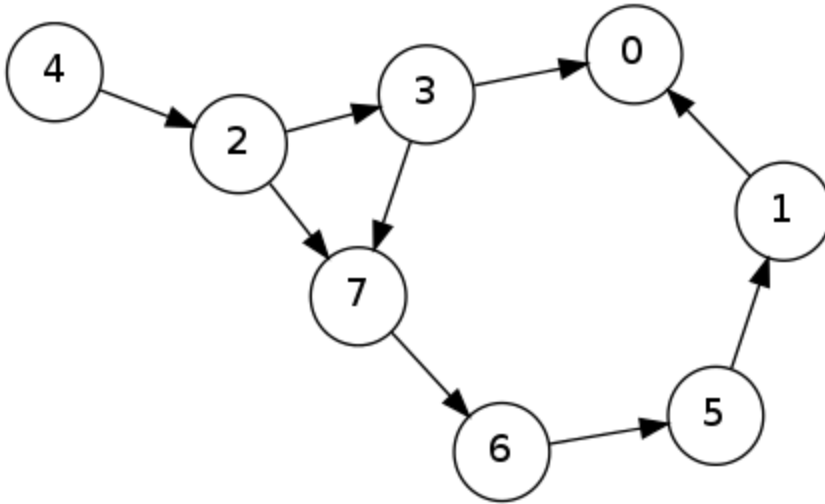
```
(define (busqueda lista func elem)
  (busquedaRec lista func elem 0)
)

(define (busquedaRec lista func elem acum)
  (if (null? lista)
      -1
      (if (func elem (car lista))
          acum
          (busquedaRec (cdr lista) func elem (+ 1 acum))
      )
  )
)
```

Parte III: Lógico (7 pts)

- 1) Una máquina generadora de números aleatorios tiene la particularidad de que no son tan aleatorios como se cree, sino que más bien, los genera con una cierta secuencia. Se ha estudiado por años esta máquina y se ha podido constatar que los números que genera están definidos por un diagrama. Este se muestra en la Figura 1 y se han descrito los hechos.

contiguo(4, 2).
contiguo(2, 3).
contiguo(2, 7).
contiguo(3, 0).
contiguo(3, 7).
contiguo(1, 0).
contiguo(7, 6).
contiguo(6, 5).
contiguo(5, 1).



Se ha solicitado a usted realizar una solución que permita preguntar por los valores que podría tomar estos números, para ello, realice en Prolog:

a. Las reglas para determinar:

i. (1 pto.) Existe un número con una determinada serie de dos dígitos contiguos.

Ejemplo:

?- serie (3 , 0) .

true .

serie(X,Y) :- contiguo(X,Y).

ii. (4 ptos.) ¿Puede haber un número aleatorio con dos números no necesariamente contiguos?.

?- noContiguo (6 ,X) .

X = 5 ;

X = 1 ;

X = 0 ;

false .

noContiguo(X,Y) :- contiguo(X,Y).

noContiguo(X,Y) :- contiguo(X,Z) , noContiguo(Z,Y).

iii. (2 ptos.) Es posible terminar con un determinado número. Ej:

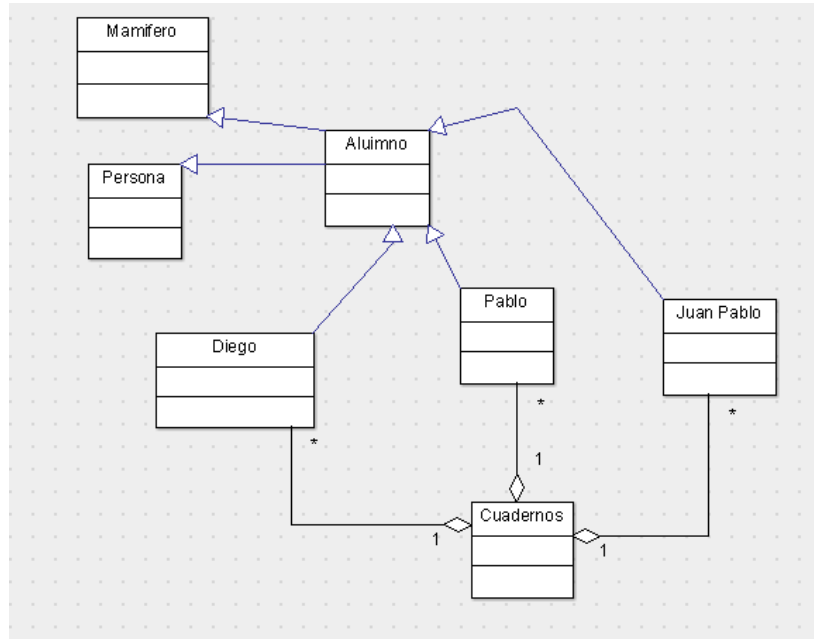
?- t e r m i n a r (4) .

false

terminar(X) :- contiguo(Y,X).

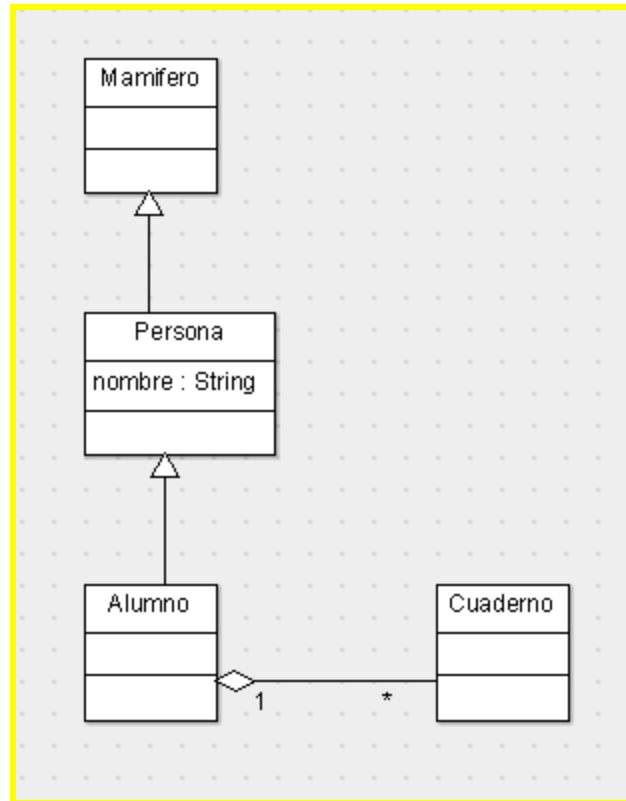
Parte IV: Orientado a Objetos (10 pts)

- 1) **(10 pts)** Considere el siguiente diagrama de clases para una implementación en C#. Indique problemas que se puedan desprender del diagrama y provea un diagrama donde se ilustre una alternativa para solucionarlos.



Una solución

- Se aprecia herencia múltiple la cual no es posible implementarla en C# **(2 pts)**
- Las clases Diego, Pablo y Juan Pablo no ameritan ser clases. Corresponden a atributos **(2 pts)**
- Las relaciones de agregación están conceptualmente invertidas. Ya que un cuaderno no tiene alumnos (o para el caso del diagrama, no posee DiegoS, PabloS o Juan PabloS. Es el alumno el que tiene CuadernoS. **(2 pts)**
- Diagrama con soluciones propuestas **(4 pts.** si y solo si cubre cada uno de los problemas señalados anteriormente)



Otra posible solución para la herencia múltiple es (en base a la misma respuesta anterior) hacer que “Mamífero” y/o “Persona” sean una interfaz, luego el “Alumno” debe implementar estas interfaces.

Parte V: Concurrente (5 pts)

- 1) Si se tienen un thread A encargado de escuchar por eventos de la interfaz gráfica en una ventana de chat y otro thread B que escucha mensajes desde la red, ¿Es posible que el thread B escriba directamente información en la vista de chat? ¿Por qué?.

No es posible porque los elementos de la vista no son thread-safe y sólo el thread de la vista puede modificarlos, C# lanza una excepción. (basta con decir que no se puede y alguno de los argumentos anteriores para tener el puntaje completo)

Parte VI: Eventos (10 pts)

- 1) Considere las clases Cliente y Tienda y el delegado NotificacionPreciosBajos.

Complete las clases Cliente y Tienda según corresponda de manera que los clientes sean notificados cuando se registre una baja de precios en los productos de la tienda.

En la hoja de respuestas puede escribir sus líneas de código señalando los números de líneas (predecesoras y sucesoras) donde deberían ir ubicadas..

```
public delegate void NotificacionPreciosBajos(string producto, float precio, precioAntiguo);
```

```

1. class Cliente {
2.     public void notificar(string producto, float precioAntiguo, float PrecioNuevo) {
3.         Console.WriteLine("El producto "+producto+" ha bajado su precio desde "+precioAntiguo+" a
4.         "+precioNuevo);
5.     }}

6. class Tienda {
7.     public event NotificacionPreciosBajos notificador; (2 pts) //Entre lineas 6 y 7
8.     List<Producto> productos = new List<Producto>();
9.     //Producto es una clase con propiedades nombreProducto y precio
10.    public void cambiarPrecio(int posicion, float precioNuevo) {
11.        Producto p = productos[posicion]; //Asuma que las condiciones de borde para la lista de
12.        //productos están controladas
13.        if (p.precio>precioNuevo) (1 pto) //Entre lineas 11 y 12
14.        if (notificador!=null) (1 pto) //Entre lineas 11 y 12
15.        notificador(p.nombreProducto, p.precio, precioNuevo); (2 pts) //Entre 11 y 12
16.    }
17.    p.precio = precioNuevo;
18. }}

19. public class Program{
20.     public static void Main(string[] args){
21.         Tienda t = new Tienda();
22.         Cliente c1 = new Cliente();
23.         Cliente c2 = new Cliente();
24.         t.notificador += new NotificacionPreciosBajos (c1.notificar); (2 pts) //Entre 18 y 19
25.         t.notificador += new NotificacionPreciosBajos (c2.notificar); (2 pts) //Entre 18 y 19
26.         t.cambiarPrecio(2, 300);
27.     }}

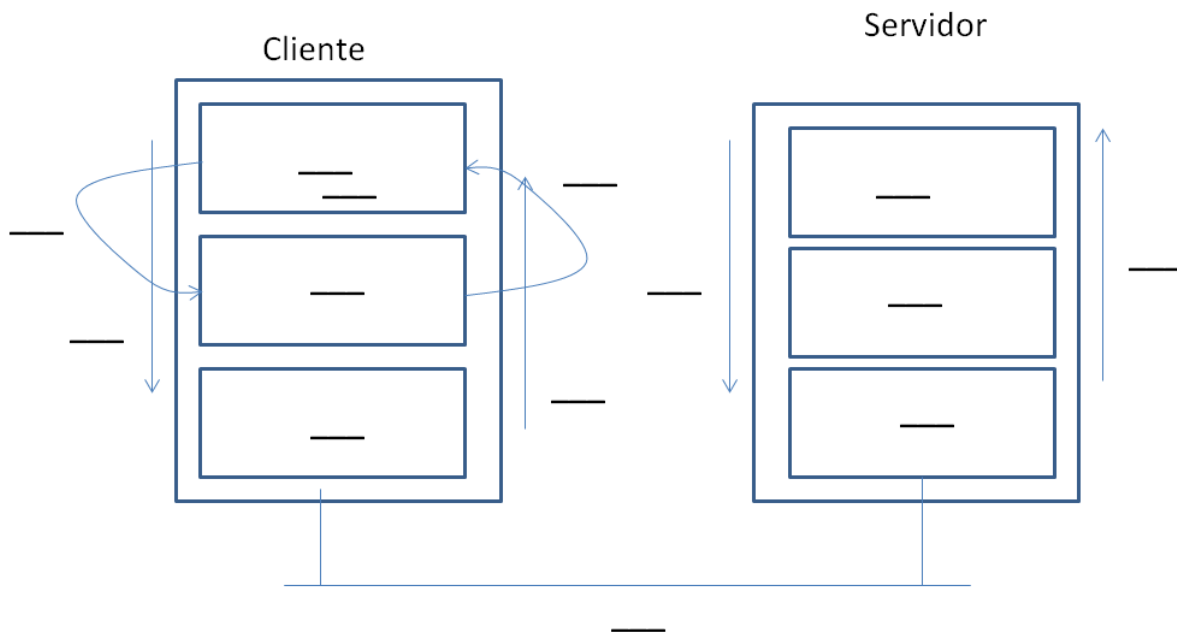
```

Parte VII: Servicios (5 pts)

- 1) A continuación se provee un listado de términos y un diagrama simplificado relativos a la implementación de un sistema distribuido mediante el uso de servicios Web. Disponer cada una de las letras asociadas a los términos (a, b, c, ...) en el diagrama de la derecha en los lugares que corresponda. Nota: Algunas letras pueden aparecer mas de una vez en el diagrama.

- a. Proxy
- b. Empaquetado Invocación

- c. Desempaquetado
- d. Protocolo
- e. Inovocación
- f. Aplicación Cliente
- g. Servicios Web
- h. Respuesta
- i. Red
- j. Sistema Operativo
- k. XML
- l. Servidor Web
- m. Empaquetado Respuesta



Solución

(0,3 puntos cada una)

