Apellido y Nombres:……. Legajo:..... Máquina:...

Segundo Parcial de Paradigmas de Programación

Objetivo

Evaluar al estudiante en la parte práctica de las unidades nro. 5 y nro. 6 (Paradigma Lógico y Paradigma Funcional, respectivamente) a partir de la resolución que guarde en los archivos más abajo especificados, correspondientes a las consignas solicitadas para los ejercicios de cada paradigma..

Condiciones de trabajo:

- Este parcial práctico consta de dos partes: programación lógica y programación funcional.
  Para cada paradigma se deberá desarrollar un programa, utilizando el correspondiente entorno.
- Para resolver las consignas correspondientes al paradigma lógico, se deberá generar un archivo con el nombre Legajo_ApellidoNombre.pl para definir los hechos y reglas, conforme se solicite en las consignas que se detallan más abajo. También se deberá generar un archivo con el nombre Legajo_ApellidoNombre.txt para formular los objetivos solicitados más abajo y su correspondiente respuesta de Prolog.
- Para resolver las consignas correspondientes al paradigma funcional, se deberá generar un archivo con el nombre Legajo_ApellidoNombre.hs para formular las funciones que más abajo serán solicitadas.
- Es responsabilidad de cada alumno ir guardando periódicamente cada archivo solicitado, como así también del contenido de los mismos, teniendo la precaución de guardarlo en el disco D: para su posterior backup.
- En caso de que máquina no funcione correctamente durante el transcurso de la evaluación, debe notificar de esta situación a cualquier docente de la mesa examinadora.
- En ningún caso debe reiniciar la máquina, ya que perderá la totalidad del examen.
- El **tiempo previsto** para la realización de este examen es de **1:30 hs**.

Paradigma Lógico

Enunciado

En una cafetería de la ciudad se necesita un programa en prolog para registrar los pedidos que reciben los mozos por parte de los clientes. Para ello registran los mismos de la siguiente manera:

Tabla 1: Pedidos

Número de Orden	Número de Mesa	Nombre del Mozo	Artículo		Código de promoción
			Código de Artículo	Cantidad	
1	11	Juan	1	1	1
2	12	Pedro	2	2	0
3	14	Lucas	5	1	1
4	11	Juan	3	3	1
5	18	Juan	5	2	3
6	12	Pedro	4	4	0
7	10	Lucas	1	3	2
8	14	Juan	4	2	2

Tabla 2: Artículos

Código de Artículo	Nombre	Precio
1	Cafe	40
2	Cafe con leche	55
3	Factura	10
4	Criollo	6
5	Desayuno completo	85
6	Desayuno light	70

Tabla 3: Promociones

Código de promoción	Descuento
0	0%
1	50%
2	25%
3	10%

Se necesita de un programa funcional que permita obtener cierta información al respecto.

Su tarea:

A partir de los hechos ya definidos que representan todos los datos de las tablas 1, 2 y 3 usted deberá definir las reglas que permitan resolver lo siguiente:

1) Obtener una lista que contenga los nombres de todos los artículos solicitados en los pedidos. En el caso de que un artículo haya sido pedido más de una vez, el mismo debe aparecer una única vez en la lista. Nombre de la regla: regla1/1 (15 puntos).

2) Indicar si se vendió o no algún artículo en particular indicado por su nombre. Nombre de la regla: regla2/2 **(15 puntos)**

3) Calcular la suma total de la facturación de un mozo indicado por su nombre. Para calcular la facturación de cada pedido debe sumarse el el importe del artículo (precio * cantidad) y luego aplicar el descuento correspondiente a la promoción si corresponde. Nombre de la regla: regla3/2 **(20 puntos)**

Paradigma Funcional

Dada la misma tabla de promociones presentada para el programa del paradigma lógico, desarrollar las siguientes funciones en Haskell:

1) Una función que reciba precio unitario y cantidad de un artículo solicitado y el código una promoción. Con esos datos la función debe calcular a abonar por el cliente, considerando el descuento correspondiente. Si el código de promoción no existe en la tabla realizar el cálculo sin aplicar descuentos. **(15 puntos)**

La siguiente lista contiene el total facturado por la cafetería durante los días del mes en curso. Con respecto a los días en los que el comercio no abre, la lista contiene 0.

```
lista :: [Integer]
lista = [23454, 12432, 7890, 10000, 0, 0, 12345, 5555]
```

2) Realizar una función que reciba por parámetro la lista de totales facturados y retorne una nueva lista que contenga la facturación de los días en los que la misma sea menor a $8000. No incluir en la lista los datos correspondientes a los días en que no abre el comercio. **(15 puntos).**

3) Realizar una función que calcule el promedio de facturación de los días con facturación mayor a $10000. Resolver utilizando recursividad. **(20 puntos)**