

---

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

## **Segundo Parcial de Paradigmas de Programación – TEMA 2**

### **Objetivo**

Evaluar al estudiante en la parte práctica de las unidades nro. 5 y nro. 6 (Paradigma Funcional y Paradigma Lógico, 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.
- 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

Una estación de servicio necesita un programa en Prolog que le brinde cierta información sobre las ventas realizadas y las promociones que ofrece.

Se disponen de 3 tablas, con información referida a las promociones que tiene disponibles, las ventas realizadas, y los tipos de combustibles que tiene a la venta.

La **tabla 1** corresponde a las diferentes promociones, las cuales se pueden asignar al azar a cada venta, y luego evaluar la posibilidad de aplicar un descuento sobre el importe de la misma, en caso de corresponder.

El porcentaje de descuento de una promoción se aplicaría al ticket de venta, sólo en caso que dicha promoción haya sido asignada a la venta en cuestión, y la cantidad de litros de combustible de la venta asociada, sea mayor o igual a la cantidad de litros de combustible que especifica la promoción. En tal caso, el porcentaje de descuento a aplicar es el que figura en la correspondiente promoción; en caso contrario, el porcentaje de descuento será 0. Se aclara que todas las promociones pueden aplicarse a cualquier tipo de combustible que se haya vendido.

**Tabla 1: Promociones**

Código de promoción	Slogan	Cantidad mínima de litros de combustible	Porcentaje de descuento
111	'Cargando 10 litros o más de cualquier combustible, descuento del 8%'	10	8
222	'Cargando 8 litros o más de cualquier combustible, descuento del 5%'	8	5
333	'Cargando 12 litros o más de cualquier combustible, descuento del 10%'	12	10

La **tabla 2** corresponde a las diferentes ventas de combustible que se realizaron. De cada venta se registran los siguientes datos:

- Código de venta: es un valor numérico, único para cada venta.
- Código de promoción asignado: se asigna siempre un código de promoción al azar a cada venta realizada. Como ya se explicó antes, el porcentaje de descuento de la promoción se aplicará sólo en caso que la cantidad de litros de combustible vendido en ese ticket de venta sea mayor o igual a la cantidad de litros de combustible especificada en la promoción. En caso contrario, el porcentaje de descuento será 0.
- Fecha de venta: la cual se expresa en términos de día, mes y año, y corresponde a la fecha en la que se realizó dicha venta de combustible.

- Surtidor: representa el código del surtidor que se utilizó para la venta de combustible.
- Litros cargados: representa la cantidad exacta de litros que se cargaron del correspondiente combustible.

**Tabla 2: Venta de combustible**

Código de venta	Código de promoción asignado	Fecha de venta			Surtidor	Litros cargados [litro]
		Día	Mes	Año		
1	111	29	9	2019	'A'	12.5
2	222	16	10	2019	'B'	10.0
3	111	16	10	2019	'D'	15.5
4	111	18	10	2019	'H'	8.0
5	333	18	10	2019	'I'	12.0

La **tabla 3** corresponde a los diferentes tipos de combustibles que se pueden vender, el importe por litro de cada uno, y una lista de surtidores que los abastecen.

**Tabla 3: Importe por litro de combustible y lista de surtidores que lo abastecen.**

Código del ítem	Descripción	Precio por litro [\$/litro]	Surtidores
1	Súper	45.67	['A', 'B', 'C']
2	Premium	52.97	['D', 'E']
3	Diésel	44.69	['F']
4	Diésel Premium	50.84	['G']
5	GNC	20.99	['H', 'I', 'J']

### 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) Conocer el código de venta, slogan de la promoción asignada, mes y año de la venta, de aquellas ventas en las que se haya vendido más de 12 litros de cualquier combustible. Predicado sugerido: regla1/4. **(15 puntos)**

*Por ejemplo:*

**regla1(CV,SloganPromo,Mes,Año).**

CV = 1,

SloganPromo = 'Cargando 10 litros o más de cualquier combustible, descuento del 8%',

Mes = 9,

Año = 2019 ;

CV = 3,

SloganPromo = 'Cargando 10 litros o más de cualquier combustible, descuento del 8%',

Mes = 10,

Año = 2019 ;

false.

- 2) Para un determinado código de venta que se especifica, conocer el porcentaje de descuento en caso de corresponderle, según la promoción asignada. Tener en cuenta que en caso de no corresponderle, este porcentaje de descuento será 0. Predicado sugerido para esta regla: regla2/2. **(15 puntos)**

*Por ejemplo:*

**regla2(1,PorcDes).**

PorcDes = 8.

**regla2(2,PorcDes).**

PorcDes = 5.

**regla2(4,PorcDes).**

PorcDes = 0.

- 3) Generar una lista con aquellos códigos de ventas en las que se haya cargado cierto combustible cuyo nombre se especifica. Luego de generar el listado, ordenar sus elementos, ascendentemente por código de venta. Predicado sugerido para esta regla: regla3/2. **(20 puntos)**

*Por ejemplo:*

**regla3('súper',Lista).**

Lista = [1, 2].

**regla3('GNC',Lista).**

Lista = [4, 5].

### Paradigma Funcional

#### Enunciado

Se requiere un conjunto de funciones en Haskell que contribuyan a la generación de un texto encriptado a partir de un conjunto acotado de reemplazos que se mencionarán más adelante, como pautas a aplicar sobre un texto que se recibe como parámetro. A continuación se detallan las funciones a codificar:

**1)** Realizar una función que reciba como parámetro de entrada el carácter correspondiente a una vocal (en minúscula y sin tilde, tal como se muestra en la tabla 1), y devuelva otro carácter correspondiente a un dígito, de acuerdo a la siguiente tabla:

**Tabla 1. Codificación de vocales**

Vocal	Código
'a'	'1'
'e'	'2'
'i'	'3'
'o'	'4'
'u'	'5'

En caso que la función reciba cualquier otro carácter que no se corresponda con ninguna de las vocales especificadas en la tabla 1, la función deberá devolver el carácter que represente el número 0. **(15 puntos)**

Ejemplos: `Main> funcion1 'a'`  
`'1'`

`Main> funcion1 'o'`  
`'4'`

`Main> funcion1 'A'`  
`'0'`

`Main> funcion1 'b'`  
`'0'`

**2)** Realizar una función que reciba como parámetro de entrada un carácter, y devuelva True en caso que se corresponda a una vocal en minúscula y sin tilde (tal como se especifica en la primera columna de la tabla 1), o un False en caso de no corresponderse a ninguna de las vocales anteriormente mencionadas en la tabla 1. **(15 puntos)**

Ejemplos: `Main> validar 'a'`  
`True`

`Main> validar 'b'`

```
False
Main> validar 'A'
False

Main> validar 'o'
True
```

**3)** Realizar una función que reciba una cadena de caracteres, la procese carácter a carácter, y genere otra cadena en la que sólo se reemplace cada carácter vocal especificado en la tabla 1, por el código correspondiente de acuerdo a la ya mencionada tabla1. Todo carácter que no se corresponda con alguna de las vocales que figuran en dicha tabla, deberán ser mantenidos inalterados en la cadena a generar. Aplicar recursividad. **(20 puntos)**

Ejemplos: `Main> codificar "Paradigmas"`  
`"P1r1d3gm1s"`

```
Main> codificar "Lenguaje Haskell"
"L2ng51j2 H1sk211"
```

```
Main> codificar "Estoy rindiendo el segundo parcial"
"Est4y r3nd32nd4 21 s2g5nd4 p1rc311"
```