

## **Trabajo Práctico Integrador Nro 2.**

### **Unidad 4: Paradigma Funcional.**

#### **Objetivos:**

El objetivo del presente trabajo práctico integrador es la evaluación de los temas de la unidad número 4: Paradigma de Programación Funcional, que se detallan a continuación: definición de funciones, uso de expresiones como guardas, if – then – else, otherwise, entre otras, listas por enumeración y por comprensión, recursividad.

#### **Caso de estudio:**

Una empresa de transporte de bienes ofrece el servicio de delivery de diversos comercios a los domicilios de los clientes. La empresa cobra por su servicio una comisión equivalente a un porcentaje del importe compra de cada ticket dependiendo del tipo de comercio. A continuación se muestra una tabla en la que se detallan las descripciones de los tipos de comercios para los cuales se realiza el delivery, su código, y el porcentaje que se aplica sobre el monto de la compra, a modo de comisión.

<b>Código comercio</b>	<b>Denominación</b>	<b>Porcentaje</b>
<b>1</b>	Gastronomía	15%
<b>2</b>	Supermercado	10%
<b>3</b>	Librerías	7%
<b>4</b>	Farmacia	1%

El servicio de transporte es solicitado por cada comercio cuando un cliente le efectúa una compra con entrega a domicilio, y en ese momento se informa al delivery: el tipo de comercio, el importe de la compra, y la cantidad de dinero en efectivo con el cual el cliente abonará. El importe total que el cliente deberá abonar por el ticket de su compra es el importe de la compra que figura en el ticket más el recargo por el delivery (cuyo porcentaje depende del tipo de comercio) que se aplica sobre el importe de la compra de acuerdo al porcentaje correspondiente al tipo de comercio. Es decir, al importe de la compra se le suma la comisión por el delivery, y así se obtiene el monto total a cobrar al cliente.

El transportista entonces se dirige hasta el comercio y retira el paquete a entregar y el dinero en efectivo necesario para dar en vuelto al cliente; posteriormente luego de finalizar la entrega en forma satisfactoria, vuelve al comercio a entregar el dinero recaudado.

Se solicitan los siguientes puntos:

1. Una función que dado un código de comercio retorne el porcentaje correspondiente a la comisión. Utilizar guardas. En caso de recibir un código no especificado, la función deberá devolver 0.
2. Una función dado el importe de compra de un ticket y el código del comercio devuelva el total a cobrar al cliente.
3. Una función que reciba el total a cobrar a un cliente y la cantidad de dinero en efectivo con el cual el cliente abonará. Con esos datos la función debe calcular el vuelto que tiene que entregarse. Validar que la cantidad de dinero en efectivo con el que se abonará, sea mayor o igual que el total a cobrar. En caso contrario, retornar -1. Usar if then else.
4. Una función que reciba una lista con el importe de compra de un conjunto de tickets y otra lista con los códigos de comercio de dichos tickets. Con esos datos debe retornar una lista conteniendo el monto total a cobrar por cada uno de esos tickets. Las dos listas que se reciben deben ser del mismo tamaño. En caso de no cumplirse esta restricción debe devolverse una lista vacía. La lista retornada también debe ser del mismo tamaño que las de entrada. Utilizar recursividad.
5. Una función que reciba una lista con el total a cobrar de un conjunto de clientes, y otra lista con la cantidad de dinero en efectivo con la cual abonará cada uno de dichos clientes. La función debe retornar el vuelto que debe entregarse a cada uno de los clientes. Si un cliente abona con la cantidad de dinero exacta del importe a cobrar, el vuelto es 0, y por lo tanto **no debe incluirse** en la lista retornada. Ambas listas de ingreso deben ser del mismo tamaño pero la retornada puede contener menos elementos. Validar que ambas listas que se reciben tengan el mismo tamaño. En caso que no tengan el mismo tamaño ambas listas, o en caso que la cantidad de dinero con la que el cliente vaya a abonar no sea mayor o igual al importe total a cobrar, se deberá devolver una lista vacía. Se deberá utilizar para resolver este punto, lista por comprensión.

### **Tabla de valoración de los ítems evaluados**

Nro. de ítem	Ítems o requerimiento a evaluar	Puntaje	Observaciones	Obtenido
1	Definición correcta de la función solicitada: argumentos y tipo de retorno. Utilización apropiada de expresiones y funciones provistas por el lenguaje.	10		
2	Definición correcta de la función solicitada: argumentos y tipo de retorno. Utilización apropiada de expresiones y funciones provistas por el lenguaje.	10		
3	Definición correcta de la función solicitada: argumentos y tipo de retorno. Utilización	10		



	apropiada de expresiones y funciones provistas por el lenguaje.			
4	Definición correcta de la función solicitada: argumentos y tipo de retorno. Utilización apropiada de expresiones y funciones provistas por el lenguaje. Utilización de recursividad.	20		
5	Definición correcta de la función solicitada: argumentos y tipo de retorno. Definición de funciones auxiliares. Composición de funciones. Utilización de listas por comprensión.	20		
	<b>Total</b>	<b>100</b>		

### Condiciones de entrega:

- Este trabajo práctico integrador **se deberá realizar en forma grupal**.
- Deberán nombrar el **archivo comprimido** con el siguiente formato: TP2\_NroLeg1erIntegranteApellido1erIntegrante\_NroLeg2doIntegranteApellido2doIntegrante\_NroLeg3erIntegranteApellido3erIntegrante. El orden en el cual deberán colocar los legajos y apellidos cada integrante en el nombre del archivo comprimido será de acuerdo al orden alfabético de los apellidos de los integrantes, desde la A hasta la Z.
- Se deberá subir una carpeta comprimida que contenga en su interior el archivo .hs correspondiente a la codificación en Haskell, además debe contener un archivo .txt con las invocaciones y resultados de las funciones solicitadas para hacer funcionar el programa;
- Plazo máximo de entrega de este trabajo: **HASTA el ##/##/## a las ##:## hs.**
- Todas las consultas de este TP2 pueden realizarlas a través del foro del aula virtual.

### Instancia de recuperación del trabajo práctico:

En caso de recuperar el trabajo práctico nro. 2 de programación funcional, se deberán presentar todas las funciones correspondientes a la instancia de entrega original, funcionando correctamente, más las funciones correspondientes a los siguientes puntos:

6. Una función que reciba dos listas con los datos de un conjunto de tickets, cada una de ellas contiene:
  - a. El importe de compra de cada ticket.
  - b. El código de comercio de cada ticket.

La función debe retornar una lista con las comisiones a cobrar por cada ticket, para aquellos cuyo tipo de comercio sea 1 o 2. Utilizar listas por comprensión.

Validar que ambas listas tengan el mismo tamaño. Si no tuvieran el mismo tamaño, se deberá devolver la lista vacía.

7. Una función que reciba tres listas con los datos de un conjunto de tickets, cada una de ellas contiene:

- a. El importe de compra de cada ticket.
- b. El código de comercio de cada ticket.
- c. La cantidad de dinero en efectivo con el que abonará cada cliente su ticket.

La función debe retornar la cantidad de tickets a los que hay que entregar al cliente correspondiente un vuelto entre \$10 y \$100. Utilizar recursividad.

Validar que las tres listas tengan el mismo tamaño. En caso que no tuvieran el mismo tamaño las tres listas, o en caso que se detecte que la cantidad de dinero con el que el cliente abone algún ticket sea menor al importe total a pagar, se deberá devolver -1.

8. Una función que reciba tres listas con los datos de un conjunto de tickets, cada una de ellas contiene:
  - a. El importe de compra de cada ticket.
  - b. El código de comercio de cada ticket.
  - c. La cantidad de dinero en efectivo con el que abonará su ticket, cada cliente.

La función debe retornar una lista con los códigos de comercio de aquellos tickets en los cuales no se requiera dar vuelto. Utilizar recursividad.

Validar que las tres listas tengan el mismo tamaño. En caso que no tuvieran el mismo tamaño las tres listas, o en caso que se detecte que la cantidad de dinero con el que el cliente abone algún ticket sea menor al importe total a pagar, se deberá devolver la lista vacía.

### **Tabla de valoración de los ítems evaluados en la instancia de recuperación**

Es condición necesaria para aprobar la instancia de recuperación que se presenten todas las funciones correspondientes a la instancia original correctamente implementadas.

Nro. de ítem	Ítems o requerimiento a evaluar	Puntaje	Observaciones	Obtenido
1 a 5	Implementación correcta de todas las funciones correspondientes a la instancia original.	30		
	Definición correcta de la función solicitada: argumentos y tipo de retorno. Definición de funciones auxiliares o validaciones. Composición de funciones. Listas por comprensión.	20		
6	Definición correcta de la función solicitada: argumentos y tipo de retorno. Definición de funciones auxiliares o validaciones. Composición de funciones. Definición de recursión.	25		
7	Definición correcta de la función solicitada: argumentos y tipo de retorno. Definición de funciones auxiliares o validaciones. Composición de funciones. Definición de recursión	25		
	<b>Total</b>	<b>100</b>		