



## Recuperatorio

Nota: ..... (      )

### Paradigmas de Programación

Fecha: 31/10/15

[Práctico]

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

### Recuperatorio 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 correspondiente 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.



## Recuperatorio

Nota: ..... ( )

### Paradigmas de Programación

Fecha: 31/10/15

[Práctico]

- **Es responsabilidad de cada estudiante el contenido que quede guardado al momento de finalizar la evaluación, en todos sus archivos generados durante el examen.**
- El **tiempo previsto** para la realización de este examen es de 1 y 30 **hs.**

#### Evaluación de Programación Lógica

##### Caso de estudio:

Una empresa de televisión satelital desea realizar un análisis de los clientes y los planes que estos poseen, para ello. La empresa ofrece la posibilidad de alquilar películas y bajadas adicionales, cuyo costo dependerá del plan que posee el cliente. Mediante programación lógica deberá resolver una serie de requerimientos, basados en la base de conocimiento a continuación tabulada y ya implementada.

Abonado							
Código	Cliente			Plan	Cantidad Alquileres	Paquete Premium	Bajada Adicional
	Teléfono	Nombre	Región				
1245	157456898	German de la Prueba	5	G	12	Y	N
1844	157434584	Carmen de la Prueba	2	S	5	N	N
87	156241789	Alberto de la Prueba	6	S	10	Y	Y
5487	154872563	Carlos de la Prueba	6	S	2	Y	Y
48854	156854963	Karina de la Prueba	2	B	1	Y	Y
4874	155847813	Juana de la Prueba	6	Pp	1	N	N
366	4254789	Hernan de la Prueba	4	Pp	5	N	N
4587	458796	Martin de la Prueba	1	B	6	Y	Y

**Recuperatorio**

Nota: ..... ( )

**Paradigmas de Programación**

Fecha: 31/10/15

[Práctico]

96352	154789321	Francisco de la Prueba	3	G	8	N	N
95825	154698325	Pablo de la Prueba	5	G	4	Y	N
1246	147856323	Claudia de la Prueba	1	S	9	N	N
3527	155123456	Natalia de la Prueba	3	G	6	N	N
6547	154217582	Carolina de la Prueba	6	S	7	N	N
351	15623489	Juan de la Prueba	2	G	2	N	N

Plan					
Codigo	Descripcion	Costo Base	Costo Alquiler	Costo Premium	Costo B Adicional
G	Paquete Gold	458	25	75	85
S	Paquete Silver	350	27,5	95	105
_,B	Paquete Black	750	17,5	0	75
PP	Prepago	100	35	0	0

Se sabe que para obtener el costo total que se le factura a un abonado es de la siguiente manera.

*Costo Total = Costo Base Plan + (Cant Alq \* Costo Alq) + (Cto Premium \* Cant Paq Premium Contratado) + Cto Bda Adic*

Su tarea es la siguiente:

- 1) Calcular la ganancia total obtenida de los abonados de un determinado plan. **(10 pts)**.



## Recuperatorio

Nota: ..... ( )

### Paradigmas de Programación

Fecha: 31/10/15

[Práctico]

- 2) Generar una lista donde cada componente sea la descripción de la región y el total ganado en esa región **(15 pts)**.
- 3) Listar código de Abonado, Nombre del Cliente, Descripción de la Región, y Nombre del Plan, de aquellos abonados que posean más de 2 paquetes premium contratados. **(10 pts)**
- 4) Obtener el Nombre del Cliente, Descripción de la Región de aquellos abonados que tengan como paquete premium contratado un valor de referencia pasada por parámetro **(10 pts)**

### Evaluación de Programación Funcional

#### Caso de estudio:

En una empresa dedicada a la observación aérea se está desarrollando un software que debe obtener información respecto de los aviones con los que se está operando.

Suponga que usted forma parte del equipo de trabajo y posee la siguiente información:

Código Cuadrante	Nombre Cuadrante
1	Norte
2	Sur
3	Este
4	Oeste

Código Tipo Avión	Nombre Tipo Avión
1	Militar
2	Presidencial
3	Turista
4	Espia



## Paradigmas de Programación

Fecha: 31/10/15

[Práctico]

Avion	Tipo	Cuadrante	Velocidad
Cessna	1	2	1000Km/hs
Ranquel	2	1	2500Km/hs
Boeing	4	2	2300Km/hs
Cessna1	3	3	1050Km/hs
Airbus	2	4	1800Km/hs
Ranquel	1	2	2500Km/hs
Pucara	4	1	1350Km/hs
Boeing	2	4	2700Km/hs
Pucara	3	3	2100Km/hs

Se le solicita realizar un programa en Haskell con las siguientes funciones:

- 1)tasaVelocidadEfectiva(tipo, velocidad):** Calcula la tasa de velocidad efectiva de la siguiente forma: Si el tipo es uno, la tasa efectiva es igual a la velocidad. Si el tipo es dos, la tasa efectiva es el triple de la velocidad. Si el tipo es tres la tasa efectiva es la mitad de la velocidad. Si el tipo es cuatro la tasa efectiva es el cuadrado de la velocidad. **(15 puntos)**
- 2)trayectoria(desde, hasta):** Obtiene una lista que comienza con el valor “desde” y termina con “hasta” avanzando de dos en dos. Por ejemplo si desde es 10 y hasta es 25 debe retornar la lista con los valores [10, 12, 14, 16, 18, 20, 22, 24]. **(15 puntos)**
- 3)puntoTrayectoria(desde, hasta, unPunto):** con desde y hasta llama a la función anterior y luego obtiene el valor de la lista que se encuentre en el lugar “unPunto”. Por ejemplo, ante esta llamada puntoTrayectoria(10, 25, 3) debe retornar 14. **(20 puntos)**