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

**Examen Final de Paradigmas de Programación**

**Objetivo**

Evaluar al estudiante en la parte práctica de la Unidad 5 Paradigma funcional y 6 Paradigma Lógico

**Condiciones de trabajo:**

* El alumno deberá desarrollar para cada paradigma un programa, utilizando el correspondiente entorno.
* Para el paradigma lógico, deberá generar un archivo con el nombre Legajo\_ApellidoNombre.pl y para el paradigma funcional, deberá generar un archivo con el nombre Legajo\_ApellidoNombre.hs.
* Los archivos generados deberán almacenarse en la unidad D:
* **Es responsabilidad de cada alumno ir guardando periódicamente su archivo, 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.
* **Es responsabilidad de cada alumnoque el contenido quede guardado al momento de finalizar la evaluación, con todos sus archivos generados durante el examen.**
* El **tiempo previsto** para la realización de este examen es de**1:45hs**.
* El alumno para aprobar el examen, deberá aprobar comomínimo un 50 % de cada paradigma evaluado**.**

**Paradigma Lógico**

**Enunciado**

Una empresa de suministro de energía eléctrica de una pequeña ciudad de la provincia de Córdoba nos ha pedido desarrollar un sistema que le permita la gestión de las facturas de sus usuarios. Para ello nos aporta la siguiente base de conocimiento:

**Usuario**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dni** | **Apellido** | **Nombre** | **Medidores** |
| 23586985 | Perez | María | [161,164] |
| 12589256 | Gomez | José | [162,163] |
| 25689452 | Garcia | Delfina | [165] |

**Factura**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nro**  **Liquidación** | **Nro**  **Medidor** | **Lectura anterior** | | | | **Lectura actual** | | | | **Código escalafón** |
| **fecha** | | | **kWh** | **fecha** | | | **kWh** |
| **Día** | **Mes** | **Año** | **Día** | **Mes** | **Año** |  |  |
| 00010-33550 | 161 | 20 | 7 | 2018 | 32993 | 20 | 8 | 2018 | 33158 | 1 |
| 00010-33551 | 162 | 20 | 7 | 2018 | 27645 | 20 | 8 | 2018 | 27895 | 2 |
| 00010-33552 | 163 | 23 | 7 | 2018 | 31895 | 23 | 8 | 2018 | 32015 | 1 |
| 00010-33553 | 164 | 27 | 7 | 2018 | 15689 | 27 | 8 | 2018 | 16000 | 1 |
| 00010-33554 | 165 | 27 | 7 | 2018 | 42589 | 27 | 8 | 2018 | 42789 | 2 |

**Escalafón de Energía**

|  |  |
| --- | --- |
| **Código de escalafón** | **Precio unitario por kWh** |
| 1 | $ 2,72 |
| 2 | $ 3,54 |

En base a estos hechos usted debe desarrollar las siguientes reglas:

1. Mostrar el número de liquidación e importe neto de cada factura. El importe neto es igual al consumo actual multiplicado por el precio unitario, a lo cual se le suma un cargo fijo. Para ello considerar:

* Cargo fijo de $ 47.84
* Consumo actual: que es igual a la resta de la cantidad de kWh de la lectura actual menos la cantidad de kWh de la lectura anterior.
* Precio unitario: depende del código de escalafón asociado al medidor.

Por ejemplo:

Para el Número de liquidación: '00010-33550'

Su importe Neto es: $ 496.64

1. Determinar cuál es el consumo promedio, medido en kWh, de los habitantes de la ciudad.

Por ejemplo, para nuestra base de conocimiento el:

Consumo promedio es: 1151.1066666666668 kWh

1. Para un Dni pasado como argumento, mostrar por cada uno de sus números de medidor el importe neto, pero solo de aquellos medidores que tengan un importe neto mayor a un valor X (siendo X un valor pasado como argumento).

Por ejemplo: Para el Dni: 23586985 y un valor de X= 1, el resultado obtenido es:

Medidor = 161, Importe = 496.64 ;

Medidor = 164, Importe = 893.7600000000001 .

**Paradigma Funcional**

**Enunciado**

**Enunciado**

Un call center almacena la cantidad de llamadas atendidas diariamente en una lista. Esta lista es retornada por la siguiente función:

lista :: [Integer]

Lista = [50, 150, 287,500, 876, 425, 890, 60, 678, 567, 480, 789, 567, 359, 345, 230, 678, 679, 236]

Se solicita implementar las siguientes funciones:

1.Realizar una función que reciba un número de llamadas atendidas y retorne el nivel de atención para ese día. La siguiente tabla detalla el nivel de atención según el intervalo en que se encuentre el número de llamadas atendida. **(15 pts.).**

|  |  |
| --- | --- |
| Rango de Llamadas atendidas | Nivel de Atención |
| 0 a 50 | Nivel Bajo |
| 51 a 300 | Nivel Aceptable |
| 301 a 701 | Nivel Medio |
| >701 | Nivel Alto |

2. A partir de la lista informar la cantidad de llamadas cuyo Nivel de Atención es Aceptable. **(15 pts.).**

3. A partir de la lista generar una segunda lista que contenga solo la cantidad de llamadas atendidas superiores a un valor “p” de referencia. Para resolver este punto el alumno podrá elegir resolverlo con recursividad (usando el operador : o el operador ++) o listas por comprensión. **(20 pts.) .**