

Evaluación Final Transversal

Ejecución práctica

Contexto

Sigla	Nombre Asignatura	Tiempo Asignado	% Ponderación
FPY1101	Fundamentos de programación	7 horas	40%

1.Situación Evaluativa 1:

X	Ejecución práctica
---	--------------------

2.Instrucciones Evaluación Final Transversal

Descripción
<p>NOTA: en los ejemplos, las letras con negrita significan que son datos ingresados por teclado.</p> <p>Una tienda <i>online</i> llamada Pybooks se ha especializado en la venta de <i>notebooks</i>. La información de cada modelo de <i>notebook</i> está registrada en un diccionario llamado “<i>productos</i>”, donde las llaves son los modelos y los valores asociados a las llaves son listas que contiene información asociada a cada modelo. Los puntos suspensivos indican que pueden existir muchos más datos. A continuación, se muestra un ejemplo del diccionario “<i>productos</i>”:</p> <pre>#productos = {modelo: [marca, pantalla, RAM, disco, GB de DD, procesador, video], ...}</pre> <pre>productos = {'8475HD': ['HP', 15.6, '8GB', 'DD', '1T', 'Intel Core i5', 'Nvidia GTX1050'], '2175HD': ['lenovo', 14, '4GB', 'SSD', '512GB', 'Intel Core i5', 'Nvidia GTX1050'], 'JjfFHD': ['Asus', 14, '16GB', 'SSD', '256GB', 'Intel Core i7', 'Nvidia RTX2080Ti'], 'fgdxFHD': ['HP', 15.6, '8GB', 'DD', '1T', 'Intel Core i3', 'integrada'], 'GF75HD': ['Asus', 15.6, '8GB', 'DD', '1T', 'Intel Core i7', 'Nvidia GTX1050'], '123FHD': ['lenovo', 14, '6GB', 'DD', '1T', 'AMD Ryzen 5', 'integrada'], '342FHD': ['lenovo', 15.6, '8GB', 'DD', '1T', 'AMD Ryzen 7', 'Nvidia GTX1050'], 'UWU131HD': ['Dell', 15.6, '8GB', 'DD', '1T', 'AMD Ryzen 3', 'Nvidia GTX1050'], ... }</pre> <p>El campo disco indica el tipo del disco, es decir, si es mecánico (DD) o de estado sólido (SSD).</p> <p>También se cuenta con el diccionario “<i>stock</i>”, donde las llaves son el modelo de los notebooks y el valor una lista que incluye el precio y <i>stock</i> del producto. Todos los notebooks en el diccionario “productos” aparecen también en el diccionario “stock”. Los puntos suspensivos indican que pueden existir muchos más datos. A continuación, se muestra un ejemplo del diccionario “<i>stock</i>”:</p>

```
#stock = {modelo: [precio, stock], ...}
```

```
stock = {'8475HD': [387990,10], '2175HD': [327990,4], 'JjfFHD': [424990,1],  
        'fgdxFHD': [664990,21], '123FHD': [290890,32], '342FHD': [444990,7],  
        'GF75HD': [749990,2], 'UWU131HD': [349990,1], 'FS1230HD': [249990,0], ...  
        }
```

Pybooks le solicita lo siguiente:

Un menú que tenga las siguientes opciones

*** MENU PRINCIPAL ***

1. Stock marca.
2. Búsqueda por precio.
3. Actualizar precio.
4. Salir.

Donde:

La opción 1 (Stock marca) debe entregar el stock de una marca particular ingresada por teclado. La marca ingresa puede estar escrita en mayúscula o minúsculas y debe funcionar de igual manera. Debe estar implementada mediante una función llamada *stock_marca(marca)* que recibe como parámetro la marca y no debe retornar nada.

La opción 2 (Búsqueda por precio) debe entregar una lista de strings de todos los modelos que están dentro de un rango de precios ingresados por teclado y que tengan stock (stock distinto de cero). La lista debe tener el nombre de la marca junto al modelo con el formato "Marca--Modelo". Estos nombres deben estar ordenados alfabéticamente. Debe asegurarse que, al momento de ingresar el rango de precios, deban ser valores enteros. Si no se cumple esta condición, el programa debe enviar el mensaje: "Debe ingresar valores enteros!!", y volver a preguntar por el rango de valores. En esta situación, el usuario puede ingresar cualquier tipo de dato, por lo que debe hacer uso del procedimiento de manejo de excepciones estudiado en clases. Por último, si no se encuentra ningún notebook dentro del rango de precios, el programa debe mostrar el mensaje: "No hay notebooks en ese rango de precios." Debe estar implementada mediante una función llamada *búsqueda_precio(p_min, p_max)* que reciba el precio mínimo y precio máximo como parámetro y no debe retornar nada.

La opción 3 (Actualizar precio) debe actualizar el precio de un modelo en particular en el diccionario "*stock*". Debe estar implementado mediante una función llamada *actualizar_precio(modelo, p)* que reciba como parámetro un modelo y el precio nuevo. Si el modelo ingresado no existe, la función debe retornar "False" y si el modelo existe entonces la función debe retornar "True". El código principal (main) debe recibir el valor de retorno y mostrar los mensajes "Precio actualizado!!" si se pudo realizar la actualización y "El modelo no existe!!" si no se pudo. Finalmente, el programa debe preguntar si desea actualizar otro precio de notebook. Si la respuesta es "sí", el proceso comienza nuevamente, si la respuesta es "no", entonces el programa vuelve al menú principal.

La opción 4 (salir), debe terminar el programa, mostrando por pantalla el mensaje: "Programa finalizado."

Estando en el menú principal, si se ingresa cualquier otro valor como opción, entonces el programa debe mostrar el mensaje: "Debe seleccionar una opción válida!!", y volver a preguntar. A continuación, se muestra un ejemplo de ejecución de cómo debería funcionar su solución.

Ejemplo:

*** MENU PRINCIPAL ***

1. Stock marca.
2. Búsqueda por precio.
3. Actualizar precio.
4. Salir.

Ingrese opción: **1**

Ingrese marca a consultar: **HP**

El stock es: 31

*** MENU PRINCIPAL ***

1. Stock marca.
2. Búsqueda por precio.
3. Actualizar precio.
4. Salir.

Ingrese opción: **1**

Ingrese marca a consultar: **acer**

El stock es: 0

*** MENU PRINCIPAL ***

1. Stock marca.
2. Búsqueda por precio.
3. Actualizar precio.
4. Salir.

Ingrese opción: **2**

Ingrese precio mínimo: **240000**

Ingrese precio máximo: **350000**

Los notebooks entre los precios consultas son: ['Dell--UWU131HD', 'lenovo--123FHD', 'lenovo--2175HD']

*** MENU PRINCIPAL ***

1. Stock marca.
2. Búsqueda por precio.
3. Actualizar precio.
4. Salir.

Ingrese opción: **2**

Ingrese precio mínimo: **hola**

Debe ingresar valores enteros!!

Ingrese precio mínimo: **900000**

Ingrese precio máximo: **1200000**

No hay notebooks en ese rango de precios.

*** MENU PRINCIPAL ***

1. Stock marca.
2. Búsqueda por precio.
3. Actualizar precio.
4. Salir.

Ingrese opción: **3**

Ingrese modelo a actualizar: **UWU131HD**

Ingrese precio nuevo: **399000**

Precio actualizado!!

Desea actualizar otro precio (s/n)?: **si**

Ingrese modelo a actualizar: **3456TYH**

Ingrese precio nuevo: **550000**

El modelo no existe!!

Desea actualizar otro precio (s/n)?: **no**

*** MENU PRINCIPAL ***

1. Stock marca.
2. Búsqueda por precio.
3. Actualizar precio.
4. Salir.

Ingrese opción: **6**

Debe seleccionar una opción válida!!

*** MENU PRINCIPAL ***

1. Stock marca.
2. Búsqueda por precio.
3. Actualizar precio.
4. Salir.

Ingrese opción: **4**

Programa finalizado.