CARRERA DE: INGENIERÍA INFORMÁTICA
MATERIA: PARADIGMA DE PROGRAMACIÓN II
FORMACIÓN TEÓRICA: GUÍA DE EJERCICIOS

GUÍA DE EJERCICIOS 02

TEMA: INTRODUCCIÓN

OBJETIVO: Aprovechar las herramientas del lenguaje para manejo de archivos y reutilizacion de codigo mediante la codificacion modularizada, junto con los conceptos mas importante del paradigma estructurado. Mostrar su utilidad y comprobar su funcionamiento.

1- Se necesita un programa que lea un archivo de texto e imprima por pantalla la cantidad de letras mayúsculas, minúsculas y otros caracteres que no sean letras.

TIPs: Recordar de las guías anteriores las funciones aplicadas sobre strings, en particular isupper() y islower() que retornan True en caso de tratarse de caracteres en mayúsculas o minúsculas, respectivamente.

Resultado esperado: Si el archivo texto.txt contiene la siguiente frase:

texto.txt

> No tios tome Lunes ELLA ojos. No secretario, escondidos, un sr ir INTENsidad Comprendio??. He cura ir un fino miro. Perdonaria antipatico al herrero. Toco sera pero don dio miro fue. Crimen OH! si lineas de tardas. Puede que Tengo y daria tanto de tarde el si. Tesorero sonriendo insegura tenorios hablando no injurias. Generosa ni claridad YO! Justicia Embestir TItubear.

Se espera recibir como resultado en pantalla:

```
La cantidad de mayusculas es: 29
La cantidad de minusculas es: 268
La cantidad de otros caracteres es: 76
```

2- ¡Usted tiene la importante tarea de crear los cartones de Bingo! Para dicho propósito, se le pide que escriba un programa con las siguientes características:

a) Los cartones están formados solamente por números enteros y positivos. No se aceptan letras ni ningún otro carácter

b) Los números posibles van del rango 00 hasta el 99 (inclusive)

c) Cada cartón consta de 5 líneas con 5 números cada línea, lo que forma un total de 25 números por carton. Para una mejor legibilidad se pide que al imprimirlo se muestren de esa forma.

Para poder dar por concluido el trabajo, usted propuso trabajar por fases. Es de esperar que el resultado de cada una de ellas sirva para la fase siguiente (siguiendo el paradigma estructurado y la reutilización de código). Por lo tanto, se le pide:

a) Crear la función generar_carton() que genere aleatoriamente los 25 números que debe llevar. Debe retornarlo en una lista. Por ejemplo: [4,7,13,45, 56, 67, 78, 89, …].

Responda: ¿Se trata de una función de Frontend o de Backend?

TIPs: el módulo random de python (import random) ofrece formas muy útiles para generar números aleatorios. En particular, analizar la funcion random.sample que genera una lista con la cantidad pedida y el rango.

b) Utilizando la función anterior, que le permite ir creando varios cartones, escriba la función imprimir_carton(lista) que reciba una lista de números (el cartón) y que la imprima por pantalla de la forma solicitada.

Por ejemplo, si se invoca imprimir_carton([4,7,13,45, 56, 67, 78, 89, …]) se debe visualizar en pantalla:

```
04 07 13 45 56
67 78 89 …
```

¿Es una función de Frontend o de Backend?

c) Para la tercera fase, se debe permitir grabar en un archivo un cartón generado. Para ello, se requiere la función grabar_carton(lista) que reciba la lista de números (el cartón) y lo grabe en un archivo de texto, siguiendo el mismo patrón de impresión por pantalla.

Por ejemplo, si se invoca grabar_carton ([4,7,13,45, 56, 67, 78, 89, …]) se debe crear el archivo carton.txt con el siguiente contenido:

carton.txt

```
04 07 13 45 56
67 78 89 …
```

¿Es una función de Frontend o de Backend?

d) Por último, crear un programa que utilice todas las funciones anteriores. Para ello, debe mostrar un menú donde el usuario puede elegir las siguientes opciones:

```
1) Generar e imprimir un cartón
2) Generar y grabar un cartón
3) Salir
```

En el caso 1, solamente debe generarlo e imprimirlo por pantalla. En el caso 2, además debe preguntarle al usuario el nombre del archivo que va a tomar al momento de grabar el cartón. En el caso 3, debe finalizar el programa.

3- Luego de tanto buscar trabajo en el rubro de Seguridad Informática, se encuentra con un pedido de un cliente nuevo. Lo que se necesita es poder llevar registro de los usuarios que acceden a su computadora, algo similar a una "fichada". Solo es necesario grabar la información de hora y fecha junto con nombre de usuario. También, por cuestiones de seguridad, se necesita que autentique al usuario. Usted está muy motivado por realizar el trabajo, y propone un esquema de autenticación basado en usuario y password. Como objetivo final, debe lograr que una vez que el usuario ingresa el usuario y el password correcto, entonces se graba la fichada tomando la fecha/hora actual.

Al igual que antes, se propuso trabajar por fases, las cuales se detallan a continuación:

a) Construya de forma modularizada un programa que pida por línea de comando el nombre de usuario y una vez ingresado, registre la información de fecha/hora y la grabe en un archivo fichadas.txt (puede utilizar el resultado del módulo datetime). Por ejemplo, si el usuario ingresa ndiaz debería ocurrir lo siguiente:

```
Bienvenido al sistema de Fichadas!
Ingrese el usuario: ndiaz
Registro grabado
```

fichadas.txt

```
2019-07-11 14:15:13.541325 – ndiaz
```

b) Ahora decide mejorar el programa, de forma tal que valide que el usuario ingresado es correcto. Para ello, existe otro archivo de texto llamado passwd.txt que contiene los nombres de usuarios y los passwords (de forma legible). Por ejemplo, si el archivo passwd.txt contiene lo siguiente:

passwd.txt

```
ndiaz:passWord!
jperez:admin1234
```

Entonces se debe observar lo siguiente:

```
Bienvenido al sistema de Fichadas!
Ingrese el usuario: admin
Usuario incorrecto

Ingrese el usuario: admin2
Usuario incorrecto

usuario: jperez
Registro grabado
```

fichadas.txt

```
2019-07-11 14:15:13.541325 – ndiaz
2019-07-11 15:14:22.681205 – jperez
```

c) Avanzando con su espectacular proyecto, decide realizar una mejora al programa, que valide que el usuario y el password coincida con el texto guardado en passwd.txt. Por ejemplo, una posible corrida del programa manteniendo el mismo archivo passwd.txt, podría ser:

```
Bienvenido al sistema de Fichadas!
Ingrese el usuario: admin
Ingrese el password: lalala
Usuario incorrecto

Ingrese el usuario: admin2
Ingrese el password: lalala
Usuario incorrecto

Ingrese el usuario: jperez
Ingrese el password: lalala
Usuario incorrecto

Ingrese el usuario: ndiaz
Ingrese el password: passWord!
Registro grabado
```

fichadas.txt

```
2019-07-11 14:15:13.541325 – ndiaz
2019-07-11 15:14:22.681205 – jperez
2019-07-12 09:05:17.587545 – ndiaz
```