

Proyecto Final #1

INTRODUCCIÓN

Una de las formas de demostrar nuestros conocimientos adquiridos es ponerlos en práctica con ejercicios que permitan desarrollar nuestras habilidades. En el presente documento se detallan 5 casos de uso a resolver con python y aplicar todo lo aprendido durante el proceso de aprendizaje.

OBJETIVOS

Objetivo General

Aplicar los conocimientos adquiridos en la cursada relacionados a programación con python.

Objetivos Específicos

- Aprender a trabajar de forma colaborativa en grupos de 5 personas (a determinar por los profesores)
- Emplear un sistema de control de versiones (Git) y un repositorio remoto (GitHub) para almacenar los ejercicios resueltos.
- Aprender habilidades de investigación para resolver problemas

PAUTAS

Los profesores formarán grupos de 5 personas para realizar un conjunto de ejercicios (uno por integrante), los cuales tienen que resolver y subir en un repositorio de GitHub para que los profesores puedan visualizar los resultados y realizar sugerencias para mejorar o corregir en caso de presentarse errores. Se debe hacer uso de funciones y, si lo consideran necesario, el uso de módulos o paquetes personalizados (no de terceros).

EJERCICIOS A RESOLVER

- **Contador de Palabras:** Crear un algoritmo que permita realizar el conteo de palabras de una frase ingresada por el usuario. Además, debe almacenar o registrar las frases ingresadas en un archivo de texto. Cada vez que ejecuta el algoritmo, debe mostrar un resumen con la cantidad de frases ingresadas y el total de palabras, siempre y cuando el archivo de texto contenga información previa. Se debe ingresar frases hasta que se ingrese una palabra o frase clave de salida (debe contemplar ambas). Por ejemplo, el algoritmo termina si se ingresa alguna de las siguientes frases: "abra cadabra", "show time", "the end".
 - Nombre del script: word_counter.py
 - Nombre del archivo de texto: phrases.txt
- **Palíndromos:** Crear un algoritmo que permita identificar si una palabra ingresada por el usuario es o no un palíndromo. Un palíndromo es una palabra que puede leerse tanto de izquierda a derecha como de derecha a izquierda. Por ejemplo: neuquen, a3l3a, okonoko, 153351. El programa finaliza cuando se ingresa alguna

palabra clave, como por ejemplo “aloha”, “bye”. Al finalizar el programa, se debe presentar un resumen con la cantidad de palabras ingresadas, cuantas eran palíndromos y cuantas no eran palíndromos.

- Nombre del script: palindromes.py
- **Fibonacci:** Crear un algoritmo que permita ingresar un número entero, y muestre en pantalla la sucesión fibonacci con la cantidad de elementos igual al número ingresado. Por ejemplo, si se ingresa el número 3, se deberían mostrar los primeros 3 elementos de la serie: 1, 1, 2. Se debe controlar que el número ingresado sea mayor que 0 y mostrar un mensaje informativo al respecto para que el usuario ingrese otro número válido.
 - Nombre del script: fibonacci.py
- **Generador de Contraseñas:** Crear un algoritmo que permita generar una contraseña aleatoria, solicitando al usuario la longitud de la misma, si desea que tenga mayúsculas, minúsculas, números y/o caracteres especiales (tomaremos únicamente los caracteres +, -, %, \$, &, #). Por defecto, si el usuario no escoge ninguna de las opciones propuestas, se debe generar una contraseña de 16 dígitos alfanuméricos (letras mayúsculas, minúsculas y números).
 - Nombre del script: password_generator.py
- **Libreta de Direcciones:** Crear un algoritmo que permita registrar información de una dirección en un archivo csv (valores separados por coma) con los siguientes campos: ID (generado aleatoriamente), nombre identificador, dirección, teléfono, fecha de registro (este último debe generarse automáticamente al realizar el registro).
 - Nombre del script: address_book.py
 - Nombre del archivo de texto: addresses.csv