# **TP7 archivos y try**

1. Existen dos errores típicos que se pueden producir al manejar archivos

* La primera, es abrir un archivo en modo lectura sin antes preguntar si existe tal archivo. (FileNotFoundError)
* La segunda es crear dos archivos con el mismo nombre. (FileExistsError)

Realizar funciones que hagan la siguiente tarea:

1. Abrir un archivo de texto en modo **escritura**, si no existe ese archivo que lo cree. La función debe recibir como parámetro el nombre del archivo de texto que se va a crear o abrir, el texto que se va a escribir y una opción para añadir más elementos o para sobre escribir lo que había en el archivo.  
   Ejemplo: **def escribirArchivo(nombreArchivo,nuevoTexto ,opcion):**

La función debe devolver un booleano, **False** si se crea un nuevo archivo y **True** si se modificó un archivo existente.

1. Abrir un archivo de texto en modo **lectura**, la función debe recibir como parámetro el nombre del archivo de texto y nada más.

Ejemplo: **def leerArchivo(nombreArchivo):**  
La función debe devolver dos valores, el primero es una variable booleana y el segundo un string, un texto.  
Ejemplo: **return** **bien**, **texto  
La salida será la siguiente:** **bien**=True si se pudo leer el archivo, además la variable **texto** debe contener todo lo que haya en el archivo de texto.  
**bien**=False si no existe ese archivo, en tal caso la variable **texto**=””

**Datazo**: recordar cerrar el archivo de texto cuando lo hayan abierto.

1. Crear un diccionario llamado codificación, donde cada letra va a estar cambiada por la siguiente en el alfabeto:  
   codificacion = {'a':'b','b':'c','c':'d','d':'e','e':'f','f':'g','g':'h','h':'i','i':'j','j':'k','k':'l','l':'m','m':'n','n':'ñ','ñ':'o','o':'p','p':'q','q':'r','r':'s','s':'t','t':'u','u':'v','v':'w','w':'x','x':'y','y':'z','z':'a'}  
   crear una función que reciba como parámetro de entrada un texto cualquiera, que se codifique y ese mensaje codificado se guarde en un archivo de texto.  
   **Ejemplo:** si a mi función le paso el texto “hola”, me debería devolver “ipmb”
2. Crear un diccionario que decodifique el mensaje del ejercicio 2, luego hacer una función que pueda abrir un archivo codificado y pueda decodificar el mensaje, mostrando el mensaje real en pantalla con un print().  
   Ejemplo: si en el archivo estaba escrito “ipmb”, me debería mostrar “hola” en pantalla.
3. Hacer una función que pueda crear una lista con valores aleatorios, los parámetros de entrada deberían ser limiteInferior, limiteSuperior y cantidadElementos. La función debería devolver una lista con valores aleatorios entre los límites superior e inferior, la lista debe tener cantidadElementos. Para ello importen la librería **random**, y usen el método randint() **(ver clase 9)**
4. **Vamos a crear el primer juego:** la idea del juego es que el programa nos va a mostrar una serie de números entre el 1 y el 100 de manera aleatoria, el usuario deberá adivinar si el siguiente número que nos muestre es mayor, menor o igual. En total el juego nos va a mostrar 11 números aleatorios. Por cada acierto tenemos que ganar 10 puntos. Al finalizar la secuencia de 11 números se debe mostrar el puntaje obtenido.   
   Realizar un menú sencillo para jugar, el puntaje se debe guardar en un archivo de texto llamado “resultados.txt”.
5. **Optativo:** diseñen el juego de piedra, papel o tijeras. Objetivo, ganarle a la computadora 2 de 3 intentos, se debe mostrar un texto adecuado si se pudo ganar o no.
6. **Optativo:** diseñen un simulador de cajero automático, con las funciones básicas, preguntar saldo, cargar saldo y extraer saldo. También deben crear una clave de acceso para un usuario.