

Fundamentos de programación.

Profesor: Roberto Martínez Román

### Misión 08. Cadenas.

Crea un archivo en Python con nombre **Misión\_08.py** (con acento y guion bajo)

Escribe cada una de las funciones que se piden a continuación, TEN CUIDADO de nombrar a las funciones como se indica. Esta tarea será calificada parcialmente por un robot. Si tus funciones tienen nombres diferentes, se tomarán como incorrectas.

1. (250 hp) Escribe la función **combinarLetras**, que recibe como parámetro una cadena y regresa una nueva alternando mayúsculas y minúsculas. Por ejemplo, si recibe "**Hola mundo**", regresa "**HoLa mUnDo**". La cadena resultante siempre inicia con mayúscula.

*Sugerencia: Recorre la cadena con un ciclo for y en un 'acumulador de cadenas' vas construyendo el resultado. También, puedes usar las funciones **lower** y **upper** para convertir los caracteres a minúsculas/mayúsculas.*

2. (200 hp) Escribe la función **contieneLasVocales**, que recibe como parámetro una cadena y regresa **True** si la cadena contiene TODAS las vocales, **False** en otro caso. Por ejemplo, si recibe "**Monterrey**" regresa **False**, si recibe "**Abuelito**" regresa **True**. No importan mayúsculas/minúsculas.

*Sugerencia: Utiliza el operador **in** de cadenas.*

3. (200 hp) Escribe la función **formarNombreUsuario**, que recibe tres parámetros: nombre, apellido paterno y matrícula numérica entera (7 dígitos). Regresa un nombre de usuario formado de: los tres primeros caracteres del nombre + los tres primeros caracteres del apellido paterno + los últimos 3 dígitos de la matrícula. Por ejemplo, si recibe "Roberto", "Martínez" y 12345678, el nombre de usuario que regresa es: "**robmar678**", note que todos los caracteres están en minúsculas.

*Sugerencia: Utiliza el operador de corte (slicing) **[ : : ]** para obtener un subconjunto de caracteres de una cadena. Para separar dígitos de un entero, puedes usar el operador módulo (%).*

4. (300 hp) Escribe la función **esCorrecto**, que recibe como parámetro una cadena que representa el nombre de una persona (nombre y dos apellidos sencillos). Regresa **True** si las reglas de mayúsculas están bien aplicadas, **False** en otro caso. Por ejemplo, si recibe "**Roberto Martínez Román**" regresa **True**, si recibe "**roberto MARTinez RoMan**" regresa **False**. Solo es correcto cuando la letra inicial de cada palabra es mayúscula y las otras son minúsculas.

*Sugerencia: Utiliza las funciones **split**, **islower** e **isupper**.*

5. (300 hp) Escribe la función **traducirTelefono**, que recibe como parámetro un número de teléfono en el formato "01800-XXX-XXXX" y regresa una cadena del número de teléfono con los dígitos correspondientes. Por ejemplo, si recibe "**01800-VOY-BIEN**", regresa "**01800-869-2436**". Puedes suponer que el teléfono que llega siempre tiene el formato correcto.

Usa la siguiente tabla:

1	ABC 2	DEF 3
GHI 4	JKL 5	MNO 6
PQRS 7	TUV 8	WXYZ 9
*	0	#

*Sugerencia: ¡Suerte! 😊*

6. **EXTRA.** (300 hp) Escribe la función **esValido**, que recibe como parámetro un password y regresa **True** si cumple con las reglas, **False** en otro caso. Por ejemplo, si recibe "Abcd-7635" regresa True, si recibe "abcd1936" regresa False.

Un password es válido si cumple las siguientes reglas:

1. Tiene 8 caracteres o más.
2. Contiene al menos una letra mayúscula y una letra minúscula.
3. Contiene al menos un carácter especial (no es letra ni dígito).
4. Tiene al menos 1 dígito.
5. Comienza con una letra.
6. No contiene dígitos consecutivos (12, 123, 5678, 345, etc.)

*Sugerencia: No hay. 😊*

Al terminar, sube a **Github** el archivo **Misión\_08.py** y abre un **Pull Request** para que te califique.