

Puesta en Producción Segura

Curso de Especialista en Ciberseguridad. IES Campanillas. Curso 21-22

Prácticas de Evaluación Unidad 1

*Incluye todos los ficheros desarrollados en esta práctica dentro de un repositorio llamado **PPS_Unidad1** dentro de tu usuario de GitHub. Únicamente deberás proporcionar la dirección completa de este repositorio en la tarea de Moodle.*

1. Utilizando la clasificación vista en clase sobre los lenguajes de programación, escoge 5 lenguajes que desees y clasifícalos en una tabla según su nivel de abstracción, su forma de ejecución y los paradigmas de programación que incorpora. No olvides incluir el año de aparición y el autor/autores del mismo como MÍNIMO. Incluye toda esta información en un fichero llamado **lenguajes.pdf**.
2. Realiza, utilizando Python 3, un programa llamado **binario.py** que pida al usuario que introduzca un número binario e imprima por pantalla el número en formato decimal. Para desarrollar el programa, es necesario que desarrolles una función con la siguiente cabecera:

```
def esBinario(strbinario)
```

```
# Devuelve True o False si la cadena de caracteres (strbinario) que se ha pasado como parámetro contiene una cadena binaria.
```

```
# Ejemplo de esBinario:
```

```
print(esBinario("1001"))
```

```
True
```

```
print(esBinario("Hola"))
```

```
False
```

3. Realiza, utilizando Python 3, el ejercicio 3 de la página 35 del libro "Introducción a Python" de Jon Vadillo e inclúyelo en un fichero llamado **lista.py**. Las funciones que debes usar en el ejercicio 3 deben utilizar OBLIGATORIAMENTE las siguientes cabeceras:

```
def estaEnRango(valor, minimo, maximo)
```

```
# Devuelve True o False determinando que valor se encuentra entre el mínimo y el máximo.
```

```
def estaEnLista(valor, lista)
```

```
# Devuelve True o False determinando si el valor está en la lista.
```

4. Crea una suite de tests mediante **UnitTest** que comprueben las 3 funciones que has desarrollado en los ejercicios anteriores. Procura que los tests unitarios cubran lo mejor posible la aparición de comportamientos no deseados.
5. Realiza el ejercicio 4 pero utilizando esta vez cualquier otro framework de terceros como por ejemplo **pytest**.