



# Curso especialización de ciberseguridad.

## Puesta en Producción Segura

### Prácticas de evaluación Unidad 1

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.

Lenguaje	Nivel Abstracción	Forma de Ejecución	Paradigma de programación
C++	ALTO	COMPILADO	MULTIPARADIGMA*
PYTHON	ALTO	INTERPRETADO	MULTIPARADIGMA*
PHP	MEDIO	INTERPRETADO	MULTIPARADIGMA*
ENSAMBLADOR	BAJO	VIRTUAL	IMPERATIVO
RUBY	ALTO	INTERPRETADO	MULTIPARADIGMA*

**C++:** Un lenguaje diseñado y desarrollado por Bjarne Stroustrup en el año 1983 influido por otros lenguajes de programación en especial C, Simula, Ada 83 y ALGOL 68 y que ha influido en el desarrollo de otros lenguajes como Perl, Lua, Ada95, Java, PHP, D, C# entre otros. Es un lenguaje multiparadigma donde se destacan el paradigma imperativo y orientado a objetos.

**PYTHON:** Este lenguaje ha sido diseñado por Guido Van Rossum en 1991 y administrado actualmente por Python Software Foundation. Se trata de un lenguaje Multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida programación funcional y lógica.

**PHP:** Es un lenguaje de programación que se adapta especialmente al desarrollo web. Fue creado por el Danés Rasmus Ledorf en el año 94, sin embargo, su primera aparición fue al año siguiente. Un Lenguaje Multiparadigma que soporta los paradigmas orientados a objetos, imperativos, funcionales, lógicos y procedimentales.

**ENSAMBLADOR:** El lenguaje ensamblador es un lenguaje de programación de bajo nivel consiste en representar instrucciones básicas para los computadores, microprocesadores y otros circuitos programables ha estado disponible desde los años 1950 y su ventaja es que es mucho más rápido que otros lenguajes.

**RUBY:** Ruby es un lenguaje de programación interpretado, lógico y orientado a objetos, creado por el programador japonés Yukihiro Matz Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995. Combina una sintaxis inspirada en Python y Perl.

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
```

```
def esbinario (strbinario):
    for elemento in range(0,len(strbinario)):
        if strbinario[elemento]!='1' and strbinario[elemento]!='0':
            return False
    return True

def extraer(binario):
    decimal=0
    for elemento in range(0,len(binario)):
        if binario[elemento]=='1':
            decimal=decimal+pow(2, len(binario)-elemento-1)
    return decimal

cadena=str(input("introduce un número binario: "))

if esbinario(cadena):
    print(f"{cadena} es binario " + str(extraer(cadena)))
else:
    print(f"{cadena} no es binario")
```

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.
```

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

    if valor >= minimo and valor <= maximo:
        return True
    else:
        return False
```

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

def estaEnLista(valor, lista):

    if valor in lista:
        return True
    else:
        return False

# Devuelve True o False determinando si el valor está en la lista.
min=1
max=20
variable=int(input(f"introduce un valor entre {min} y {max}: "))
lista=[6,14,11,3,2,1,15,19]
if estaEnRango(variable, min, max):

    if estaEnLista(variable, lista):
        print(f"el numero {variable} está dentro de la lista")
    else:
        print(f"el numero {variable} está en el rango pero no está en la
lista")
else:
    print(f"en número {variable} no está entre {min} y {max}" )
```