

Trabajo Práctico 6: Recursividad Programación I

Profesor/es: Julieta Trapé

Tutor: Tomás Ferro

Estudiante: Karen Lauk

Ejercicios:

1. Factorial de un número

```
def factorial(n):  
    if n == 0 or n == 1:  
        return 1  
    return n * factorial(n - 1)
```

2. Serie de Fibonacci

```
def fibonacci(n):  
    if n == 0:  
        return 0  
    elif n == 1:  
        return 1  
    return fibonacci(n - 1) + fibonacci(n - 2)
```

3. Potencia recursiva

```
def potencia(base, exponente):  
    if exponente == 0:  
        return 1  
    return base * potencia(base, exponente - 1)
```

4. Decimal a Binario

```
def decimal_a_binario(n):  
    if n == 0:  
        return ""  
    return decimal_a_binario(n // 2) + str(n % 2)
```

5. Palíndromo recursivo

```
def es_palindromo(palabra):  
    if len(palabra) <= 1:  
        return True  
    if palabra[0] != palabra[-1]:  
        return False  
    return es_palindromo(palabra[1:-1])
```

6. Suma de dígitos

```
def suma_digitos(n):  
    if n < 10:  
        return n  
    return n % 10 + suma_digitos(n // 10)
```

7. Contar bloques de una pirámide

```
def contar_bloques(n):  
    if n == 1:  
        return 1  
    return n + contar_bloques(n - 1)
```

8. Contar dígitos

```
def contar_digito(numero, digito):  
    if numero == 0:  
        return 0  
    return (1 if numero % 10 == digito else 0) + contar_digito(numero  
// 10, digito)
```