

Nombre y Apellido:

Legajo:

Examen Parcial de Programación 2

Tiempo mínimo para el examen: 1 hora reloj.

Tiempo máximo para el examen: 2 horas reloj.

El examen se compone los siguientes ejercicios. Se solicita documentar bien su código (por ej., cuando deba definir funciones/métodos indique claramente los tipos de los parámetros, el tipo del valor de retorno y explique brevemente qué hace la función/método).

1. Implemente una función recursiva llamada `es_palindromo_rec` que recibe como parámetro una cadena de caracteres y devuelve un valor booleano indicando si la cadena es un palíndromo (es decir, se lee igual de izquierda a derecha que de derecha a izquierda). También implemente una función iterativa `es_palindromo_it` que realice la misma tarea.

Ejemplo de Uso:

- `es_palindromo_rec("radar")` debería devolver `True`.
- `es_palindromo_it("mundo")` debería devolver `False`

Aclaración: trabaje en todo momento con cadenas, no transforme a otro tipo de dato.

2. Considere la siguiente función:

```
def g(lista: list) -> int:
    resp = 1
    n = len(lista)
    i = 0
    while i < n:
        for j in range(n):
            if (i+j) % 2 == 0:
                resp *= ( lista[i] + lista[j] )
        i += 1
    return resp
```

y responda las siguientes preguntas:

1. ¿Cuántas asignaciones se realizan en total en la función `g`?
 2. ¿Cuál es el orden de complejidad temporal de la función?
3. Implementación de un sistema de gestión de biblioteca utilizando POO. Para ello, implemente las siguientes clases:

1. **Clase Libro:**

- Atributos:

- `codigo_libro (str)`: código único del libro.
- `titulo (str)`: título del libro.
- `autor (str)`: autor del libro.

- Métodos:

- `__init__`: inicializa los atributos.
- `__str__`: devuelve una representación en cadena del libro.

2. Clase Ejemplares:

■ Atributos:

- `libro (Libro)`: el libro al que hace referencia.
- `total (int)`: cantidad total de ejemplares del libro en la biblioteca.
- `prestados (int)`: cantidad de ejemplares que están actualmente prestados.

■ Métodos:

- `__init__`: inicializa los atributos.
- `prestar`: actualiza el número de ejemplares prestados.
- `devolver`: actualiza el número de ejemplares prestados.
- `agregar_ejemplares`: actualiza el número total de ejemplares.
- `disponibles`: devuelve la cantidad de ejemplares disponibles.

3. Clase Biblioteca:

■ Atributos:

- `ejemplares (dict[str, Ejemplares])`: diccionario de ejemplares en la biblioteca (clave: código del libro, valor: objeto Ejemplares).
- `prestamos (dict[str, list[str]])`: diccionario que almacena los libros prestados a personas (clave: DNI, valor: lista de códigos de libros prestados a la persona del DNI).

■ Métodos:

- `__init__`: inicializa los atributos.
- `agregar_ejemplares`: agrega ejemplares de un libro particular a la biblioteca.
- `prestamo_libro`: permite prestar un libro (identificado por su código) a una persona (identificada por su DNI).
- `devolver_libro`: permite devolver un libro.

```
# Ejemplo de uso
```

```
# Creando libros
```

```
novela1 = Libro("N001", "Cien años de soledad", "Gabriel García Márquez")
```

```
texto1 = Libro("T001", "Matemáticas para Todos", "José Pérez")
```

```
# Creando ejemplares
```

```
ejemplar_novela1 = Ejemplares(novela1, 5) # 5 ejemplares de la novela
```

```
ejemplar_texto1 = Ejemplares(texto1, 3) # 3 ejemplares del libro de texto
```

```
# Creando biblioteca
```

```
biblioteca = Biblioteca()
```

```
biblioteca.agregar_ejemplares(ejemplar_novela1)
```

```
biblioteca.agregar_ejemplares(ejemplar_texto1)
```

```
# Prestando libros a personas identificadas por su DNI
```

```
biblioteca.prestamo_libro("12345678", "N001") # DNI de la persona y código del libro
```

```
biblioteca.prestamo_libro("12345678", "T001") # Otro libro prestado
```

```
# Devolviendo un libro
```

```
biblioteca.devolver_libro("12345678", "N001")
```