



**Instituto Politécnico  
Nacional**



## **Escuela Superior de Cómputo**

Unidad académica: Análisis y Diseño de Algoritmos

Actividad:

**“Algoritmos de la jarra de agua”**

Equipo:

Solares Velasco Arturo Misael

Solis Lugo Mayra

Grupo: 3CV1

Profesor: García Floriano Andrés

Fecha:

21 de Mayo de 2024

# Reporte de Práctica: Código de Jarras de Agua en Python

## Introducción

Este reporte describe el código Python implementado para resolver el clásico problema de las jarras de agua. El objetivo del problema es alcanzar una cantidad específica de agua en una de las jarras utilizando dos jarras de capacidades conocidas, realizando operaciones de llenar, vaciar y transferir agua entre ellas.

## Código

El código se estructura en una clase `Jarra` que representa una jarra con las siguientes características:

- `capacidad`: Capacidad máxima de la jarra (en litros).
- `contenido`: Cantidad actual de agua en la jarra (en litros).

La clase `Jarra` define tres métodos:

- `llenar()`: Llena la jarra hasta su capacidad máxima.
- `vaciar()`: Vacía la jarra por completo.
- `transferir_a(otra_jarra)`: Transfiere agua de la jarra actual a `otra_jarra` hasta que una de las siguientes condiciones se cumpla:
  - La jarra actual se vacía por completo.
  - `otra_jarra` se llena a su capacidad máxima.

La función `resolver_problema()` implementa el algoritmo para alcanzar la cantidad de agua deseada en una de las jarras. La función sigue estos pasos:

1. Inicializa dos jarras con las capacidades especificadas.
2. Crea una lista `pasos` para almacenar las descripciones de cada operación realizada.
3. Realiza una serie de operaciones de llenar, vaciar y transferir agua entre las jarras hasta alcanzar la cantidad objetivo en una de ellas.
4. Registra cada operación en la lista `pasos`.
5. Devuelve la lista `pasos` con la secuencia de operaciones realizadas.

## Algoritmo en python

Se define la clase Jarra:

```
1 class Jarra:
2     def __init__(self, capacidad):
3         self.capacidad = capacidad
4         self.contenido = 0
5
6     def llenar(self):
7         self.contenido = self.capacidad
8
9     def vaciar(self):
10        self.contenido = 0
11
12    def transferir_a(self, otra_jarra):
13        cantidad_transferir = min(self.contenido, otra_jarra.capacidad - otra_jarra.contenido)
14        self.contenido -= cantidad_transferir
15        otra_jarra.contenido += cantidad_transferir
16
```

Función de resolución de problema:

```
17 def resolver_problema():
18     jarra_5 = Jarra(5)
19     jarra_3 = Jarra(3)
20
21     pasos = []
22
23     # Paso 1: Llenar la jarra de 5 litros
24     jarra_5.llenar()
25     pasos.append(f"Llenar jarra de 5 litros: 5L, {jarra_5.contenido}L; 3L, {jarra_3.contenido}L")
26
27     # Paso 2: Verter el agua de la jarra de 5 litros a la jarra de 3 litros
28     jarra_5.transferir_a(jarra_3)
29     pasos.append(f"Transferir de 5L a 3L: 5L, {jarra_5.contenido}L; 3L, {jarra_3.contenido}L")
30
31     # Paso 3: Vaciar la jarra de 3 litros
32     jarra_3.vaciar()
33     pasos.append(f"Vaciar jarra de 3 litros: 5L, {jarra_5.contenido}L; 3L, {jarra_3.contenido}L")
34
35     # Paso 4: Verter el agua de la jarra de 5 litros a la jarra de 3 litros
36     jarra_5.transferir_a(jarra_3)
37     pasos.append(f"Transferir de 5L a 3L: 5L, {jarra_5.contenido}L; 3L, {jarra_3.contenido}L")
38
39     # Paso 5: Llenar la jarra de 5 litros
40     jarra_5.llenar()
41     pasos.append(f"Llenar jarra de 5 litros: 5L, {jarra_5.contenido}L; 3L, {jarra_3.contenido}L")
42
43     # Paso 6: Verter el agua de la jarra de 5 litros a la jarra de 3 litros
44     jarra_5.transferir_a(jarra_3)
45     pasos.append(f"Transferir de 5L a 3L: 5L, {jarra_5.contenido}L; 3L, {jarra_3.contenido}L")
46
47     return pasos
```

## Ejemplo de Ejecución

El código incluye un bloque `if __name__ == "__main__":` que ejecuta la función `resolver_problema()` y muestra la secuencia de pasos para alcanzar 2 litros de agua en una jarra de 5 litros y otra de 3 litros.

```
49
50 if __name__ == "__main__":
51     pasos_solucion = resolver_problema()
52     for paso in pasos_solucion:
53         print(paso)
54
```

## Resultados

```
arturo ➤ master ? :1 ➤ ~/Documents/ESCOM/algoritmos_escom_2024/practica_lab_10/main (3).py"
Llenar jarra de 5 litros: 5L, 5L; 3L, 0L
Transferir de 5L a 3L: 5L, 2L; 3L, 3L
Vaciar jarra de 3 litros: 5L, 2L; 3L, 0L
Transferir de 5L a 3L: 5L, 0L; 3L, 2L
Llenar jarra de 5 litros: 5L, 5L; 3L, 2L
Transferir de 5L a 3L: 5L, 4L; 3L, 3L
```

## Análisis y Discusión

El código implementa una solución eficiente al problema de las jarras de agua utilizando una clase `Jarra` para representar las jarras y funciones para realizar las operaciones de llenar, vaciar y transferir agua. La función `resolver_problema()` utiliza un algoritmo simple pero efectivo para alcanzar la cantidad objetivo de agua en una de las jarras.

El código es claro, conciso y fácil de entender. Utiliza comentarios para explicar las funciones y variables, lo que facilita su comprensión y mantenimiento.

## Conclusiones

El código presentado proporciona una implementación efectiva y didáctica del problema de las jarras de agua en Python. Demuestra la capacidad del lenguaje para modelar problemas y desarrollar soluciones algorítmicas. El código puede servir como base para explorar diferentes algoritmos de búsqueda y optimización para resolver problemas similares.