

Ejemplos LeetCode (Python)

Ejemplo 1: Suma de dos números (Two Sum)

```
def two_sum(nums, target):  
    # Crear un diccionario para almacenar el índice de cada  
    número  
    num_map = {}  
    # Iterar sobre la lista de números  
    for i, num in enumerate(nums):  
        # Calcular el complemento que, junto con el número  
        actual, suma el objetivo  
        complement = target - num  
        # Si el complemento ya está en el diccionario, hemos  
        encontrado la solución  
        if complement in num_map:  
            return [num_map[complement], i]  
        # De lo contrario, almacenar el número actual y su  
        índice en el diccionario  
        num_map[num] = i  
    return []
```

Ejemplo 2: Longest substring without repeating characters

```
def length_of_longest_substring(s):  
    # Diccionario para almacenar la última posición de cada  
    carácter  
    char_index = {}  
    max_length = 0  
    start = 0
```

```

# Iterar sobre la cadena de caracteres
for i, char in enumerate(s):
    # Si el carácter ya ha sido encontrado y está en el
rango de la subcadena actual
    if char in char_index and start <= char_index[char]:
        start = char_index[char] + 1
    else:
        max_length = max(max_length, i - start + 1)
    # Actualizar la última posición del carácter actual
    char_index[char] = i
return max_length

```

Ejemplo 3: Valid parentheses

```

def is_valid(s):
    # Mapa de pares de paréntesis
    bracket_map = {"(": ")", "[": "]", "{": "}"
    stack = []
    # Iterar sobre cada carácter en la cadena
    for char in s:
        # Si es un paréntesis de cierre
        if char in bracket_map:
            # Comparar con el último paréntesis de apertura en
la pila
            top_element = stack.pop() if stack else '#'
            if bracket_map[char] != top_element:
                return False
        else:
            stack.append(char)
    return not stack

```

Ejemplo 4: Merge two sorted lists

```

class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val

```

```

        self.next = next

def merge_two_lists(l1, l2):
    dummy = ListNode()
    current = dummy
    # Comparar los nodos de ambas listas enlazadas
    while l1 and l2:
        if l1.val < l2.val:
            current.next = l1
            l1 = l1.next
        else:
            current.next = l2
            l2 = l2.next
        current = current.next
    # Adjuntar el resto de la lista que aún tiene nodos
    current.next = l1 or l2
    return dummy.next

```

Ejemplo 5: Median of two sorted arrays

```

def find_median_sorted_arrays(nums1, nums2):
    # Unir ambas listas y ordenarlas
    nums = sorted(nums1 + nums2)
    length = len(nums)
    # Calcular la mediana
    if length % 2 == 1:
        return nums[length // 2]
    else:
        return (nums[length // 2 - 1] + nums[length // 2]) / 2

```