Taller 4

Jose David Ramirez Beltran - 506222723

10 de septiembre de 2023

1. Introducción

La recursividad evita el uso de bucles y otros iteradores mediante funciones que se llaman a sí mismas.

Calcular el factorial de un número entero es un ejemplo sencillo que se utiliza con frecuencia. El factorial de un número se define como ese número multiplicado por el que le precede, por el que le sigue, y así sucesivamente hasta que es igual a 1.

Por ejemplo, el factorial del número 5 sería 5x4x3x2x1 = 120.

"Tomando el factorial como base para un ejemplo, ¿cómo podemos crear una función que calcule el factorial de un número? Bueno, existen multitud de formas. La más obvia quizá sería simplemente usar un bucle determinado para hacerlo, algo así en JavaScript:

```
function factorial(n){
  var res = 1;
  for(var i=n; i>=1; i--){
    res = res * i;
  }
  return res;
}
```

Sin embargo hay otra forma de hacerlo sin necesidad de usar ninguna estructura de bucle que es mediante recursividad. Esta versión de la función hace exactamente lo mismo, pero es más corta, más simple y más elegante:

```
function factorial(n) {
    if (n<=1) return 1;
    return n* factorial(n-1);
}
```

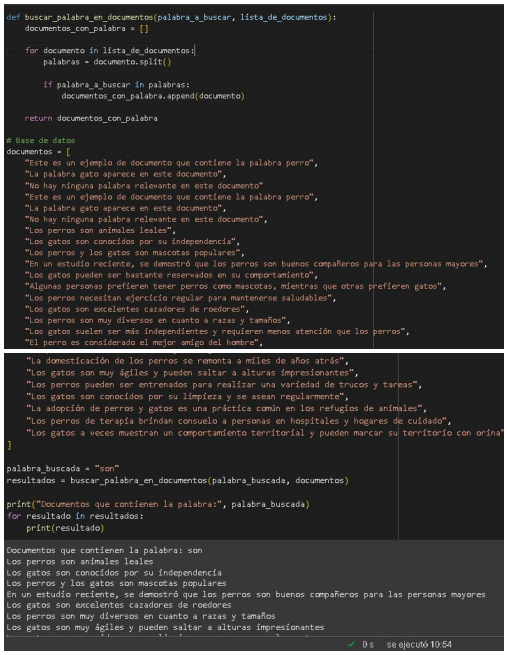
Aquí lo que se hace es que la función se llama a sí misma (recursividad), y deja de llamarse cuando se cumple la condición de parar en este caso que el argumento sea menor o igual que 1 (condicional)."

2. Desarrollo

Para llevar el desarrollo del algoritmo, se puede plantear de la siguiente manera:

1. Definir una lista de documentos.
2. Definir una función que tome como parámetros la palabra a buscar y la lista de documentos.
3. Dentro de la función, crear una lista vacía para almacenar los documentos que contienen la palabra.
4. Iterar a través de cada documento en la lista de documentos.
5. Para cada documento, dividir el texto en palabras utilizando el método split().
6. Verificar si la palabra a buscar está en la lista de palabras del documento utilizando el operador ïn".
7. Si la palabra está en el documento, agregar ese documento a la lista de resultados.
8. Al final de la función, que devuelva la lista de documentos que contienen la palabra.

3. Ejecucion



```
def buscar_palabra_en_documentos(palabra_a_buscar, lista_de_documentos):
    documentos_con_palabra = []

    for documento in lista_de_documentos:
        palabras = documento.split()

        if palabra_a_buscar in palabras:
            documentos_con_palabra.append(documento)

    return documentos_con_palabra

# Base de datos
documentos = [
    "Este es un ejemplo de documento que contiene la palabra perro",
    "La palabra gato aparece en este documento",
    "No hay ninguna palabra relevante en este documento"
    "Este es un ejemplo de documento que contiene la palabra perro",
    "La palabra gato aparece en este documento",
    "No hay ninguna palabra relevante en este documento",
    "Los perros son animales leales",
    "Los gatos son conocidos por su independencia",
    "Los perros y los gatos son mascotas populares",
    "En un estudio reciente, se demostró que los perros son buenos compañeros para las personas mayores",
    "Los gatos pueden ser bastante reservados en su comportamiento",
    "Algunas personas prefieren tener perros como mascotas, mientras que otras prefieren gatos",
    "Los perros necesitan ejercicio regular para mantenerse saludables",
    "Los gatos son excelentes cazadores de roedores",
    "Los perros son muy diversos en cuanto a razas y tamaños",
    "Los gatos suelen ser más independientes y requieren menos atención que los perros",
    "El perro es considerado el mejor amigo del hombre",
    "La domesticación de los perros se remonta a miles de años atrás",
    "Los gatos son muy ágiles y pueden saltar a alturas impresionantes",
    "Los perros pueden ser entrenados para realizar una variedad de trucos y tareas",
    "Los gatos son conocidos por su limpieza y se asean regularmente",
    "La adopción de perros y gatos es una práctica común en los refugios de animales",
    "Los perros de terapia brindan consuelo a personas en hospitales y hogares de cuidado",
    "Los gatos a veces muestran un comportamiento territorial y pueden marcar su territorio con orina"
]

palabra_buscada = "son"
resultados = buscar_palabra_en_documentos(palabra_buscada, documentos)

print("Documentos que contienen la palabra:", palabra_buscada)
for resultado in resultados:
    print(resultado)
```

```
Documentos que contienen la palabra: son
Los perros son animales leales
Los gatos son conocidos por su independencia
Los perros y los gatos son mascotas populares
En un estudio reciente, se demostró que los perros son buenos compañeros para las personas mayores
Los gatos son excelentes cazadores de roedores
Los perros son muy diversos en cuanto a razas y tamaños
Los gatos son muy ágiles y pueden saltar a alturas impresionantes
```

0 s se ejecutó 10:54

4. Funcionamiento

El código emplea la recursividad para buscar la palabra especificada en una lista de documentos, ya que cada vez que se realiza una llamada recursiva, se examina un documento en la posición actual (indicada por el índice). El contenido del documento se descompone en palabras y se comprueba la presencia de la palabra buscada.

Si se encuentra, el documento se añade a una lista de resultados, luego la función se llama a sí misma de forma recursiva con el siguiente documento de la lista, aumentando el indicador; ese proceso continuará hasta que se hayan revisado todos los documentos y al finalizar el término de las llamadas recursivas, se devuelven los resultados acumulados, imprimiendo una lista de documentos que contienen el término buscado.

5. Coclusiones

En conclusión, la comprensión de estas dimensiones y su adecuada aplicación en la programación es esencial para el éxito de un proyecto de desarrollo de software en un mundo cada vez más dependiente de la tecnología, así como para la eficacia de las propias aplicaciones y que podamos ver que tan eficiente es un codigo comparado con otro, así mismo como sus diferencias entre equipos y las capacidades y requerimientos que cada programa exige a un dispositivo o equipo.

6. Bibliografia

https://github.com

https://www.campusmvp.es/recursos/post/Que-es-la-recursividad-o-recursion-Un-ejemplo-con-JavaScript.aspx

https://colab.research.google.com/drive/1HMJJESce4BVmIoxJGSr