

# Taller 5, Buscador de Palabras

Miguel Daniel Ruiz Silva, Camila Andrea Galindo Ruiz, Daniel Felipe Gonzalez Beltran

## I. INTRODUCCION

En este taller, desarrollaremos un microservicio utilizando el framework FastAPI que implemente un algoritmo de índices invertidos. Además, aprenderemos a desplegar este microservicio en un servidor en la nube, como DigitalOcean. La importancia de hacer uso de índices invertidos es por que son como el GPS de la programación. Nos ayudan a encontrar información en grandes conjuntos de datos de manera rápida y eficiente. Son la clave detrás de los motores de búsqueda, bases de datos y sistemas de recomendación, haciendo que nuestras aplicaciones funcionen de manera ágil y precisa. De la misma manera entender el funcionamiento de las API'S y de como desplegar un micro servicio, es fundamental para entender de manera directa que las API son como puentes entre aplicaciones, y los microservicios son piezas ágiles de software que construyen sistemas modernos.

## II. ALGORITMO DE ÍNDICES INVERTIDOS

Este es el código que hace uso de índices invertidos para la búsqueda de una palabra en un archivo secundario llamado mook, en donde esta el documento a revisar. El retorno esperado de este algoritmo, son las líneas del documento en donde aparece su repetición.

```

1  from fastapi import FastAPI
2
3  import mook
4
5  app = FastAPI()
6
7  @app.get("/")
8  def root():
9      return{
10         "Servicio": "Estructuras de datos"
11     }
12
13 @app.post("/indices-invertidos")
14 def indices_invertidos(palabra: dict):
15     cache = {}
16     for index, documento in enumerate(mook.my_documents):
17         words = documento.lower().split()
18         for word in words:
19             if word in cache:
20                 cache[word].append(mook.my_documents[index])
21             else:
22                 cache[word] = [mook.my_documents[index]]
23
24     return cache.get(palabra["palabra"], "No se encontro")
25
26

```

Fig. 1. Algoritmo Implementador de Índices Invertidos

El funcionamiento del algoritmo de búsqueda es el siguiente: - El algoritmo tiene 2 partes, en la línea 7 tenemos lo

que va a devolver nuestro microservicio al hacer una petición de tipo get, al hacer una consulta de este tipo nos retorna:

"Servicio": "Estructuras de datos"

- La segunda parte del algoritmo es la que hace la búsqueda de la palabra deseada en nuestro archivo mook, este algoritmo hace uso de memorización en la función índices invertidos, la función toma un parámetro llamado palabra, que se espera que sea un diccionario, el for recorre el archivo mook y dentro del bucle, se convierte el contenido del documento a minúsculas y se divide en palabras individuales. Estas palabras se almacenan en la lista words.

En el siguiente for se recorre sobre cada palabra del diccionario, se verifica si ya esta en el cache sino esta en el cache se le agrega al diccionario cache y por ultimo en la línea 22 se utiliza para construir un índice invertido en el diccionario cache, donde cada palabra se asocia con una lista de documentos que la contienen.

## III. MICROSERVICIO

El siguiente paso en nuestro proceso es enlazar nuestro algoritmo de búsqueda con nuestra fast api de esta manera ya tendríamos un gran porcentaje de nuestro taller realizado, este proceso se hace mediante el uso de la consola de python en donde vamos a acceder a nuestro servidor y enlazarlo con el repositorio en github de nuestro proyecto.

## IV. DIGITALOCEAN

Una vez ya tenido nuestro micro servicio que es completamente independiente, autónomo y que realiza una tarea específica, el siguiente paso es conectarlo con digital ocean que es una plataforma que nos permite implementar y administrar los distintos microservicios que tengamos, las ventajas que nos da digital ocean es la escalabilidad y flexibilidad con la que nos permite ajustar los recursos y necesidades de nuestro micro servicio, aparte de ello ofrece servicios de administración que simplifican la gestión de múltiples microservicios, este es el principal motivo por el cual usamos esta plataforma para poder administrar todos los microservicios que vamos a ir agregando a nuestro servicio en la medida del tiempo.

## V. REFERENCIAS

DigitalOcean. (2023). Digital Ocean Download. Recuperado de <https://www.digitalocean.com/>

FastAPI. (2023).Fast API Download. Recuperado de <https://fastapi.tiangolo.com/>