

RETO 1

Alejandra Melo

Cod: 202021526

Correo: a.melo4@uniandes.edu.co

Obed Cabanzo

Cod: 201911749

Correo: jo.cabanzo@uniandes.edu.co

- **REQUERIMIENTO 1 (Equipo de trabajo)**

La complejidad del método es: $O(N^2)$

```
def getTrendingVideos(catalog, category_name, country, n):
    categorys = catalog['category']
    count = lt.size(categorys)
    inicio = 0
    while inicio <= count:
        elemento = lt.getElement(categorys, inicio)
        if elemento['name'] == category_name.strip():
            categ = elemento.copy()
            videos = categ['videos']
            tamañoVideos = lt.size(videos)
            cont = 0
            listaVideos = lt.newList('ARRAY_LIST')
            tamañolv = size(listaVideos)
            while cont <= tamañoVideos:
                video = lt.getElement(videos, cont)
                if video['country'] == country:
                    lt.addLast(listaVideos, video)
                cont += 1
            listaOrdenada = sa.sort(listaVideos, cmpVideosByViews)
            if int(n) <= tamañolv:
                listaFinal = lt.subList(listaOrdenada, 0, tamañolv)
            elif listaVideos == None:
                listaFinal == None
            elif int(n) >= tamañolv:
                listaFinal = lt.subList(listaOrdenada, 0, int(n))

        inicio += 1

    return listaFinal
```

- REQUERIMIENTO 2 (Alejandra Melo)

La complejidad del método es: $O(3N)$, lo cual se aproxima a $O(N)$ -> lineal

```
def getVideosByCountry(catalog, countryname):
    lista_videos_pais = []
    dict = {}
    dict_repeticiones = {}

    for video in lt.iterator(catalog["videos"]):
        if video["country"] == countryname:
            dict["video"] = video["title"]
            dict["canal"] = video["channel_title"]
            dict["id"] = video["video_id"]
            lista_videos_pais.append(dict)
            if video["video_id"] not in dict_repeticiones:
                dict_repeticiones[video["video_id"]] = 1
            else:
                dict_repeticiones[video["video_id"]] += 1
            dict = {}

    mas_trending = ""
    dias_trending = 0
    for video in dict_repeticiones:
        if dict_repeticiones[video] > dias_trending:
            dias_trending = dict_repeticiones[video]
            mas_trending = video

    for pos in range(0, len(lista_videos_pais)):
        if lista_videos_pais[pos]["id"] == mas_trending:
            return (lista_videos_pais[pos], dias_trending)
```

- REQUERIMIENTO 3 (Obed Cabanzo)

La complejidad del método es: $O(N^2)$

```
def getVideosByCategory (catalog, category_name):  
    categorys = catalog['category']  
    count = lt.size(categorys)  
    inicio = 0  
    while inicio <= count:  
        elemento = lt.getElement(categorys, inicio)  
        if elemento['name'] == category_name.strip():  
            categ = elemento.copy()  
            videos = categ['videos']  
            tamañoVideos = lt.size(videos)  
            diasMayor = 0  
            cont = 0  
            video = None  
            while cont <= tamañoVideos:  
                video = lt.getElement(videos, cont)  
                dias = contarDias(video)  
                if dias >= diasMayor:  
                    resultado = [video, dias]  
                    diasMayor = dias  
                cont = cont + 1  
            inicio = count + 1  
        else:  
            inicio += 1  
            resultado = 'No se encontró la categoria. '  
    return resultado
```

Método auxiliar:

La complejidad del método es: $O(k)$

```
def contarDias(video):  
  
    listaPos = video  
    ultimoDia = listaPos['publish_time']  
    primerDia = listaPos['trending_date']  
  
    uDia = ultimoDia.split('T')  
    uSeparado = uDia[0].split('-')  
    sAño = int(uSeparado[0])  
    sMes = int(uSeparado[1])  
    sDia = int(uSeparado[2])  
  
    diaTrend = date(sAño, sMes, sDia)  
  
    pSeparado = primerDia.split('.')  
    pAño = 2000 + int(pSeparado[0])  
    pMes = int(pSeparado[2])  
    pDia = int(pSeparado[1])  
  
    diaPubli = date(pAño, pMes, pDia)  
  
    diferencia = diaPubli - diaTrend  
    resultado = diferencia.days  
    return int(resultado)
```

- REQUERIMIENTO 4 (Trabajo en equipo)

La complejidad del método es: $O(2N + 3/2N^2)$ que se aproxima a $O(N^2)$ -> cuadrática

```
def getVideosByLikes(catalog, n, countryname, tag):
    list_videos_pais = []
    dict = {}
    lista_videos_likes = []

    for video in lt.iterator(catalog["videos"]):
        if video["country"] == countryname:
            if tag in video["tags"]:
                dict["id"] = video["video_id"]
                dict["title"] = video["title"]
                dict["cannel_title"] = video["channel_title"]
                dict["publish_time"] = video["publish_time"]
                dict["views"] = video["views"]
                dict["likes"] = video["likes"]
                dict["dislikes"] = video["dislikes"]
                dict["tags"] = video["tags"]
                dict["trending_date"] = video["trending_date"]
                lista_videos_likes.append((video["likes"], video["video_id"]))
                list_videos_pais.append(dict)
                dict = {}

    lista_videos_likes.sort()

    lista_id_elegido = []
    lista_likes_elegido = []

    pos = len(lista_videos_likes) - 1
    x = len(lista_videos_likes) - n

    if x > 0:
        while pos >= x and x >= 0:
            if lista_videos_likes[pos][1] not in lista_id_elegido:
                lista_id_elegido.append(lista_videos_likes[pos][1])
                lista_likes_elegido.append(lista_videos_likes[pos][0])
            else:
                x -= 1
                pos -= 1

    else:
        while pos >= 0:
            if lista_videos_likes[pos][1] not in lista_id_elegido:
                lista_id_elegido.append(lista_videos_likes[pos][1])
                lista_likes_elegido.append(lista_videos_likes[pos][0])
            pos -= 1

    respuesta = getInfoVideos(list_videos_pais, lista_id_elegido, lista_likes_elegido)
    return respuesta
```

Método auxiliar:

La complejidad del método es: $O()$

```
def getInfoVideos(lista1, lista2, lista3):  
    lista_final = []  
    for elemento in range(0, len(lista2)):  
        for d in range(0, len(lista1)):  
            if lista1[d]["id"] == lista2[elemento] and lista1[d]["likes"] == lista3[elemento]:  
                lista_final.append(lista1[d])  
  
    return lista_final
```