Análisis de Complejidad temporal en notación O

Req 1

```
def instancesPerCharact(catalog, charact, valmax, valmin):
    valores = om.valueSet(catalog['byDates'])
    artists = om.newMap(omaptype='BST',comparefunction=compareIds)
    trackcount = 0
    artistcount = 0
    for index in range(0, lt.size(valores)):
        data = lt.getElement(valores, int(index))
        if float(data[charact]) >= valmin and float(data[charact]) <= valmax:
            trackcount += 1
            if om.contains(artists, data['artist_id']) == False:
                om.put(artists, data['artist_id'], data)


    print("Cuenta reproducciones: "+str(trackcount))
    print("Cuenta artistas: "+str(om.size(artists)))
    return None
```

O(n) donde n= # de eventos

Total:O(2n) $\approx o(n)$

Req 2(santiago rodriguez cruz) y 3(juan pablo tatis goenaga)

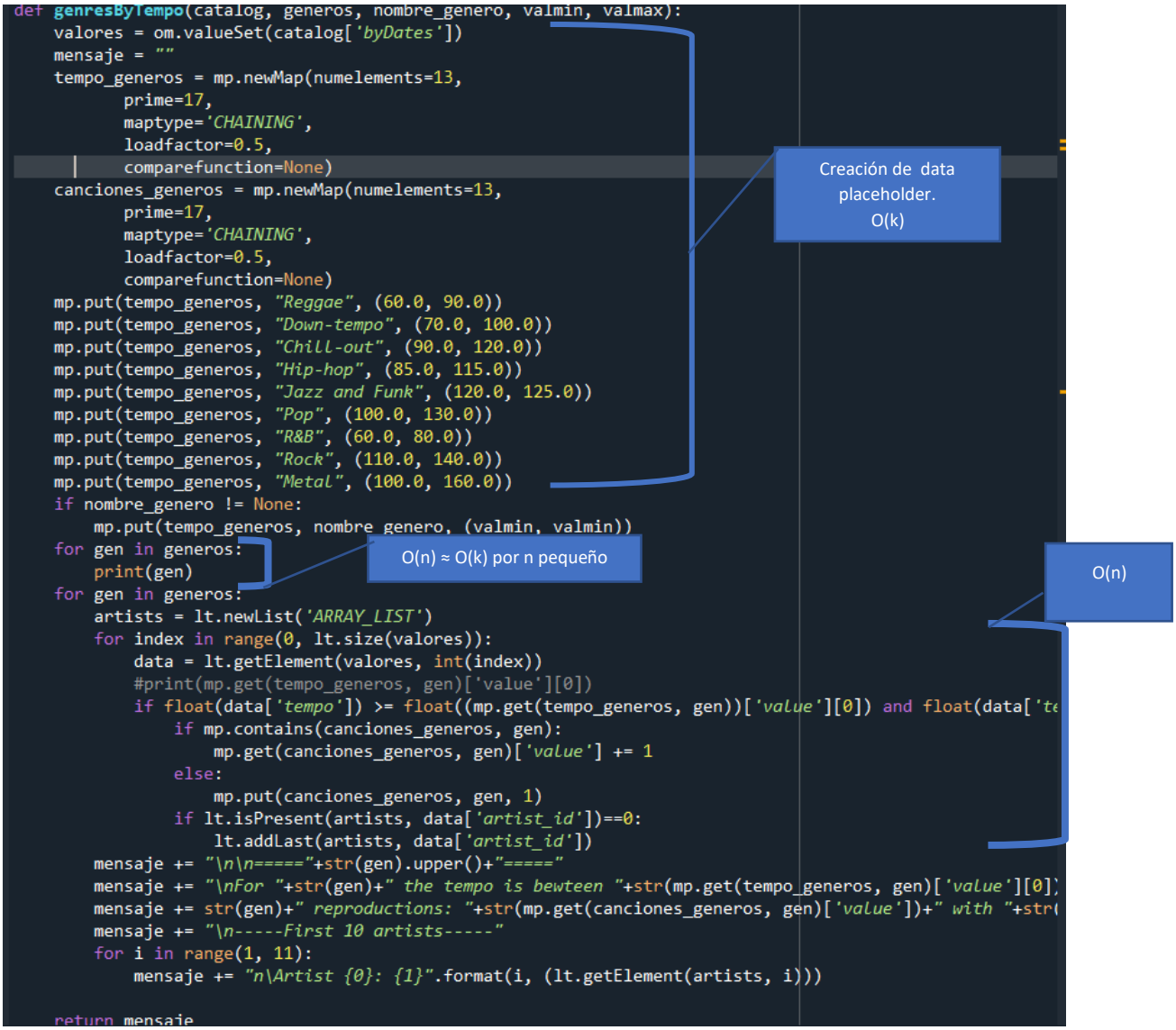
```
def studyRecomend(catalog, valmaxtemp, valmintemp, valmaxinst, valmininst):
    valores = om.valueSet(catalog['byDates'])
    tracklist = lt.newList('ARRAY_LIST')
    trackcheck = lt.newList('ARRAY_LIST')
    generated = 1
    for index in range(0, lt.size(valores)):
        data = lt.getElement(valores, int(index))
        if float(data['instrumentalness']) >= valmininst and float(data['instrumentalness']) <= valmaxinst \
        and float(data['tempo']) <= valmaxtemp and float(data['tempo']) >= valmintemp and lt.isPresent(trackcheck, data['
            lt.addLast(trackcheck, data['track_id'])
            lt.addLast(tracklist, data)

    mensaje = ('Total of unique tracks in events:'+str(lt.size(tracklist)))
    print(mensaje)
    repeated = []
    while generated < 6:
        trackindex = int(rdm.randint(0, int(lt.size(tracklist))))
        if (trackindex in repeated) == False:
            trackdata = lt.getElement(tracklist, int(trackindex))
            mensaje += ('\nTrack {0}: {1} with instrumentalness of {2} and tempo of {3}.'.format(str(generated), str(trac
            generated += 1

    return mensaje
```

O(n)

O(k) donde k=7

Total:O(n)+7 ≈ $o(n)$

Req 4:




```
def genresByTempo(catalog, generos, nombre_genero, valmin, valmax):
    valores = om.valueSet(catalog['byDates'])
    mensaje = ""
    tempo_generos = mp.newMap(numelements=13,
           prime=17,
           maptype='CHAINING',
           loadfactor=0.5,
           comparefunction=None)
    canciones_generos = mp.newMap(numelements=13,
           prime=17,
           maptype='CHAINING',
           loadfactor=0.5,
           comparefunction=None)
    mp.put(tempo_generos, "Reggae", (60.0, 90.0))
    mp.put(tempo_generos, "Down-tempo", (70.0, 100.0))
    mp.put(tempo_generos, "Chill-out", (90.0, 120.0))
    mp.put(tempo_generos, "Hip-hop", (85.0, 115.0))
    mp.put(tempo_generos, "Jazz and Funk", (120.0, 125.0))
    mp.put(tempo_generos, "Pop", (100.0, 130.0))
    mp.put(tempo_generos, "R&B", (60.0, 80.0))
    mp.put(tempo_generos, "Rock", (110.0, 140.0))
    mp.put(tempo_generos, "Metal", (100.0, 160.0))
    if nombre_genero != None:
        mp.put(tempo_generos, nombre_genero, (valmin, valmin))
    for gen in generos:
        print(gen)
    for gen in generos:
        artists = lt.newList('ARRAY_LIST')
        for index in range(0, lt.size(valores)):
            data = lt.getElement(valores, int(index))
            #print(mp.get(tempo_generos, gen)['value'][0])
            if float(data['tempo']) >= float((mp.get(tempo_generos, gen))['value'][0]) and float(data['te
                if mp.contains(canciones_generos, gen):
                    mp.get(canciones_generos, gen)['value'] += 1
                else:
                    mp.put(canciones_generos, gen, 1)
                if lt.isPresent(artists, data['artist_id'])==0:
                    lt.addLast(artists, data['artist_id'])
        mensaje += "\n\n=====">+str(gen).upper()+"====="
        mensaje += "\nFor "+str(gen)+" the tempo is bewteen "+str(mp.get(tempo_generos, gen)['value'][0])
        mensaje += str(gen)+" reproductions: "+str(mp.get(canciones_generos, gen)['value'])+" with "+str(
        mensaje += "\n-----First 10 artists-----"
        for i in range(1, 11):
            mensaje += "n\Artist {0}: {1}".format(i, (lt.getElement(artists, i)))

    return mensaje
```

Total=O(n)+o(k) ≈ O(n)

Req 5