

Reto 1

Nombres:

Gabriela Carvajal Rojas – 202111058 – g.carvajal@uniandes.edu.co

Juanita Gil Arango – 202111556 – j.gila2@uniandes.edu.co

Análisis de complejidad:

Requerimiento #1:

- Complejidad: $O(n)$

```
56 def orgartistasCro(catalog, inicial, final):
57     artistas=lt.newList()
58     for artista in lt.iterator(catalog['Artists']):
59         if artista['BeginDate']>= inicial and artista['BeginDate']<= final:
60             informacion= lt.newList()
61             lt.addLast(informacion, artista['DisplayName'])
62             lt.addLast(informacion, artista['BeginDate'])
63             lt.addLast(informacion, artista['EndDate'])
64             lt.addLast(informacion, artista['Nationality'])
65             lt.addLast(informacion, artista['Gender'])
66             lt.addLast(artistas,informacion)
67     totalArtistas= lt.size(artistas)
68     return (artistas, totalArtistas)
```

Requerimiento #2:

- Complejidad: $O(1)$:

```
81 def compararIDayo(catalog, id):
82     #en id entraria el constituint ID del artworks
83     for artist in lt.iterator(catalog['Artist']):
84         if id == artist['ConstituentID']:
85             nomArtista = artist['DisplayName']
86             return nomArtista
87
```

- Complejidad: $O(n)$

```
88 def orgObrasCro(catalog, inicial, final):
89     obras =lt.newList()
90     conteoObras = 0
91     for obra in lt.iterator(catalog['Artwoks']):
92         if obra['DateAcquired']>= inicial and obra['DateAcquired']<= final:
93             conteoObras += 1
94             informacion= lt.newList()
95             lt.addLast(informacion, obra['Title'])
96             lt.addLast(informacion, obra[compararIDayo(obra['ConstituentID'])])
97             lt.addLast(informacion, obra['Date'])
98             lt.addLast(informacion, obra['DateAcquired'])
99             lt.addLast(informacion, obra['Medium'])
100             lt.addLast(informacion, obra['Dimensions'])
101             lt.addLast(obras,informacion)
102     tupla = (obras, conteoObras)
103     """
104     obras->tupla[0], obras es la lista de las obras en el rango
105     conteoObras->tupla[1], conteoObras es el numero de obras en el rango
106     """
107     return tupla
```

- Complejidad: $O(n)$:

```

113 ✓ def numPurchase(catalog):
114     conteoPu = 0
115     for obra in lt.iterator(catalog['Artworks']):
116         if obra['CreditLine'] == 'Purchase':
117             conteoPu += 1
118     return conteoPu

```

Requerimiento #3

- Complejidad: $O(n-1)$:

```

124 def enconID(catalog, nombre):
125     i=0
126     f=len(catalog['Artists'])-1
127     pos=-1
128     id= False
129     while i <= f and id == False:
130         m=(i+f)//2
131         if catalog['Artists'][m]== nombre:
132             pos=m
133             id=True
134         elif catalog['Artists'][m] > nombre:
135             f=m-1
136         else:
137             i=m+1
138     encontrarid= catalog['Artists'][pos]['Constituent ID']
139     return encontrarid

```

- Complejidad: $O(n)$:

```

140 def tecnicasartista(catalog, encontrarid):
141     cantidadobras=0
142     tecnicas={}
143     for obra in lt.iterator(catalog['Artworks']):
144         if obra['Constituent ID'] == encontrarid:
145             cantidadobras+=1
146             tecnica= obra['Medium']
147             if tecnica in tecnicas:
148                 lt.addLast(tecnicas[tecnica],obra['Title'])
149             else:
150                 tecnicas[tecnica]=lt.newList()
151                 lt.addLast(tecnicas[tecnica],obra['Title'])
152     return (cantidadobras, tecnicas)

```

- Complejidad: $O(n^2)$:

```

158 ✓ def tecnicasusada(tecnicas:dict):
159     mayor=0
160     for categoria in tecnicas:
161         size= lt.size(categoria)
162         if size > mayor:
163             mayor= size
164         for key in tecnicas.keys():
165             if key == tecnicas['categoria']:
166                 masusada = key
167     return masusada

```

- Complejidad: $O(n)$:

```

169 def listaObras(catalog, masusada, tecnicas):
170     for obra in lt.iterator(catalog['Artworks']):
171         if obra['Title'] in tecnicas[masusada]:
172             info=lt.newList()
173             lt.addLast(info, obra['Title'])
174             lt.addLast(info, obra['Date'])
175             lt.addLast(info, obra['Medium'])
176             lt.addLast(info, obra['Dimensions'])
177             obras=lt.newList()
178             lt.addLast(obras,info)
179     return obras

```

Requerimiento #4:

- Complejidad: $O(1)$:

```

182 def idArtists(catalog):
183     for artist in lt.iterator(catalog['Artists']):
184         id = artist['ConstituentID']
185     return id

```

- Complejidad: $O(n)$:

```

186 def idyNacio(catalog, id):
187     nacioNombre = {}
188     #en id entraria el constituent ID de artists (return idArtists)
189     for obra in lt.iterator(catalog['Artworks']):
190         if id == obra['ConstituentID']:
191             for artista in lt.iterator(catalog['Artists']):
192                 nacionalidad = artista['Nationality']
193                 nacioNombre[nacionalidad]= ''
194     return nacioNombre

```

- Complejidad: $O(n)$:

```

195 def contNacio(catalog, nacioNombre: dict):
196     conteoNa = 0
197     for artist in lt.iterator(catalog['Artists']):
198         nacionalidad = artist['Nationality']
199         if nacionalidad in nacioNombre.keys():
200             conteoNa += 1
201             nacioNombre[nacionalidad] = conteoNa
202     return nacioNombre

```

- Complejidad: $O(n)$:

```

204 def Top10(nacioNombre: dict):
205     valoresNacio=lt.newList()
206     top10= lt.newList()
207     for nacio in nacioNombre:
208         lt.addLast(valoresNacio, nacio)
209     valoresOrdenados= sa.sort(valoresNacio)
210     valorestop10= lt.subList(valoresOrdenados, 1, 10)
211     for valor in lt.iterator(valorestop10):
212         for valornacionalidad in nacioNombre.keys():
213             if nacioNombre[valornacionalidad] == valor:
214                 nacionalidad= valornacionalidad
215                 lt.addLast(top10, nacionalidad)
216     return top10

```

- Complejidad: $O(n^2)$:

```

218 def nacioMasObras(top10, catalog):
219     uno = lt.getElement(top10,1)
220     for artista in lt.iterator(catalog['Artists']):
221         nacionalidad = artista['Nationality']
222         if nacionalidad == uno:
223             id= artista['ConstituentID']
224             for obra in lt.iterator(catalog['Artworks']):
225                 if id == obra['ConstituentID']:
226                     for obra in top10:
227                         x = lt.newList
228                         lt.addLast(x, obra['Title'])
229                         lt.addLast(x, obra['Date'])
230                         lt.addLast(x, obra['Medium'])
231                         lt.addLast(x, obra['Dimensions'])
232                         lt.addLast(x, obra[compararIDayo(obra['ConstituentID'])])
233                     obrasNa = lt.newList
234                     lt.addLast(obrasNa,x)
235     return obrasNa

```

- Complejidad: $O(n^2)$:

```

237 def lista_nacionalidades(nacioNombre: dict):
238     mayor=0
239     top10= 0
240     lst_nacio_ord = lt.newList
241     for obra in nacioNombre:
242         size= lt.size(obra)
243         if size > mayor:
244             mayor= size
245             while top10 <= 10:
246                 key = nacioNombre.keys()
247                 if key == nacioNombre['Nationality']:
248                     nacionalidad_mas_repetida = key
249                     top10+= 1
250                     lst_top10_final = lt.addLast(lst_nacio_ord,nacionalidad_mas_repetida)
251     return lst_top10_final

```

Requerimiento #5:

- Complejidad: $O(n)$:

```

256 def obrasDepartamento(departamento, catalog):
257     lista= lt.newList()
258     for obra in lt.iterator(catalog['Artworks']):
259         if catalog['Artworks']['Department'] == departamento:
260             lt.addLast(lista, obra)
261     return lista

```

- Complejidad: $O(n)$:

```

262 def listafechas(lista):
263     listafechas= lt.newList('SINGLE_LINKED')
264     for obra in lista:
265         f= obra['Date']
266         t= obra['Title']
267         lt.addLast(listafechas, {'fecha': f, 'titulo': t})
268     return listafechas

```

- Complejidad: $O(n)$:

```

270 def ordenar(o1,o2):
271     return o1['fecha']<o2['fecha']

```

- Complejidad: $O(n)$:

```
273 def ordenarlista(listafechas):
274     listaOrdenada=sa.sort(listafechas, ordenar)
275     return listaOrdenada
```

- Complejidad: $O(n)$:

```
277 def listaprecios(costoObras:dict):
278     listaprecios= lt.newList('SINGLE_LINKED')
279     for llave in costoObras.keys():
280         costo= costoObras[llave]
281         lt.addLast(listafechas, {'costo': costo, 'titulo': llave})
282     return listaprecios
```

- Complejidad: $O(n)$:

```
287 def ordenarlista(listaprecios):
288     listaOrdenadaprecios=sa.sort(listaprecios, ordenar)
289     return listaOrdenadaprecios
```

- Complejidad: $O(n)$:

```
291 def pesototal(lista):
292     peso=0
293     for obra in lt.iterator(lista):
294         pesoObra= int(obra['Weight'])
295         peso+= pesoObra
296     return peso
```

- Complejidad: $O()$:

```

302 def dictCostos(lista):
303     costoObras={}
304     for obra in lt.iterator(lista):
305         altura=obra['Height']
306         longitud=obra['Length']
307         peso=obra['Weigth']
308         ancho= obra['Width']
309         if (altura== '' or longitud=='') and peso=='':
310             costoObras[obra['Title']] = 48
311         else:
312             mayor=0
313             costos=lt.newList()
314             if longitud != '' and altura!= '':
315                 area= (altura*longitud)/100
316                 precioArea= 72/area
317                 lt.addLast(costos, precioArea)
318                 if ancho!='':
319                     volumen= (altura*longitud*ancho)/100
320                     precioVolumen= 72/volumen
321                     lt.addLast(costos, precioVolumen)
322             if peso != '':
323                 precioPeso= 72/peso
324                 lt.addLast(costos, precioPeso)
325             for precio in lt.iterator(costos):
326                 if precio> mayor:
327                     mayor= precio
328             costoObras[obra['Title']] = mayor
329     return costoObras

```

Tiempos de ejecución:

Requerimientos individuales:

Req. 3: Gabriela Carvajal

Req. 4: Juanita Gil