

Proyecto - SEGQL

Nuestra solución al proyecto SEGQL, se encuentra almacenada en un repositorio en Github, al cual se puede acceder por medio del siguiente enlace: <https://github.com/JCibeira/SEGQL>

En dicho repositorio podrá encontrar los diferentes archivos que conforman la solución, los cuales se nombran a continuación:

- Gemfile.
- Globals.rb.
- Query.rb
- README.md.
- SEGQL.rb.
- file.json.

Tal y como se menciona anteriormente, los archivos .rb necesarios para poder ejecutar la solución son: Globals.rb, Query.rb, SEGQL.rb.

Adicionalmente, se hizo uso de un archivo llamado "file.json" (el cual se tenía guardado de laboratorios previos), este archivo fue de mucho utilidad para poder realizar las pruebas necesarias a la solución implementada en los casos en los que no funcionaba el sistema "bom" (que tiene un tiempo considerable sin funcionar). Este archivo json es el resultado de una segmentación ya hecha en una página específica y con parámetros específicos que se realizaron en alguno de los laboratorios de la materia.

Para ejecutar la solución implementada usando el sistema "bom" debe comentar la línea 260 y descomentar las líneas de la 251 a la 259 del archivo Query.rb, como se muestra en la siguiente imagen:

```
250     def get_blocks()
251         if($STATUS != 'data stored in Cache')
252             if(get_request('segment', @segment_params))
253                 $STATUS = 'data stored in Cache'
254             else
255                 return false
256             end
257         end
258
259         return bom_json(@segment_params['key'])
260         #return file_json()
261     end
```

Para ejecutar la solución implementada usando el archivo "file.json" debe comentar las líneas de la 251 a la 259 y descomentar la línea 260 del archivo Query.rb, como se muestra en la siguiente imagen:

```

250     def get_blocks()
251         #if($STATUS != 'data stored in Cache')
252         #     if(get_request('segment', @segment_params))
253         #         $STATUS = 'data stored in Cache'
254         #     else
255         #         return false
256         #     end
257         #end
258
259         #return bom_json(@segment_params['key'])
260         return file_json()
261     end

```

Puntos extra:

El mecanismo propuesto para no recalcular el pA cada vez que se unen dos bloques es el siguiente:

- Lo primero que se realiza es el ordenamiento de forma ascendente por el área de los bloques que intervienen en la segmentación.
- Luego de esto se procede a calcular área acumulada y con ello los valores máximos y mínimos de esta.
- Se divide en 10 intervalos usando un incremento de $(\text{max_acum} - \text{min_acum}) / 10$, desde min_acum hasta max_acum, para ello se crea un arreglo de números flotantes donde en cada posición se encontrará el límite superior de cada uno de ellos.
- Luego se suman las áreas de los bloques a mezclar y se busca el pA correspondiente en el intervalo generado, siendo el índice + 1 del arreglo anterior.