

Análisis del proceso de optimización

Manejando los joins desde ORACLE:

PUT

http://localhost:8080/RotondAndes/rest/restaurantes/consumo

Params

Send

Save

AuthorizationHeaders (1)BodyPre-request ScriptTests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "cedula":1000050,
3   "local":366520,
4   "fechaInicio":"01/01/17",
5   "fechaFin":"31/12/17"
6 }
```

BodyCookiesHeaders (6)Tests

Status: 200 OKTime: 2239 msSize: 457 B

PrettyRawPreviewJSON

```
1 [
2   {
3     "email": "email650549@correo.com",
4     "rol": "CLIENTE",
5     "nombre": "usuario650549",
6     "cedula": 650549
7   },
8   {
9     "email": "email57523@correo.com",
10    "rol": "CLIENTE",
11    "nombre": "usuario57523",
12    "cedula": 57523
13  },
14  {
15    "email": "email92061@correo.com",
16    "rol": "CLIENTE",
17    "nombre": "usuario92061",
18    "cedula": 92061
19  }
20 ]
```

Manejando los joins desde java:

PUT

http://localhost:8080/RotondAndes/rest/restaurantes/consumo

Params

Send

AuthorizationHeaders (1)BodyPre-request ScriptTests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "cedula":1000050,
3   "local":366520,
4   "fechaInicio":"01/01/17",
5   "fechaFin":"31/12/17"
6 }
```

BodyCookiesHeaders (6)Tests

Status: 500 Internal Server ErrorTime: 815803 ms

PrettyRawPreviewJSON

```
1 {
2   "ERROR": "Index: 692775, Size: 692775"
3 }
```

Análisis:

Como se puede observar en las imágenes el realizar el join en el código es increíblemente ineficiente (después de más de 800 segundos se lanzó un error que indicaba que se acabó la memoria del GC), por el contrario ejecutarlo con sentencias de ORACLE demora 2 segundos. Esto es debido a que ORACLE SQL DEVELOPER utiliza los algoritmos de join que optimizan al máximo el uso de la memoria (no hacen tantos accesos) lo contrario a java que hace un acceso por tupla.