Iteración 3

Carlos Mario Alba Rodríguez, Juan Diego Trujillo Montoya
Reporte técnico de la iteración 3
Universidad de los Andes, Bogotá, Colombia
{cm.alba10, jd.trujillom}@uniandes.edu.co
Fecha de presentación: mayo 20 de 2018

Tabla de contenido

1. Introducción 1
2. Análisis (1 %) 1
3. Diseño de la aplicación (64 %) 5
3.1. Análisis de impacto (1%) 5
3.2. Diseño físico. (63 %) 5
3.2.1. Documentación del diseño físico. (19 %) 5

1. Introducción

El presente documento demuestra el desarrollo de la iteración 3. En el mismo se podrá encontrar, en primer lugar, el análisis que se realiza del impacto que representa la adición de los nuevos requerimientos funcionales y las recomendaciones realizadas por los monitores en las anteriores entregas. Posteriormente, se presenta el diseño físico de la base de datos, donde se justifica el uso de los índices a favor de mejorar el rendimiento de las consultas SQL, correspondientes a los requerimientos funcionales de esta entrega. Después, se analizan los escenarios de pruebas propuestas y se realiza un análisis de eficiencia, todo esto sobre los requerimientos de consulta solicitados. Finalmente, se muestran evidencias de la implementación de la aplicación por medio de capturas donde se evidencia la carga de datos a la base de datos y se compara el desempeño en el desarrollo de las consultas por parte de manejador de la base datos de ORACLE y el uso de sentencias de control en Java.

2. Análisis (1 %)

Ajuste el modelo del mundo (modelo conceptual: diagrama de clases UML) propuesto en la iteración anterior, si lo requiere. Indique cuáles clases del modelo del mundo fueron actualizadas o creadas en esta iteración.

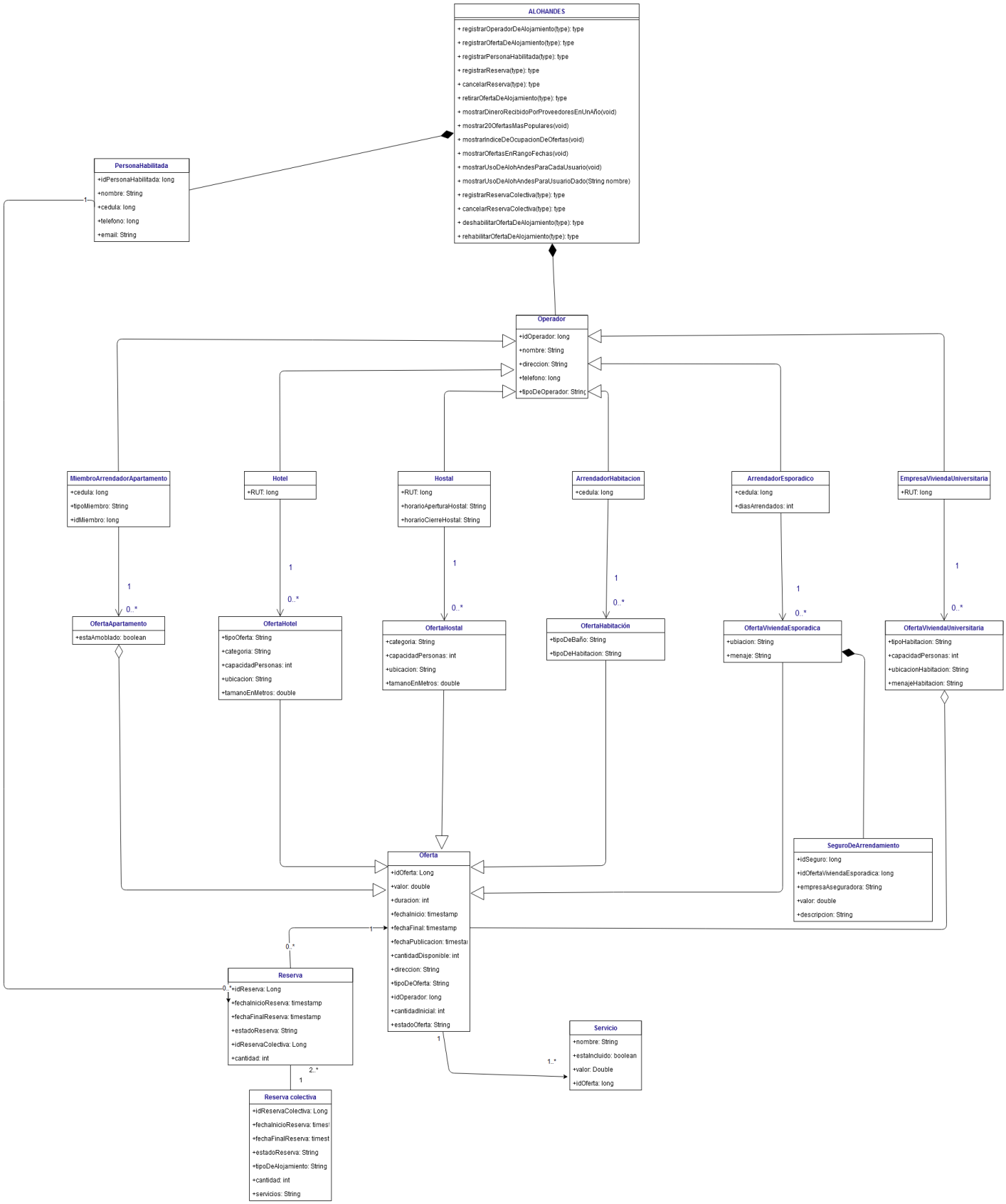
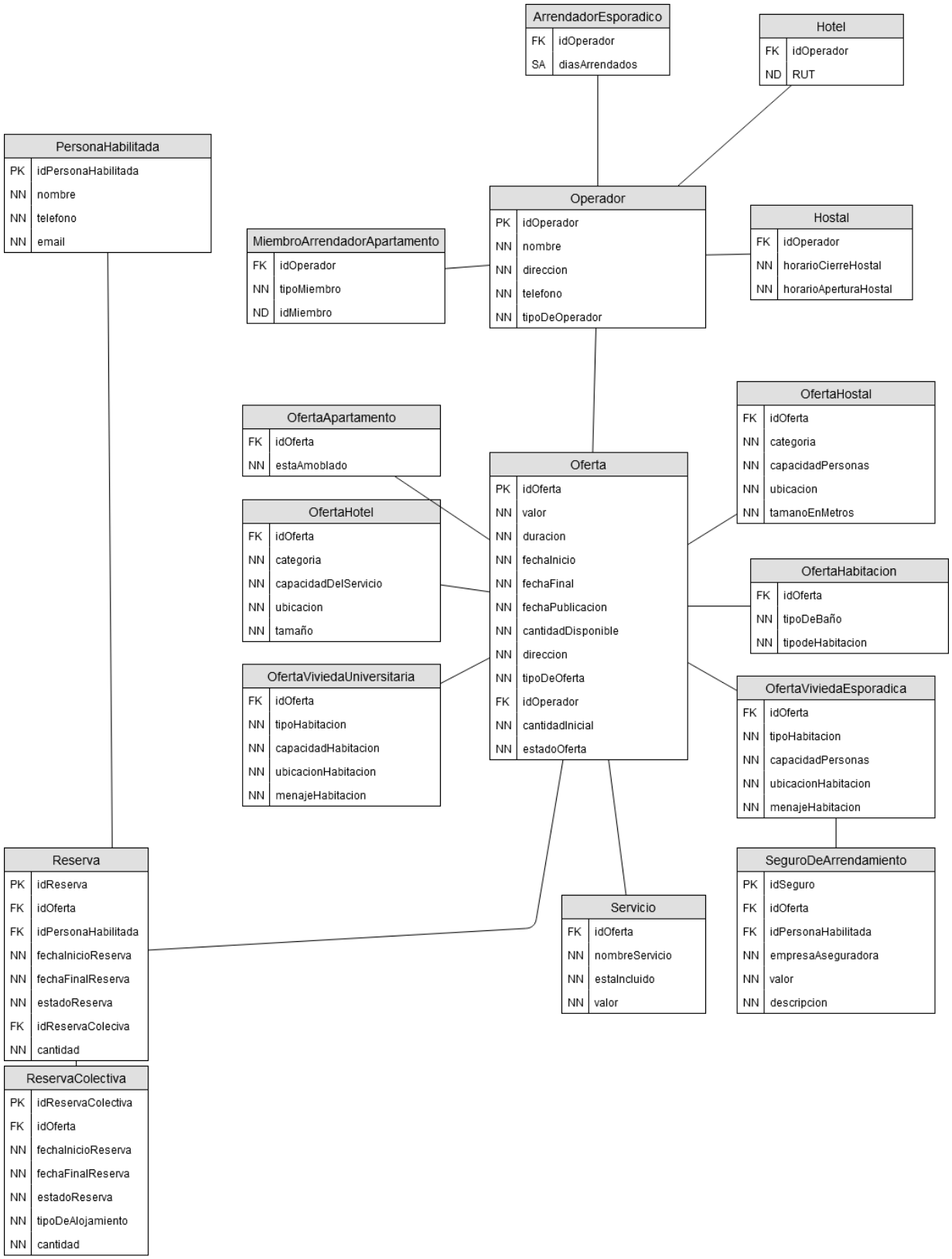


Figura 1. *Modelo conceptual*

Figura 2. Modelo relacional


ArrendadorEsporadico
FK idOperador
SA diasArrendados
Hotel
FK idOperador
ND RUT
PersonaHabilitada
PK idPersonaHabilitada
NN nombre
NN telefono
NN email
Operador
PK idOperador
NN nombre
NN direccion
NN telefono
NN tipoDeOperador
Hostal
FK idOperador
NN horarioCierreHostal
NN horarioAperturaHostal
MiembroArrendadorApartamento
FK idOperador
NN tipoMiembro
ND idMiembro
OfertaHostal
FK idOferta
NN categoria
NN capacidadPersonas
NN ubicacion
NN tamanoEnMetros
OfertaApartamento
FK idOferta
NN estaAmoblado
Oferta
PK idOferta
NN valor
NN duracion
NN fechaInicio
NN fechaFinal
NN fechaPublicacion
NN cantidadDisponible
NN direccion
NN tipoDeOferta
FK idOperador
NN cantidadInicial
NN estadoOferta
OfertaHotel
FK idOferta
NN categoria
NN capacidadDelServicio
NN ubicacion
NN tamaño
OfertaHabitacion
FK idOferta
NN tipoDeBaño
NN tipodeHabitacion
OfertaViviedaUniversitaria
FK idOferta
NN tipoHabitacion
NN capacidadHabitacion
NN ubicacionHabitacion
NN menajeHabitacion
OfertaViviedaEsporadica
FK idOferta
NN tipoHabitacion
NN capacidadPersonas
NN ubicacionHabitacion
NN menajeHabitacion
Reserva
PK idReserva
FK idOferta
FK idPersonaHabilitada
NN fechaInicioReserva
NN fechaFinalReserva
NN estadoReserva
FK idReservaColeciva
NN cantidad
Servicio
FK idOferta
NN nombreServicio
NN estaIncluido
NN valor
SeguroDeArrendamiento
PK idSeguro
FK idOferta
FK idPersonaHabilitada
NN empresaAseguradora
NN valor
NN descripcion
ReservaColectiva
PK idReservaColectiva
FK idOferta
NN fechaInicioReserva
NN fechaFinalReserva
NN estadoReserva
NN tipoDeAlojamiento
NN cantidad

Se actualizó la clase reserva, a esta se añadió los atributos fechaInicioReserva, fechaFinalReserva y cantidad, del mismo modo a la clase reserva colectiva se le añadieron los atributos fechaInicioReserva y fechaFinalReserva.

3. Diseño de la aplicación (64 %)

3.1. Análisis de impacto (1%)

A partir del diseño existente, analice el impacto que representa la introducción de los nuevos requerimientos y restricciones a nivel del modelo conceptual. Sea claro en mencionar explícitamente los cambios relevantes entre su diseño entregado en iteraciones anteriores y el actual.

Para el RFC10 fue necesario añadir a la clase reserva los atributos fecha inicio y fecha final, ya que anteriormente solo existía una fecha llamada fecha reserva. Todo lo demás requerido para este requerimiento ya existía anteriormente. Para el requerimiento número once fue necesario añadir en atributo cantidad en la clase oferta ya que anteriormente no existía y resulta necesario para conocer los usuarios que no realizaron al menos una reserva.

Para el RFC11, RFC12 y RFC13 no fue necesario añadir nada mas ya que con los atributos anteriores existentes fue suficiente.

3.2. Diseño físico. (63 %)

Analice la aplicación completa resultante de la iteración anterior y de los nuevos requerimientos para realizar el diseño físico correspondiente. En particular, diseñe los índices necesarios para el adecuado rendimiento global de la aplicación.

3.2.1. Documentación del diseño físico. (19 %)

Justifique la selección de índices desde el punto de vista de cada uno de los requerimientos funcionales. Indique claramente cuál es el tipo de índice utilizado (B+, Hash, …, primario, secundario)

Para el RFC10 Y RFC11, se decidió utilizar un índice B+ sobre el atributo que modela el inicio de una reserva, esto se debe a que se hace una selección a partir de un rango de fechas.

	SEGMENT_NAME	Size in MB
1	FECHAINICIORESERVA_INDEX	32
2	RESERVA	88

De igual forma, para el RFC11, se decidió implementar un índice de tipo Hash sobre el atributo que modela el tipo de oferta, en la tabla oferta, debido a que se usa para la comparación con valores muy precisos.

	SEGMENT_NAME	Size in MB
1	OFERTA	12
2	TIPODEOFERTA_INDEX	0,1875

Para el RFC12, no se define ningún índice, esto se debe a que ya hay una definido por ORACLE, un índice primario sobre el id de la oferta.

Finalmente, para el RFC13, se define un índice sobre el atributo que modela el valor de una oferta en la tabla oferta, este es de tipo B+, debido a que se hace una búsqueda por rango en el mismo.

	SEGMENT_NAME	Size in MB
1	OFERTA	12

- Según su modelo de datos, para los índices creados de forma automática por Oracle
  - Incluya una foto de pantalla con la información generada por Oracle asociada a los índices existentes.

    Tabla Reserva

	OWNER	INDEX_NAME	INDEX_TYPE
1	ISIS2304A721810	FECHAINICIORESERVA_INDEX	NORMAL
2	ISIS2304A721810	RESERVA_PK1	NORMAL

    Tabla Oferta

	OWNER	INDEX_NAME	INDEX_TYPE	T
1	ISIS2304A721810	OFERTA_PK	NORMAL	IS
2	ISIS2304A721810	TIPODEOFERTA_INDEX	BITMAP	IS
3	ISIS2304A721810	VALOR_OFERTA_INDEX	NORMAL	IS

    Tabla persona habilitada

	OWNER	INDEX_NAME	INDEX_TYPE
1	ISIS2304A721810	PERSONAHABILITADA_PK	NORMAL

- Analice los índices encontrados. Específicamente, analice por qué fueron creados por Oracle y si ayudan al rendimiento de los requerimientos funcionales

  Como se puede ver, estos índices son primarios y son sobre las claves primarias, por lo que estos índices son útiles cuando se hacen consultas relacionadas con las PK's o se hacen hash con las FK's.

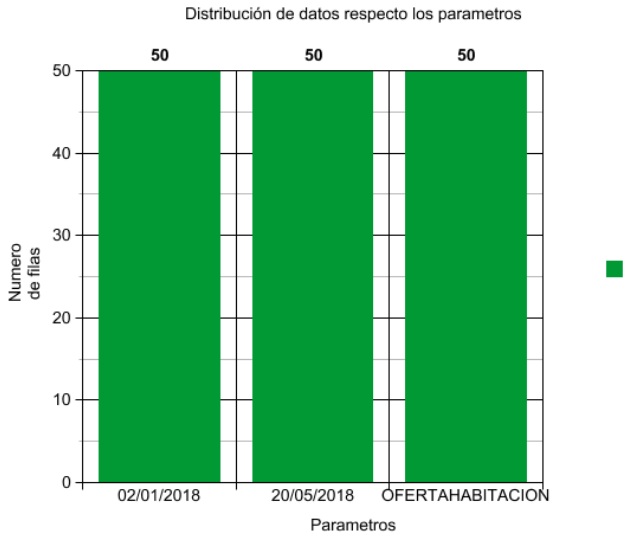
❖ **(44%)** Documente plenamente el análisis realizado, incluyendo los siguientes aspectos para cada requerimiento funcional de consulta solicitado

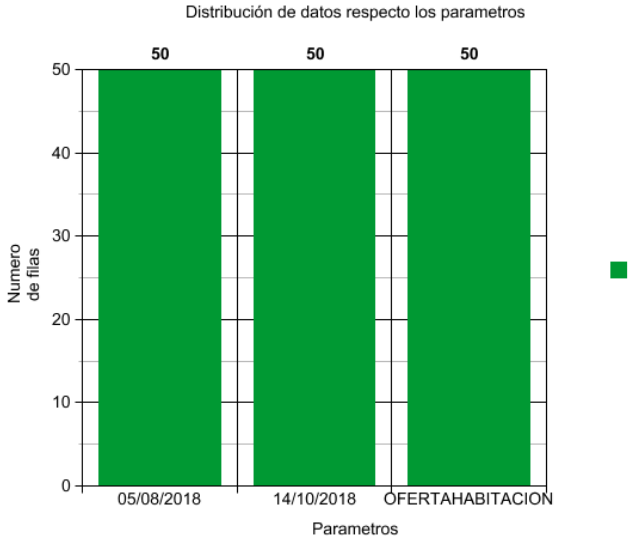
- Documentación del escenario de pruebas
- RCF10
  - Sentencias SQL que responden el requerimiento y que fueron analizadas.

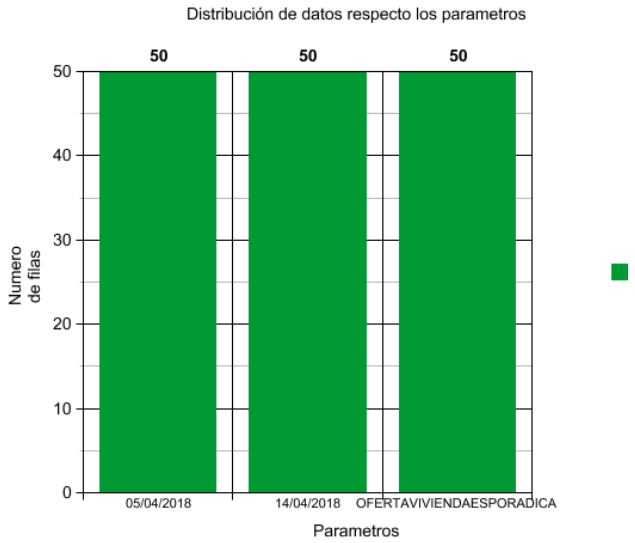
```
// obtiene los datos de las personas que han hecho reservas en el rango de fechas
        indicado y del tipo de oferta definido y ordenado por cualquier tipo de dato del
        cliente (en la ultima linea)
SELECT DISTINCT PERSONAHABILITADA.IDPERSONAHABILITADA,
        PERSONAHABILITADA.NOMBRE, PERSONAHABILITADA.EMAIL,
        PERSONAHABILITADA.TELEFONO
FROM
((PERSONAHABILITADA INNER JOIN RESERVA ON
        PERSONAHABILITADA.IDPERSONAHABILITADA =
        RESERVA.IDPERSONAHABILITADA)
        INNER JOIN OFERTA ON RESERVA.IDOFERTA = OFERTA.IDOFERTA )
WHERE OFERTA.TIPODEOFERTA = 'OFERTAHOTEL'
AND RESERVA.FECHARESERVA>='01/03/18' AND
        RESERVA.FECHARESERVA<='30/12/18'
ORDER BY PERSONAHABILITADA.NOMBRE;

// obtiene los datos de las personas que han hecho reservas en el rango de fechas
        indicado y agrupado por el tipo de oferta
SELECT DISTINCT OFERTA.TIPODEOFERTA,
        PERSONAHABILITADA.IDPERSONAHABILITADA,
        PERSONAHABILITADA.NOMBRE,
PERSONAHABILITADA.EMAIL, PERSONAHABILITADA.TELEFONO
FROM
((PERSONAHABILITADA INNER JOIN RESERVA ON
        PERSONAHABILITADA.IDPERSONAHABILITADA =
RESERVA.IDPERSONAHABILITADA)
        INNER JOIN OFERTA ON RESERVA.IDOFERTA = OFERTA.IDOFERTA )
WHERE RESERVA.FECHARESERVA>='01/03/18' AND
        RESERVA.FECHARESERVA<='30/12/18'
GROUP BY
        OFERTA.TIPODEOFERTA,PERSONAHABILITADA.IDPERSONAHABILITA
        DA, PERSONAHABILITADA.NOMBRE,
PERSONAHABILITADA.EMAIL, PERSONAHABILITADA.TELEFONO
ORDER BY OFERTA.TIPODEOFERTA;
```

  - Distribución de los datos con respecto a los parámetros de entrada utilizados en el requerimiento funcional. En particular se quiere un análisis de distribución que permita ver cómo puede cambiar el tamaño de la respuesta según el valor de los parámetros utilizados y la configuración de los datos de prueba.

- Valores de los parámetros utilizados en el análisis y que constituyen diferenciadores en los planes de ejecución obtenidos.







- Planes de consulta obtenidos en Oracle para la ejecución del requerimiento. Para ello, documente con una foto de pantalla los planes de consulta obtenidos en SQLDeveloper.

PLAN_TABLE_OUTPUT

```
Plan hash value: 1071258402

-----------------------------------------------------------------------------------------------------
| Id  | Operation              | Name               | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT       |                    |  424K|   37M|       | 29675   (1)| 00:00:02 |
|   1 |  SORT ORDER BY         |                    |  424K|   37M|   72M| 29675   (1)| 00:00:02 |
|   2 |   HASH GROUP BY        |                    |  424K|   37M|   72M| 29675   (1)| 00:00:02 |
|*  3 |    HASH JOIN           |                    |  693K|   61M| 5184K|  7499   (1)| 00:00:01 |
|   4 |     TABLE ACCESS FULL  | PERSONAHABILITADA  |  100K| 4004K|       |   171   (1)| 00:00:01 |
|*  5 |     HASH JOIN          |                    |  698K|   34M| 3504K|  4959   (1)| 00:00:01 |
|   6 |      TABLE ACCESS FULL| OFERTA             | 99656 | 2335K|       |   410   (1)| 00:00:01 |
|*  7 |      TABLE ACCESS FULL| RESERVA            |  704K|   18M|       |  3043   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   3 - access("PERSONAHABILITADA"."IDPERSONAHABILITADA"="RESERVA"."IDPERSONAHABILITADA")
   5 - access("RESERVA"."IDOFERTA"="OFERTA"."IDOFERTA")
   7 - filter("RESERVA"."FECHAINICIORESERVA">=TO_TIMESTAMP('01/03/18') AND
              "RESERVA"."FECHAFINALRESERVA"<=TO_TIMESTAMP('30/12/18'))

Note
-----
   - dynamic statistics used: dynamic sampling (level=2)
   - 2 Sql Plan Directives used for this statement
```

- Tiempos obtenidos con la ejecución de cada uno de los planes. Estos tiempos son medidos desde el núcleo de la aplicación, es decir, no incluyen la parte de interacción con el usuario, ingreso de datos ni despliegue de resultados.

Hoja de Trabajo | Generador de Consultas

```
1 SELECT DISTINCT PERSONAHABILITADA.IDPERSONAHABILITADA, PERSONAHABILITADA.NOMBRE, PERSONAHABILITADA.EMAIL, PERSONAHABILI
2 FROM
3 ((PERSONAHABILITADA INNER JOIN RESERVA ON PERSONAHABILITADA.IDPERSONAHABILITADA = RESERVA.IDPERSONAHABILITADA)
4     INNER JOIN OFERTA ON RESERVA.IDOFERTA = OFERTA.IDOFERTA )
5 WHERE OFERTA.TIPODEOFERTA = 'OFERTAHOTEL'
6 AND RESERVA.FECHAINICIORESERVA>='02/01/2018' AND RESERVA.FECHAINICIORESERVA<='03/01/2018'
7 ORDER BY PERSONAHABILITADA.NOMBRE;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,224 segundos

	IDPERSONAHABILITADA	NOMBRE	EMAIL	TELEFONO
1	89472	Aaron Austin	sasuh@zi.ug	34768
2	65024	Aaron Bailey	aj@dep.vi	93946
3	28216	Aaron Bennett	tessajim@mu.cc	94235
4	22299	Aaron Berry	koh@sir.sk	33370
5	40382	Aaron Bowen	liddiire@suzo.cl	51461
6	49712	Aaron Brewer	zod@lor.mx	81556
7	32557	Aaron Brewer	evozu@piha.bd	94739
8	59498	Aaron Caldwell	mot@lec.tz	35504
9	44572	Aaron Carlson	keh@zu.ru	11462
10	30566	Aaron Carroll	neken@wutavoac.tn	43116
11	85249	Aaron Chambers	ad@cib.ir	1710
12	10154	Aaron Chambers	boem@eketip.de	1420
13	54053	Aaron Collier	miw@te.sk	26635

Consulta del RFC 10 gráfica 1

Hoja de Trabajo | Generador de Consultas

```
1 SELECT DISTINCT PERSONAHABILITADA.IDPERSONAHABILITADA, PERSONAHABILITADA.NOMBRE, PERSONAHABILITADA.EMAIL, PERSONAHABILI
2 FROM
3 ((PERSONAHABILITADA INNER JOIN RESERVA ON PERSONAHABILITADA.IDPERSONAHABILITADA = RESERVA.IDPERSONAHABILITADA)
4     INNER JOIN OFERTA ON RESERVA.IDOFERTA = OFERTA.IDOFERTA )
5 WHERE OFERTA.TIPODEOFERTA = 'OFERTAHABITACION'
6 AND RESERVA.FECHAINICIORESERVA>='05/08/2018' AND RESERVA.FECHAINICIORESERVA<='14/10/2018'
7 ORDER BY PERSONAHABILITADA.NOMBRE;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,174 segundos

	IDPERSONAHABILITADA	NOMBRE	EMAIL	TELEFONO
1	28216	Aaron Bennett	tessajim@mu.cc	94235
2	36723	Aaron Bishop	pe@difwag.cd	58126
3	40382	Aaron Bowen	liddiire@suzo.cl	51461
4	33928	Aaron Caldwell	bojished@je.do	77375
5	75715	Aaron Carson	zofli@co.md	19856
6	15293	Aaron Christensen	nedilpu@pinuklev.np	61137
7	54053	Aaron Collier	miw@te.sk	26635
8	15948	Aaron Crawford	zinol@zues.ae	37784
9	65219	Aaron Dunn	utarog@ipe.co	84680
10	45472	Aaron Hines	mi@ihje.bf	51789
11	73582	Aaron Hodges	nucfisot@jikhahra.mr	78496
12	87803	Aaron Larson	gocujehu@rikneno.bd	37328
13	39328	Aaron Luna	af@iswu.gm	54177

tecla Control pulsada para ejecutar "Ir a Declaración" | Línea 7 Columna 35 | Insertar | Modificado | Windows: CR

Consulta del RFC 10 gráfica 2

Hoja de Trabajo | Generador de Consultas

```
1 SELECT DISTINCT PERSONAHABILITADA.IDPERSONAHABILITADA, PERSONAHABILITADA.NOMBRE, PERSONAHABILITADA.EMAIL, PERSONAHABILI
2 FROM
3 ((PERSONAHABILITADA INNER JOIN RESERVA ON PERSONAHABILITADA.IDPERSONAHABILITADA = RESERVA.IDPERSONAHABILITADA)
4     INNER JOIN OFERTA ON RESERVA.IDOFERTA = OFERTA.IDOFERTA )
5 WHERE OFERTA.TIPODEOFERTA = 'OFERTAVIVIENDAESPORADICA'
6 AND RESERVA.FECHAINICIORESERVA>='05/04/2018' AND RESERVA.FECHAINICIORESERVA<='14/04/2018'
7 ORDER BY PERSONAHABILITADA.NOMBRE;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,229 segundos

	IDPERSONAHABILITADA	NOMBRE	EMAIL	TELEFONO
1	67220	Aaron Brown	bipos@ten.tl	93567
2	59498	Aaron Caldwell	mot@lec.tz	35504
3	85733	Aaron Carpenter	faho@jaz.bq	24122
4	83774	Aaron Castro	vevhu@fuvige.ma	902
5	42409	Aaron Douglas	giem@wuvup.ke	96610
6	80992	Aaron Harris	elarew@jozba.my	8240
7	30079	Aaron Harrison	suwoggoh@ala.np	44351
8	47506	Aaron Haynes	cuzen@ragcep.tg	19892
9	13957	Aaron Luna	kowca@musa.bj	94471
10	39328	Aaron Luna	af@iswu.gm	54177
11	15836	Aaron Mitchell	soewedos@pulebju.il	68984
12	79523	Aaron Murphy	buzmaej@gam.cy	95341
13	36750	Aaron Ramsey	ahcop@haowu.gl	58907

tecla Control pulsada para ejecutar "Ir a Declaración" | Línea 7 Columna 35 | Insertar | Modificado | Windows: CR

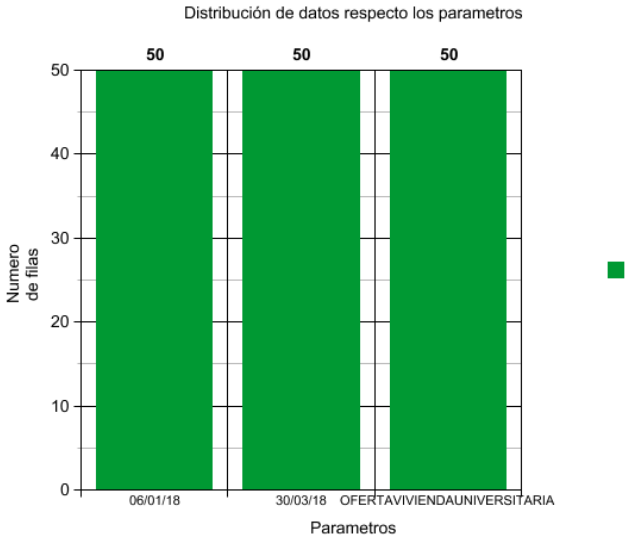
Consulta del RFC 10 gráfica 3

- RCF11
  - Sentencias SQL que responden el requerimiento y que fueron analizadas.

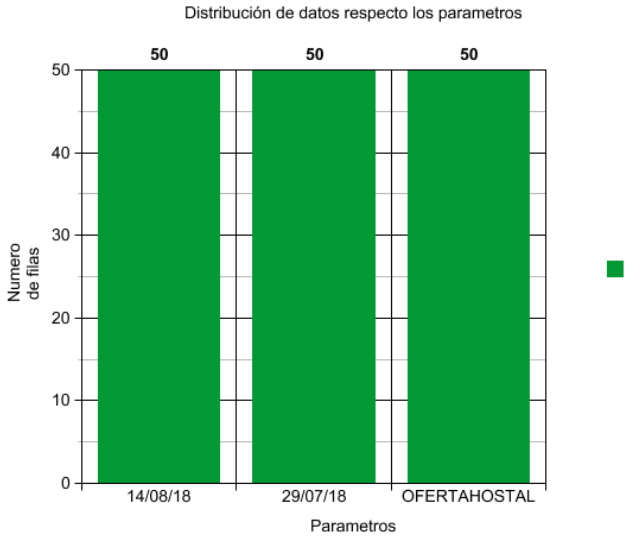
// obtiene los datos de las personas que NO han hecho reservas en el rango de fechas indicado y del tipo de oferta definido y ordenado por cualquier tipo de dato del cliente (en la ultima linea)

```
SELECT PERSONAHABILITADA.IDPERSONAHABILITADA,
        PERSONAHABILITADA.NOMBRE,
PERSONAHABILITADA.EMAIL, PERSONAHABILITADA.TELEFONO
FROM (PERSONAHABILITADA LEFT JOIN
(SELECT DISTINCT RESERVA.IDPERSONAHABILITADA ID_NO_MOSTRAR
        FROM
        RESERVA INNER JOIN OFERTA ON RESERVA.IDOFERTA =
        OFERTA.IDOFERTA
        WHERE OFERTA.TIPODEOFERTA = 'OFERTAHOTEL'
AND RESERVA.FECHARESERVA>='06/01/18' AND
        RESERVA.FECHARESERVA<='30/12/18') TABLAMALA

ON PERSONAHABILITADA.IDPERSONAHABILITADA =
        TABLAMALA.ID_NO_MOSTRAR)
WHERE TABLAMALA.ID_NO_MOSTRAR IS NULL
ORDER BY PERSONAHABILITADA.NOMBRE;
```

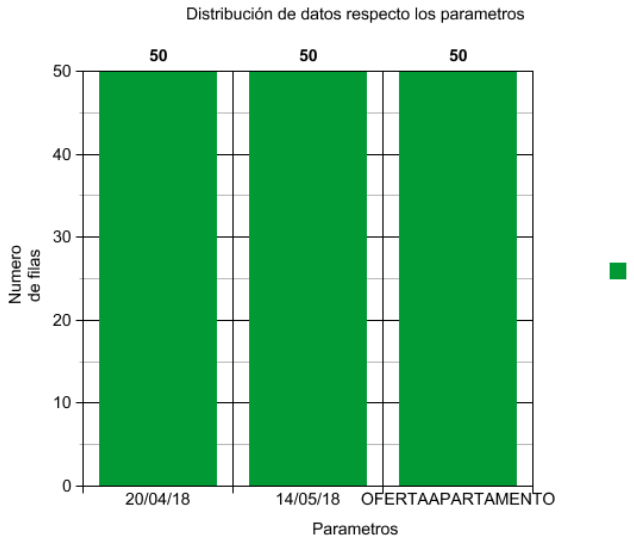
- Distribución de los datos con respecto a los parámetros de entrada utilizados en el requerimiento funcional. En particular se quiere un análisis de distribución que permita ver cómo puede cambiar el tamaño de la respuesta según el valor de los parámetros utilizados y la configuración de los datos de prueba.
- Valores de los parámetros utilizados en el análisis y que constituyen diferenciadores en los planes de ejecución obtenidos.



RFC 11 gráfica 1



RFC 11 gráfica 2



RFC 11 gráfica 3

- Planes de consulta obtenidos en Oracle para la ejecución del requerimiento. Para ello, documente con una foto de pantalla los planes de consulta obtenidos en SQLDevelopper.

```
PLAN_TABLE_OUTPUT
Plan hash value: 3870756435

-------------------------------------------------------------------------------------------------
| Id  | Operation                | Name               | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT         |                    | 36974 |  1660K|       |  5018   (1)| 00:00:01 |
|   1 |  SORT ORDER BY           |                    | 36974 |  1660K|  1896K|  5018   (1)| 00:00:01 |
|*  2 |   HASH JOIN RIGHT ANTI   |                    | 36974 |  1660K|       |  4587   (1)| 00:00:01 |
|   3 |    VIEW                  |                    | 63039 |   307K|       |  4416   (1)| 00:00:01 |
|   4 |     HASH UNIQUE          |                    | 63038 |  2524K|  7208K|  4416   (1)| 00:00:01 |
|*  5 |      HASH JOIN           |                    |   141K|  5657K|       |  3453   (1)| 00:00:01 |
|*  6 |       TABLE ACCESS FULL| OFERTA             | 15957 |   373K|       |   410   (1)| 00:00:01 |
|*  7 |       TABLE ACCESS FULL| RESERVA             |   889K|    14M|       |  3040   (1)| 00:00:01 |
|   8 |    TABLE ACCESS FULL   | PERSONAHABILITADA  |   100K|  4004K|       |   171   (1)| 00:00:01 |
-------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("PERSONAHABILITADA"."IDPERSONAHABILITADA"="TABLAMALA"."ID_NO_MOSTRAR")
   5 - access("RESERVA"."IDOFERTA"="OFERTA"."IDOFERTA")
   6 - filter("OFERTA"."TIPODEOFERTA"='OFERTAHOTEL')
   7 - filter("RESERVA"."FECHAINICIORESERVA">=TO_TIMESTAMP('06/01/18') AND
              "RESERVA"."FECHAINICIORESERVA"<=TO_TIMESTAMP('30/12/18'))

Note
-----
   - dynamic statistics used: dynamic sampling (level=2)
   - 4 Sql Plan Directives used for this statement
```

- Tiempos obtenidos con la ejecución de cada uno de los planes. Estos tiempos son medidos desde el núcleo de la aplicación, es decir, no incluyen la parte de interacción con el usuario, ingreso de datos ni despliegue de resultados.

Hoja de Trabajo | Generador de Consultas

```
SELECT PERSONAHABILITADA.IDPERSONAHABILITADA, PERSONAHABILITADA.NOMBRE,
PERSONAHABILITADA.EMAIL, PERSONAHABILITADA.TELEFONO
FROM (PERSONAHABILITADA LEFT JOIN
(SELECT DISTINCT RESERVA.IDPERSONAHABILITADA ID_NO_MOSTRAR
    FROM
        RESERVA INNER JOIN OFERTA ON RESERVA.IDOFERTA = OFERTA.IDOFERTA
        WHERE OFERTA.TIPODEOFERTA = 'OFERTAVIVIENDAUNIVERSITARIA'
AND RESERVA.FECHAINICIORESERVA>='06/01/18' AND RESERVA.FECHAINICIORESERVA<='30/03/18') TABLAMALA

ON PERSONAHABILITADA.IDPERSONAHABILITADA = TABLAMALA.ID_NO_MOSTRAR)
WHERE TABLAMALA.ID_NO_MOSTRAR IS NULL
ORDER BY PERSONAHABILITADA.NOMBRE;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,492 segundos

	IDPERSONAHABILITADA	NOMBRE	EMAIL	TELEFONO
1	65024	Aaron Bailey	aj@dep.vi	93946
2	47120	Aaron Barber	fiadutu@noke.re	29827
3	33539	Aaron Barber	tod@uk.by	40968
4	67270	Aaron Barnes	lu@nodsezu.ad	88319
5	99435	Aaron Beck	foz@hepekda.ki	51460
6	28216	Aaron Bennett	tessajim@mu.cc	94235
7	22376	Aaron Bennett	fed@tawajuba.hk	87402
8	22299	Aaron Berry	koh@sir.sk	33370
9	36723	Aaron Bishop	pe@difwag.cd	58126
10	33435	Aaron Boone	ta@ba.au	72105
11	40382	Aaron Bowen	liddiire@suzo.cl	51461

tecla Control pulsada para ejecutar "Ir a Declaración" | Línea 12 Columna 35 | Insertar | Modificado | Windows: C

Consulta del RFC 11 gráfica 1

Hoja de Trabajo | Generador de Consultas

```
SELECT PERSONAHABILITADA.IDPERSONAHABILITADA, PERSONAHABILITADA.NOMBRE,
PERSONAHABILITADA.EMAIL, PERSONAHABILITADA.TELEFONO
FROM (PERSONAHABILITADA LEFT JOIN
(SELECT DISTINCT RESERVA.IDPERSONAHABILITADA ID_NO_MOSTRAR
    FROM
        RESERVA INNER JOIN OFERTA ON RESERVA.IDOFERTA = OFERTA.IDOFERTA
        WHERE OFERTA.TIPODEOFERTA = 'OFERTAAPARTAMENTO'
AND RESERVA.FECHAINICIORESERVA>='20/04/18' AND RESERVA.FECHAINICIORESERVA<='14/05/18') TABLAMALA

ON PERSONAHABILITADA.IDPERSONAHABILITADA = TABLAMALA.ID_NO_MOSTRAR)
WHERE TABLAMALA.ID_NO_MOSTRAR IS NULL
ORDER BY PERSONAHABILITADA.EMAIL;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,4 segundos

	IDPERSONAHABILITADA	NOMBRE	EMAIL	TELEFONO
1	18023	Jeremy Garrett	ababutod@cirzevup.ba	17930
2	55201	Rena Franklin	abacoz@tomkiz.tt	13661
3	71258	Jon Santos	abaehona@gifuvob.co.uk	36964
4	77208	Chris Rios	aba@es.py	81401
5	92219	Gabriel Knight	aba@fo.ru	55860
6	60194	Julian Kelly	aba@go.be	4170
7	32860	Jordan Harvey	ababa@hauhati.cv	47748
8	85314	Lucinda Bishop	abajeg@udo.nu	97228
9	58896	Ruth Jennings	abak@maditca.ai	72282
10	49742	Roy Soto	abalbo@guhzihno.kp	47706
11	67110	Violet Lloyd	aba@ma.cg	90699

Consulta del RFC 11 gráfica 2

Hoja de Trabajo | Generador de Consultas

```
SELECT PERSONAHABILITADA.IDPERSONAHABILITADA, PERSONAHABILITADA.NOMBRE,
PERSONAHABILITADA.EMAIL, PERSONAHABILITADA.TELEFONO
FROM (PERSONAHABILITADA LEFT JOIN
(SELECT DISTINCT RESERVA.IDPERSONAHABILITADA ID_NO_MOSTRAR
    FROM
        RESERVA INNER JOIN OFERTA ON RESERVA.IDOFERTA = OFERTA.IDOFERTA
        WHERE OFERTA.TIPODEOFERTA = 'OFERTAHOSTAL'
AND RESERVA.FECHAINICIORESERVA>='29/07/18' AND RESERVA.FECHAINICIORESERVA<='14/08/18') TABLAMALA

ON PERSONAHABILITADA.IDPERSONAHABILITADA = TABLAMALA.ID_NO_MOSTRAR)
WHERE TABLAMALA.ID_NO_MOSTRAR IS NULL
ORDER BY PERSONAHABILITADA.IDPERSONAHABILITADA;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,122 segundos

	IDPERSONAHABILITADA	NOMBRE	EMAIL	TELEFONO
1	1	Dollie Herrera	izhir@nerarad.ne	26398
2	2	Brent Rodriguez	hiko@luwelu.re	56535
3	3	Lulu Potter	vofihop@todunmap.tz	17097
4	4	Lucile Neal	tot@puc.vc	50533
5	5	Ricky Lloyd	zojibez@haehape.az	66370
6	6	Jose James	bo@fadanfo.li	59854
7	7	Stella Murphy	fof@hiczo.bj	94787
8	8	Minerva Johnston	nagmaz@ikare.cg	83628
9	9	Maggie Rivera	ibigim@itosoeba.bf	38526
10	10	Susie Obrien	ovu@okob.pg	29931
11	11	Rose Adams	edetij@amalur.mu	60547

tecla Control pulsada para ejecutar "Ir a Declaración" | Línea 12 Columna 48 | Insertar | Modificado | Windows: CI

Consulta del RFC 11 gráfica 3

- RCF12
  - Sentencias SQL que responden el requerimiento y que fueron analizadas.

```
//para cada semana del año da las ofertas con MAYOR ocupacion
SELECT  NRO_SEMANA, OFERTA.IDOFERTA, OFERTA.VALOR,
        OFERTA.DIRECCION, OFERTA.TIPODEOFERTA,
        OFERTA.IDOPERADOR, CANT AS CANTIDAD
        FROM( OFERTA INNER JOIN (
        SELECT SEMANA_INICIO AS NRO_SEMANA, OFER AS OFERTA_ID, CANT
        FROM(
        SELECT SEMANA_INICIO, OFER,ROW_NUMBER() OVER(PARTITION BY
        SEMANA_INICIO ORDER BY CANT DESC) AS POSICION, CANT
        FROM(
                SELECT TO_NUMBER(TO_CHAR(TO_DATE(
        CAST(RESERVA.FECHAINICIORESERVA AS DATE), 'DD/MM/YY'),'WW'))
        AS SEMANA_INICIO,
                RESERVA.IDOFERTA AS OFER,
        RESERVA.CANTIDAD AS CANT
                FROM RESERVA
                ORDER BY SEMANA_INICIO
        )
        )
        WHERE POSICION = 1)
        ON OFERTA.IDOFERTA= OFERTA_ID)
```

```
ORDER BY NRO_SEMANA;

//para cada semana del año da las ofertas con MENOR ocupacion
SELECT  NRO_SEMANA, OFERTA.IDOFERTA, OFERTA.VALOR,
        OFERTA.DIRECCION, OFERTA.TIPODEOFERTA,
        OFERTA.IDOPERADOR, CANT AS CANTIDAD
        FROM( OFERTA INNER JOIN (
        SELECT SEMANA_INICIO AS NRO_SEMANA, OFER AS OFERTA_ID, CANT
        FROM(
        SELECT SEMANA_INICIO, OFER,ROW_NUMBER() OVER(PARTITION BY
        SEMANA_INICIO ORDER BY CANT ASC) AS POSICION, CANT
        FROM(
                                SELECT TO_NUMBER(TO_CHAR(TO_DATE(
        CAST(RESERVA.FECHAINICIORESERVA AS DATE), 'DD/MM/YY'),'WW'))
        AS SEMANA_INICIO,
                                RESERVA.IDOFERTA AS OFER,
        RESERVA.CANTIDAD AS CANT
                                FROM RESERVA
                                ORDER BY SEMANA_INICIO
        )
        )
        WHERE POSICION = 1)
        ON OFERTA.IDOFERTA= OFERTA_ID)
ORDER BY NRO_SEMANA;
)

//para cada semana del año da los operadores MAS solicitados
SELECT NRO_SEMANA, OPERADOR.IDOPERADOR, OPERADOR.NOMBRE,
        OPERADOR.DIRECCION, OPERADOR.TELEFONO,
        OPERADOR.TIPODEOPERADOR, CANTIDAD
        FROM (OPERADOR INNER JOIN (
        SELECT  NRO_SEMANA, OFERTA.IDOFERTA, OFERTA.IDOPERADOR AS
        OPERADOR_ID, CANT AS CANTIDAD
        FROM( OFERTA INNER JOIN (
        SELECT SEMANA_INICIO AS NRO_SEMANA, OFER AS OFERTA_ID, CANT
        FROM(
                                SELECT SEMANA_INICIO, OFER,ROW_NUMBER()
        OVER(PARTITION BY SEMANA_INICIO ORDER BY CANT DESC) AS
        POSICION, CANT
                                FROM(
                                SELECT TO_NUMBER(TO_CHAR(TO_DATE(
        CAST(RESERVA.FECHAINICIORESERVA AS DATE), 'DD/MM/YY'),'WW'))
        AS SEMANA_INICIO,
```

```
                                RESERVA.IDOFERTA AS OFER,
        RESERVA.CANTIDAD AS CANT
                                FROM RESERVA
                                ORDER BY SEMANA_INICIO
                                )
        )
        WHERE POSICION = 1)
        ON OFERTA.IDOFERTA= OFERTA_ID)
        ) ON OPERADOR.IDOPERADOR = OPERADOR_ID)
ORDER BY NRO_SEMANA;


//para cada semana del año da los operadores MENOS solicitados
SELECT NRO_SEMANA, OPERADOR.IDOPERADOR, OPERADOR.NOMBRE,
        OPERADOR.DIRECCION, OPERADOR.TELEFONO,
        OPERADOR.TIPODEOPERADOR, CANTIDAD
        FROM (OPERADOR INNER JOIN (
        SELECT  NRO_SEMANA, OFERTA.IDOFERTA, OFERTA.IDOPERADOR AS
        OPERADOR_ID, CANT AS CANTIDAD
        FROM( OFERTA INNER JOIN (
        SELECT SEMANA_INICIO AS NRO_SEMANA, OFER AS OFERTA_ID, CANT
        FROM(
                                SELECT SEMANA_INICIO, OFER,ROW_NUMBER()
        OVER(PARTITION BY SEMANA_INICIO ORDER BY CANT ASC) AS
        POSICION, CANT
                                FROM(
                                SELECT TO_NUMBER(TO_CHAR(TO_DATE(
        CAST(RESERVA.FECHAINICIORESERVA AS DATE), 'DD/MM/YY'),'WW'))
        AS SEMANA_INICIO,
                                RESERVA.IDOFERTA AS OFER,
        RESERVA.CANTIDAD AS CANT
                                FROM RESERVA
                                ORDER BY SEMANA_INICIO
                                )
        )
        WHERE POSICION = 1)
        ON OFERTA.IDOFERTA= OFERTA_ID)
        ) ON OPERADOR.IDOPERADOR = OPERADOR_ID)
ORDER BY NRO_SEMANA;
```

- Distribución de los datos con respecto a los parámetros de entrada utilizados en el requerimiento funcional. En particular se quiere un análisis de distribución que permita

ver cómo puede cambiar el tamaño de la respuesta según el valor de los parámetros utilizados y la configuración de los datos de prueba.

Estos son siempre 52 ya que son 52 las semanas que tiene el año 2018 y en cada semana ha habido mínimo una reserva y esta consulta no depende de filtros sino de los datos existentes en la Base de Datos.

- Valores de los parámetros utilizados en el análisis y que constituyen diferenciadores en los planes de ejecución obtenidos.

  No hay ningún valor debido a que este requerimiento funcional de consulta no requiere de parámetros

- Planes de consulta obtenidos en Oracle para la ejecución del requerimiento. Para ello, documente con una foto de pantalla los planes de consulta obtenidos en SQLDevelopper.

```
PLAN_TABLE_OUTPUT
Plan hash value: 2699422374

-----------------------------------------------------------------------------------------------------
| Id  | Operation                      | Name       | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |            |     1 |    95 |       | 25674   (1)| 00:00:02 |
|   1 |  NESTED LOOPS                  |            |     1 |    95 |       | 25674   (1)| 00:00:02 |
|   2 |   NESTED LOOPS                 |            |     1 |    95 |       | 25674   (1)| 00:00:02 |
|*  3 |    VIEW                        |            |     1 |    44 |       | 25673   (1)| 00:00:02 |
|*  4 |     WINDOW SORT PUSHED RANK    |            |  1494K|    55M|    74M| 25673   (1)| 00:00:02 |
|   5 |      VIEW                      |            |  1494K|    55M|       | 10683   (1)| 00:00:01 |
|   6 |       SORT ORDER BY            |            |  1494K|    21M|    40M| 10683   (1)| 00:00:01 |
|   7 |        TABLE ACCESS FULL       | RESERVA    |  1494K|    21M|       |  3026   (1)| 00:00:01 |
|*  8 |    INDEX UNIQUE SCAN           | OFERTA_PK  |     1 |       |       |     0   (0)| 00:00:01 |
|   9 |   TABLE ACCESS BY INDEX ROWID| OFERTA     |     1 |    51 |       |     1   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   3 - filter("POSICION"=1)
   4 - filter(ROW_NUMBER() OVER ( PARTITION BY "SEMANA_INICIO" ORDER BY
              INTERNAL_FUNCTION("CANT") DESC )<=1)
   8 - access("OFERTA"."IDOFERTA"="OFER")
```

- Tiempos obtenidos con la ejecución de cada uno de los planes. Estos tiempos son medidos desde el núcleo de la aplicación, es decir, no incluyen la parte de interacción con el usuario, ingreso de datos ni despliegue de resultados.

Hoja de Trabajo | Generador de Consultas

```
SELECT NRO_SEMANA, OFERTA.IDOFERTA, OFERTA.VALOR, OFERTA.DIRECCION, OFERTA.TIPODEOFERTA, OFERTA.IDOPERADOR, CANT AS C
    FROM( OFERTA INNER JOIN (
        SELECT SEMANA_INICIO AS NRO_SEMANA, OFER AS OFERTA_ID, CANT
        FROM(
                SELECT SEMANA_INICIO, OFER,ROW_NUMBER() OVER(PARTITION BY SEMANA_INICIO ORDER BY CANT DESC) AS POSICIC
                FROM(
                    SELECT TO_NUMBER(TO_CHAR(TO_DATE( CAST(RESERVA.FECHAINICIORESERVA AS DATE), 'DD/MM/YY'),'WW')) AS
                    RESERVA.IDOFERTA AS OFER, RESERVA.CANTIDAD AS CANT
                    FROM RESERVA
                    ORDER BY SEMANA_INICIO
                )
        )
    WHERE POSICION = 1)
    ON OFERTA.IDOFERTA= OFERTA_ID)
ORDER BY NRO_SEMANA;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 2,122 segundos

	NRO_SEMANA	IDOFERTA	VALOR	DIRECCION	TIPODEOFERTA	IDOPERADOR	CANTIDAD
1	1	96825	17007	1888 Zapzat Heights	OFERTAHABITACION	78682	1
2	2	20180003	50000	calle 73#24	OFERTAHOTEL	9876543200	6
3	3	20180003	50000	calle 73#24	OFERTAHOTEL	9876543200	5
4	5	88010	68921	108 Puhne Point	OFERTAHOTEL	48000	1
5	6	88796	21288	817 Foho Parkway	OFERTAHABITACION	15278	1
6	7	97159	3301	40 Lurelu Street	OFERTAHOTEL	48399	1
7	8	94881	60305	291 Welid Mill	OFERTAVIVIENDAUNIVERSITARIA	11744	1
8	9	20180009	180000	calle 16#36	OFERTAVIVIENDAESPORADICA	1234567804	3
9	10	20180010	140000	calle 84#62	OFERTAVIVIENDAESPORADICA	1234567805	4
10	11	466	76326	1175 Botma Point	OFERTAAPARTAMENTO	94841	1

Semanas con ofertas mayor ocupación

Hoja de Trabajo | Generador de Consultas

```
SELECT NRO_SEMANA, OFERTA.IDOFERTA, OFERTA.VALOR, OFERTA.DIRECCION, OFERTA.TIPODEOFERTA, OFERTA.IDOPERADOR, CANT AS C
    FROM( OFERTA INNER JOIN (
        SELECT SEMANA_INICIO AS NRO_SEMANA, OFER AS OFERTA_ID, CANT
        FROM(
                SELECT SEMANA_INICIO, OFER,ROW_NUMBER() OVER(PARTITION BY SEMANA_INICIO ORDER BY CANT ASC) AS POSICION
                FROM(
                    SELECT TO_NUMBER(TO_CHAR(TO_DATE( CAST(RESERVA.FECHAINICIORESERVA AS DATE), 'DD/MM/YY'),'WW')) AS
                    RESERVA.IDOFERTA AS OFER, RESERVA.CANTIDAD AS CANT
                    FROM RESERVA
                    ORDER BY SEMANA_INICIO
                )
        )
    WHERE POSICION = 1)
    ON OFERTA.IDOFERTA= OFERTA_ID)
ORDER BY NRO_SEMANA;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 2,136 segundos

	NRO_SEMANA	IDOFERTA	VALOR	DIRECCION	TIPODEOFERTA	IDOPERADOR	CANTIDAD
1	1	96825	17007	1888 Zapzat Heights	OFERTAHABITACION	78682	1
2	2	91653	85664	572 Acuker Junction	OFERTAHABITACION	49594	1
3	3	43921	1795	1052 Nofji Glen	OFERTAVIVIENDAUNIVERSITARIA	57477	1
4	5	88010	68921	108 Puhne Point	OFERTAHOTEL	48000	1
5	6	88796	21288	817 Foho Parkway	OFERTAHABITACION	15278	1
6	7	97159	3301	40 Lurelu Street	OFERTAHOTEL	48399	1
7	8	94881	60305	291 Welid Mill	OFERTAVIVIENDAUNIVERSITARIA	11744	1
8	9	97150	94003	399 Miin Mill	OFERTAVIVIENDAESPORADICA	49812	1
9	10	94880	65266	1177 Cezu Road	OFERTAHOTEL	45127	1
10	11	466	76326	1175 Botma Point	OFERTAAPARTAMENTO	94841	1

Semanas con ofertas menor ocupación

Hoja de Trabajo | Generador de Consultas

```
5            SELECT SEMANA_INICIO AS NRO_SEMANA, OFER AS OFERTA_ID, CANT
6            FROM(
7                    SELECT SEMANA_INICIO, OFER,ROW_NUMBER() OVER(PARTITION BY SEMANA_INICIO ORDER BY CANT DESC) AS POS
8                    FROM(
9                        SELECT TO_NUMBER(TO_CHAR(TO_DATE( CAST(RESERVA.FECHAINICIORESERVA AS DATE), 'DD/MM/YY'),'WW'))
10                       RESERVA.IDOFERTA AS OFER, RESERVA.CANTIDAD AS CANT
11                       FROM RESERVA
12                       ORDER BY SEMANA_INICIO
13                   )
14           )
15       WHERE POSICION = 1)
16       ON OFERTA.IDOFERTA= OFERTA_ID)
17       ) ON OPERADOR.IDOPERADOR = OPERADOR_ID)
18 ORDER BY NRO_SEMANA;
19
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 2,165 segundos

	NRO_SEMANA	IDOPERADOR	NOMBRE	DIRECCION	TELEFONO	TIPODEOPERADOR	CANTIDAD
1	1	78682	Ina Armstrong	1972 Jibe Circle	35279	EmpresaViviendaUniversitaria	1
2	2	9876543200	Hotel Alfa	calle 15#45	3330003	Hotel	6
3	3	9876543200	Hotel Alfa	calle 15#45	3330003	Hotel	5
4	5	48000	Raymond Lawson	81 Fivkeb Terrace	51287	ArrendadorHabitacion	1
5	6	15278	Allen Farmer	1874 Dova Ridge	44551	MiembroArrendadorApartamento	1
6	7	48399	Etta Barton	1309 Uwzi River	88398	MiembroArrendadorApartamento	1
7	8	11744	Emma Powell	879 Ofekec Road	80706	Hotel	1
8	9	1234567804	Rodrigo Cardoso Rodriguez	calle 21#01	3330009	ArrendadorEsporadico	3
9	10	1234567805	Jairo Emilio Bautista	calle 22#12	3330010	ArrendadorEsporadico	4
10	11	94841	Callie Bridges	1154 Mifoz Circle	23141	MiembroArrendadorApartamento	1

Semanas con operadores mayor demanda

Hoja de Trabajo | Generador de Consultas

```
4            FROM( OFERTA INNER JOIN (
5            SELECT SEMANA_INICIO AS NRO_SEMANA, OFER AS OFERTA_ID, CANT
6            FROM(
7                    SELECT SEMANA_INICIO, OFER,ROW_NUMBER() OVER(PARTITION BY SEMANA_INICIO ORDER BY CANT ASC) AS POSI
8                    FROM(
9                        SELECT TO_NUMBER(TO_CHAR(TO_DATE( CAST(RESERVA.FECHAINICIORESERVA AS DATE), 'DD/MM/YY'),'WW'))
10                       RESERVA.IDOFERTA AS OFER, RESERVA.CANTIDAD AS CANT
11                       FROM RESERVA
12                       ORDER BY SEMANA_INICIO
13                   )
14           )
15       WHERE POSICION = 1)
16       ON OFERTA.IDOFERTA= OFERTA_ID)
17       ) ON OPERADOR.IDOPERADOR = OPERADOR_ID)
18 ORDER BY NRO_SEMANA;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 2,217 segundos

	NRO_SEMANA	IDOPERADOR	NOMBRE	DIRECCION	TELEFONO	TIPODEOPERADOR	CANTIDAD
1	1	78682	Ina Armstrong	1972 Jibe Circle	35279	EmpresaViviendaUniversitaria	1
2	2	49594	Hulda Howell	861 Amizu Street	25743	ArrendadorHabitacion	1
3	3	57477	Isabel Hopkins	803 Catca Grove	36754	MiembroArrendadorApartamento	1
4	5	48000	Raymond Lawson	81 Fivkeb Terrace	51287	ArrendadorHabitacion	1
5	6	15278	Allen Farmer	1874 Dova Ridge	44551	MiembroArrendadorApartamento	1
6	7	48399	Etta Barton	1309 Uwzi River	88398	MiembroArrendadorApartamento	1
7	8	11744	Emma Powell	879 Ofekec Road	80706	Hotel	1
8	9	49812	Addie Hicks	1933 Zeeca Grove	17196	Hotel	1
9	10	45127	Lena Barnett	1521 Goidu Park	91013	EmpresaViviendaUniversitaria	1
10	11	94841	Callie Bridges	1154 Mifoz Circle	23141	MiembroArrendadorApartamento	1

Semanas con operadores menor demanda

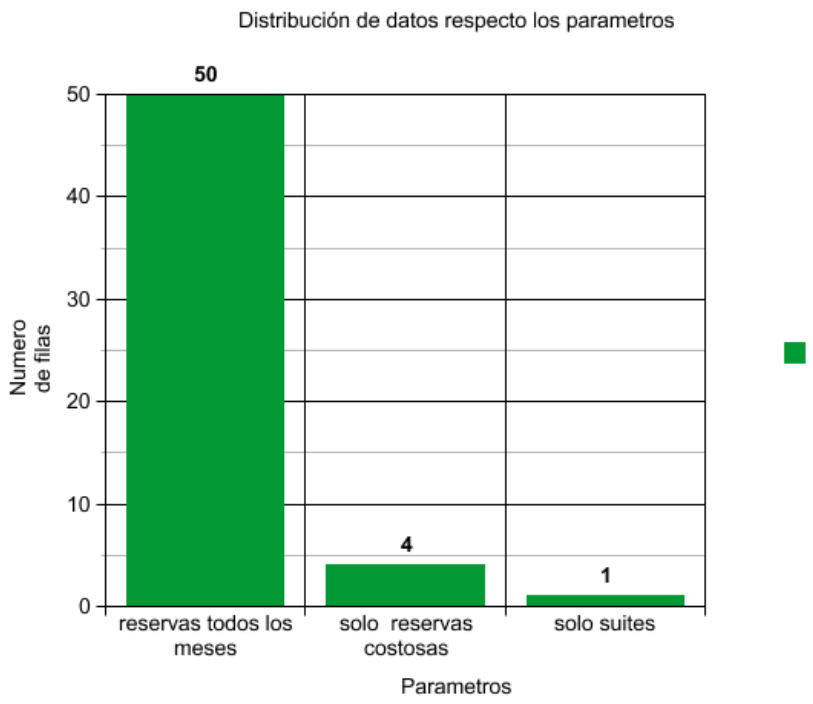
- RCF13
  - Sentencias SQL que responden el requerimiento y que fueron analizadas.

//obtiene la info de los usuario que han reservado en cada uno de los meses del año

```
SELECT PERSONAHABILITADA.IDPERSONAHABILITADA,
        PERSONAHABILITADA.NOMBRE, PERSONAHABILITADA.TELEFONO,
        PERSONAHABILITADA.EMAIL
FROM (PERSONAHABILITADA INNER JOIN (
        SELECT ID_USUARIO, CANTIDAD_MESES
        FROM (
        SELECT DISTINCT RESERVA.IDPERSONAHABILITADA AS ID_USUARIO,
        COUNT (EXTRACT(MONTH FROM RESERVA.FECHAINICIORESERVA))
        AS CANTIDAD_MESES
        FROM RESERVA GROUP BY  RESERVA.IDPERSONAHABILITADA )
        WHERE CANTIDAD_MESES=4)
        ON PERSONAHABILITADA.IDPERSONAHABILITADA = ID_USUARIO);

//obtiene la info de los usuario que han reservado SOLO alojamientos costosos
SELECT PERSONAHABILITADA.IDPERSONAHABILITADA,
        PERSONAHABILITADA.NOMBRE, PERSONAHABILITADA.TELEFONO,
        PERSONAHABILITADA.EMAIL
FROM (PERSONAHABILITADA
        INNER JOIN (
        SELECT ID_USUARIO2 AS ID_VALIOSO FROM (
        (SELECT RESERVA.IDPERSONAHABILITADA AS ID_USUARIO2,
        COUNT(RESERVA.IDRESERVA) AS CANT_RESERVAS_VALIOSAS
        FROM (
        RESERVA INNER JOIN OFERTA ON RESERVA.IDOFERTA =
        OFERTA.IDOFERTA) WHERE OFERTA.VALOR >= 435000
        GROUP BY RESERVA.IDPERSONAHABILITADA
        )
        INNER JOIN
        (SELECT RESERVA.IDPERSONAHABILITADA AS ID_USUARIO1,
        COUNT(RESERVA.IDRESERVA) AS CANT_RESERVAS_TOTALES
        FROM (
        RESERVA INNER JOIN OFERTA ON RESERVA.IDOFERTA =
        OFERTA.IDOFERTA)
        GROUP BY RESERVA.IDPERSONAHABILITADA) THAT ON
        ID_USUARIO2= THAT.ID_USUARIO1)
        WHERE CANT_RESERVAS_VALIOSAS = CANT_RESERVAS_TOTALES)
        ON PERSONAHABILITADA.IDPERSONAHABILITADA= ID_VALIOSO);

//obtiene la info de los usuario que han reservado SOLO suites
SELECT PERSONAHABILITADA.IDPERSONAHABILITADA,
        PERSONAHABILITADA.NOMBRE, PERSONAHABILITADA.TELEFONO,
        PERSONAHABILITADA.EMAIL
FROM (PERSONAHABILITADA
```

```
INNER JOIN (
SELECT ID_USUARIO2 AS ID_VALIOSO FROM (
(SELECT RESERVA.IDPERSONAHABILITADA AS ID_USUARIO2,
COUNT(RESERVA.IDRESERVA) AS CANT_RESERVAS_VALIOSAS
FROM (
RESERVA INNER JOIN (
                    SELECT OFERTA.IDOFERTA AS
ID_OFERTA_SUITE FROM (
                    OFERTA INNER JOIN OFERTAHOTEL ON
OFERTA.IDOFERTA = OFERTAHOTEL.IDOFERTA)
                    WHERE OFERTAHOTEL.CATEGORIA = 'suite')
ON RESERVA.IDOFERTA = ID_OFERTA_SUITE)
GROUP BY RESERVA.IDPERSONAHABILITADA
)
INNER JOIN
(SELECT RESERVA.IDPERSONAHABILITADA AS ID_USUARIO1,
COUNT(RESERVA.IDRESERVA) AS CANT_RESERVAS_TOTALES
FROM (
RESERVA INNER JOIN OFERTA ON RESERVA.IDOFERTA =
OFERTA.IDOFERTA)
GROUP BY RESERVA.IDPERSONAHABILITADA) THAT ON
ID_USUARIO2= THAT.ID_USUARIO1)
WHERE CANT_RESERVAS_VALIOSAS = CANT_RESERVAS_TOTALES)
ON PERSONAHABILITADA.IDPERSONAHABILITADA= ID_VALIOSO);
```

- Distribución de los datos con respecto a los parámetros de entrada utilizados en el requerimiento funcional. En particular se quiere un análisis de distribución que permita ver cómo puede cambiar el tamaño de la respuesta según el valor de los parámetros utilizados y la configuración de los datos de prueba.



- Valores de los parámetros utilizados en el análisis y que constituyen diferenciadores en los planes de ejecución obtenidos.
- **En si no se ingresaron parámetros sino que se distinguieron diferentes criterios de búsqueda y asi se obtuvieron diferentes datos**

- Planes de consulta obtenidos en Oracle para la ejecución del requerimiento. Para ello, documente con una foto de pantalla los planes de consulta obtenidos en SQLDevelopper.

PLAN_TABLE_OUTPUT

```
Plan hash value: 2604836516

-----------------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name                  | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                       |   142 | 13206 |       |  8420   (1)| 00:00:01 |
|   1 |  NESTED LOOPS                 |                       |   142 | 13206 |       |  8420   (1)| 00:00:01 |
|   2 |   NESTED LOOPS                |                       |   142 | 13206 |       |  8420   (1)| 00:00:01 |
|*  3 |    HASH JOIN                  |                       |   142 |  7384 |       |  8279   (1)| 00:00:01 |
|   4 |     VIEW                      |                       | 14155 |   359K|       |  3436   (1)| 00:00:01 |
|   5 |      HASH GROUP BY            |                       | 14155 |   276K|       |  3436   (1)| 00:00:01 |
|*  6 |       HASH JOIN               |                       | 24023 |   469K|       |  3435   (1)| 00:00:01 |
|*  7 |        TABLE ACCESS FULL      | OFERTA                |  1615 | 16150 |       |   410   (1)| 00:00:01 |
|   8 |        TABLE ACCESS FULL      | RESERVA               |  1494K|    14M|       |  3021   (1)| 00:00:01 |
|   9 |     VIEW                      |                       |   100K|  2560K|       |  4842   (1)| 00:00:01 |
|  10 |      HASH GROUP BY            |                       |   100K|   492K|    17M|  4842   (1)| 00:00:01 |
|  11 |       TABLE ACCESS FULL       | RESERVA               |  1494K|  7296K|       |  3021   (1)| 00:00:01 |
|* 12 |    INDEX UNIQUE SCAN          | PERSONAHABILITADA_PK  |     1 |       |       |     0   (0)| 00:00:01 |
|  13 |   TABLE ACCESS BY INDEX ROWID| PERSONAHABILITADA     |     1 |    41 |       |     1   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   3 - access("from$_subquery$_003"."CANT_RESERVAS_VALIOSAS"="THAT"."CANT_RESERVAS_TOTALES" AND
              "ID_USUARIO2"="THAT"."ID_USUARIO1")
   6 - access("RESERVA"."IDOFERTA"="OFERTA"."IDOFERTA")
   7 - filter("OFERTA"."VALOR">=435000)
  12 - access("PERSONAHABILITADA"."IDPERSONAHABILITADA"="from$_subquery$_003"."ID_USUARIO2")

Note
-----
   - dynamic statistics used: dynamic sampling (level=2)
   - this is an adaptive plan
   - 3 Sql Plan Directives used for this statement
```

-
- Tiempos obtenidos con la ejecución de cada uno de los planes. Estos tiempos son medidos desde el núcleo de la aplicación, es decir, no incluyen la parte de interacción con el usuario, ingreso de datos ni despliegue de resultados.

Hoja de Trabajo | Generador de Consultas

```
1  SELECT PERSONAHABILITADA.IDPERSONAHABILITADA, PERSONAHABILITADA.NOMBRE, PERSONAHABILITADA.TELEFONO, PERSONAHABILITADA.EM
2  FROM (PERSONAHABILITADA INNER JOIN (
3      SELECT ID_USUARIO, CANTIDAD_MESES
4      FROM (
5          SELECT DISTINCT RESERVA.IDPERSONAHABILITADA AS ID_USUARIO, COUNT (EXTRACT(MONTH FROM RESERVA.FECHAINICIORESERVA
6          FROM RESERVA GROUP BY  RESERVA.IDPERSONAHABILITADA )
7          WHERE CANTIDAD_MESES=4)
8      ON PERSONAHABILITADA.IDPERSONAHABILITADA = ID_USUARIO);
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,416 segundos

	IDPERSONAHABILITADA	NOMBRE	TELEFONO	EMAIL
1	428	Louisa Gardner	73098	puehi@uhza.bd
2	449	Edwin Bennett	98780	cizhi@sa.fr
3	490	Ian Hoffman	70304	sakicu@fa.hu
4	571	Annie Sandoval	61089	goznul@dij.ch
5	573	Edward Jones	13262	ufru@efnocaz.bz
6	630	Lelia Fox	58414	fovsem@re.to
7	512345601	Harold Enrique Castro	2220001	harold01@gmail.com
8	280	Isabella Cummings	19890	ba@ama.fo
9	1457	Melvin Daniels	24604	mafigve@ti.vi
10	1582	Nelle Davis	57241	dusronet@mafovis.rw
11	1595	Marion Caldwell	4154	niuj@pamfadse.re

Consulta del RFC 13 - búsqueda 1

Hoja de Trabajo | Generador de Consultas

```
4          SELECT ID_USUARIO2 AS ID_VALIOSO FROM (
5              (SELECT RESERVA.IDPERSONAHABILITADA AS ID_USUARIO2, COUNT(RESERVA.IDRESERVA) AS CANT_RESERVAS_VALIOSAS
6                  FROM (
7                  RESERVA INNER JOIN OFERTA ON RESERVA.IDOFERTA = OFERTA.IDOFERTA) WHERE OFERTA.VALOR >= 435000
8                  GROUP BY RESERVA.IDPERSONAHABILITADA
9              )
10             INNER JOIN
11             (SELECT RESERVA.IDPERSONAHABILITADA AS ID_USUARIO1, COUNT(RESERVA.IDRESERVA) AS CANT_RESERVAS_TOTALES
12                 FROM (
13                 RESERVA INNER JOIN OFERTA ON RESERVA.IDOFERTA = OFERTA.IDOFERTA)
14                 GROUP BY RESERVA.IDPERSONAHABILITADA) THAT ON ID_USUARIO2= THAT.ID_USUARIO1)
15         WHERE CANT_RESERVAS_VALIOSAS = CANT_RESERVAS_TOTALES)
16     ON PERSONAHABILITADA.IDPERSONAHABILITADA= ID_VALIOSO);
```

Salida de Script | Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 4 en 0,422 segundos

	IDPERSONAHABILITADA	NOMBRE	TELEFONO	EMAIL
1	69059	Grace Buchanan	25729	netwiju@bauk.ir
2	512345613	Maria Alejandra Salazar	2220013	mariale83@gmail.com
3	89292	Evan Warner	32042	ron@ve.tz
4	512345612	Oscar Santiago Nieto	2220012	oscarni64@gmail.com

Consulta del RFC 13 - búsqueda 2

```
Hoja de Trabajo   Generador de Consultas
4   SELECT ID_USUARIO2 AS ID_VALIOSO FROM (
5       (SELECT RESERVA.IDPERSONAHABILITADA AS ID_USUARIO2, COUNT(RESERVA.IDRESERVA) AS CANT_RESERVAS_VALIOSAS
6           FROM (
7           RESERVA INNER JOIN (
8                   SELECT OFERTA.IDOFERTA AS ID_OFERTA_SUITE FROM (
9                   OFERTA INNER JOIN OFERTAHOTEL ON OFERTA.IDOFERTA = OFERTAHOTEL.IDOFERTA)
10                  WHERE OFERTAHOTEL.CATEGORIA = 'suite')
11          ON RESERVA.IDOFERTA = ID_OFERTA_SUITE)
12          GROUP BY RESERVA.IDPERSONAHABILITADA
13      )
14      INNER JOIN
15      (SELECT RESERVA.IDPERSONAHABILITADA AS ID_USUARIO1, COUNT(RESERVA.IDRESERVA) AS CANT_RESERVAS_TOTALES
16          FROM (
17          RESERVA INNER JOIN OFERTA ON RESERVA.IDOFERTA = OFERTA.IDOFERTA)
18          GROUP BY RESERVA.IDPERSONAHABILITADA) THAT ON ID_USUARIO2= THAT.ID_USUARIO1)
19  WHERE CANT_RESERVAS_VALIOSAS = CANT_RESERVAS_TOTALES)
20 ON PERSONAHABILITADA.IDPERSONAHABILITADA= ID_VALIOSO);

Salida de Script   Resultado de la Consulta
SQL | Todas las Filas Recuperadas: 1 en 0,4 segundos

| | IDPERSONAHABILITADA | NOMBRE | TELEFONO | EMAIL |
|---|---|---|---|---|
| 1 | 512345611 | Gustavo Adolfo Fernandez | 2220011 | gustavoad754@gmail.com |
```

Consulta del RFC 13 - búsqueda 3

- **Análisis de eficiencia**

  - Establezca escenarios de datos que le permitan validar diferentes selectividades.

  - Para cada requerimiento funcional, seleccione un escenario de análisis y diseñe el plan de ejecución de consulta propuesto por el grupo, de acuerdo con su conocimiento del modelo y de la aplicación.

  - Compare y analice el plan de ejecución propuesto por usted y el obtenido en Oracle.

(15 %) Construcción de la aplicación, ejecución de pruebas y análisis de resultados

- Analice la diferencia entre la ejecución de consultas delegada al manejador de bases de datos como Oracle y compárelo con una ejecución donde la aplicación trae los datos a memoria principal y resuelve con instrucciones de control (if, while, etc.), los operadores involucrados en las consultas como joins, selecciones y proyecciones.

La diferencia que existe es que debido al uso de intermediarios como es el caso de una aplicación de java reduce drásticamente el desempeño de la consulta, en este caso se aumenta drásticamente el tiempo de respuesta ya que el manejador de base de datos está optimizado para realizar consultas y

gracias al uso de índices y demás es posible tener tiempos de respuesta mucho menores. El hecho de hacer cargar desde java datos a memoria principal es una de las peores tácticas a utilizar en el caso de estas consultas ya que hace un mal uso de los recursos computacionales.