Mateo Pedroza 201531957
David Acuña 201517879

Iteración 3

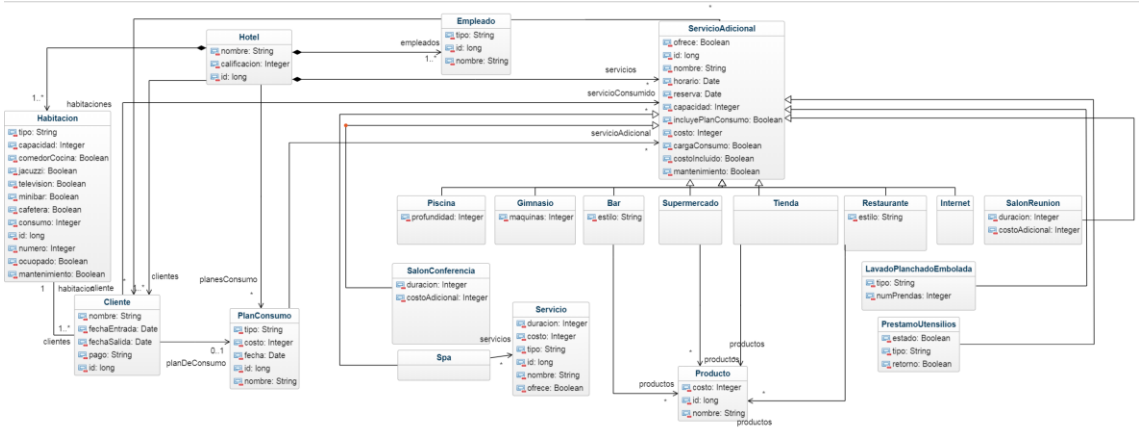
Cambios al modelo:

Análisis: Tras la retroalimentación de la iteración 2, se corrigieron los errores pertinentes y se ajustó el modelo del mundo para soportar la carga masiva de archivos. Se corrigieron aspectos de organización en los datos, clases y relaciones planteadas para permitir mayor facilidad de los requerimientos planteados en la iteración 3. Entre los cambios realizados, se incluyeron mas atributos en la clase de Servicio Adicional para simplificar las clases que heredan de esta, como el costo y también se cambió la variable de consumo acumulable para ser atributo de la clase Habitación.

Diseño de la aplicación: Ante la introducción de los nuevos requerimientos, en especial, la carga masiva de datos, tuvieron un fuerte impacto en el modelo planteado, fue necesario realizar los cambios resaltados anteriormente para asegurar buenas temporalidades en la duración de las consultas y a su vez simplificar la inserción de los datos.

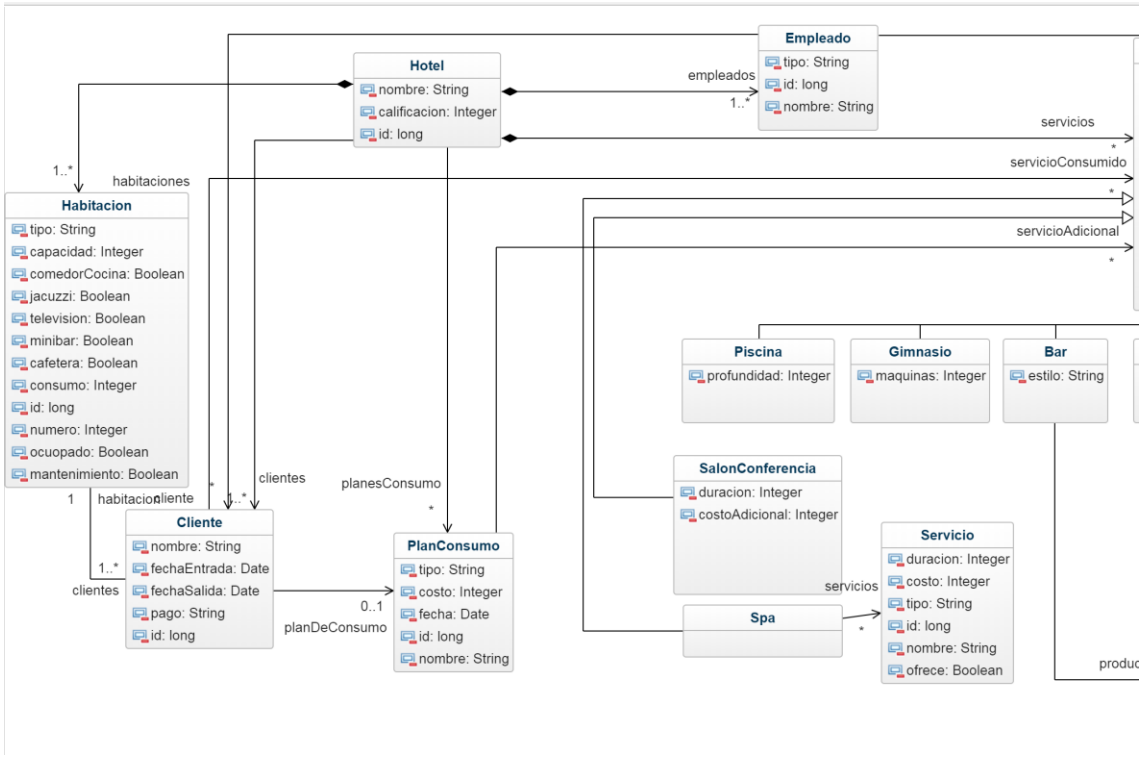
A continuación, se presenta entonces el modelo UML, junto a sus tablas relacionales:

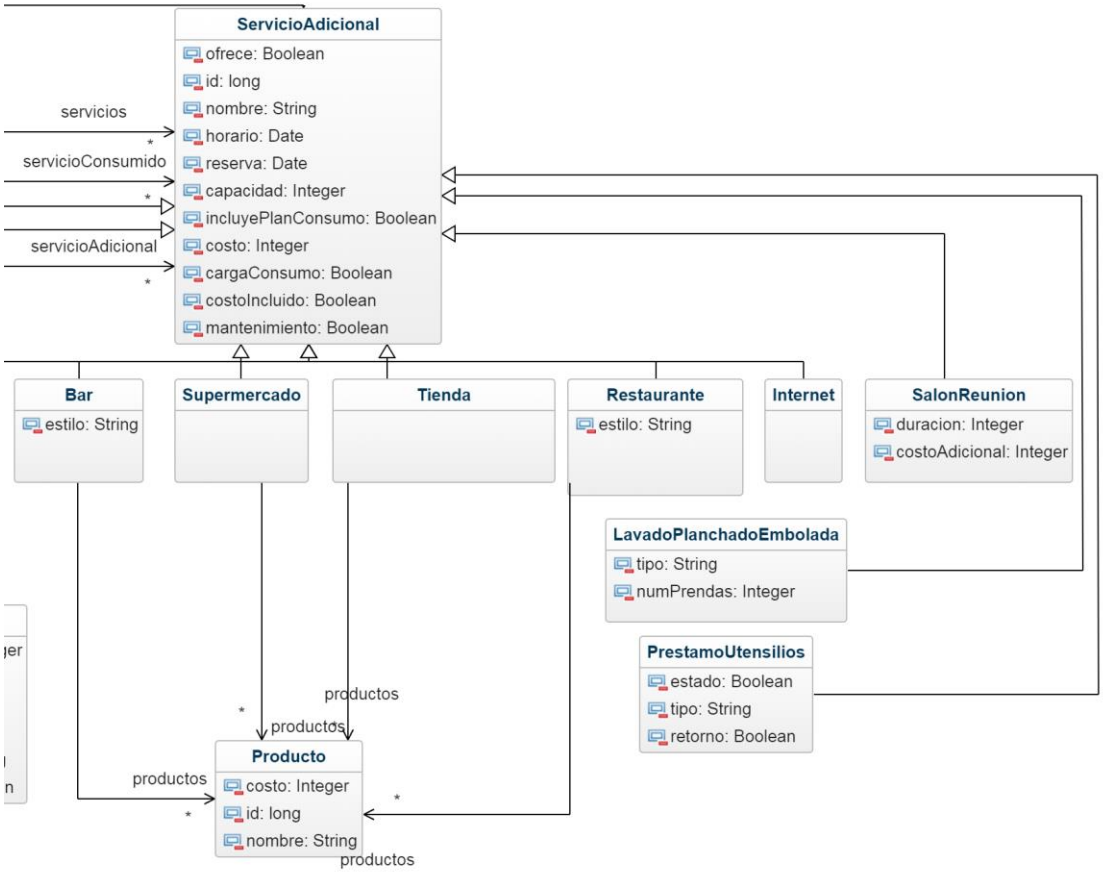
Modelo UML:



Mateo Pedroza 201531957
David Acuña 201517879

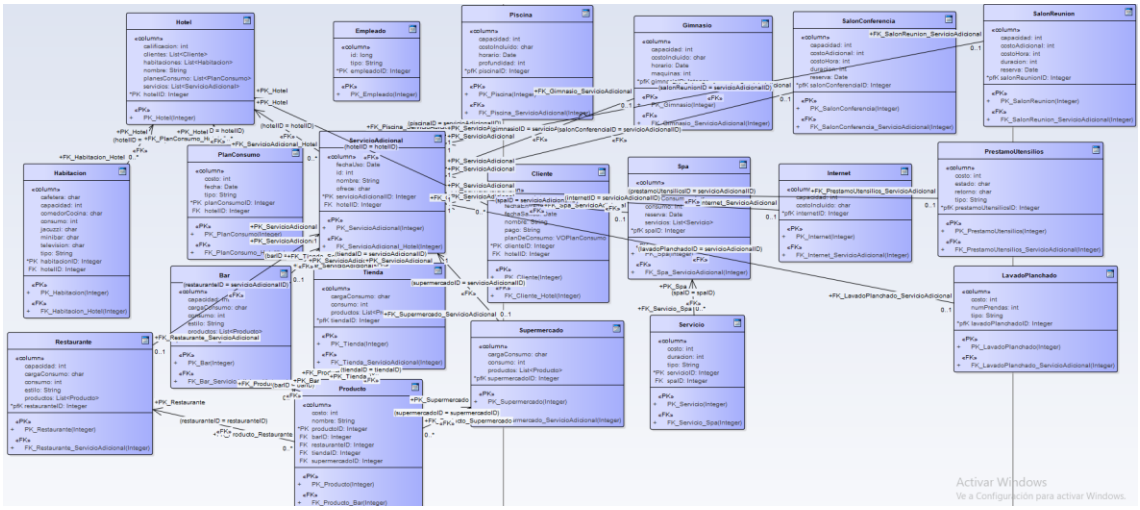
Modelo cortado, por motivos de visualización:





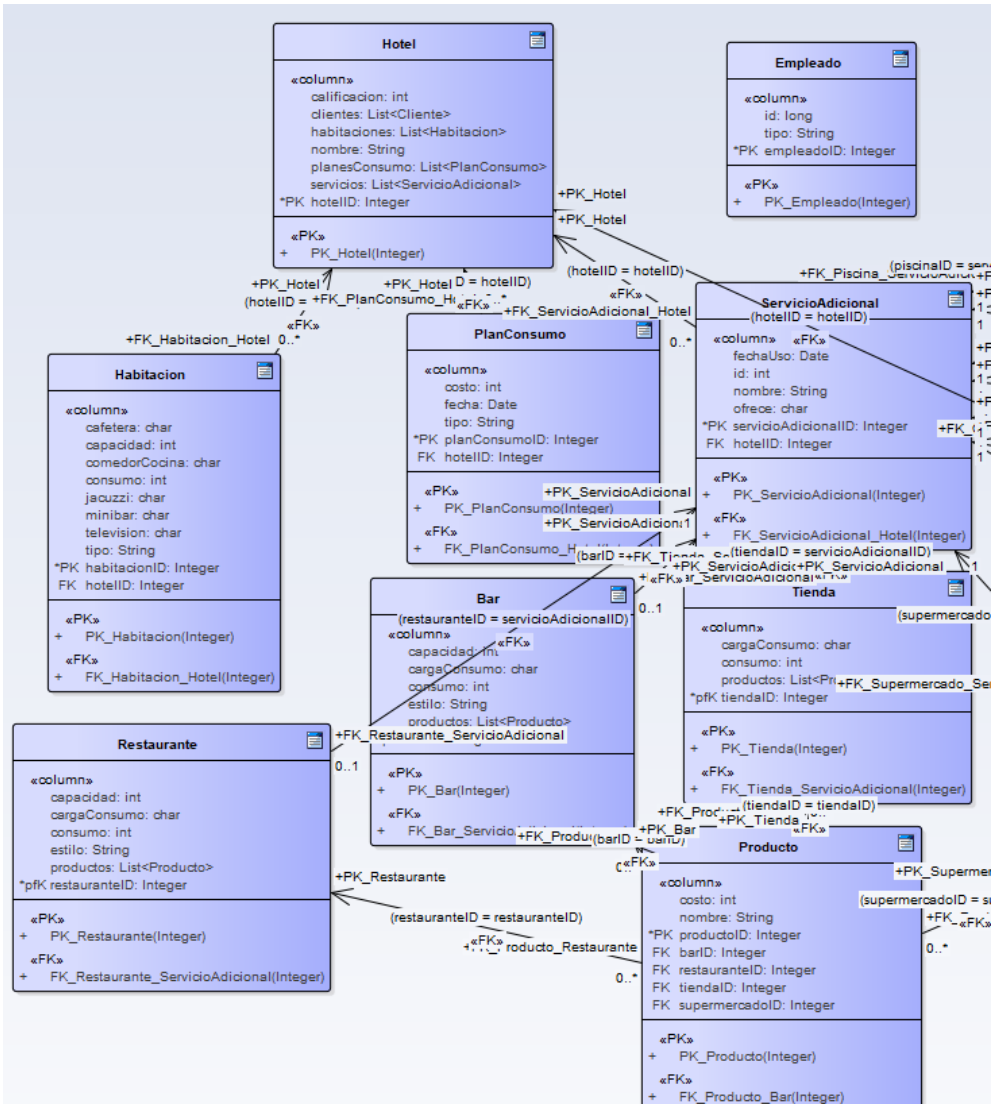
Mateo Pedroza 201531957
David Acuña 201517879

Modelo Relacional:

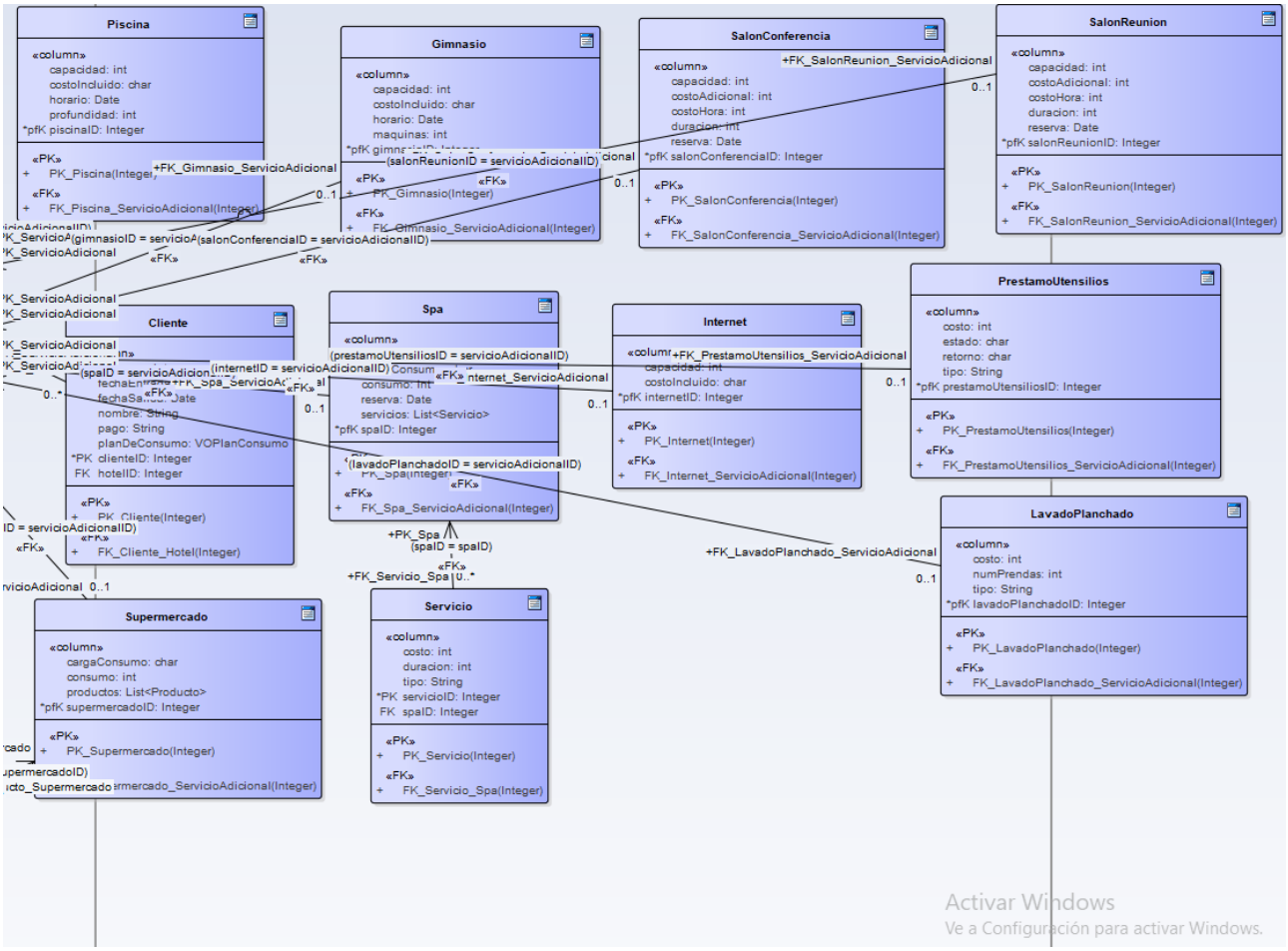


Modelo cortado, por motivos de visualización:

Mateo Pedroza 201531957
David Acuña 201517879



Mateo Pedroza 201531957
David Acuña 201517879



Modelo Relacional 2:

Hotel		
Id	nombre	calificación
PK	NN	CK > 0

Habitación											
Id	tipo	capacidad	comedorCocina	jacuzzi	television	minibar	cafetera	consumo	ocupado	Id_Hotel	Mantemineitno
PK	NN	CK > 0	NN	NN	NN	NN	NN	NN	NN	FK	NN

Cliente						
Id	nombre	fechaEntrada	fechaSalida	pago	Id_Hotel	id_Habitacion
PK	NN	NN	NN	NN	FK	FK

PlanConsumo					
Id	tipo	costo	fecha	Id_Hotel	Id_Cliente
PK	NN	CK > 0	NN	FK	FK

Mateo Pedroza 201531957
David Acuña 201517879

ServicioAdicional											
Id	nombre	ofrece	reserva	capacidad	incluyePlanConsumo	costo	costoIncluido	horario	Id_cliente	Id_Hotel	Mantemineitno
PK	NN	NN		CK>0		NN	NN	CK>0	FK	FK	NN

Piscina										
Id_ServicioAdicional	nombre_ServicioAdicional	ofrece_ServicioAdicional	reserva_ServicioAdicional	capacidad_ServicioAdicional	incluyePlanConsumo_ServicioAdicional	costo_ServicioAdicional	costoIncluido_ServicioAdicional	horario_ServicioAdicional	profundidad	Mantemineitno
FK,PK	FK,NN	FK,NN	FK	FK,CK>0	FK	FK,NN	FK,NN	FK,CK>0	CK > 0	FK, NN

Gimnasio										
Id_ServicioAdicional	nombre_ServicioAdicional	ofrece_ServicioAdicional	reserva_ServicioAdicional	capacidad_ServicioAdicional	incluyePlanConsumo_ServicioAdicional	costo_ServicioAdicional	costoIncluido_ServicioAdicional	horario_ServicioAdicional	maquinas	Mantemineitno
FK,PK	FK,NN	FK,NN	FK	FK,CK>0	FK	FK,NN	FK,NN	FK,CK>0	CK > 0	FK, NN

Bar										
Id_ServicioAdicional	nombre_ServicioAdicional	ofrece_ServicioAdicional	reserva_ServicioAdicional	capacidad_ServicioAdicional	incluyePlanConsumo_ServicioAdicional	costo_ServicioAdicional	costoIncluido_ServicioAdicional	horario_ServicioAdicional	estilo	Mantemineitno
FK,PK	FK,NN	FK,NN	FK	FK,CK>0	FK	FK,NN	FK,NN	FK,CK>0	NN	FK, NN

SuperMercado									
Id_ServicioAdicional	nombre_ServicioAdicional	ofrece_ServicioAdicional	reserva_ServicioAdicional	capacidad_ServicioAdicional	incluyePlanConsumo_ServicioAdicional	costo_ServicioAdicional	costoIncluido_ServicioAdicional	horario_ServicioAdicional	Mantemineitno
FK,PK	FK,NN	FK,NN	FK	FK,CK>0	FK	FK,NN	FK,NN	FK,CK>0	FK, NN

Tienda									
Id_ServicioAdicional	nombre_ServicioAdicional	ofrece_ServicioAdicional	reserva_ServicioAdicional	capacidad_ServicioAdicional	incluyePlanConsumo_ServicioAdicional	costo_ServicioAdicional	costoIncluido_ServicioAdicional	horario_ServicioAdicional	Mantemineitno
FK,PK	FK,NN	FK,NN	FK	FK,CK>0	FK	FK,NN	FK,NN	FK,CK>0	FK, NN

Restaurante										
Id_ServicioAdicional	nombre_ServicioAdicional	ofrece_ServicioAdicional	reserva_ServicioAdicional	capacidad_ServicioAdicional	incluyePlanConsumo_ServicioAdicional	costo_ServicioAdicional	costoIncluido_ServicioAdicional	horario_ServicioAdicional	estilo	Mantemineitno
FK,PK	FK,NN	FK,NN	FK	FK,CK>0	FK	FK,NN	FK,NN	FK,CK>0	NN	FK, NN

Mateo Pedroza 201531957
David Acuña 201517879

Internet									
Id_ServicioAdicional	nombre_ServicioAdicional	ofrece_ServicioAdicional	reserva_ServicioAdicional	capacidad_ServicioAdicional	incluyePlanConsumo_ServicioAdicional	costo_ServicioAdicional	costoIncluido_ServicioAdicional	horario_ServicioAdicional	Mantemineitno
FK,PK	FK,NN	FK,NN	FK	FK,CK>0	FK	FK,NN	FK,NN	FK,CK>0	FK, NN

SalonReunion											
Id_ServicioAdicional	nombre_ServicioAdicional	ofrece_ServicioAdicional	reserva_ServicioAdicional	capacidad_ServicioAdicional	incluyePlanConsumo_ServicioAdicional	costo_ServicioAdicional	costoIncluido_ServicioAdicional	horario_ServicioAdicional	duracion	costoAdicional	Mantemineitno
FK,PK	FK,NN	FK,NN	FK	FK,CK>0	FK	FK,NN	FK,NN	FK,CK>0	CK>0	NN	FK, NN

SalonConferencia											
Id_ServicioAdicional	nombre_ServicioAdicional	ofrece_ServicioAdicional	reserva_ServicioAdicional	capacidad_ServicioAdicional	incluyePlanConsumo_ServicioAdicional	costo_ServicioAdicional	costoIncluido_ServicioAdicional	horario_ServicioAdicional	duracion	costoAdicional	Mantemineitno
FK,PK	FK,NN	FK,NN	FK	FK,CK>0	FK	FK,NN	FK,NN	FK,CK>0	CK>0	NN	FK, NN

LavadoPlanchadoEmbolada											
Id_ServicioAdicional	nombre_ServicioAdicional	ofrece_ServicioAdicional	reserva_ServicioAdicional	capacidad_ServicioAdicional	incluyePlanConsumo_ServicioAdicional	costo_ServicioAdicional	costoIncluido_ServicioAdicional	horario_ServicioAdicional	tipo	numPrendas	Mantemineitno
FK,PK	FK,NN	FK,NN	FK	FK,CK>0	FK	FK,NN	FK,NN	FK,CK>0	NN	CK>0	FK, NN

PrestamosUtensilios												
Id_ServicioAdicional	nombre_ServicioAdicional	ofrece_ServicioAdicional	reserva_ServicioAdicional	capacidad_ServicioAdicional	incluyePlanConsumo_ServicioAdicional	costo_ServicioAdicional	costoIncluido_ServicioAdicional	horario_ServicioAdicional	estado	tipo	retorno	Mantemineitno
FK,PK	FK,NN	FK,NN	FK	FK,CK>0	FK	FK,NN	FK,NN	FK,CK>0	NN	NN	NN	FK, NN

Spa									
Id_ServicioAdicional	nombre_ServicioAdicional	ofrece_ServicioAdicional	reserva_ServicioAdicional	capacidad_ServicioAdicional	incluyePlanConsumo_ServicioAdicional	costo_ServicioAdicional	costoIncluido_ServicioAdicional	horario_ServicioAdicional	Mantemineitno
FK,PK	FK,NN	FK,NN	FK	FK,CK>0	FK	FK,NN	FK,NN	FK,CK>0	FK, NN

Servicio							
Id	duracion	costo	tipo	ofrece	Id_Spa	Id_Hotel	id_Cliente
PK	CK > 0	CK > 0	NN	NN	FK	FK	FK

Mateo Pedroza 201531957
David Acuña 201517879

Producto								
Id	costo	nombre	Id_Bar	Id_Supermercado	Id_Tienda	Id_Restaurante	Id_Hotel	id_Cliente
PK	CK > 0	NN	FK	FK	FK	FK	FK	FK

BCNF: Cabe resaltar que la estructura del modelo creado cumple con la forma normal BC, pues no existen dependencias parciales ni transitivas entre los atributos no primos, y todos los recursos dependen de una clave "PK" que es la única determinante para el recurso. En otras palabras, podemos afirmar la tercera forma normal en el modelo establecido ya que no existen dependencias funcionales dentro de los recursos planteados y a su vez, cada uno de ellos tiene una llave primaria única para acceder a su información.

Justificación de la selección de índices:

Para seleccionar un buen índice que asegure buena velocidad de consulta en la base de datos, es necesario conocer la selectividad del mismo, esto es, el porcentaje o porción que dicho índice captura de los datos. Así pues, se considera un buen índice aquel con un nivel de selectividad menor al 25%. Para la aplicación del proyecto, se consideraron los siguientes índices: En la tabla de 'cliente', el índice 'nombre cliente' Con un nivel de selectividad respectivo con valores de: 5%. Los índices fueron seleccionados con los requerimientos de consulta en mente pues la función de estos es disminuir el tamaño general de la tabla guardado en memoria un apuntador o lista de índices con los cuales se podrán acceder a los datos, disminuyendo así el espacio en memoria requerida para realizar un join y con esto reduciendo el tiempo de duración de una consulta particular.

Por otro lado, el tipo de índice usado fue un secundario. Dicha selección es a causa de la magnitud de datos y la selectividad general de los índices planteados. Se seleccionó este índice secundario, pues entre todas las otras opciones, asegura un menor tiempo de consulta ya que ocupa una menor magnitud de memoria principal dado que permite acceder a un grupo de datos con ciertas características dadas por la selectividad, en lugar de la tabla completa.

El costo de almacenamiento y mantenimiento asociado a los índices es el siguiente:

Índices generados por Oracle:

	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	ISIS2304B011910	NOMBRE	NONUNIQUE	VALID	NORMAL	N	NO	(null)	NO	NOMBRE
2	ISIS2304B011910	CLIENTE_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	ID

Oracle inicialmente crea un índice primario basado en el id de cada tabla para cada una de estas. Esto sucede porque, al menos una de las dos tablas que están haciendo join, cabe en memoria.

Adicionalmente, agregamos un índice secundario en la tabla de cliente, que es el que apareceré en imagen adjunta.

Mateo Pedroza 201531957
David Acuña 201517879

Sentencias SQL:

```
--RFC9
    --Cualquiera
SELECT DISTINCT cliente.*
FROM cliente,servicioAdicional
WHERE  fechaUso BETWEEN TO_DATE('22/Marzo/2018','DD/MON/YY') AND
TO_DATE('23/Abril/2018','DD/MON/YY')
AND cliente.id = servicioAdicional.id_Cliente;

    --Organizador
SELECT DISTINCT cliente.*
FROM cliente,servicioAdicional
WHERE  fechaUso BETWEEN TO_DATE('22/Marzo/2018','DD/MON/YY') AND
TO_DATE('23/Abril/2018','DD/MON/YY')
AND cliente.id = servicioAdicional.id_Cliente
AND servicioAdicional.reserva = 9;

--RFC10
    --Cualquiera
SELECT cliente.*,b.fechaUso
FROM cliente LEFT OUTER JOIN (
SELECT *
FROM servicioAdicional
WHERE  fechaUso BETWEEN TO_DATE('22/Marzo/2018','DD/MON/YY') AND
TO_DATE('23/Abril/2018','DD/MON/YY'))b
ON cliente.id = b.id_cliente
WHERE b.costo is null;

    --Organizador
    SELECT cliente.*,b.fechaUso
FROM cliente LEFT OUTER JOIN (
SELECT *
FROM servicioAdicional
WHERE  fechaUso BETWEEN TO_DATE('22/Marzo/2018','DD/MON/YY') AND
TO_DATE('23/Abril/2018','DD/MON/YY')
AND servicioAdicional.reserva = 9)b
ON cliente.id = b.id_cliente
WHERE b.costo is null;

--RFC11
     --Mayor uso servicio
Select a.nombre as NombreServicio, Count(a.nombre) as
NumeroDeUsos, to_number (to_char (a.fechaUso, 'w')) as semana,
to_number (to_char (a.fechaUso, 'mm')) as mes
From servicioadicional a
where to_number (to_char (a.fechaUso, 'w')) = 1
AND to_number (to_char (a.fechaUso, 'mm')) = 1
group by nombre, fechauso
order by Count(a.nombre) DESC
fetch first row only;

     --Menor uso servicio
Select a.nombre as NombreServicio, Count(a.nombre) as
NumeroDeUsos, to_number (to_char (a.fechaUso, 'w')) as semana,
to_number (to_char (a.fechaUso, 'mm')) as mes
```

```
From servicioadicional a
where to_number (to_char (a.fechaUso, 'w')) = 2
AND to_number (to_char (a.fechaUso, 'mm')) = 2
group by nombre, fechauso
order by Count(a.nombre) ASC
fetch first row only;
```

--Mayor uso Habitación

```
Select count(a.id) as NumeroDeUsoDeHabitacion, a.id as
numeroHabitacion
From habitacion a inner join cliente c on a.id = c.id_habitacion
where to_number (to_char (c.fechaentrada, 'w')) = 1
AND to_number (to_char (c.fechaentrada, 'mm')) = 1
group by a.id
order by count (a.id) DESC
fetch first row only;
```

--Menor Uso habitación

```
Select count(a.id) as NumeroDeUsoDeHabitacion, a.id as
numeroHabitacion
From habitacion a inner join cliente c on a.id = c.id_habitacion
where to_number (to_char (c.fechaentrada, 'w')) = 1
AND to_number (to_char (c.fechaentrada, 'mm')) = 1
group by a.id
order by count (a.id) ASC
fetch first row only;
```

--RFC12

--Clientes que van cada trimestre de un año

```
SELECT DISTINCT a.nombre
FROM(
SELECT cliente.*
FROM cliente
WHERE to_number (to_char (cliente.fechaentrada, 'qy')) = 16
)a
INNER JOIN
(SELECT cliente.*
FROM cliente
WHERE to_number (to_char (cliente.fechaentrada, 'qy')) = 26)   b
on a.nombre = b.nombre
INNER JOIN
(SELECT cliente.*
FROM cliente
WHERE to_number (to_char (cliente.fechaentrada, 'qy')) = 36)   c
on a.nombre = b.nombre
INNER JOIN
(SELECT cliente.*
FROM cliente
WHERE to_number (to_char (cliente.fechaentrada, 'qy')) = 46)   d
on c.nombre = d.nombre;
```

--Clientes que consumen servicios mayores a 400.000

```
SELECT DISTINCT cliente.nombre
FROM cliente, servicioAdicional
WHERE cliente.id = servicioAdicional.id
```

```
AND servicioAdicional.costo >400000;

     --Clientes que consumen servicios por al menos de 4 horas
SELECT DISTINCT cliente.*
FROM cliente, servicioAdicional
WHERE cliente.id = servicioAdicional.id
AND servicioadicional.duracion >3;
```

Cambio en tamaño de respuesta con diferentes parámetros:

Hubo cambios al variar la cantidad de datos para las consultas, pero no hubo cambios en el tiempo al variar el rango de las fechas, ya que el plan de ejecución de Oracle hace un full table scan.

Valores de los parámetros usados:

- **RFC9**
  Fecha de uso (01/01/2018,01/02/2018),(01/01/2018,01/01/2018)

- **RFC10**
  Fecha de uso (01/01/2018,01/02/2018),(01/01/2018,01/01/2018)

- **RFC11**
  to_number (to_char (c.fechaentrada, 'w')) = 1
  to_number (to_char (c.fechaentrada, 'mm')) = 1,
  to_number (to_char (c.fechaentrada, 'w')) = 1
  to_number (to_char (c.fechaentrada, 'mm')) = 2

- **RFC12**
  to_number (to_char (cliente.fechaentrada, 'qy')) = 16,26,36,46
  to_number (to_char (cliente.fechaentrada, 'qy')) = 17,27,37,47

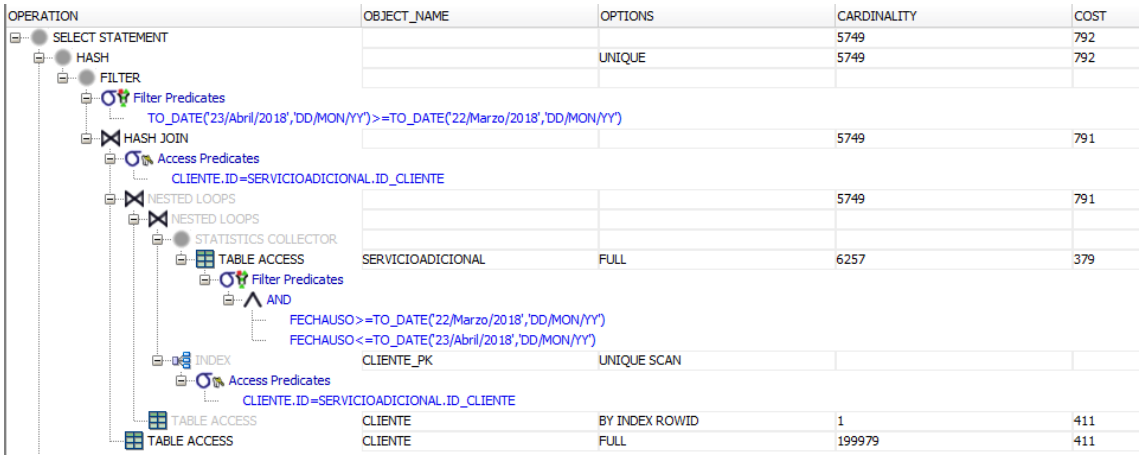
Análisis de eficiencia:

Plan de consulta obtenido por Oracle y tiempos de ejecución:

Para una carga de 200.000 datos tanto en la tabla cliente, como en servicio adicional, se obtuvo el siguiente plan de ejecución para cada RFC de esta iteración

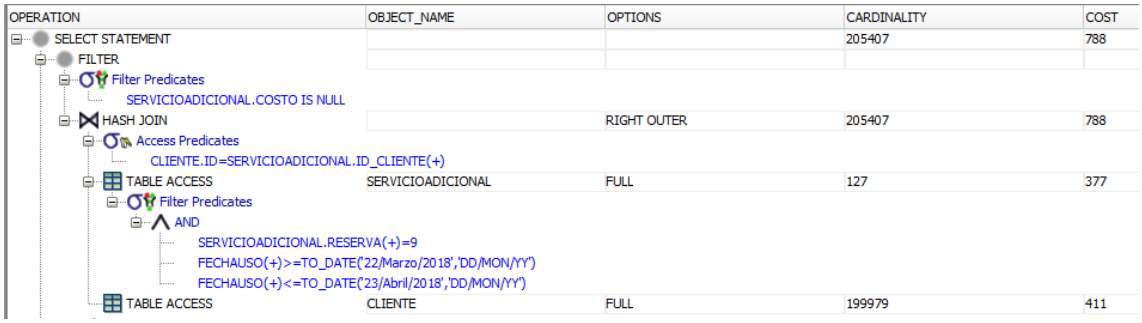
Mateo Pedroza 201531957
David Acuña 201517879

- **RFC9**



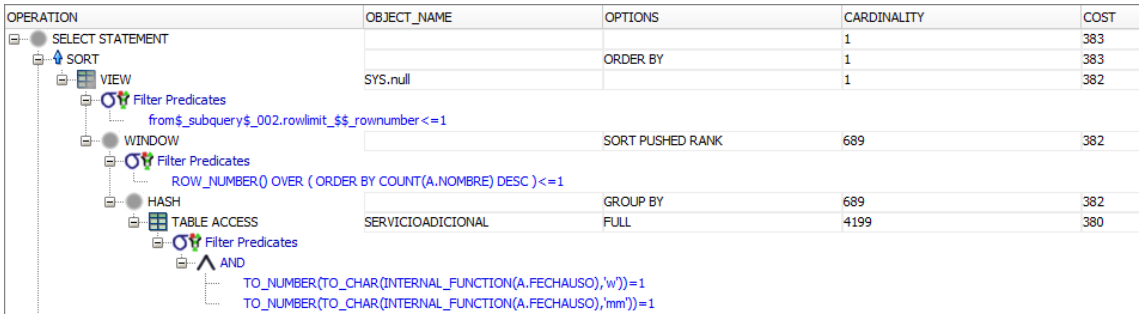
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			5749	792
HASH		UNIQUE	5749	792
FILTER				
Filter Predicates				
TO_DATE('23/Abril/2018','DD/MON/YY')>=TO_DATE('22/Marzo/2018','DD/MON/YY')				
HASH JOIN			5749	791
Access Predicates				
CLIENTE.ID=SERVICIOADICIONAL.ID_CLIENTE				
NESTED LOOPS			5749	791
NESTED LOOPS				
STATISTICS COLLECTOR				
TABLE ACCESS	SERVICIOADICIONAL	FULL	6257	379
Filter Predicates				
AND				
FECHAUSO>=TO_DATE('22/Marzo/2018','DD/MON/YY')				
FECHAUSO<=TO_DATE('23/Abril/2018','DD/MON/YY')				
INDEX	CLIENTE_PK	UNIQUE SCAN		
Access Predicates				
CLIENTE.ID=SERVICIOADICIONAL.ID_CLIENTE				
TABLE ACCESS	CLIENTE	BY INDEX ROWID	1	411
TABLE ACCESS	CLIENTE	FULL	199979	411

- **RFC10**

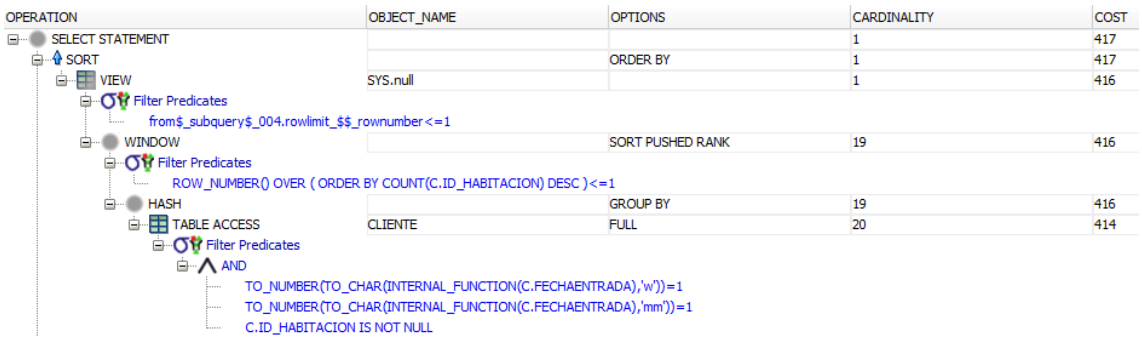


OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			205407	788
FILTER				
Filter Predicates				
SERVICIOADICIONAL.COSTO IS NULL				
HASH JOIN		RIGHT OUTER	205407	788
Access Predicates				
CLIENTE.ID=SERVICIOADICIONAL.ID_CLIENTE(+)				
TABLE ACCESS	SERVICIOADICIONAL	FULL	127	377
Filter Predicates				
AND				
SERVICIOADICIONAL.RESERVA(+)=9				
FECHAUSO(+)>=TO_DATE('22/Marzo/2018','DD/MON/YY')				
FECHAUSO(+)<=TO_DATE('23/Abril/2018','DD/MON/YY')				
TABLE ACCESS	CLIENTE	FULL	199979	411

- **RFC11**



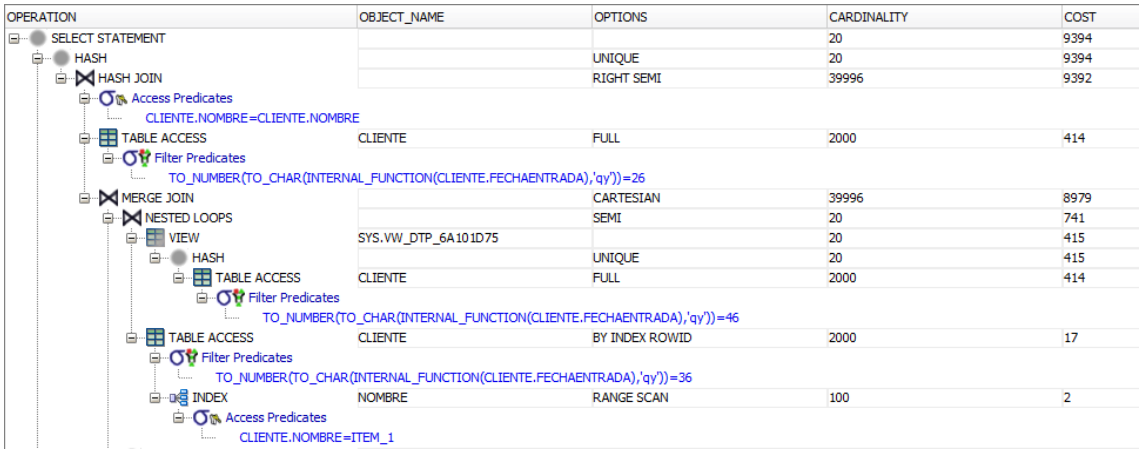
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	383
SORT		ORDER BY	1	383
VIEW	SYS.null		1	382
Filter Predicates				
from$_subquery$_002.rowlimit_$$_rownumber<=1				
WINDOW		SORT PUSHED RANK	689	382
Filter Predicates				
ROW_NUMBER() OVER (ORDER BY COUNT(A.NOMBRE) DESC)<=1				
HASH		GROUP BY	689	382
TABLE ACCESS	SERVICIOADICIONAL	FULL	4199	380
Filter Predicates				
AND				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(A.FECHAUSO),'w'))=1				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(A.FECHAUSO),'mm'))=1				



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	417
SORT		ORDER BY	1	417
VIEW	SYS.null		1	416
Filter Predicates				
from$_subquery$_004.rowlimit_$$_rownumber<=1				
WINDOW		SORT PUSHED RANK	19	416
Filter Predicates				
ROW_NUMBER() OVER (ORDER BY COUNT(C.ID_HABITACION) DESC)<=1				
HASH		GROUP BY	19	416
TABLE ACCESS	CLIENTE	FULL	20	414
Filter Predicates				
AND				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(C.FECHAENTRADA),'w'))=1				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(C.FECHAENTRADA),'mm'))=1				
C.ID_HABITACION IS NOT NULL				

Mateo Pedroza 201531957
David Acuña 201517879

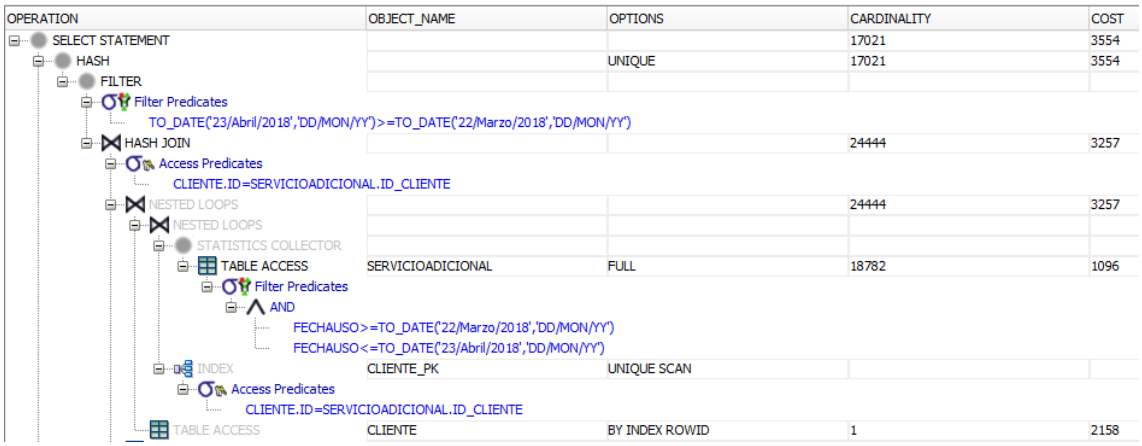
- **RFC12**



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			20	9394
HASH		UNIQUE	20	9394
HASH JOIN		RIGHT SEMI	39996	9392
Access Predicates				
CLIENTE.NOMBRE=CLIENTE.NOMBRE				
TABLE ACCESS	CLIENTE	FULL	2000	414
Filter Predicates				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(CLIENTE.FECHAENTRADA),'qy'))=26				
MERGE JOIN		CARTESIAN	39996	8979
NESTED LOOPS		SEMI	20	741
VIEW	SYS.VW_DTP_6A101D75		20	415
HASH		UNIQUE	20	415
TABLE ACCESS	CLIENTE	FULL	2000	414
Filter Predicates				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(CLIENTE.FECHAENTRADA),'qy'))=46				
TABLE ACCESS	CLIENTE	BY INDEX ROWID	2000	17
Filter Predicates				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(CLIENTE.FECHAENTRADA),'qy'))=36				
INDEX	NOMBRE	RANGE SCAN	100	2
Access Predicates				
CLIENTE.NOMBRE=ITEM_1				

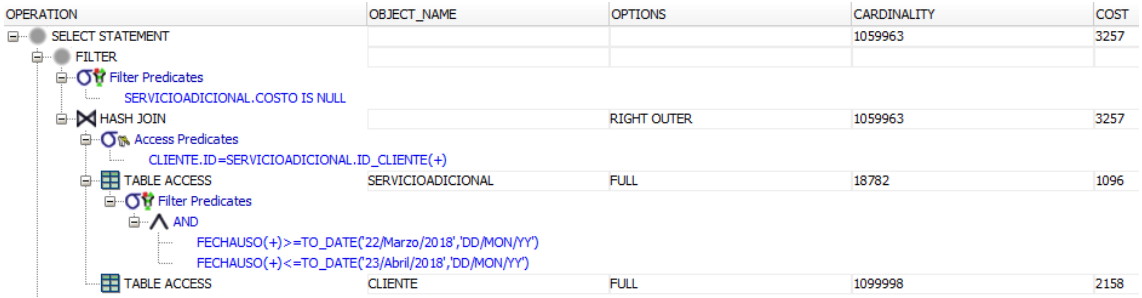
Después de una carga masiva de datos (1.1M en clientes y 200K de servicios)

- **RFC9**



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			17021	3554
HASH		UNIQUE	17021	3554
FILTER				
Filter Predicates				
TO_DATE('23/Abril/2018','DD/MON/YY')>=TO_DATE('22/Marzo/2018','DD/MON/YY')				
HASH JOIN			24444	3257
Access Predicates				
CLIENTE.ID=SERVICIOADICIONAL.ID_CLIENTE				
NESTED LOOPS			24444	3257
NESTED LOOPS				
STATISTICS COLLECTOR				
TABLE ACCESS	SERVICIOADICIONAL	FULL	18782	1096
Filter Predicates				
AND				
FECHAUSO>=TO_DATE('22/Marzo/2018','DD/MON/YY')				
FECHAUSO<=TO_DATE('23/Abril/2018','DD/MON/YY')				
INDEX	CLIENTE_PK	UNIQUE SCAN		
Access Predicates				
CLIENTE.ID=SERVICIOADICIONAL.ID_CLIENTE				
TABLE ACCESS	CLIENTE	BY INDEX ROWID	1	2158

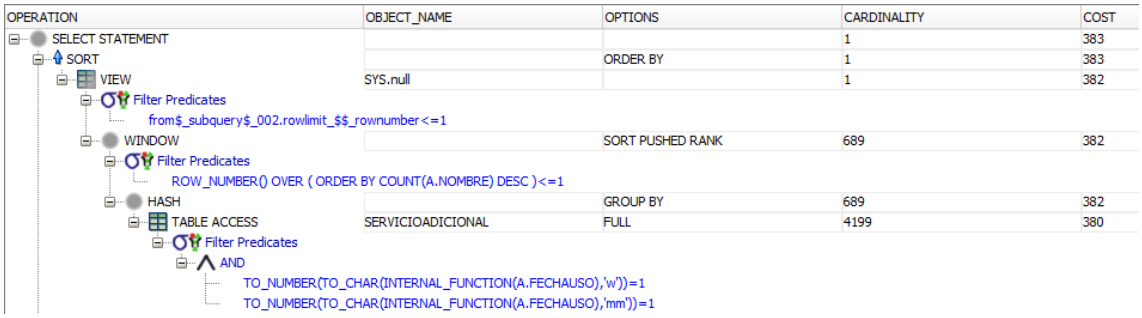
- **RFC10**



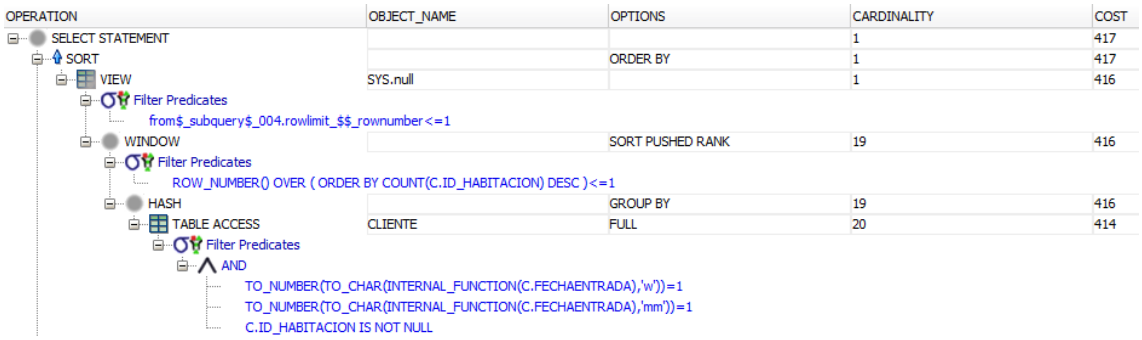
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1059963	3257
FILTER				
Filter Predicates				
SERVICIOADICIONAL.COSTO IS NULL				
HASH JOIN		RIGHT OUTER	1059963	3257
Access Predicates				
CLIENTE.ID=SERVICIOADICIONAL.ID_CLIENTE(+)				
TABLE ACCESS	SERVICIOADICIONAL	FULL	18782	1096
Filter Predicates				
AND				
FECHAUSO(+)>=TO_DATE('22/Marzo/2018','DD/MON/YY')				
FECHAUSO(+)<=TO_DATE('23/Abril/2018','DD/MON/YY')				
TABLE ACCESS	CLIENTE	FULL	1099998	2158

Mateo Pedroza 201531957
David Acuña 201517879

- **RFC11**

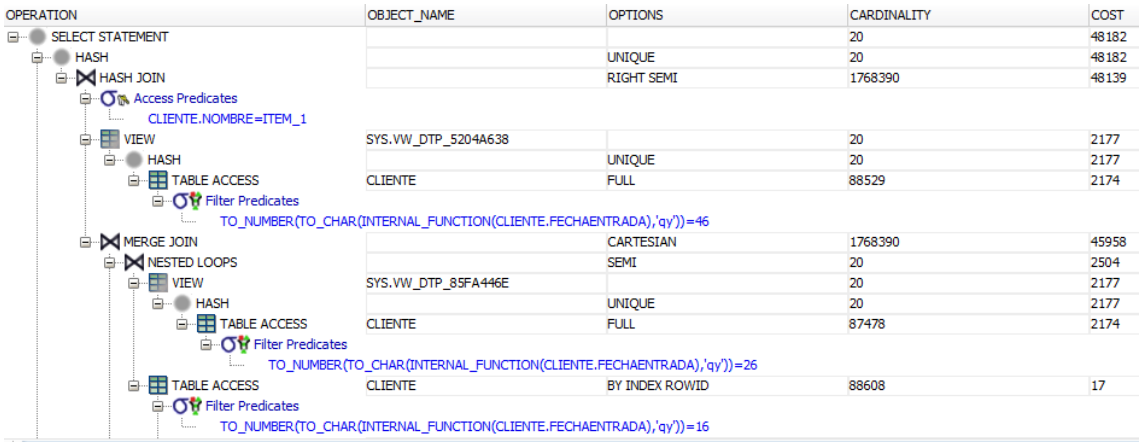


OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	383
SORT		ORDER BY	1	383
VIEW	SYS.null		1	382
Filter Predicates				
from$_subquery$_002.rowlimit_$$_rownumber<=1				
WINDOW		SORT PUSHED RANK	689	382
Filter Predicates				
ROW_NUMBER() OVER (ORDER BY COUNT(A.NOMBRE) DESC)<=1				
HASH		GROUP BY	689	382
TABLE ACCESS	SERVICIOADICIONAL	FULL	4199	380
Filter Predicates				
AND				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(A.FECHAUSO),'w'))=1				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(A.FECHAUSO),'mm'))=1				



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	417
SORT		ORDER BY	1	417
VIEW	SYS.null		1	416
Filter Predicates				
from$_subquery$_004.rowlimit_$$_rownumber<=1				
WINDOW		SORT PUSHED RANK	19	416
Filter Predicates				
ROW_NUMBER() OVER (ORDER BY COUNT(C.ID_HABITACION) DESC)<=1				
HASH		GROUP BY	19	416
TABLE ACCESS	CLIENTE	FULL	20	414
Filter Predicates				
AND				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(C.FECHAENTRADA),'w'))=1				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(C.FECHAENTRADA),'mm'))=1				
C.ID_HABITACION IS NOT NULL				

- **RFC12**



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			20	48182
HASH		UNIQUE	20	48182
HASH JOIN		RIGHT SEMI	1768390	48139
Access Predicates				
CLIENTE.NOMBRE=ITEM_1				
VIEW	SYS.VW_DTP_5204A638		20	2177
HASH		UNIQUE	20	2177
TABLE ACCESS	CLIENTE	FULL	88529	2174
Filter Predicates				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(CLIENTE.FECHAENTRADA),'qy'))=46				
MERGE JOIN		CARTESIAN	1768390	45958
NESTED LOOPS		SEMI	20	2504
VIEW	SYS.VW_DTP_85FA446E		20	2177
HASH		UNIQUE	20	2177
TABLE ACCESS	CLIENTE	FULL	87478	2174
Filter Predicates				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(CLIENTE.FECHAENTRADA),'qy'))=26				
TABLE ACCESS	CLIENTE	BY INDEX ROWID	88608	17
Filter Predicates				
TO_NUMBER(TO_CHAR(INTERNAL_FUNCTION(CLIENTE.FECHAENTRADA),'qy'))=16				

Solo hubo un cambio en la sentencia para 12a.

Plan de ejecución de consulta propuesto para los RFC:

Para sugerir un plan de ejecución, es primero necesario trabajar bajo 2 supuestos, el primero que las tablas a realizar join caben en memoria y segundo que las tablas no caben en memoria principal. Así pues, para cada uno de los dos casos se realizo el mismo procedimiento para cada consulta que requiera de un join, según el modelo planteado. El primero paso consiste en calcular el tamaño del archivo en memoria dados los índices propuestos, y comprar este tamaño con la memoria principal del sistema que corre las consultas para determinar si estas caben o no en dicha memoria. El calculo se realiza con la división del tamaño del bloque en el tamaño del archivo. Una vez se tiene este valor, se calcula el numero de bloques requeridos para almacenar la información, esto se calcula dividiendo el número total de tuplas por el calculo realizado anteriormente. Finalmente

se multiplica dicho número de bloques por el tamaño del bloque en bytes lo que resulta en la información del tamaño requerido para almacenar esta información. Asi pues los resultados son los siguientes:

Pesos en Bytes de las tablas creadas:

Cliente							
ID	Nombre	FechaEntrada	FechaSalida	Pago	ID_Hotel	ID_Habitacion	ID_Empleado
21	30	7	7	21	21	21	21

Habitacion														
ID	Tipo	Capacidad	ComedorCocina	Jacuzzi	Minibar	Television	Cafetera	Consumo	ID_Hotel	Reserva	InicioMantenimiento	FinMantenimiento	Ocupado	Id_Reserva
21	20	21	20	20	20	20	20	21	21	7	7	7	20	21

Servicio Adicional							
ID	Nombre	ID_Hotel	FechaUso	ID_Cliente	Costo	Reserva	Duracion
21	30	21	7	21	21	21	21

Datos del sistema						
ROWID	TB	S	R	BTT	EBT	MEM
20	2400	0.016	0.0083	0.0008	0.00084	16000000000

Datos calculados para cada tabla

Servicio Adicional tabla completa			
N	RpB	b	TA
2.00E+05	10	20000	4.80E+07

Servicio Adicional con indice en Nombre			
N	RpB	b	TA
2.00E+05	33	6061	1.45E+07

Habitacion tabla completa			
N	RpB	b	TA
3.00E+02	6	50	1.20E+05

Habitacion con indice en ID			
N	RpB	b	TA

Mateo Pedroza 201531957
David Acuña 201517879

3.00E+02	40	8	1.92E+04

Cliente tabla completa			
N	RpB	b	TA
1.10E+06	11	100000	2.40E+08

Cliente con indice en ID			
N	RpB	b	TA
1.10E+06	40	27500	6.60E+07

Plan de ejecución propuesto bajo los dos posibles escenarios junto a su costo pertinente:

RFC9:

Si cabe en memoria:

```
SELECT STATEMENT
NESTED LOOP JOIN
        FULL INDEX SCAN CLIENTE
         FULL INDEX SCAN  SERVICIOADICIONAL
                    FILTER PREDICATE FECHAUSO BETWEEN '22/Marzo/2018' AND '23/Abril/2018'
```

B(C)	23.1243
B(SA)	5.11554
B(C) + B(SA)	28.23984

Si no cabe en memoria

```
SELECT STATEMENT
HASH JOIN
        FULL INDEX SCAN CLIENTE
         FULL INDEX SCAN  SERVICIOADICIONAL
                    FILTER PREDICATE FECHAUSO BETWEEN '22/Marzo/2018' AND '23/Abril/2018'
```

B(C)	23.1243
B(SA)	5.11554
3*(B(C) + B(SA))	84.71952

RFC10:

Si cabe en memoria:

```
SELECT STATEMENT
NESTED LOOP JOIN
        FULL INDEX SCAN CLIENTE
         FULL INDEX SCAN  SERVICIOADICIONAL
                    FILTER PREDICATE FECHAUSO BETWEEN '22/Marzo/2018' AND '23/Abril/2018'
```

Mateo Pedroza 201531957
David Acuña 201517879

B(C)	23.1243
B(SA)	5.11554
B(C) + B(SA)	28.23984

Si no cabe en memoria

```
SELECT STATEMENT
HASH JOIN
      FULL INDEX SCAN CLIENTE
       FULL INDEX SCAN  SERVICIOADICIONAL
                  FILTER PREDICATE FECHAUSO BETWEEN '22/Marzo/2018' AND '23/Abril/2018'
```

B(C)	23.1243
B(SA)	5.11554
3*(B(C) + B(SA))	84.71952

RFC11:

Si cabe en memoria:

```
SELECT STATEMENT
NESTED LOOP JOIN
      FULL INDEX SCAN CLIENTE
                  FILTER PREDICATE to_number (to_char (fechaentrada, 'w')) = 1 AND to_number (to_char (fechaentrada, 'mm')) = 1
      FULL INDEX SCAN HABITACION
```

B(C)	23.1243
B(H)	0.03102
B(C) + B(H)	23.15532

Si no cabe en memoria

```
SELECT STATEMENT
HASH LOOP JOIN
      FULL INDEX SCAN CLIENTE
                  FILTER PREDICATE to_number (to_char (fechaentrada, 'w')) = 1 AND to_number (to_char (fechaentrada, 'mm')) = 1
      FULL INDEX SCAN HABITACION
```

B(C)	23.1243
B(H)	0.03102
3*(B(C) + B(H))	69.46596

Mateo Pedroza 201531957
David Acuña 201517879

RFC12:

Si cabe en memoria:

```
SELECT STATEMENT
NESTED LOOP JOIN
      FULL INDEX SCAN CLIENTE
            FILTER PREDICATE to_number (to_char (cliente.fechaentrada, 'qy')) = 16
      FULL INDEX SCAN CLIENTE
            FILTER PREDICATE to_number (to_char (cliente.fechaentrada, 'qy')) = 26
      FULL INDEX SCAN CLIENTE
            FILTER PREDICATE to_number (to_char (cliente.fechaentrada, 'qy')) = 36
       FULL INDEX SCAN CLIENTE
            FILTER PREDICATE to_number (to_char (cliente.fechaentrada, 'qy')) = 46
```

```
SELECT STATEMENT
NESTED LOOP JOIN
      FULL INDEX SCAN CLIENTE

      FULL INDEX SCAN SERVICIOADICIONAL
            FILTER PREDICATE COSTO > 400000
```

```
SELECT STATEMENT
NESTED LOOP JOIN
      FULL INDEX SCAN CLIENTE

      FULL INDEX SCAN SERVICIOADICIONAL
            FILTER PREDICATE DURACION > 3
```

B(C)	23.1243
B(C) + B(C)+ B(C) + B(C)	92.4972

B(C)	23.1243
B(SA)	5.11554
B(C) + B(SA)	28.23984

B(C)	23.1243
B(SA)	5.11554
B(C) + B(SA)	28.23984

Mateo Pedroza 201531957
David Acuña 201517879

Si no cabe en memoria

```
SELECT STATEMENT
HASH JOIN
      FULL INDEX SCAN CLIENTE
                  FILTER PREDICATE to_number (to_char (cliente.fechaentrada, 'qy')) = 16
      FULL INDEX SCAN CLIENTE
                  FILTER PREDICATE to_number (to_char (cliente.fechaentrada, 'qy')) = 26
      FULL INDEX SCAN CLIENTE
                  FILTER PREDICATE to_number (to_char (cliente.fechaentrada, 'qy')) = 36
       FULL INDEX SCAN CLIENTE
                  FILTER PREDICATE to_number (to_char (cliente.fechaentrada, 'qy')) = 46
```

```
SELECT STATEMENT
MERGE JOIN
      FULL INDEX SCAN CLIENTE

      FULL INDEX SCAN SERVICIOADICIONAL
                  FILTER PREDICATE COSTO > 400000
```

```
SELECT STATEMENT
MERGE JOIN
      FULL INDEX SCAN CLIENTE

      FULL INDEX SCAN SERVICIOADICIONAL
                  FILTER PREDICATE DURACION > 3
```

B(C)	23.1243
3*(B(C) + B(C)+ B(C) + B(C))	277.4916

B(C)	23.1243
B(SA)	5.11554
3*(B(C) + B(SA))	84.71952

B(C)	23.1243
B(SA)	5.11554
3*(B(C) + B(SA))	84.71952

Proceso de carga de datos:

Los datos referentes a los productos, servicios y habitaciones fueron cargados con la herramienta “Mokaroo” la cual ofrece un servicio de generación de datos que no superen las 1000 tuplas. La herramienta ofrece campos incluyendo nombres humanos, rangos, fechas y todo lo necesario para los datos de interés y permite exportar un archivo “.sql” que se cargó directamente a la base de datos.

Para las tablas de cliente y servicioAdicional, se creó un programa en eclipse que llena con valores aleatorios todos los campos de estas tablas, y guarda el script en un archivo

".sql". La fecha de entrada siempre es anterior a la fecha de salida, y los índices siempre van aumentando de uno en uno.

Análisis del proceso de optimización y el modelo de ejecución de consultas:

Para el proceso que maneja Oracle, él hace uso de índices que minimizan al máximo la carga de información a memoria principal. Como nuestras consultas son en rangos, Oracle puede organizar en rangos y no cargar toda la información a memoria principal al momento de hacer el join de las tablas. Adicionalmente, los índices ya están cargados en memoria principal, por lo que se ahorra aún más tiempo.

A diferencia de este proceso, cuando la aplicación trae los datos a memoria principal, los trae todos sin organizar, haciendo más pesado este proceso. Gracias a ello, no puede hacer uso de una 'función domino' y termina revisando cada una de las tuplas. Esto aumenta considerablemente la complejidad temporal del proceso, haciendo que sea un proceso de 'fuerza bruta'. Adicionalmente, la aplicación debe crear nuevas estructuras para almacenar los datos, aumentando ligeramente su complejidad temporal, pero sí considerablemente la disponibilidad de la memoria principal.

Resultados logrados:

Al finalizar la iteración, se pude demostrar que cumplimos con los requerimientos funcionales de consulta que propone el proyecto, se logro cargar un total estimado de 1 500 000 datos a la base de datos y los tiempos de consulta duran el tiempo indicado.

Resultados no logrados:

No se logró implementar exitosamente la conexión del proyecto en eclipse en algunos de los requerimientos funcionales de la iteración anterior. Se tuvo problemas con la asignación de la clase en el modelo y los resultado obtenidos en las consultas.