

(1 %) Análisis 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.

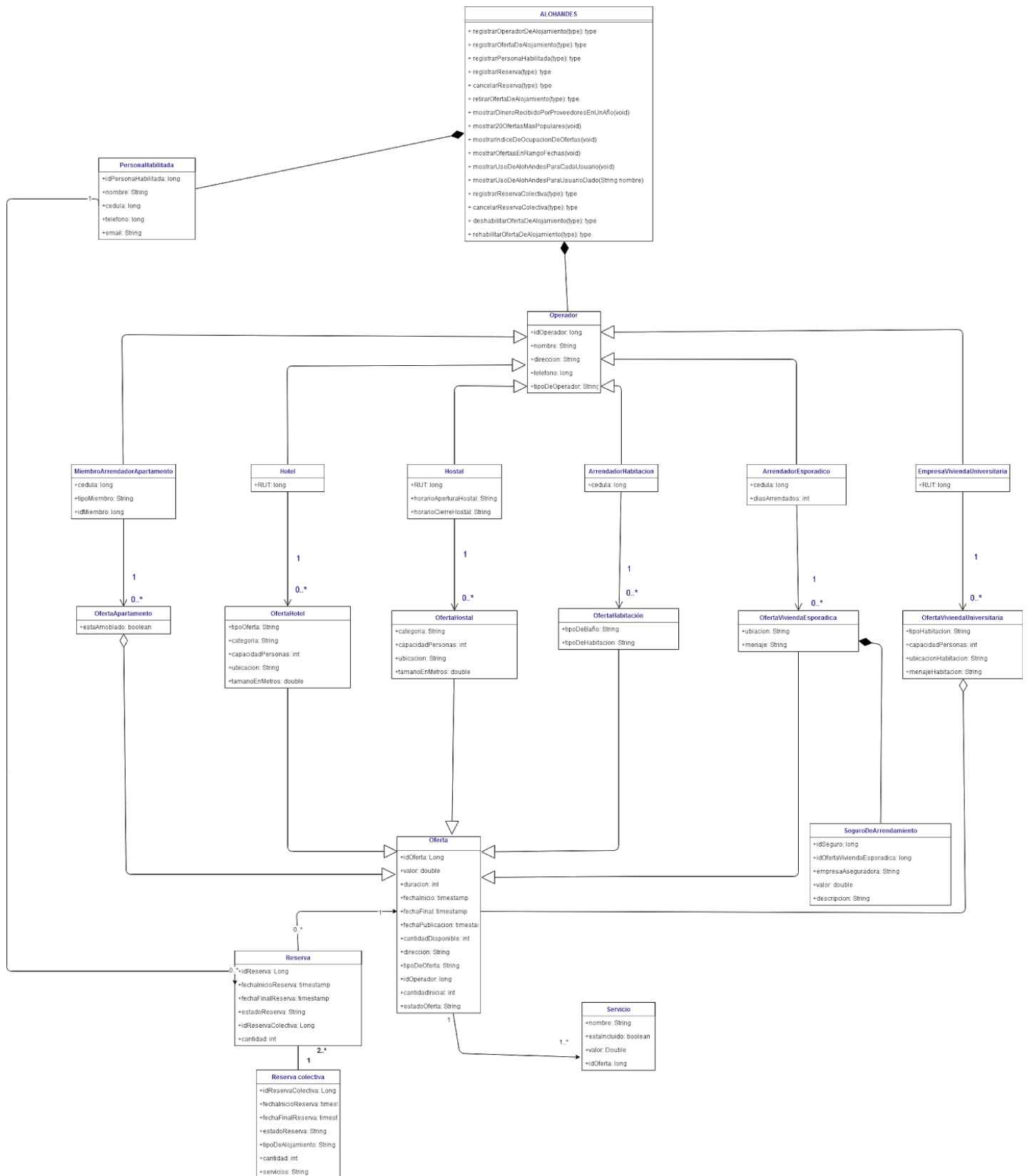


Figura 1 (Modelo conceptual)

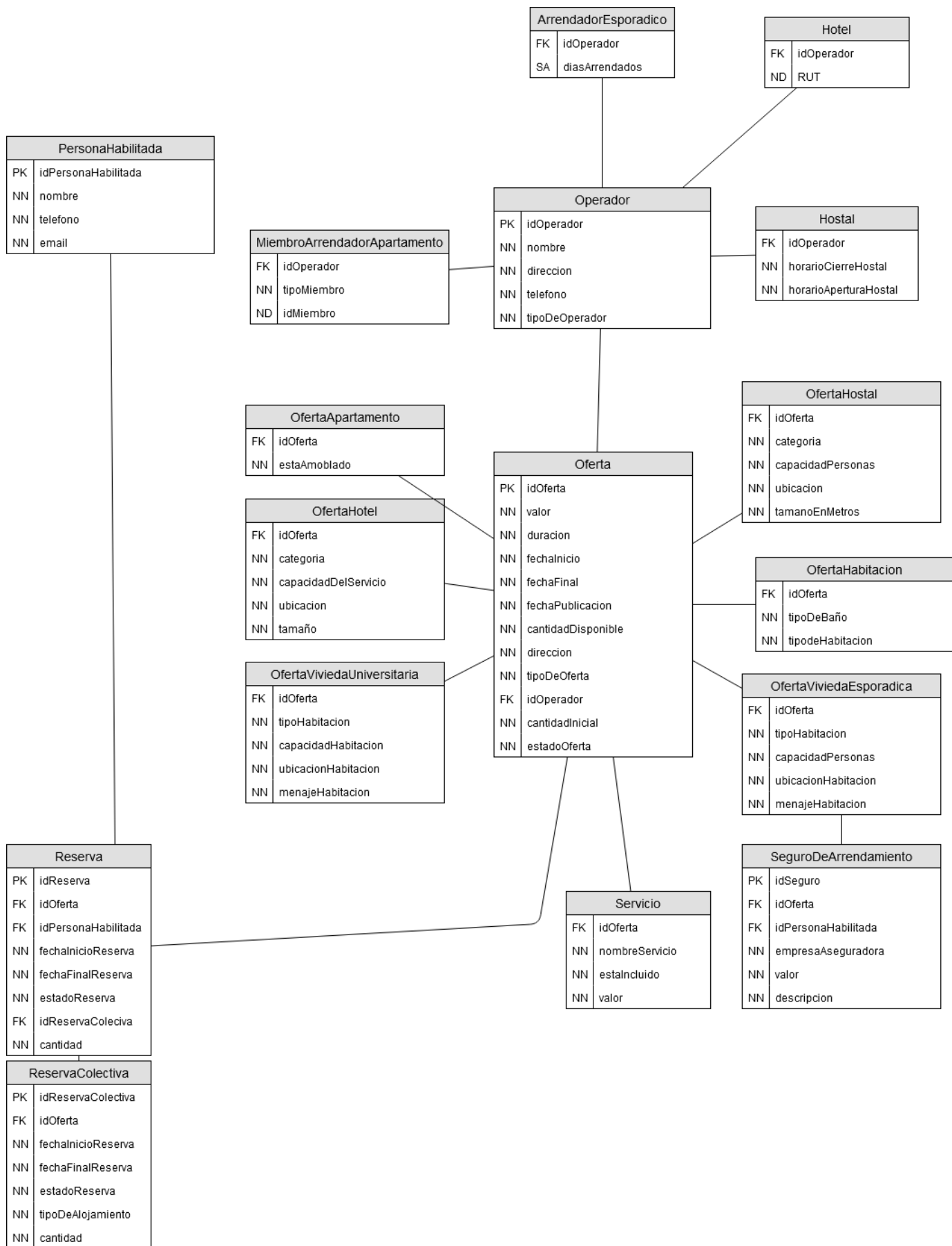


Figura 2 (Modelo relacional)

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.

(64 %) Diseño de la aplicación

➤ (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 requerimiento número diez 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 requerimiento número doce y número trece no fue necesario añadir nada mas ya que con los atributos anteriores existentes fue suficiente.

(63 %) **Diseño físico.** 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.

❖ (19%) Documente su diseño físico

- 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)

//AQUÍ JUSTIFICAS ASI BIEN PRRON EN FULL HD

//AQUÍ JUSTIFICAS ASI BIEN PRRON EN FULL HD

//AQUÍ JUSTIFICAS ASI BIEN PRRON EN FULL HD

//AQUÍ JUSTIFICAS ASI BIEN PRRON EN FULL HD

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

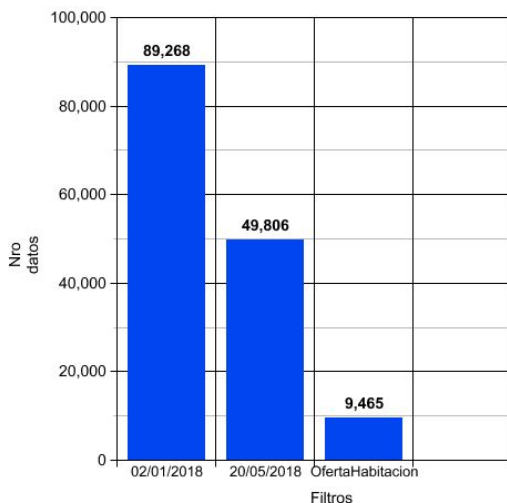
```
SELECT DISTINCT OFERTA.TIPODEOFERTA,
PERSONAHABILITADA.IDPERSONAHABILITADA,
PERSONAHABILITADA.NOMBRE,
PERSONAHABILITADA.EMAIL, PERSONAHABILITADA.TELEFONO
FROM
```

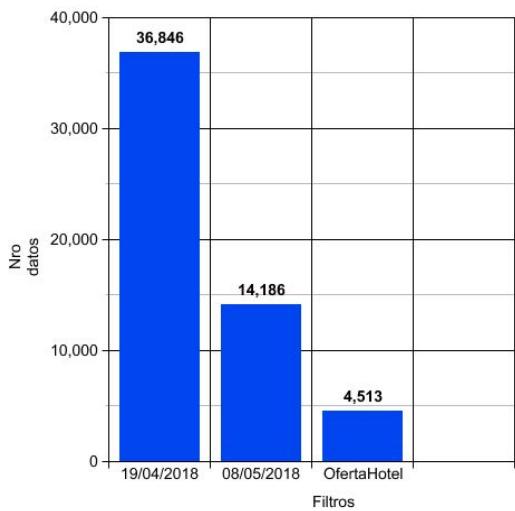
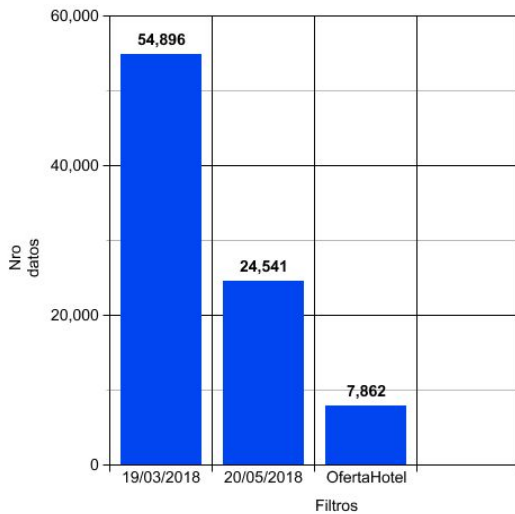
```

((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.IDPERSONAHABI
    LITADA, 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.

Prueba 1: 02/01/2018 -20/05/2018 -OfertaHabitacion

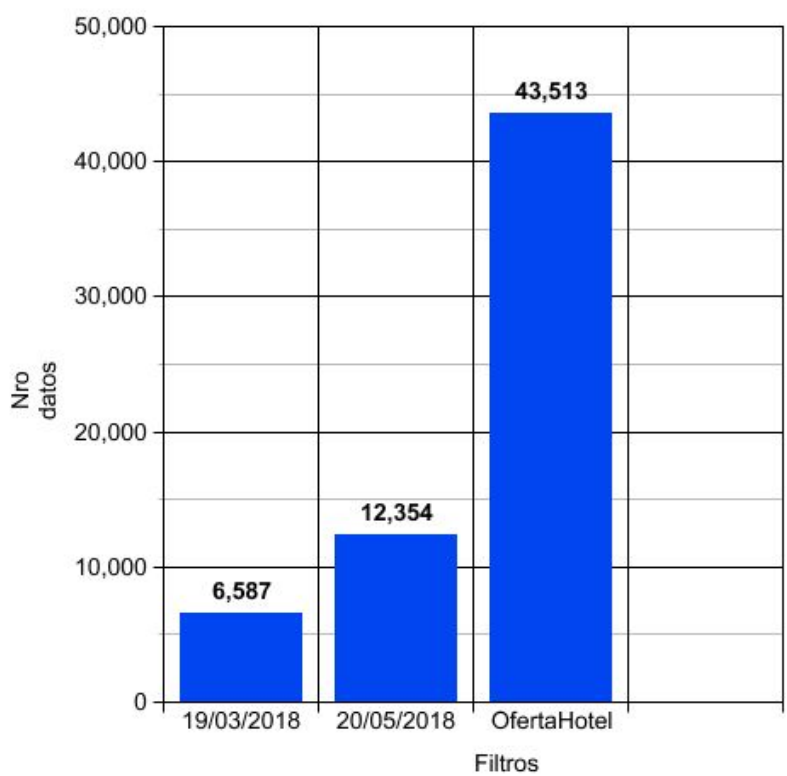
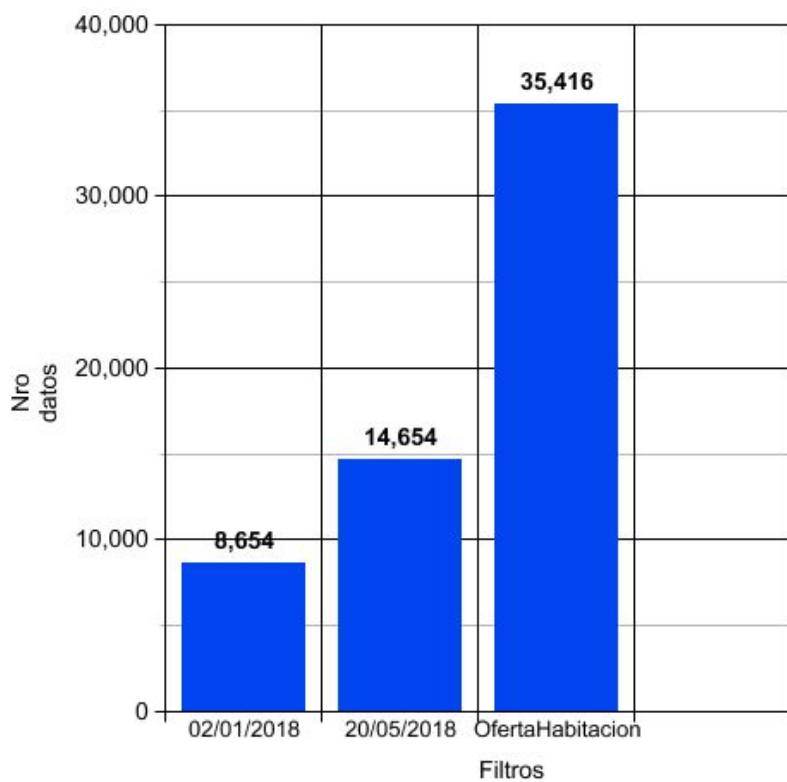
Prueba 2: 02/01/2018- 20/05/2018 -OfertaHabitacion

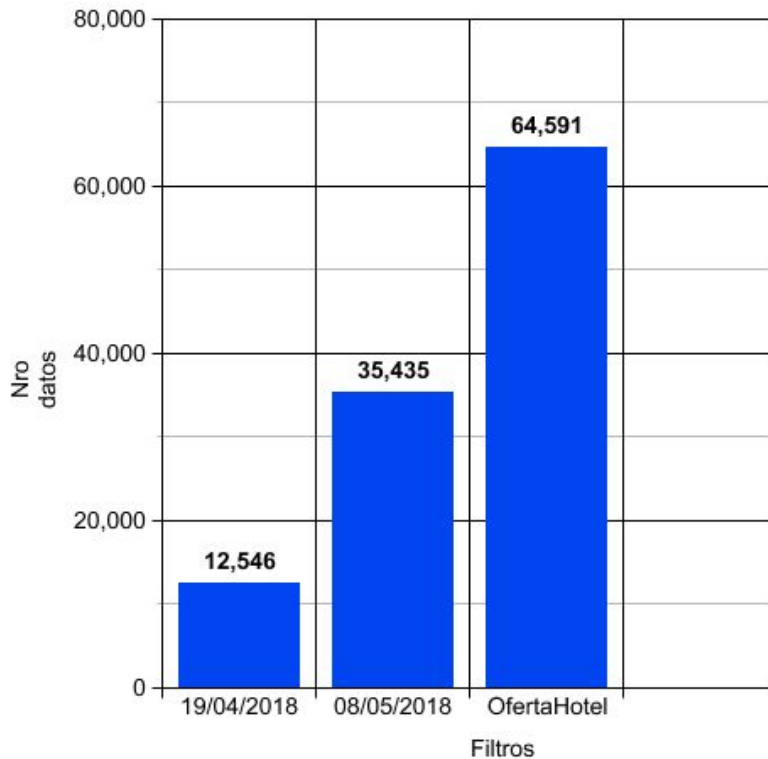
Prueba 3: 19/04/2018 -08/05/2018 -OfertaHotel

-
- 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.

- 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.
- 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.

Prueba 1: 02/01/2018 -20/05/2018 -OfertaHabitacion

Prueba 2: 02/01/2018- 20/05/2018 -OfertaHabitacion

Prueba 3: 19/04/2018 -08/05/2018 -OfertaHotel

- 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.
- 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.
- 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 SQLDeveloper.
 - 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.
- 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);

```

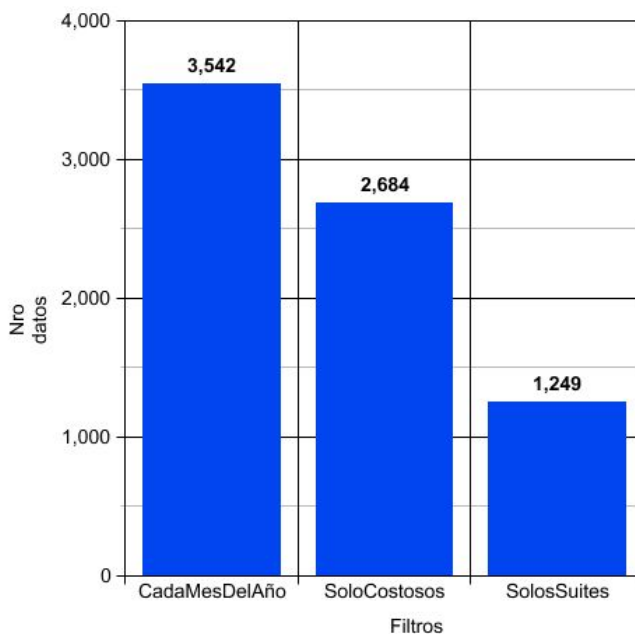
```

//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 así se obtuvieron diferentes datos**
- Planes de consulta obtenidos en Oracle para la ejecución del requerimiento. Para ello,

documento con una foto de pantalla los planes de consulta obtenidos en SQLDeveloper.

- 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.

- **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.