# Sistemas transaccionales – Caso de estudio Rotond Andes

Iteración 4

1-Cambios realizados de las iteraciones pasadas para cumplir con los objetivos de la nueva. Plasmar cambios tanto explícitamente como en los diagramas UML y relacional.

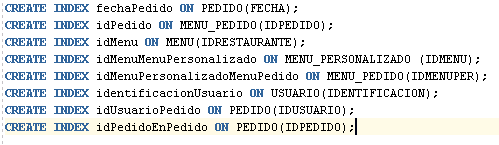
2-**Diseño Físico**

-> Justificar la selección de índices para cada uno de los requerimientos funcionales, indicar cuál es el tipo de índice utilizado (B+,Hash,Primario o secundario) Agregar costos.

**Requerimiento 9:** Para el requerimiento nueve se utilizaron los siguientes índices:

* Índice en la fecha de la tabla pedido.
* Índice en el id de un Pedido de la tabla menú pedido.\*
* Índice en el id de restaurante de la tabla menú.
* Índice en el id de menú de la tabla menú personalizado.
* Índice en el id de menú personalizado en la tabla menú pedido.
* Índice la identificación del usuario.\*
* Índice en el id de usuario de la tabla pedido.
* Índice en el id de pedido de la tabla pedido.\*

**Nota:** Índices marcados con \* son creados automáticamente por SQLdeveloper por formar parte de la llave primaria.



-> Para los índices creados de forma automática por Oracle: Incluya pantallazo con la información generada por Oracle sobre los índices existentes.

-> Analizar por qué fueron creados los índices por Oracle y si ayudan al rendimiento de los requerimientos funcionales.

**Documentación análisis para cada requerimiento**

-> Documentación del escenario de pruebas

-> Sentencia SQL que responde al requerimiento

-> Distribución de los datos con respecto a los parámetros de entrada.

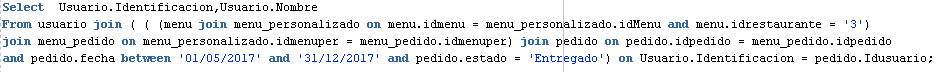
-> Planes de consulta obtenidos por Oracle para la ejecución del requerimiento. Agregar pantallazo del plan.

-> Tiempos obtenidos con la ejecución de cada uno de los planes. Desde el núcleo.

**Requerimiento 9:**

## Escenario de prueba:

Se desea consultar la identificación y el nombre de los usuarios que al menos hicieron un pedido (que fue entregado) en el restaurante con id 3 en el rango de fechas '01/05/2017' y '31/12/2017’. Dicha consulta se realiza con la siguiente sentencia SQL:



Que retorna como respuesta:



Para demostrar el funcionamiento de la anterior consulta veamos lo siguiente:

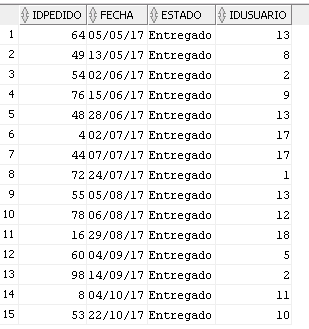
1. Se tienen 30 Usuarios registrados en la base de datos.





1. Vamos a consultar los pedidos que se han realizado en el rango de fechas establecido en la sentencia.





Se puede observar que se realizaron 15 pedidos (en estado entregado) dentro del rango de fechas '01/05/2017' y '31/12/2017’.

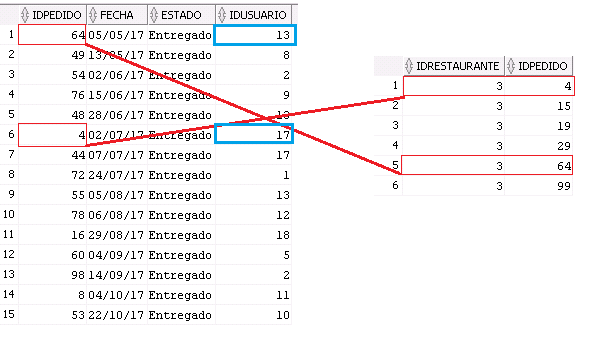
1. Ahora, debemos encontrar los pedidos que fueron realizados en el restaurante con id 3. Para esto se ejecuta la siguiente sentencia SQL:





Se pueden observar los ID de los 6 pedidos que se han realizado en el restaurante con id 3.

1. Entonces, vamos a comparar los resultados de la consulta (2) con los de la (3):

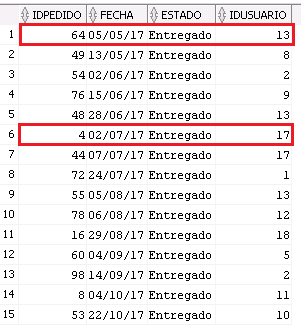


Se puede evidenciar que:

El pedido con id 64 fue realizado por el usuario con id 13 en el restaurante con id 3.

El pedido con id 4 fue realizado por el usuario con id 17 en el restaurante con id 3.

1. Finalmente, vamos a verificar que estos pedidos se hayan realizado en el rango de fechas establecido. Para ello se observa la tabla (2).

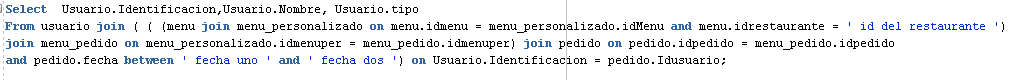


Se observa que estos pedidos fueron realizados entre las fechas '01/05/2017' y '31/12/2017’.

Entonces, se puede concluir que los usuarios con id 13 y 17, nombres Oberon y Beltran, hicieron al menos un pedido en el restaurante con id 3 en el rango de fechas '01/05/2017' y '31/12/2017’. Que coincide con la respuesta de la consulta planteada para la solución.



Sentencia SQL usada: La sentencia usada para resolver este requerimiento es la siguiente:



Donde se tienen tres parámetros que son: el id del restaurante al cual se desea consultar el consumo, y las dos fechas que representan el rango por el que se desea hacer la consulta.

**Análisis de eficiencia**

-> Escenarios de datos que permitan validar diferentes selectividades.

-> Seleccionar un escenario de análisis y diseñe un plan de ejecución de consulta propuesto por el grupo.

-> Comparar el plan propuesto por el grupo y por Oracle.

3- Construcción de la aplicación y análisis de resultados.

* Diseño del escenario de pruebas de eficiencia. Cargar los datos suficientes para realizar un análisis de eficiencia. Estado de la base de datos.
* Documentar como fue el proceso de carga de datos, como se realizo como se logró el volumen de datos esperado.
* Ajustes servicios REST para los nuevos requerimientos.
* Cambios y desarrollo de las transacciones en RotondAndesMaster.
* Cambios en los Dao.
* Análisis del proceso de optimización y el modelo de ejecución de consultas. Analice la diferencia entre la ejecución de consultas delegada al manejador de bases de datos como Oracle 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.

