Universidad de Los Andes.  
Departamento de Ingeniería de Sistemas y Computación.  
Sistemas Transaccionales.  
Iteración IV “RotondAndes”

Christian Chavarro Espejo 201613724  
Juan Sanmiguel Mateus 201617603

1 (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.

No fueron realizados cambios en el modelo de mundo o esquema de las tablas, ellas continúan como en las iteraciones anteriores.

2 (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. Realice los cambios necesarios en su modelo relacional para respetar las reglas de negocio y asegurar la calidad del mismo. Tenga en cuenta los comentarios recibidos en la sustentación de los talleres anteriores. Documente el diseño y las decisiones tomadas para crear los elementos de la base de datos que da el respaldo de persistencia a la aplicación, a partir del modelo conceptual.

* Sea claro en mencionar explícitamente los cambios relevantes entre su diseño entregado en iteraciones anteriores y este

No fueron realizados cambios en el modelo de mundo o esquema de las tablas, ellas continúan como en las iteraciones anteriores.

El aumento en el tamaño de los datos, comparado con un mal manejo de los índices, podría afectar el desempeño de la aplicación.

(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) y tenga en cuenta el costo de almacenamiento y mantenimiento asociado a los índices
* Según su modelo de datos, para los índices creados de forma automática por Oracle

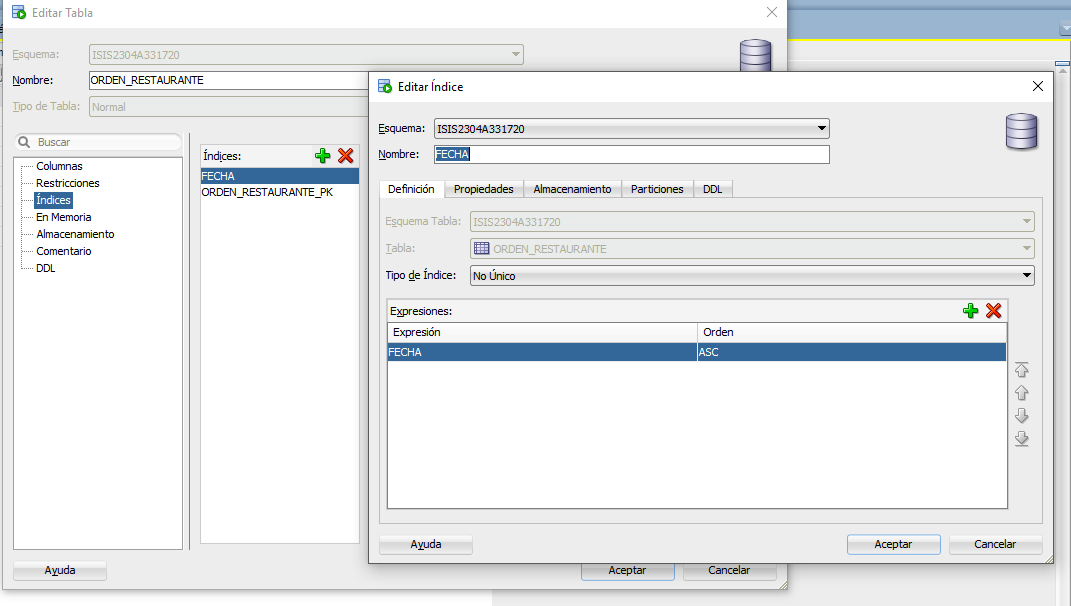
Incluya una foto de pantalla con la información generada por Oracle asociada a los índices existentes.

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

Índices

1. fecha en OrdenRestaurante. “create index fecha on orden\_restaurante(fecha)”

Este índice, a la hora de realizar el RFC11, disminuye el costo asociado al mismo y facilita el ordenamiento de los mismos por fecha. Además, en los RFC9-10 también es útil.

  
Imagen x: información asociada al índice.

En cada requerimiento, este índice es utilizado.

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

• Documentación del escenario de pruebas

* Sentencias SQL que responden el requerimiento y que fueron analizadas.
* 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 SQLDevelopper.
* 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.

NOTA: El porcentaje de evaluación correspondiente a cada uno de los requerimientos solicitados es proporcional al número de los requerimientos.

NOTA: La nota para cada uno de los requerimientos depende de los escenarios de ejecución definidos.

RFC9

RFC10

RFC11

Sentencia SQL:

SELECT \* FROM (SELECT PRODUCTO.ID,PRODUCTO.NOMBRE\_RESTAURANTE, ORDEN\_RESTAURANTE.FECHA, COUNT(\*) AS CONTEO FROM PRODUCTO LEFT JOIN (MENU LEFT JOIN ORDEN\_RESTAURANTE ON MENU.ID = ORDEN\_RESTAURANTE.ID\_MENU) ON (MENU.POSTRE=PRODUCTO.ID OR MENU.PLATOFUERTE = PRODUCTO.ID OR MENU.ACOMPANIAMIENTO = PRODUCTO.ID OR MENU.BEBIDA=PRODUCTO.ID OR MENU.ENTRADA=PRODUCTO.ID) WHERE ORDEN\_RESTAURANTE.FECHA >='10-12-2018' AND ORDEN\_RESTAURANTE.FECHA <='17-12-2018' GROUP BY PRODUCTO.ID,PRODUCTO.NOMBRE\_RESTAURANTE, ORDEN\_RESTAURANTE.FECHA ORDER BY ORDEN\_RESTAURANTE.FECHA)  
WHERE CONTEO>=3 OR (CONTEO<2 AND CONTEO >0);

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.

De acuerdo al rango de fechas dado por parámetro, el tamaño de la respuesta y el tiempo de ejecución de la sentencia puede variar. El valor más alto obtenido es de 0,8 segundos, los valores promedios, rondan los 0,4.

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

GET /administradorrotonda/consultarFuncionamiento/fechaMin/fechaMax/

GET [/administradorrotonda/consultarFuncionamiento/9-12-18/29-12-18/](http://localhost:8080/RotondAndes/rest/administradorrotonda/consultarFuncionamiento/9-12-18/29-12-18/)

Estos, son la url principal junto y el verbo para pruebas de Postman.

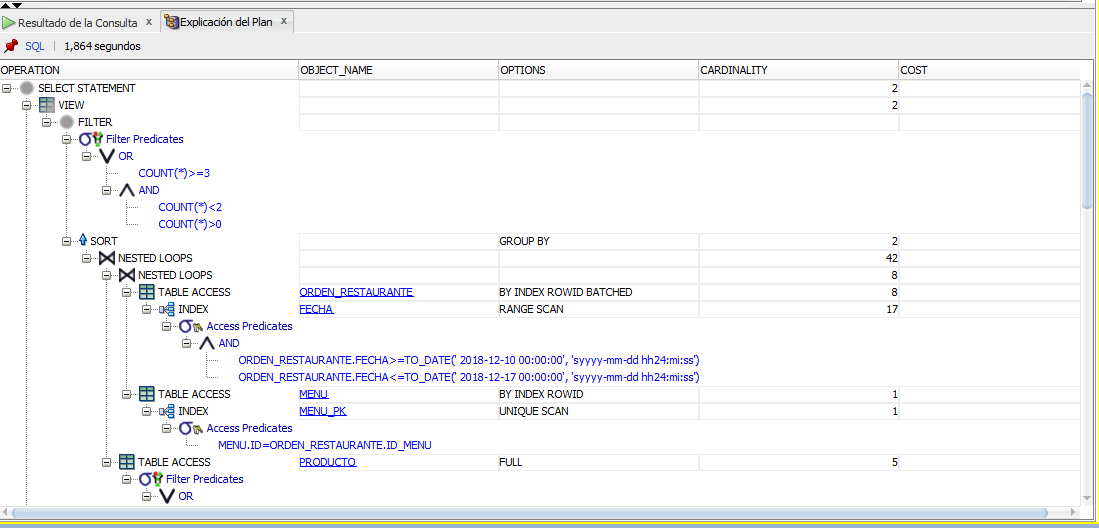
Algunos parámetros validos son:

/9-12-18/29-12-18/ - /9-2-18/29-12-18/- /9-12-10/29-12-18/- /9-12-13/29-12-16/

Donde básicamente, se modifica la fecha de inicio o final del rango de búsqueda.

Como se describió anteriormente, de acuerdo al rango de fechas, varían las respuestas, estas, vienen ordenadas por fecha descendentemente.

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.



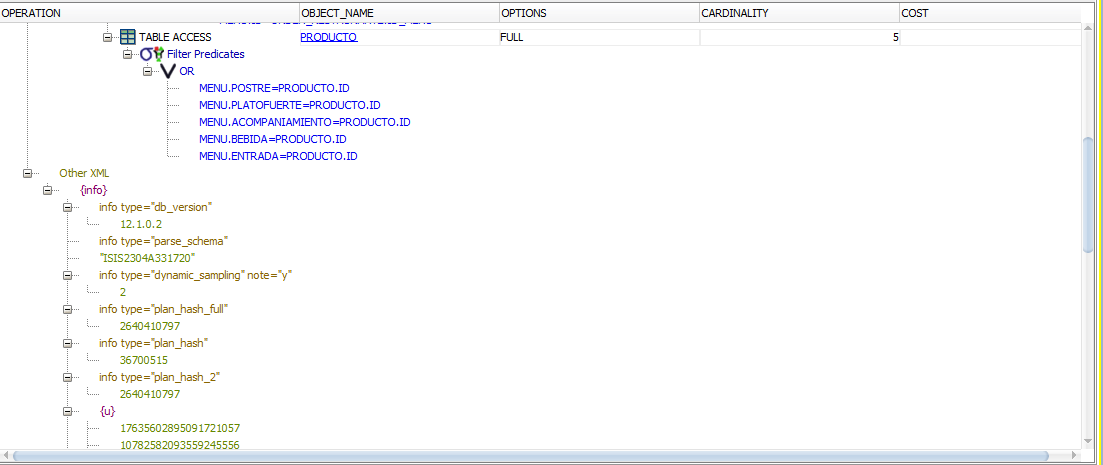
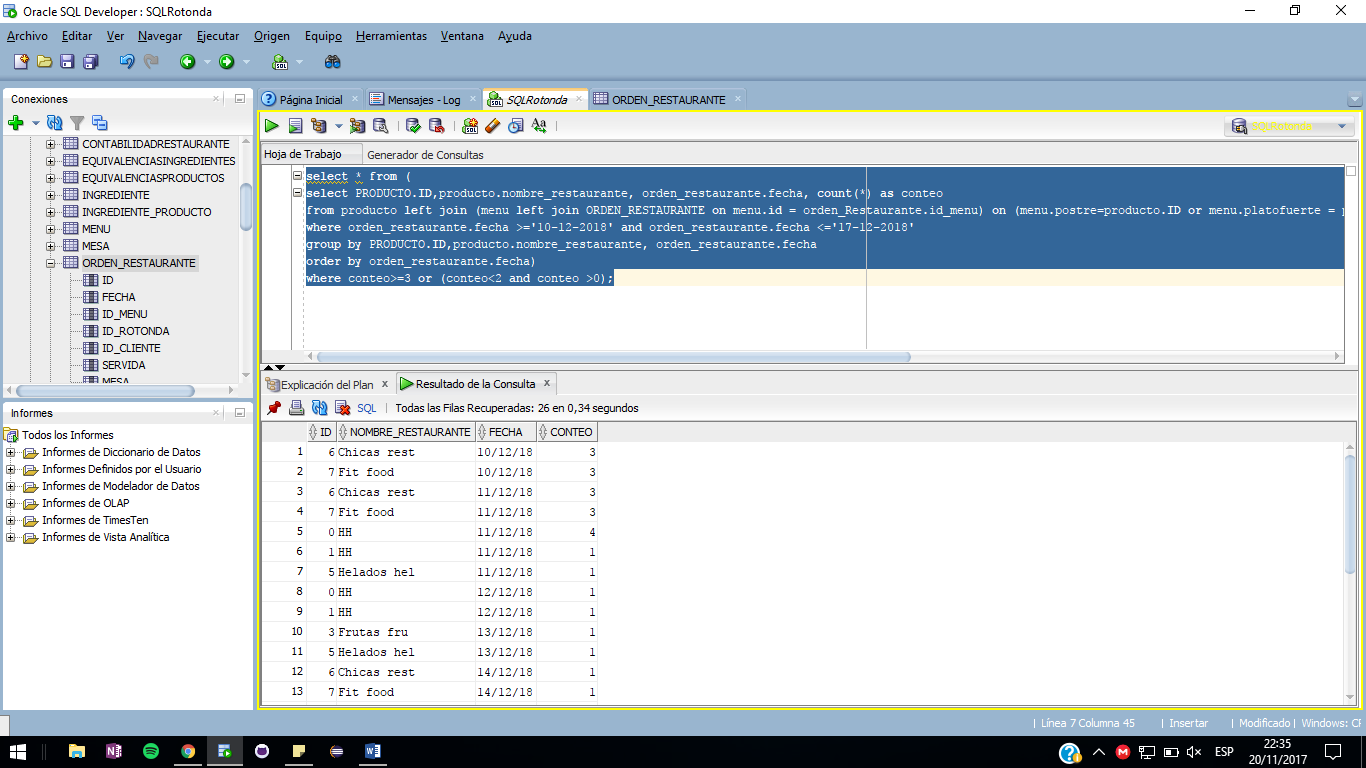
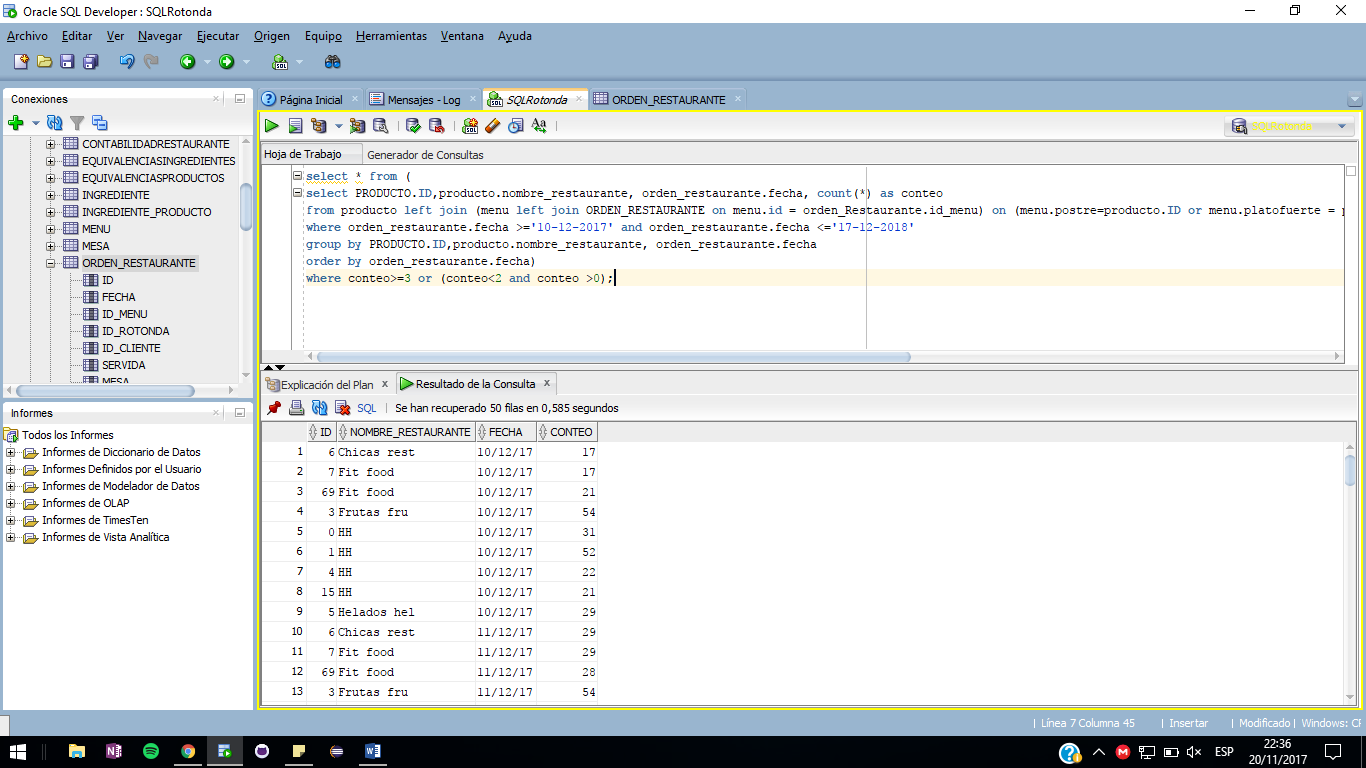


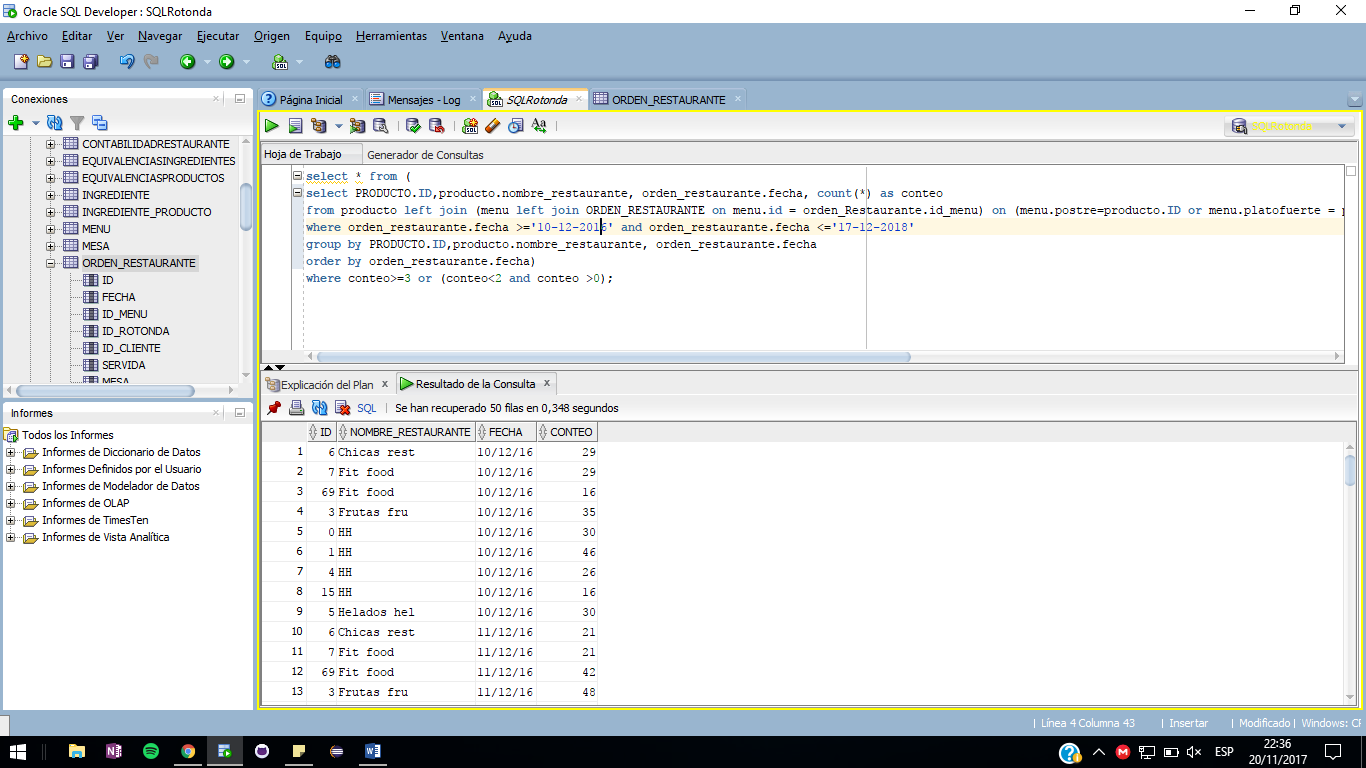
Imagen x: plan de trabajo para el RFC11.

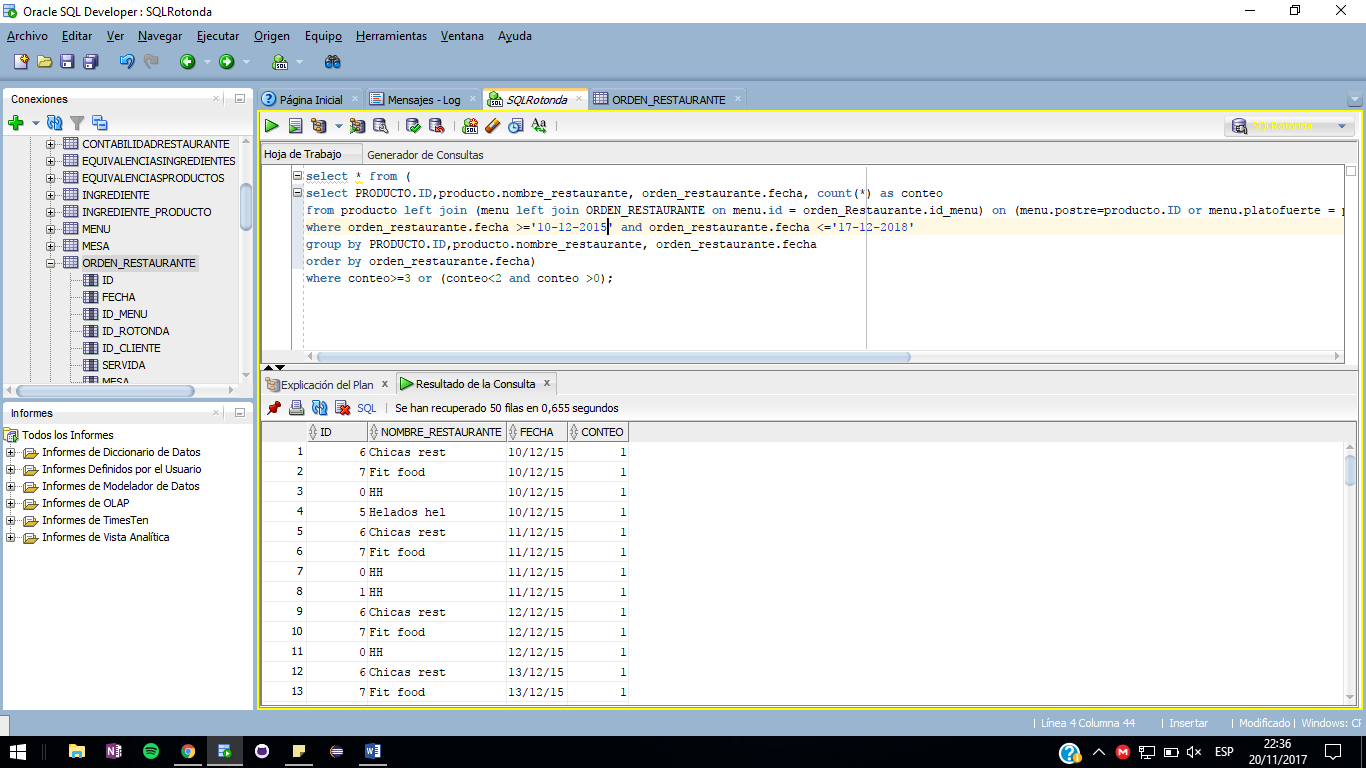
Utiliza los índices creados para él.

Tiempos RFC11.







  
Imágenes x-x: tiempos de ejecución del RFC11 para diferentes rangos de fechas. En todos, el tiempo está dentro de 0,8 segundos.

RFC12

Sentencia SQL:

SELECT CLIENTE.CEDULA,CLIENTE.NOMBRE,CLIENTE.CORREO, CLIENTE.ID\_ROTONDA, CAST ((COUNT(ORDEN\_RESTAURANTE.FECHA)/52) AS INTEGER ) AS NUMEROORDENES, MAX(PRODUCTO.PRECIO) AS PRECIOMINIMO FROM CLIENTE LEFT JOIN   
(ORDEN\_RESTAURANTE LEFT JOIN (MENU RIGHT JOIN PRODUCTO ON MENU.PLATOFUERTE = PRODUCTO.ID) ON ORDEN\_RESTAURANTE.ID\_MENU = MENU.ID) ON CLIENTE.CEDULA = ORDEN\_RESTAURANTE.ID\_CLIENTE GROUP BY CLIENTE.CEDULA, CLIENTE.NOMBRE, CLIENTE.CORREO, CLIENTE.ID\_ROTONDA ORDER BY NUMEROORDENES DESC, PRECIOMINIMO DESC;

Distribución de los datos con respecto a los parámetros de entrada utilizados en el requerimiento funcional.

Este requerimiento funcional, no tiene parámetros en entrada, sus variaciones se dan de acuerdo al volumen de datos. No posee valores de parámetros, no pueden darse ejemplos.

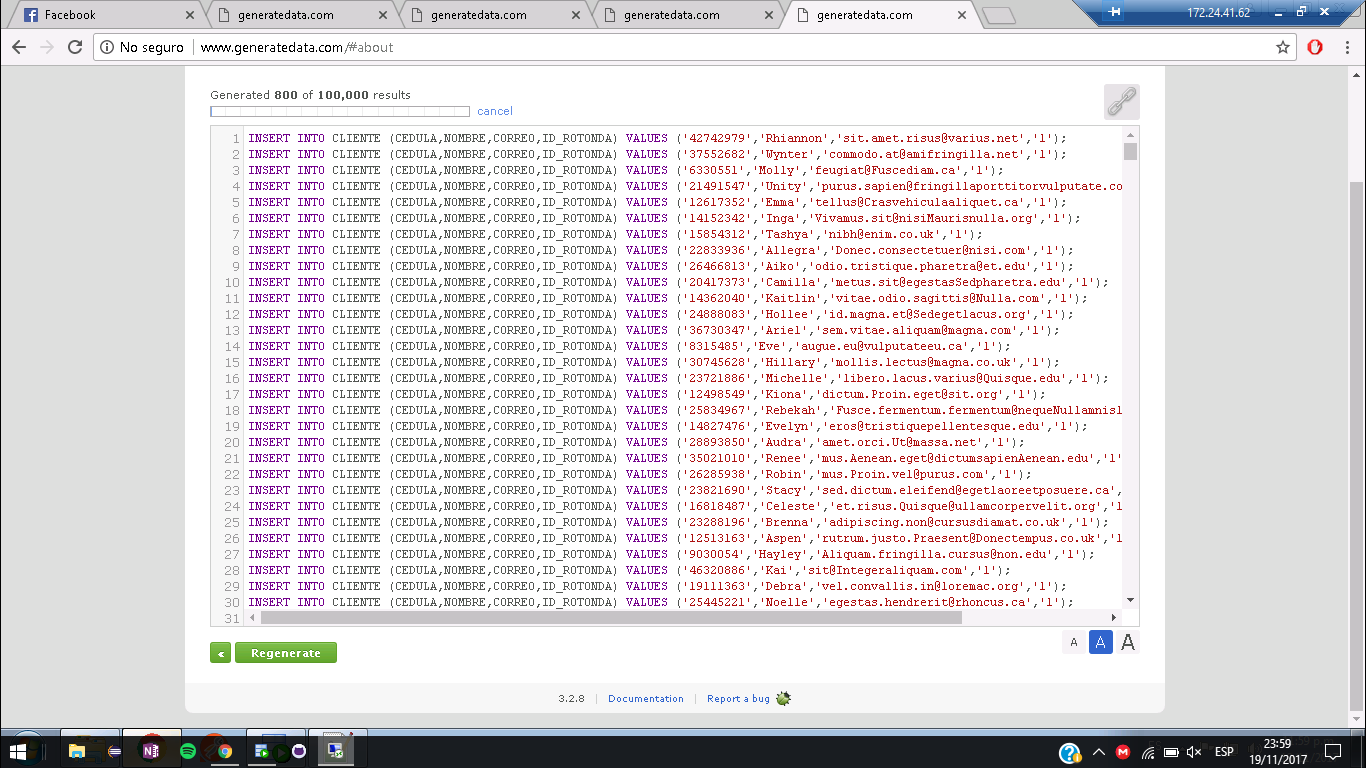
**Análisis de eficiencia**

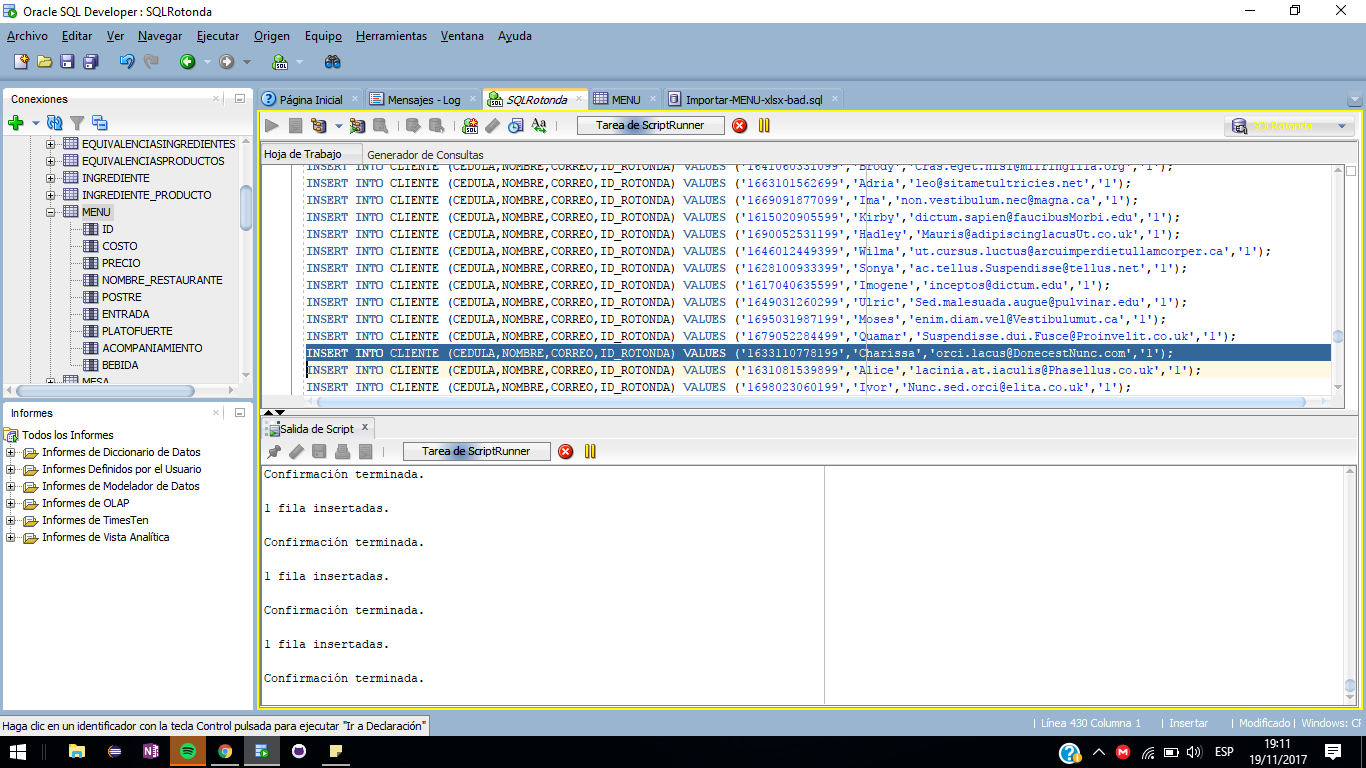
* Establezca escenarios de datos que le permitan validar diferentes selectividades.
* Para cada requerimiento funcional, seleccione un escenario de análisis y diseñe el plan de ejecución de consulta propuesto por el grupo, de acuerdo con su conocimiento del modelo y de la aplicación.
* Compare y analice el plan de ejecución propuesto por usted y el obtenido en Oracle.

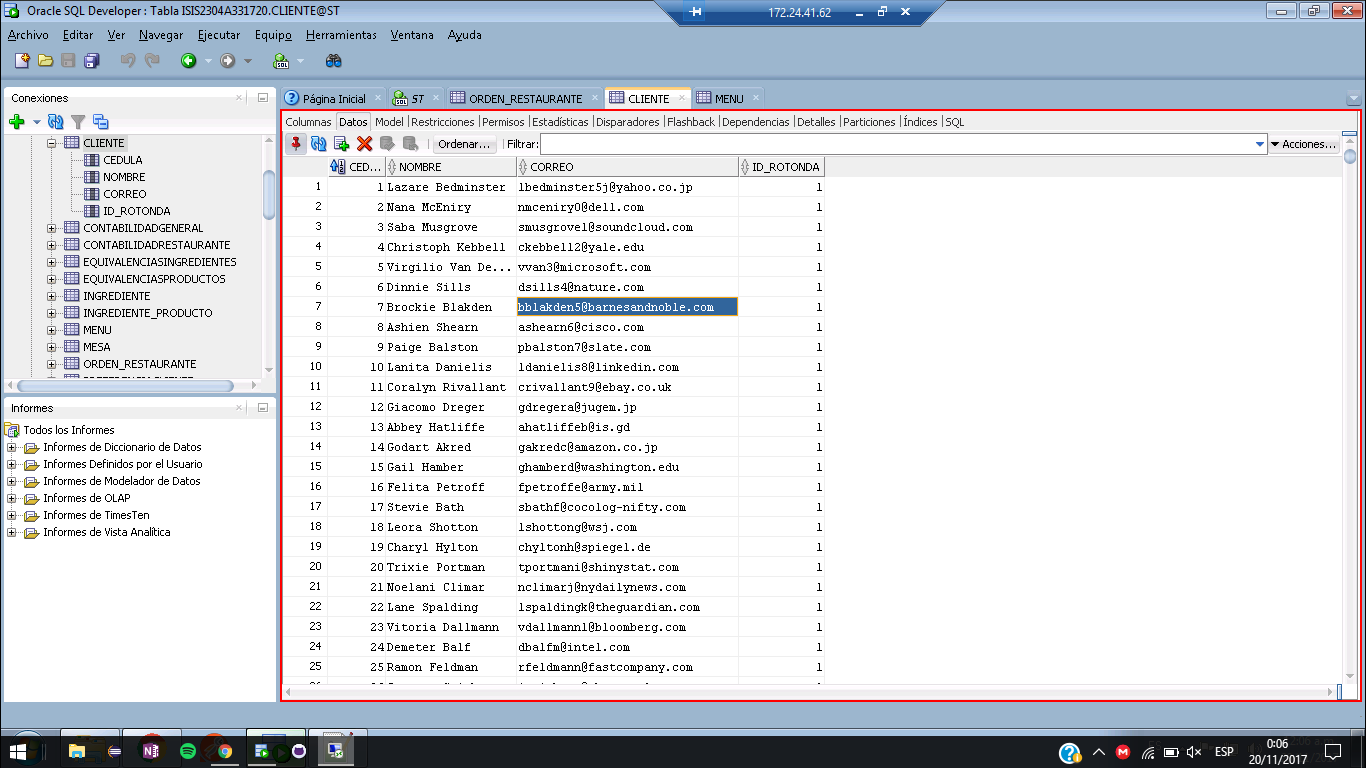
3 (35%) Construcción de la aplicación y análisis de resultados

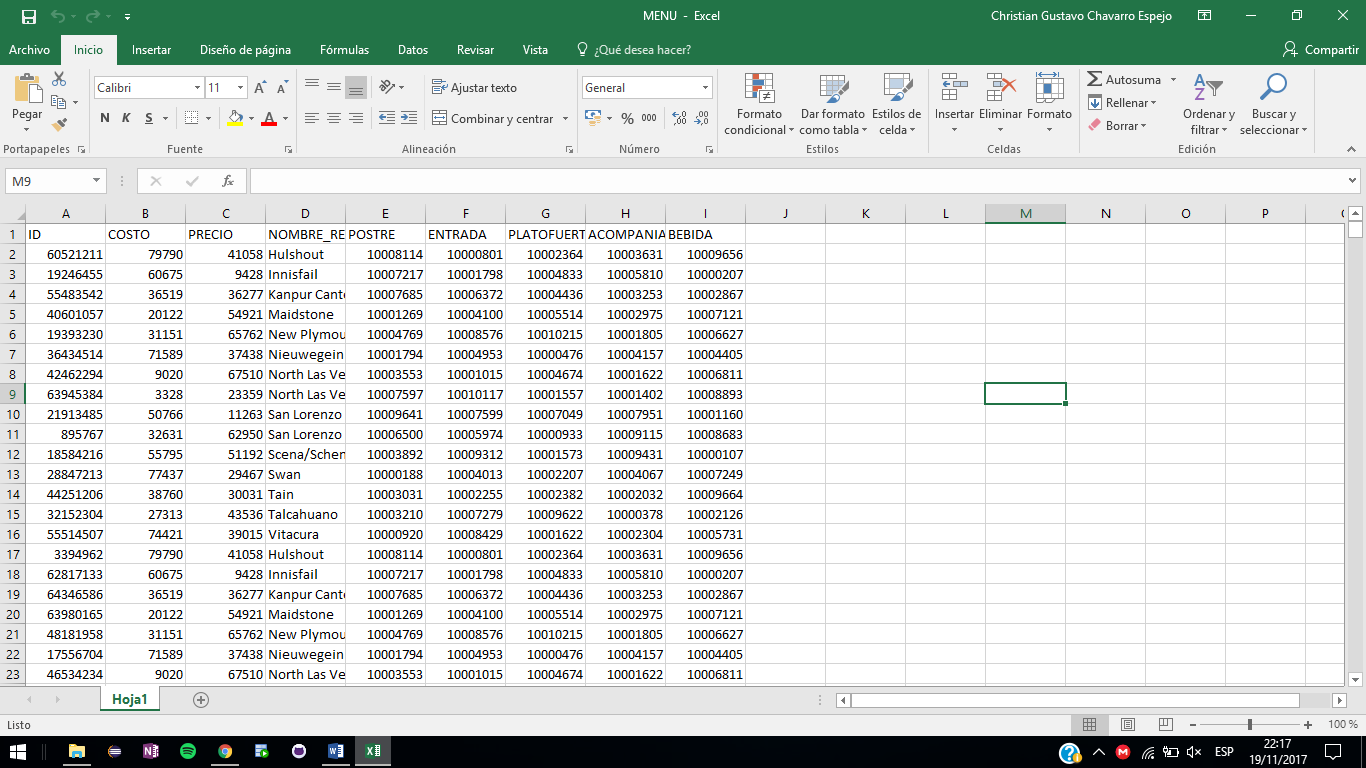
* Ajuste las tablas creadas en Oracle de acuerdo a las decisiones del punto anterior
* Diseño del escenario de pruebas de eficiencia. Cargue de datos necesarios para hacer el estudio de eficiencia de la aplicación.
* Diseñe los datos que le permitan verificar adecuadamente las reglas de negocio. Note que es importante generar adecuadamente los datos y para esta iteración lo es también el obtener un número muy grande de ellos. Se debe generar un volumen de datos tal que algunas tablas no quepan en la memoria principal de la máquina. El no cumplimiento de este requisito implica la invalidez de este componente de la evaluación
* Puede escribir un programa de generación automática de datos acorde al diseño establecido para los mismos.
* Para la población de las tablas utilice herramientas de carga masiva como SQLLoader o las disponibles en SQLDeveloper. Consulte el tutorial disponible en la wiki del curso sobre SQLLoader
* (5%) Documente claramente el proceso de carga de datos: Cómo fue realizado, cómo logró el volumen de datos solicitado

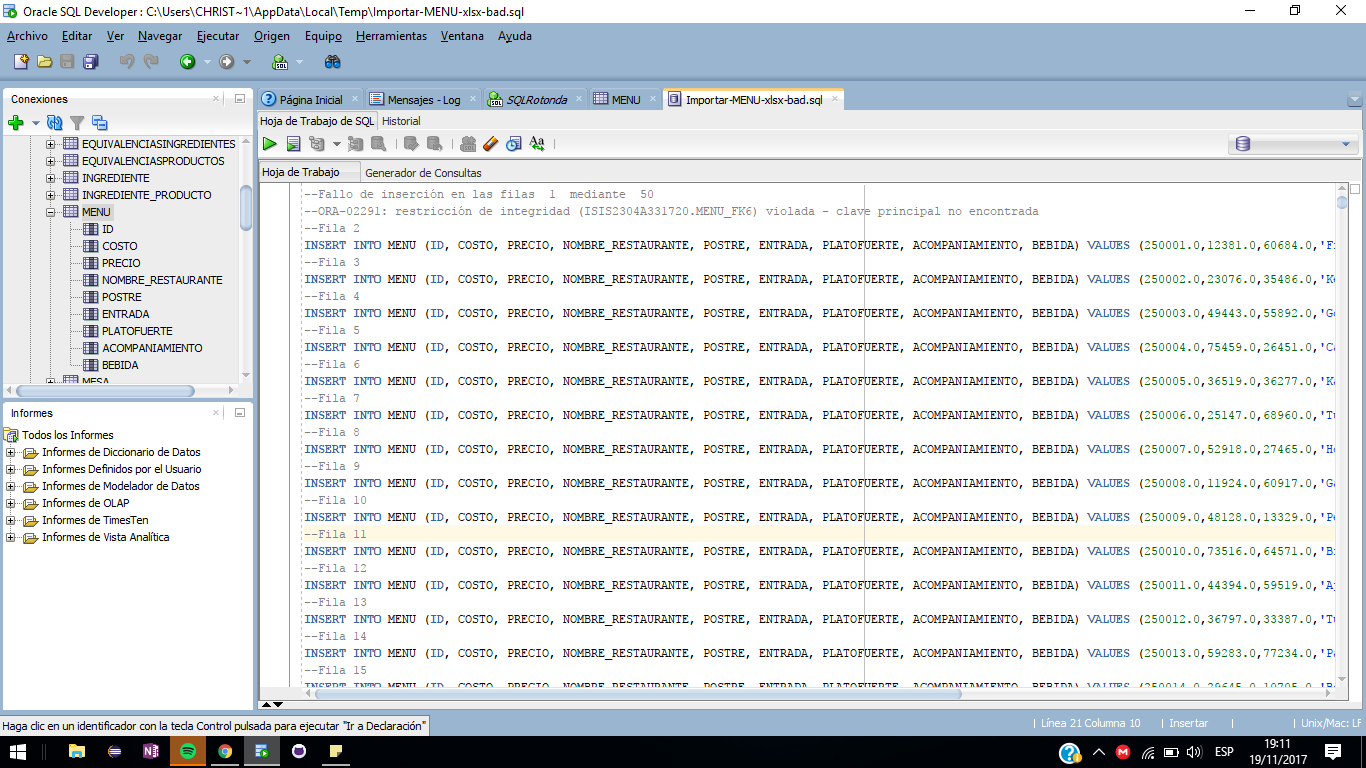
En una etapa de la carga de datos, se realizó mediante archivos de Excel y generadores de datos en línea.

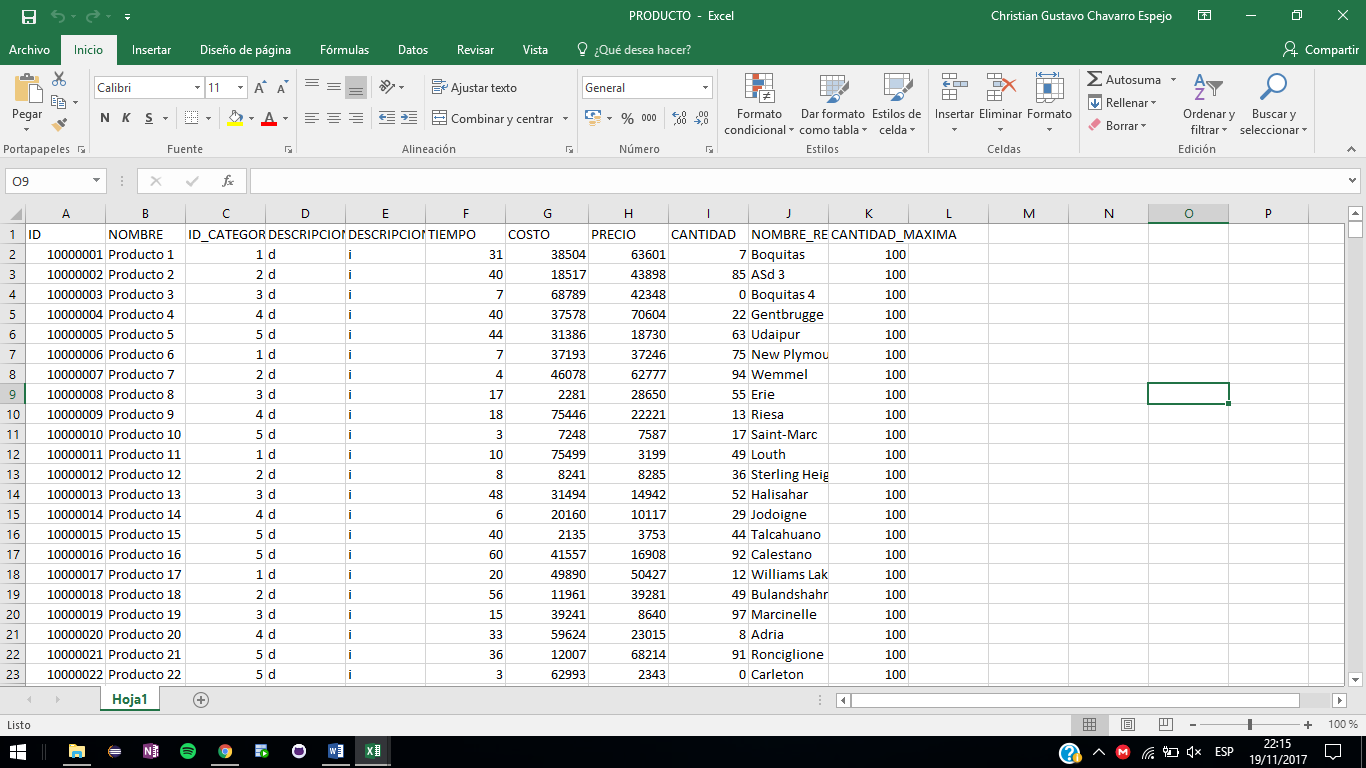
  
Imagen x: Generación de sentencias para la Inserción de clientes mediante sentencias por un generador de datos.

  
Imagen x: Inserción de clientes mediante sentencias ya realizadas en un generador de datos.

  
Imagen x: Clientes una vez insertados.

  
Imagen x: Ejemplos de menús que se insertan mediante Excel.

  
Imagen x: Inserción de los menús mediante el Excel.

  
Imagen x: productos en un archivo de Excel.

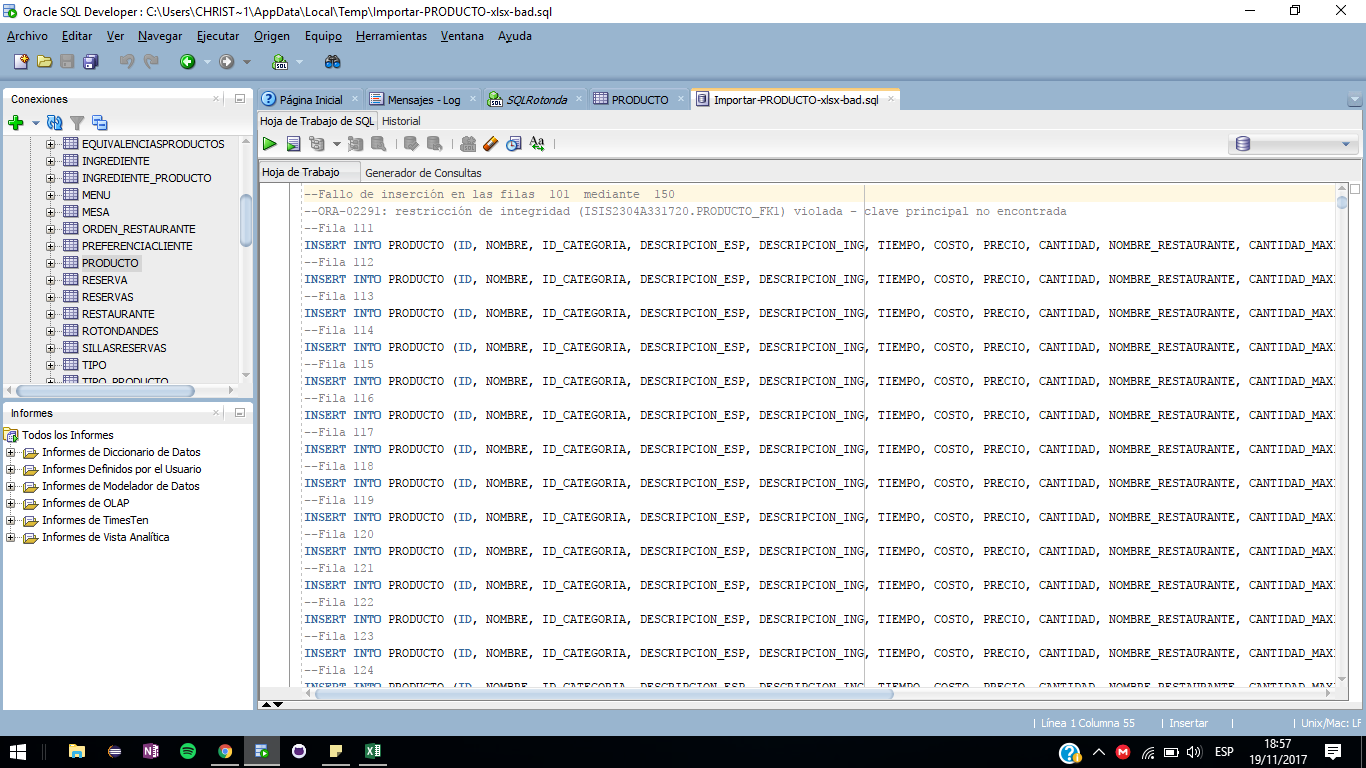


Imagen x: Inserción de productos mediante el archivo de Excel.

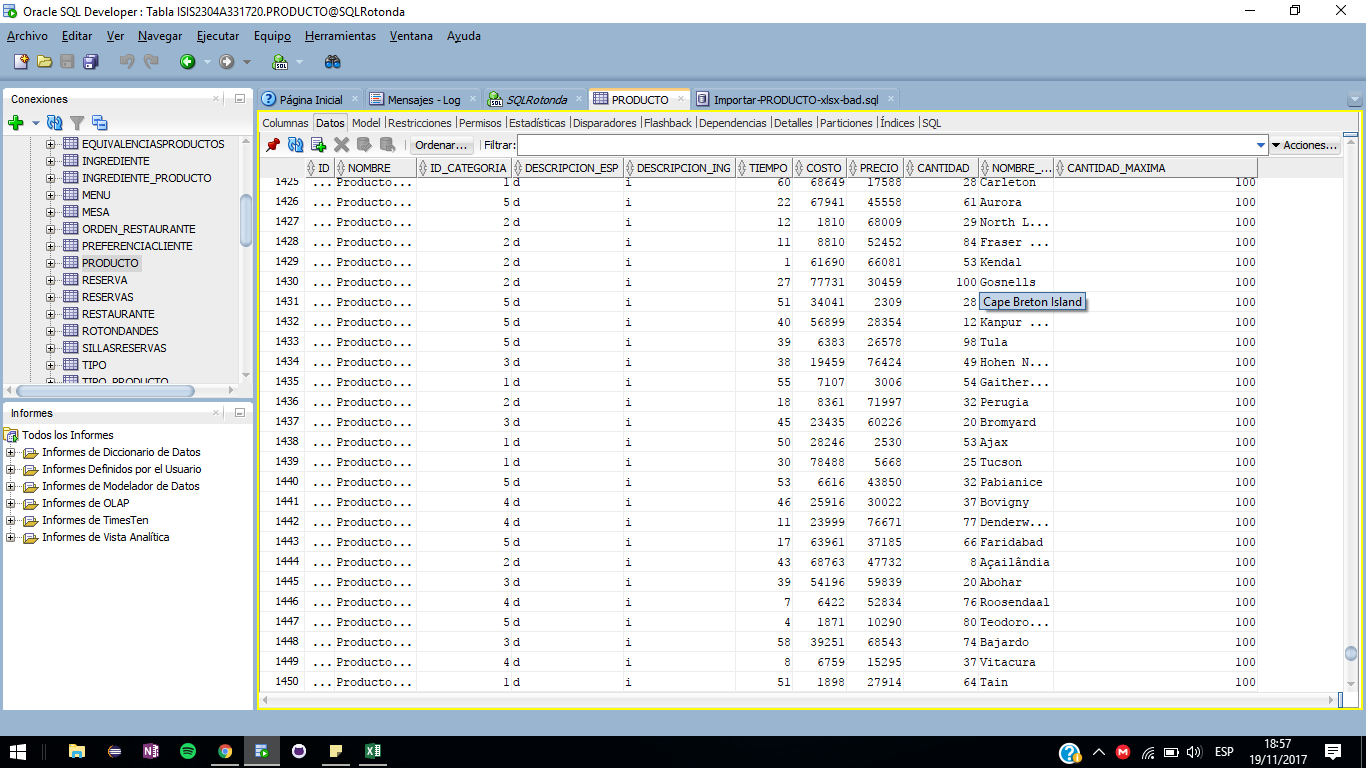
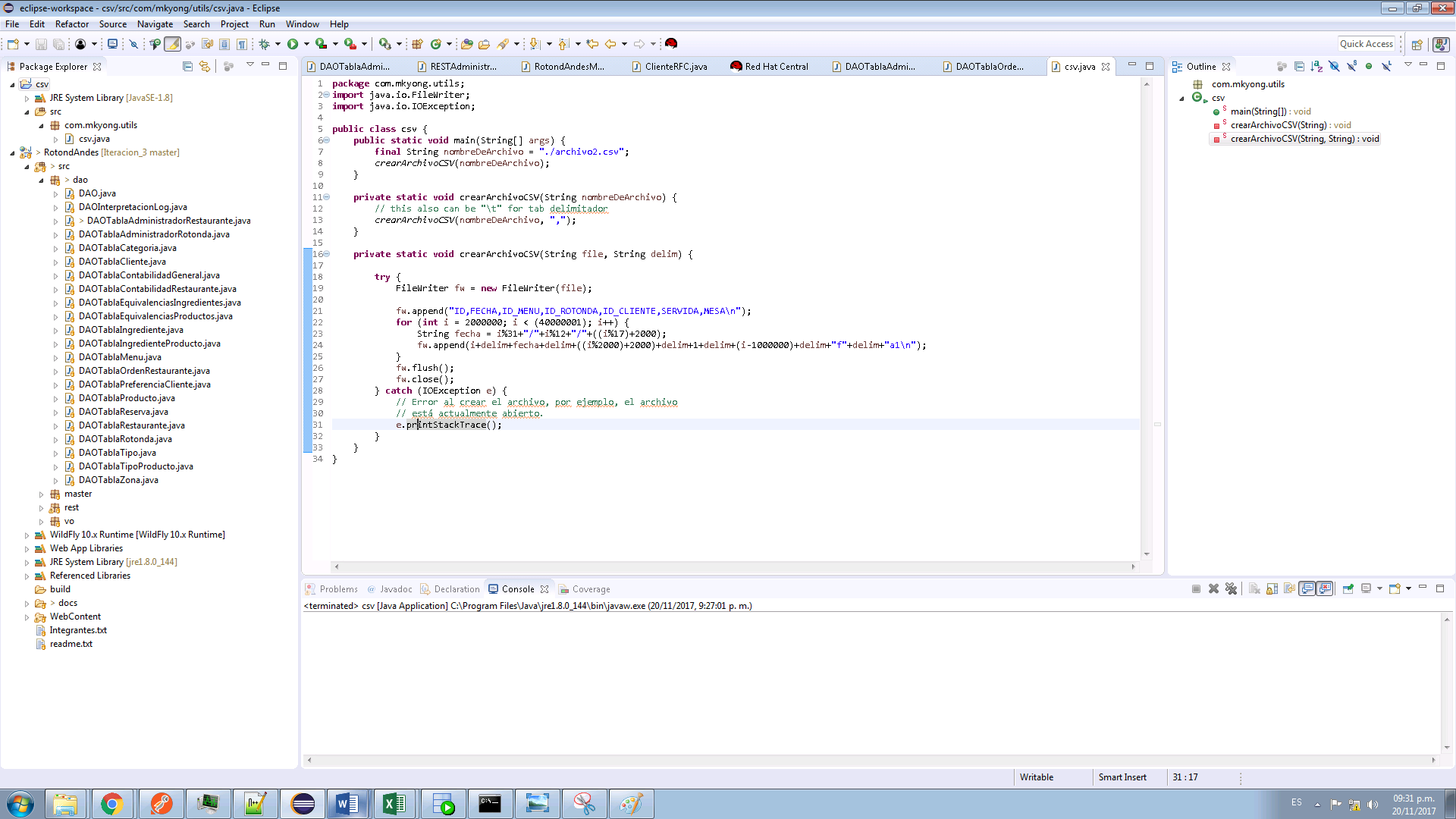
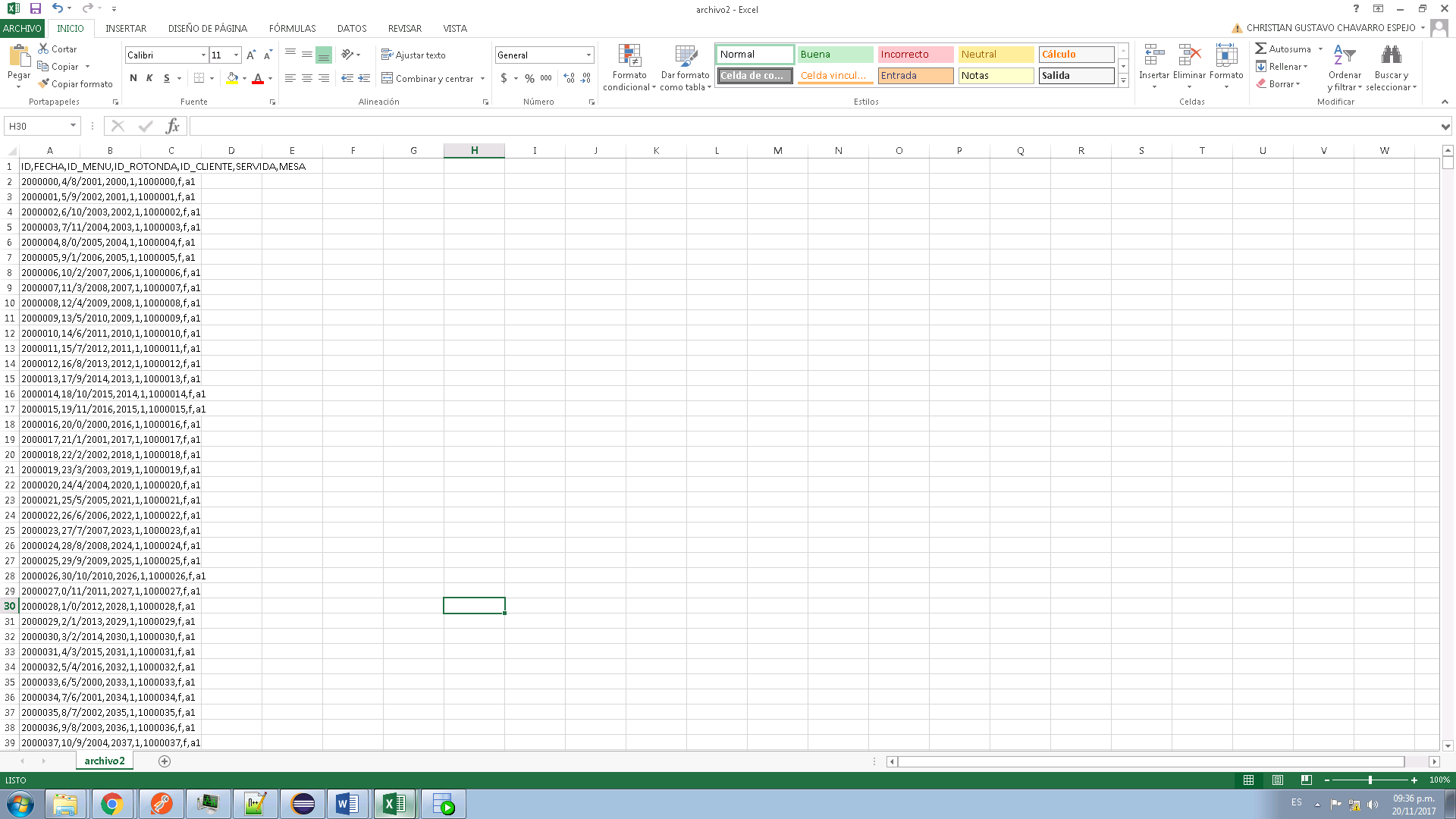
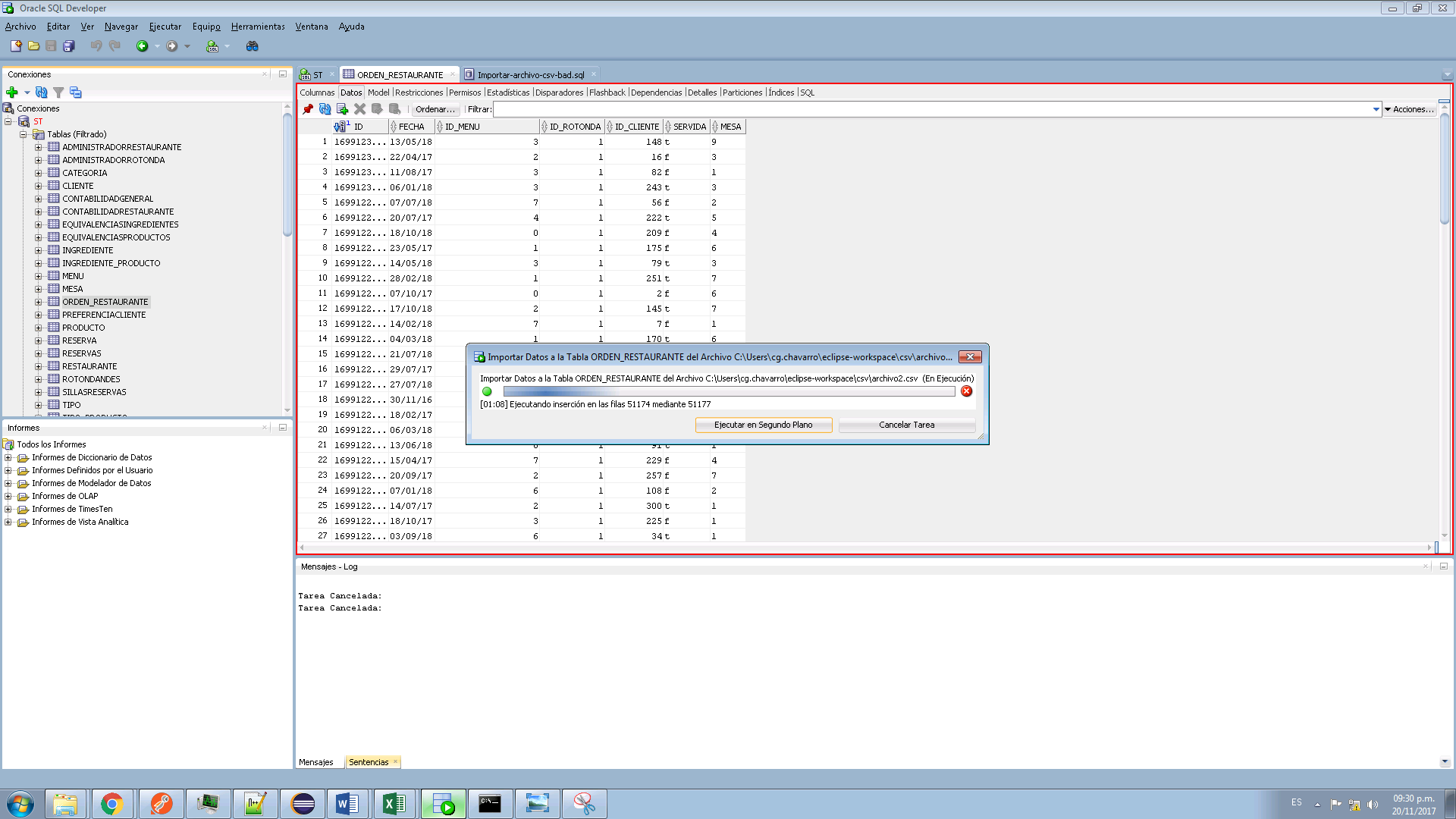


Imagen x: Productos después de ser insertados.

Imagen x: descripción del programa en java para generar los datos.  
Imagen x: respuesta del programa

  
Imagen x: inserción de los datos del csv.

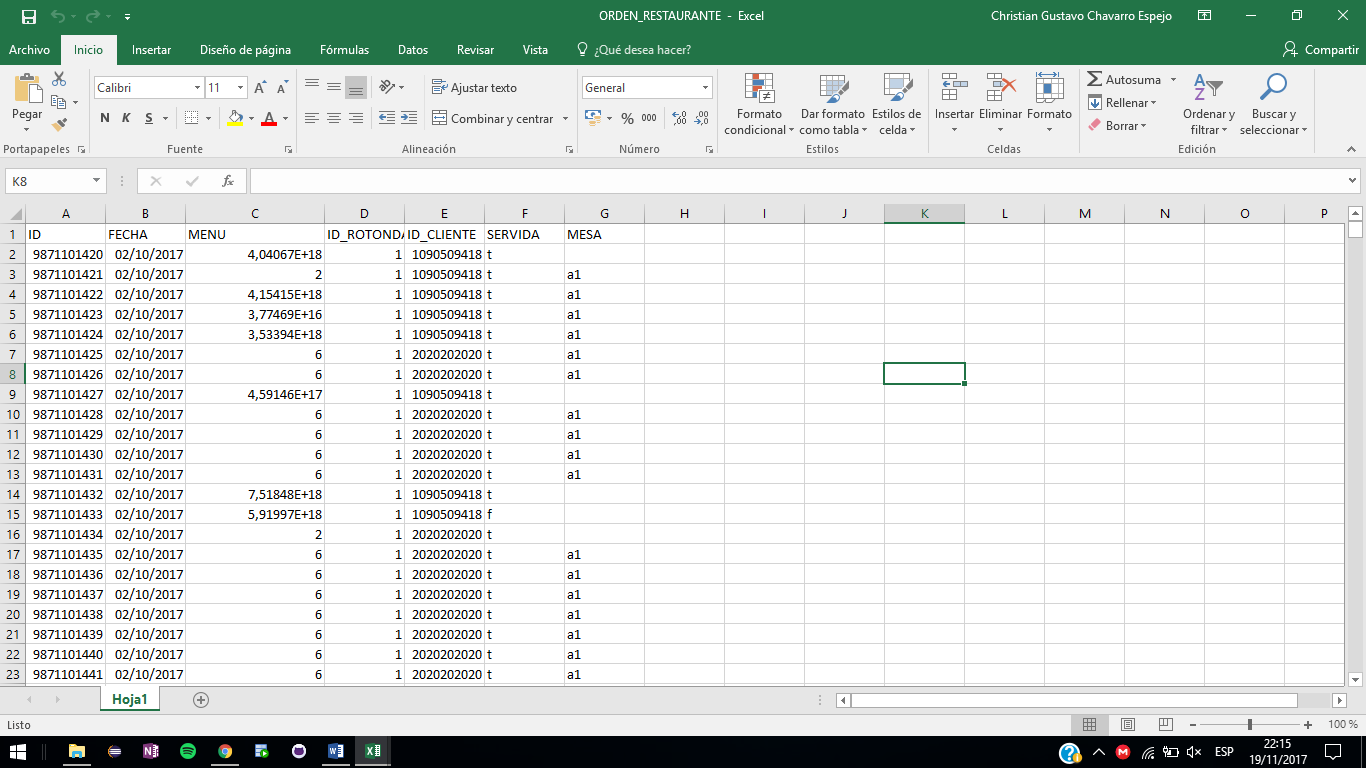
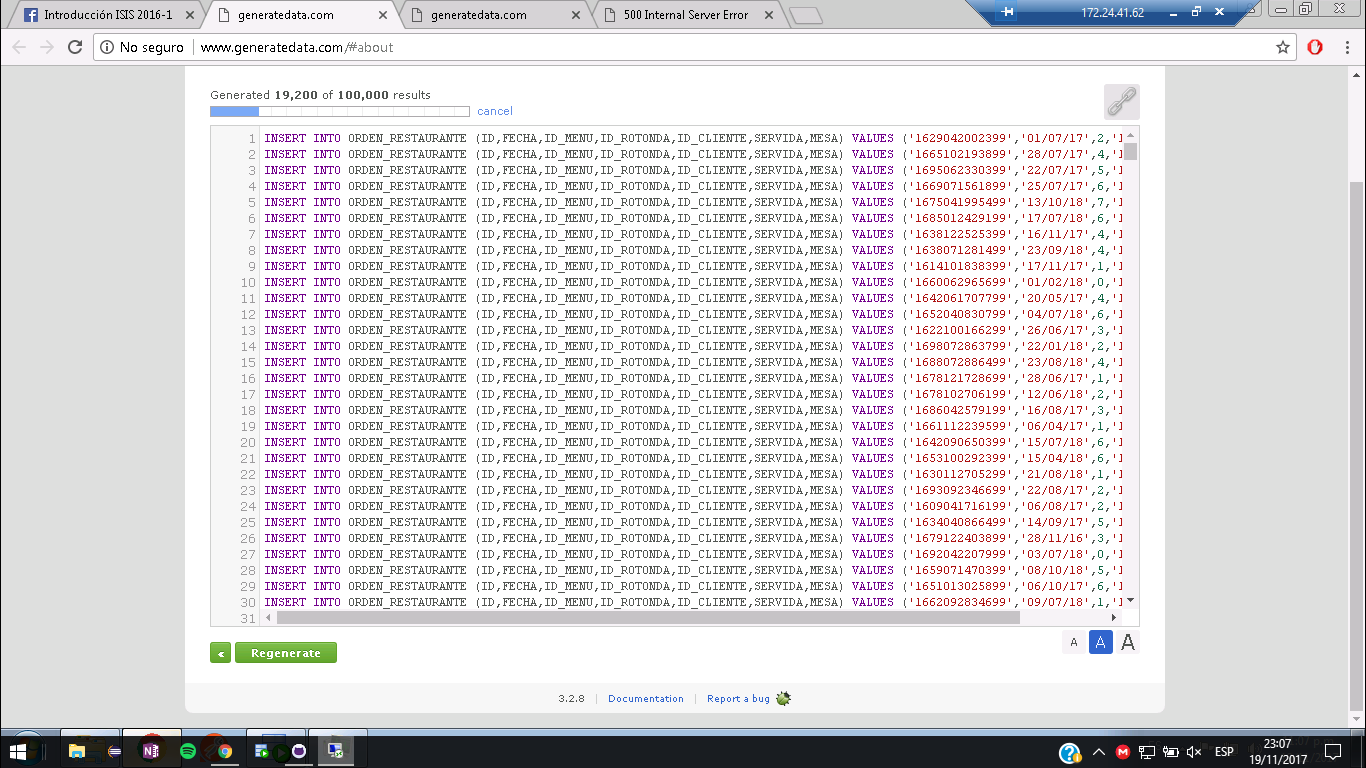


Imagen x: Ordenes a ser agregadas en la base de datos.

Imagen x: Ordenes a ser agregadas en la base de datos mediante sentencias generadas.

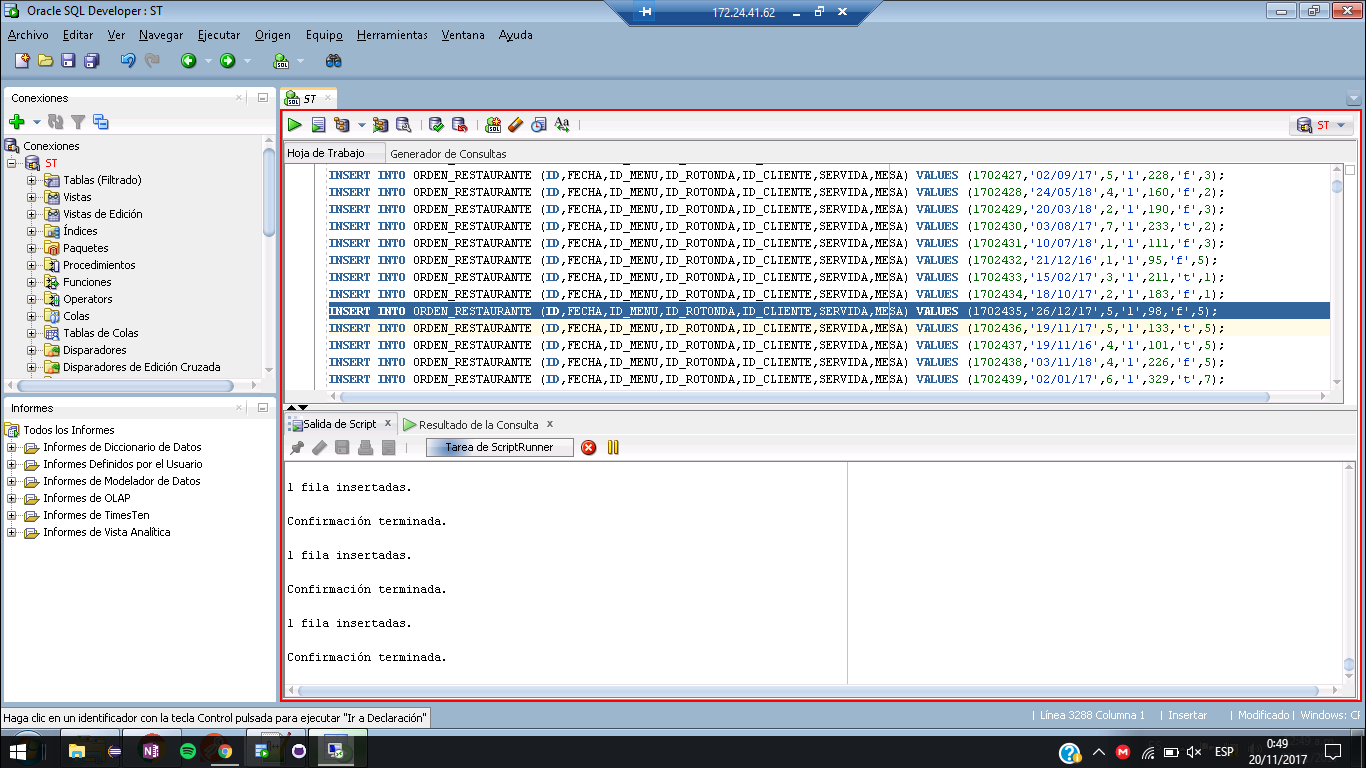


Imagen x: Ordenes siendo agregadas a la DB.

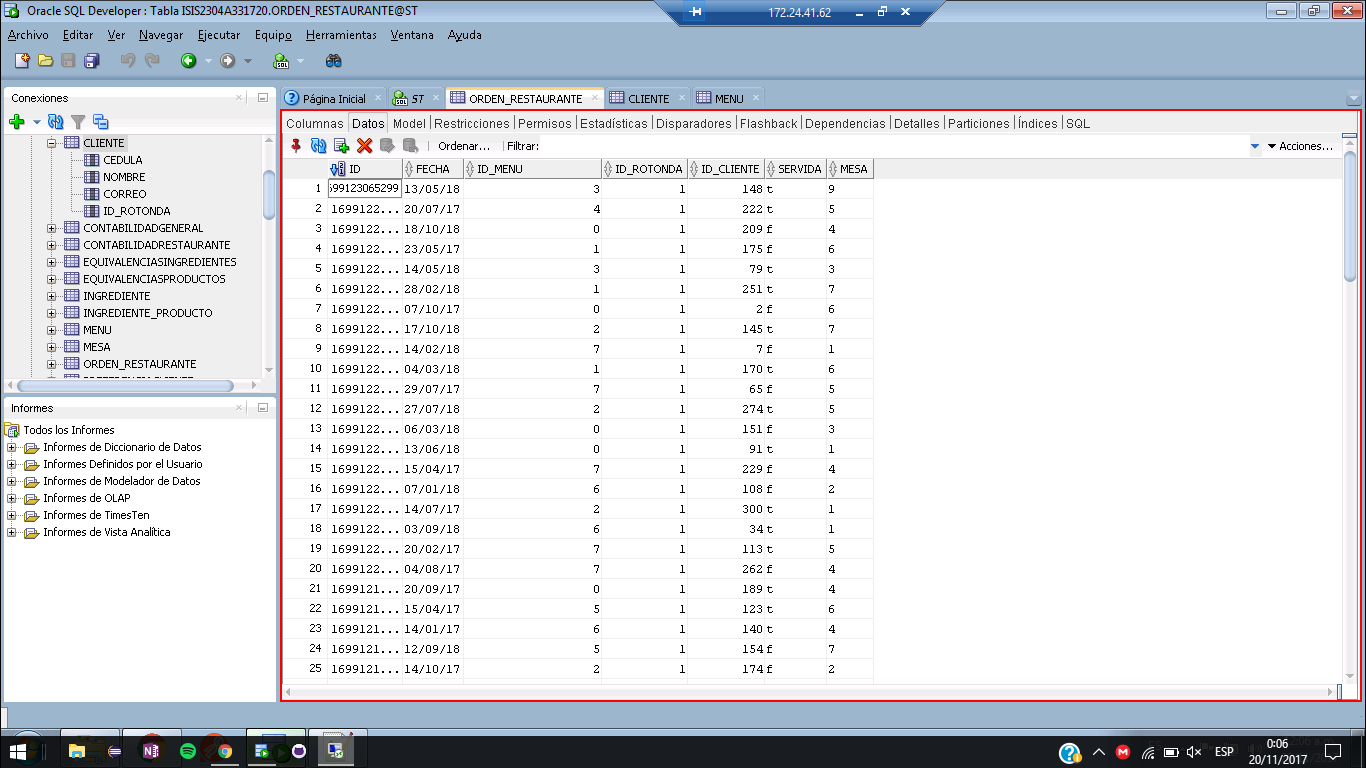


Imagen x: Ordenes una vez agregadas a la DB.

En las imágenes anteriores, se describen procesos de carga masiva de datos y solo corresponden a algunos ejemplos de la carga de estos mismos. En la aplicación, al momento de esta sustentación, el volumen de datos será más alto. Comparados con los pocos datos que se ve, son insertados en estos ejemplos.

También se muestra cómo se agrega y se crea el archivo de CSV para la creación de un millón de órdenes y de clientes, solo es mostrada la versión de la creación de órdenes.

Desarrolle o ajuste las clases involucradas en los nuevos requerimientos, de forma que complete o modifique los requerimientos funcionales y cumpla con las restricciones de negocio. Realice los cambios sobre las clases que corresponden a:

No se realizaron cambios en las tablas.

* (5%) Desarrollo y/o ajustes a los servicios REST para cumplir con los nuevos requerimientos.
* (5%) Cambios y desarrollo de las transacciones en RotondAndesMaster
* (5%) Cambios en los Dao.

(15 %) 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 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.
* Documente el análisis realizado, de forma clara y concisa.

En este punto, primero se describirá cómo trabaja cada proceso de manera breve. Tras esto, se comparará la eficiencia y qué conlleva cada uno de estos trabajos, realizar las operaciones mediante la sentencia de SQL o trabajar con los datos en memoria principal.

SQL.

Imagine el proceso de búsqueda y catalogación de los clientes, para él, se involucran como mínimo las tablas de clientes, ordenes, menús y productos. El SMBD, en este caso ORACLE, gestiona los joins entre estas tablas mediante los índices, árboles que tiene creados en su memoria principal, la del servidor. Tras concatenar estas tablas, en la sentencia SQL se piden solo las filas que interesan para el requerimiento, de las cerca de 22 filas resultantes de los joins. Con estas filas, acompañadas de una operación de agregación, se puede retornar la respuesta a la consulta.

La complejidad asociada, variará de acuerdo al tipo de join que utilice la sentencia y si esta tiene algún índice asociado.

JAVA.

Antes de iniciar esta descripción, se presenta un problema de antemano, las tablas no logran ser procesadas en memoria principal, los objetos no caben en ella.

Aún, si todos los objetos entrasen en la memoria del dispositivo sobre el cual se realiza la acción, objetos de tipo: Cliente, Orden, Menú, Producto. Realizar las operaciones de obtener las órdenes para cada cliente, obtener el plato fuerte del menú de cada orden, verificar la cantidad de ordenes por cliente y responder solo la información del cliente, costaría como mínimo un recorrido a cada uno de esos arreglos. Asumiendo que cada uno es de tamaño n, sería o(n4)

Además, los objetos deberían estar en un inicio en algún sitio de memoria principal, si están en el ordenador o en un servidor de DB, en este caso, de Oracle. El costo de carga de estos objetos también se asociaría.

Después de realizar estas descripciones, esta tabla resume cómo se realizaría cada proceso en cuanto a ejecutar la sentencia y pedir los datos por la conexión o realizar todas las acciones en memoria principal.

|  |  |  |
| --- | --- | --- |
|  | SQL | JAVA + Memoria principal. |
| Ubicación datos. | Están ubicados en el servidor, el mismo que ejecuta las sentencias. | Se ubican en el servidor de la DB, puede ser local o remoto. Remoto en este caso. |
| Procesamiento datos. | Realizado tras convertir a algebra relacional, utiliza los índices y estructuras de datos optimizadas. | Manual, con funciones como while, if. Aumenta enormemente el costo del algoritmo. |
| Selección respuesta. | El select, toma algunas columnas de las involucradas en la sentencia. Se puede adaptar a lo ya existente en el programa, clientes para el RF12. | Se debe realizar sobre los objetos realizados en los recorridos, regularmente, creando otros. VO para esta implementación. |
| ¿Cuándo se pasan los objetos a memoria del programa? | Se pasan a memoria principal, al programa, una vez hayan sido procesados, los clientes por sus criterios en el RF12. | Todos son transferidos después de iniciar la transacción y procesados en la memoria del sistema. |

Tabla x: comparativa SQL vs trabajar todo en memoria principal con java.