

## **GDD1C2013 – FRBAbus – DEL\_NAVAL - Consideraciones**

Exponemos a continuación detalles y consideraciones que tuvimos en cuenta en el diseño y desarrollo del trabajo práctico.

### **Deshabilitación de Roles**

Luego de deshabilitar el rol correspondiente, desasignamos los roles a los usuarios que lo tengan asignados y deshabilitamos los usuarios para evitar que un usuario sin rol pueda logearse en el sistema

### **Cancelación de Recorridos**

Se realiza en cascada la cancelación, de recorrido, viajes, pasajes y encomiendas, cada uno tiene su procedimiento asociado que puede a la vez reutilizarse para futuras funcionalidades en forma individual.

Los parámetros que reciben estos procedimientos se van sucediendo unos a otros, por ejemplo para llamar al procedimiento cancelarRecorrido es necesario enviar además del id\_recorrido, un código de cancelación, una fecha y un motivo. Estos parámetros se traspasan al procedimiento de cancelarViaje que a su vez se encarga de traspasárselo tanto a los procedimientos de cancelar pasajes como encomienda que son en definitiva quienes utilizan esta información

**Nota:** Al utilizar cursores en esta funcionalidad detectamos un posible inconveniente de performance a la hora de eliminar recorridos para los que se hayan vendido una gran cantidad de viajes que todavía no han partido.

Ver anexo para mayor información.

### **Devolución o cancelación de Encomiendas y Pasajes**

Suponemos que las devoluciones, tanto de encomiendas como de pasajes, solo pueden realizarse solo si la fecha de cancelación o devolución es anterior a la fecha de salida del viaje y sólo si no se encuentra cancelado o devuelto con anterioridad.

### **Bajas definitivas y por servicio de Micros**

Al realizar un reemplazo de un micro por otro, consideramos candidato como reemplazante a un micro que tenga, las misma marca, tipo de servicio, al menos la misma capacidad de bodega y al menos la misma cantidad de butacas, no se verificará que las butacas puedan reemplazarse una a una, es decir, mismo tipo, piso y número.

**NOTA:** Al no haber micros de reemplazo, el sistema brinda la opción de crear un nuevo micro. En este momento el usuario podría cargar un micro que no cumpla con los requisitos necesarios, en ese caso, cuando vuelva a intentar dar de baja nuevamente el micro, tampoco le aparecerá la opción de reemplazar, sino que le aparecerá otra vez la

opción de crear micro. En síntesis, NO SE VALIDA QUE EL MICRO QUE SE ESTA CREANDO CUMPLA LOS REQUISITOS MINIMOS PARA SER REEMPLAZANTE DEL QUE SE ESTABA INTENTANDO DAR DE BAJA.

## **Rangos de fechas**

Al utilizar rangos de fechas, por ejemplo en bajas de micros por servicio, se ingresaran fechas en el formato “DD/MM/AAAA” teniendo presente que el sistema entenderá que nos referimos a la hora 00:00:00 de ese día.

Con lo cual , si queremos dar de baja durante dos días un micro, es decir desde las 0 horas del día 01/01/2013 hasta la ultima hora del día 02/01/2013. el rango debe estar indicado de la siguiente forma:

(01/01/2013 a 03/01/2013) dado que nos estaríamos refiriendo a la hora 00:00:00 del día 3

Esto se debe a que utilizamos tipo de dato DATETIME para todas las fechas.

## **Resumen de objetos relevantes de base de datos utilizados**

### **Funciones**

**PesoLibreEncomiendasXviaje** Funcion que devuelve un escalar, representa cuanto peso entra en la bodega del micro de un viaje.

**ButacasDisponiblesXviaje** función que devuelve una tabla de butacas desocupadas del micro perteneciente a un viaje

### **Procedures**

**inhabilitar\_rol** deshabilita rol y usuarios que tienen asignado el rol que se esta deshabilitando ademas de dejar en NULL el rol para esos usuarios.

**devolverPasaje** procedimiento que realiza la cancelacion de un pasaje y devuelve el monto a devolverle al cliente, actualizando el status del sistema (ejemplo, actualiza status del pasaje, ocupacion de butaca y registra la cancelacion correspondiente)

**devolverEncomienda** procedimiento que funciona análogamente a la devolucion del pasaje, esta vez liberando peso en la bodega del micro involucrado

**cancelarViaje** realiza la devolución de todos los pasajes y encomiendas asociados a un viaje llamando a los 2 procedimientos anteriores por cada pasaje o encomienda asociados al viaje

**cancelarRecorrido** para todos los viajes que tengan asignado el recorrido a cancelar, llama al procedimiento cancelarViaje, realizando así la cancelación también de todos sus pasajes y encomiendas. Este proceso puede tardar varios minutos dependiendo de la cantidad de viajes que haya que eliminar.

**intentarBajarMicro** Si el micro no tiene viajes asignados en el rango de fechas indicado da la baja correspondiente (esta situación se informa devolviendo -1.

En caso de que tenga viajes asignados devuelve un id\_micro candidato que pueda reemplazarlo.

Si tiene viajes asignados pero no encuentra un micro de reemplazo, devuelve -2

**bajaOserviceMicro** este procedimiento es llamado por el procedimiento anterior para realizar la baja del micro propiamente dicha en caso de que no tenga viajes asignados, ya se baja definitiva o por servicio.

**cancelarViajesDeUnMicro** cuando el usuario decide cancelar todos los viajes asociados a un micro para poder darlo de baja, la aplicación llama a este procedimiento quien se encarga de dar de llamar al procedimiento cancelarViaje para cada Viaje que el micro a dar de baja tiene asignado. Al igual que en cancelarRecorrido, esto genera la cancelación de todos los Pasajes y Encomiendas de esos viajes, con lo cual se puede demorar mucho cuando se cancelan varios viajes.

**reemplazarMicro** cuando el usuario decide utilizar el micro propuesto por el procedimiento intentarBajarMicro, este procedimiento se encarga de realizar dicho reemplazo. Reemplaza el micro viejo por el nuevo en todos los viajes involucrados y además se encarga de ingresar pasaje por pasaje para actualizar las butacas del micro viejo al micro nuevo.

## Triggers

**DEL\_NAVAL.bloqueo\_usuarios** este trigger realiza el bloqueo de usuarios cuando se sobrepasa el límite de intentos de login

**DEL\_NAVAL.actualizar\_butacas\_ocupadas** cuando se realiza un insert en la tabla de pasajes, automáticamente actualiza el estado de las butacas.

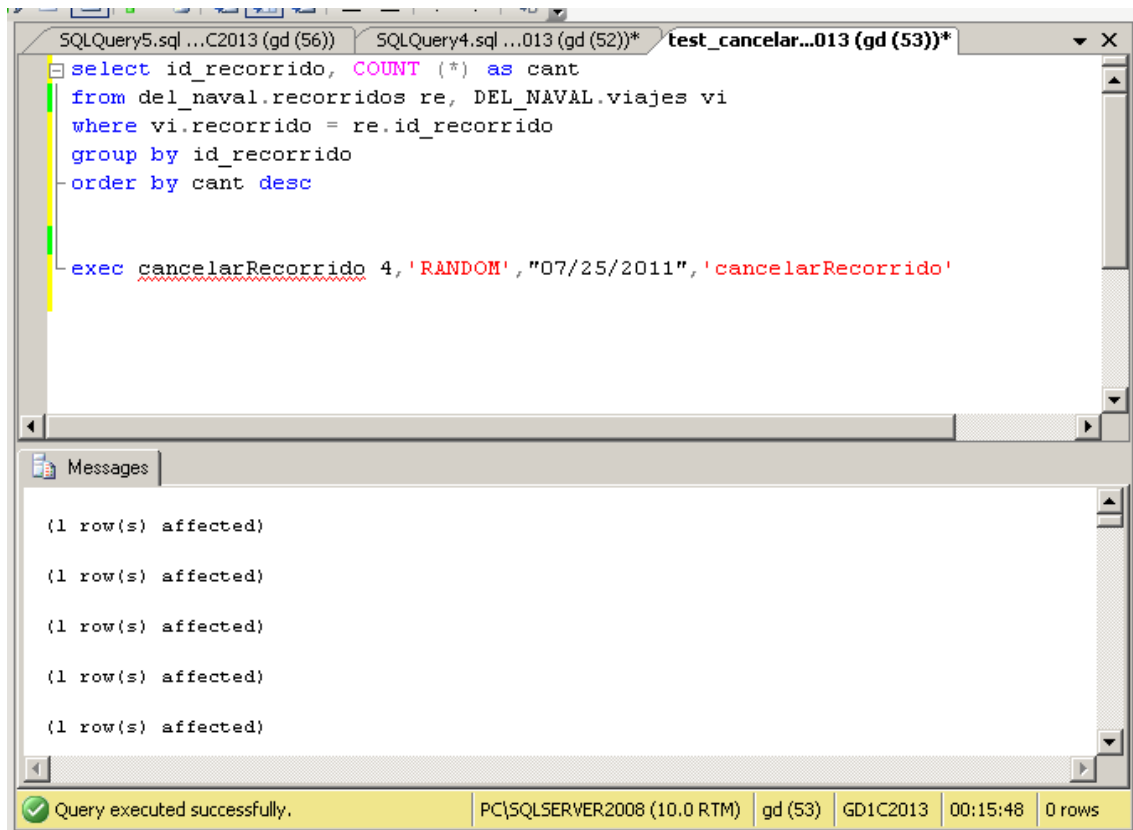
## Anexo

**Comentarios de performance relativo a la cancelación masiva de viajes, tanto para cancelación de recorrido como para cancelación de viajes por baja de un micro**

Tener en cuenta que en un caso real solo se podrían cancelar viajes, encomiendas y pasajes que aun no hayan sido realizados, nosotros decidimos no validar la existencia o no de “fecha de llegada” para permitirles a los encargados de testear la funcionalidad cancelar viajes utilizando los ya migrados, para que no se encuentren obligados a cargar

nuevos viajes (esto es porque si pusiéramos esta validación de fecha de llegada, ningún viaje migrado podría cancelarse dado que TODOS tienen fecha de llegada, es decir, todos ya fueron realizados)

**Anexamos a continuación el análisis de performance realizado:**



**Aquí ejecutamos una consulta para determinar por cada recorrido que cantidad de viajes tenía cada uno y detectamos que estaban entre 74 y 209.**

**A continuación ejecutamos el procedimiento de cancelación del recorrido que tenía 74 viajes (esto canceló el recorrido, sus 74 viajes y cada uno de sus pasajes y encomiendas) demoró aproximadamente 15 minutos. Entendemos que en la operatoria normal del sistema, es poco probable la existencia de un mayor número de viajes que no hayan salido aún para un mismo recorrido. (en este caso de test contamos todos los viajes existentes históricamente que para ese recorrido eran 74). Asimismo, decidimos incluir en esta funcionalidad un mensaje que le indique al usuario la posible demora en esta transacción.**

Por lo antes expuesto sugerimos realizar pruebas donde se eliminen entre 5 y 10 viajes para evitar timeouts en la aplicación. En la aplicación configuramos command time out en 10 minutos.