

# **Sistema de Base De Datos (Global Service Desk)**

**Víctor Manuel Miñambres Chamorro**  
Grado en Ingeniería Informática

**Base De Datos**

**Jordi Ferrer Duran**  
11/06/2018



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](#)

**Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)**

**A) Creative Commons:**



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](#)

## **B) GNU Free Documentation License (GNU FDL)**

Copyright © 2018 VÍCTOR MANUEL  
MINÁMBRES CHAMORRO.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

## **C) Copyright**

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<b>Sistema de Base De Datos (Global Service Desk)</b>
<b>Nombre del autor:</b>	<b>Víctor Manuel Miñambres Chamorro</b>
<b>Nombre del consultor/a:</b>	<b>Jordi Ferrer Duran</b>
<b>Nombre del PRA:</b>	<b>Jordi Ferrer Duran</b>
<b>Fecha de entrega (mm/aaaa):</b>	<b>06/2018</b>
<b>Titulación:</b>	<b>Grado en Ingeniería Informática</b>
<b>Área del Trabajo Final:</b>	<b>Base de Datos</b>
<b>Idioma del trabajo:</b>	<b>Español</b>
<b>Palabras clave:</b>	<b>Instalación, configuración y puesta en marcha de un sistema de base de datos (Global Service Desk)</b>
<b>Resumen del trabajo (máximo 250 palabras):</b> <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.</i>	
<p>En este proyecto se instala, diseña y configura un sistema de bases de datos donde se alojan todos los componentes necesarios para ofrecer una solución profesional a la aplicación Global Service Desk.</p> <p>Es por ello, que se definen y se muestran todos los pasos necesarios, para la instalación y configuración de un sistema gestor de base de datos con tecnología de IBM basado en Informix.</p> <p>Además, se realiza una profunda labor de configuración haciendo funcionar la aplicación Global Service Desk en un entorno real OLTP con alta concurrencia de usuarios.</p> <p>Así mismo, se establecen mecanismos necesarios de alta disponibilidad (HA) haciendo que la aplicación sea escalable, ofrezca alta disponibilidad y balanceo de carga.</p> <p>También, debido a la información sensible almacenada se configuran las herramientas necesarias de auditoría que aseguran el rastreo ante borrados o alteraciones de información no deseadas, tanto para su posterior investigación por auditorías externas como internas.</p> <p>En esta línea, se establecen las políticas de <i>backups</i> tanto de nivel 0 como incrementales necesarias para ofrecer soluciones eficaces en entornos reales.</p> <p>Como consecuencia, se presenta una solución integral de un sistema gestor de base de datos para la aplicación Global Service Desk en un entorno de producción real, ofreciendo alta disponibilidad, gestión de carga y eficacia.</p>	

**Abstract (in English, 250 words or less):**

In this project, a database system is installed, designed and configured, where all the necessary components are housed to offer a professional solution to the Global Service Desk application.

That is why all the necessary steps are defined and shown for the installation and configuration of a database management system with IBM technology based on Informix.

In addition, a deep configuration work is done by running the Global Service Desk application in a real OLTP environment with high concurrency of users.

Likewise, necessary high availability (HA) mechanisms are established, making the application scalable, offering high availability and load balancing.

Also, due to the sensitive information stored, the necessary audit tools are configured to ensure the tracking of unwanted deletions or alterations of information, both for further investigation by external and internal audits.

In this line, the backup policies of level 0 as well as the incremental ones necessary to offer effective solutions in real environments are established.

As a consequence, an integral solution of a database management system for the Global Service Desk application is presented in a real production environment, offering high availability, load management and efficiency.

# Índice

<b>1. INTRODUCCIÓN</b>	<b>1</b>
1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO	1
1.2 OBJETIVOS DEL TRABAJO	1
1.3 ENFOQUE Y MÉTODO SEGUIDO	2
1.4 PLANIFICACIÓN DEL TRABAJO	2
1.5 BREVE SUMARIO DE PRODUCTOS OBTENIDOS	8
1.6 BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA	8
<b>2. REQUISITOS DE LA APLICACIÓN GLOBAL SERVICE DESK</b>	<b>9</b>
2.1 CARACTERÍSTICAS A CUMPLIR POR EL APPLICATIVO GLOBAL SERVICE DESK	11
2.2 MEJORAR LA CONCURRENCIA DEL APPLICATIVO GLOBAL SERVICE DESK	11
2.3 PAUTAS PARA MEJORAR EL RENDIMIENTO DE LA BASE DE DATOS	12
<b>3. MODELO DE DATOS GLOBAL SERVICE DESK</b>	<b>14</b>
3.1 DETALLE DEL MODELO DE DATOS	14
3.2 DIAGRAMA UML DEL MODELO DE GLOBAL SERVICE DESK	23
3.3 CREACIÓN DE PROTOTIPO	23
3.3.1 Cálculo de extents	24
3.3.2 Modelo de usuarios y permisos	26
3.3.3 Creación de la base de datos Global Service Desk	28
3.3.4 Integridad del modelo de base de datos	29
<b>4. FUNCIONAMIENTO DEL PROTOTIPO</b>	<b>31</b>
4.1 GESTIÓN DE USUARIOS	33
4.1.1 Consulta de usuarios	33
4.1.2 Alta de usuarios	35
4.1.3 Baja de usuarios	36
4.1.4 Actualización de usuarios	37
4.2 GESTIÓN DE TICKETS	40
4.2.1 Consulta de tickets	40
4.2.2 Alta de tickets	42
4.2.3 Baja de un ticket	44
4.2.4 Actualización de tickets	45
4.3 GESTIÓN DE ESTADOS DE LOS TICKETS	48
4.3.1 Consulta del estado de un ticket	49
4.3.2 Alta del estado de un ticket	50
4.3.3 Baja de un estado de un ticket	52
4.3.4 Actualización del estado de un ticket	54
4.4 GESTIÓN DE MENSAJES DE LOS TICKETS	57
4.4.1 Consultar los mensajes de un ticket	58
4.4.2 Alta de un mensaje de un ticket	59
4.4.3 Baja de un mensaje de un ticket	60
4.4.4 Actualización del mensaje de un ticket	62
<b>5. REPOSITORIO ESTADÍSTICO</b>	<b>63</b>
5.1 DADO UN MES CUALQUIERA, TIEMPO MEDIO DE RESOLUCIÓN DE INCIDENCIAS POR CADA TIPO DE PRIORIDAD	65
RESPONDER A LA PREGUNTA	68
<b>6. CONCLUSIONES</b>	<b>70</b>
6.1. AJUSTES DE PLANIFICACIÓN	70
6.2. SOLUCIONES ADOPTADAS PARA EL SISTEMA GLOBAL SERVICE DESK	73
6.2.1. Instalación de Informix	73

6.2.2. Configuración de Informix.....	74
6.2.3. Instalación y configuración de HDR (HA) de Informix.....	76
6.2.4. Instalación y configuración de auditoría de Informix.....	78
6.3. SITUACIÓN FUTURA DESEADA .....	80
<b>7. GLOSARIO .....</b>	<b>80</b>
<b>8. BIBLIOGRAFÍA/WEBGRAFÍA .....</b>	<b>81</b>
<b>9. ANEXO A: INSTALACIÓN DE INFORMIX .....</b>	<b>83</b>
<b>10. ANEXO B: CONFIGURACIÓN DE INFORMIX.....</b>	<b>94</b>
<b>11. ANEXO C: GENERACIÓN DE SCRIPTS SQL DEL MODELO .....</b>	<b>109</b>
CREACIÓN DE LA ESTRUCTURA VACÍA Y LOS PERMISOS: .....	109
INSERCIÓN DE LOS DATOS DE PRUEBA INICIALES DEL MODELO:.....	119
CREACIÓN DE LOS PROCEDIMIENTOS ALMACENADOS DE GESTIÓN DE USUARIOS .....	127
CREACIÓN DE LOS PROCEDIMIENTOS ALMACENADOS DE GESTIÓN DE TICKETS .....	127
CREACIÓN DE LOS PROCEDIMIENTOS ALMACENADOS DE GESTIÓN DE ESTADOS DE LOS TICKETS.....	127
CREACIÓN DE LOS PROCEDIMIENTOS ALMACENADOS DE GESTIÓN DE MENSAJES DE LOS TICKETS .....	127
<b>12. ANEXO D: INSTALACIÓN Y CONFIGURACIÓN DE HDR (HA) .....</b>	<b>129</b>
<b>13. ANEXO E: INSTALACIÓN Y CONFIGURACIÓN DE AUDITORÍA .....</b>	<b>133</b>
<b>14. ANEXO F: COMANDOS DE MONITORIZACIÓN DE INFORMIX .....</b>	<b>135</b>
<b>15. ANEXO G: BATERÍA DE PRUEBAS REALIZADAS.....</b>	<b>144</b>
1. INSERCIÓN DE ESTADO DE UN TICKET 'RESUELTO'.....	144
2. BORRADO DEL ESTADO DE UN TICKET CON CONDICIÓN 'RESUELTO' .....	149
3. ACTUALIZACIÓN DEL ESTADO DE UN TICKET CON CONDICIÓN 'RESUELTO' .....	151
4. GESTIÓN DE USUARIOS: .....	159
5. GESTIÓN DE TICKETS: .....	166
6. GESTIÓN DE ESTADOS DE TICKETS: .....	175
7. GESTIÓN DE MENSAJES DE TICKETS: .....	184
<b>16. ANEXO H: GENERACIÓN DE SCRIPTS SQL DEL REPOSITORIO ESTADÍSTICO .....</b>	<b>193</b>
CREACIÓN DE LA ESTRUCTURA DEL REPOSITORIO ESTADÍSTICO.....	193
CREACIÓN DE LOS DATOS INICIALES DEL REPOSITORIO ESTADÍSTICO .....	194
CREACIÓN DE LOS PROCEDIMIENTOS DEL REPOSITORIO ESTADÍSTICO.....	200

# 1. Introducción

## 1.1 Contexto y justificación del trabajo

Se plantea la necesidad de crear un entorno de base de datos con todos los objetos necesarios para hacer funcionar una aplicación de gestión de incidencias llamada Global Service Desk.

Esto conlleva, instalar y aplicar todas las configuraciones necesarias para hacer funcionar un entorno gestor de bases de datos en producción, en base a los requerimientos solicitados, que son:

- Alta carga OLTP incluso desde entornos móviles.
- El sistema debe ser totalmente trazable.
- Debe ser un sistema escalable.
- Soporte de sentencias propias de estadísticas incluso dimensionar la base de datos para soportar minería de datos.

De esta forma, gestionando la solución integral Global Service Desk, se distribuyen los hitos y los pasos necesarios desde la instalación del motor de bases de datos, así como las políticas de *backups*, alta disponibilidad o auditoría.

En consecuencia, se pretende conseguir una solución integral y profesional vendible a cualquier organización.

## 1.2 Objetivos del trabajo

El objetivo principal de este proyecto es ofrecer una solución completa de base de datos para la aplicación Global Service Desk en un entorno real y en función de los requerimientos solicitados.

Por lo tanto, los objetivos a cumplir son:

1. Descripción del proyecto a realizar.
2. Instalación del gestor de bases de datos.
3. Configuración del gestor de bases de datos.
4. Diseño, modelado y preparación de los *scripts* del aplicativo.
5. Pruebas del modelo (ciclo de vida de un *ticket*).
6. Políticas de *backup*, estadísticas y seguridad de la base de datos.
7. Configuración del sistema en alta disponibilidad.
8. Configuración del sistema de auditoría.
9. Plan de contingencia.
10. Resolución de problemas.



### 1.3 Enfoque y método seguido

El proyecto se divide tres partes a integrar que son:

1. Elección e instalación de Informix como solución profesional de base de datos.
2. Configuración de una solución que atienda a los requisitos de uso que se tiene que garantizar.
3. Diseño y creación del modelo de objetos para la aplicación Global Service Desk.
4. Instalación y configuración de las herramientas necesarias para ofrecer una solución sólida, completa e integral en un entorno profesional, escalable y con perspectiva de futuro, como son HDR (HA; Alta disponibilidad), *backup*, *restore* y auditoría.

Por lo tanto, al elegir Informix se asegura una solución conocida y con solidez profesional, siendo además su sistema de HA y de auditoría una solución sencilla de implantar, robusta, fiable y con ejemplos amplios en el mercado.

Otro aspecto importante, es que la HA (HDR) es conocida en el mercado por su robustez, así como todo el gestor y sus herramientas con tecnología de IBM que aporta tranquilidad a cualquier cliente, teniendo un soporte de referencia en el ámbito profesional.

Además, la adquisición del gestor de Informix incluye en la misma licencia todas las herramientas necesarias para este proyecto integral, no siendo necesario costes en licencias adicionales.

### 1.4 Planificación del trabajo

#### Tarea 1: Análisis y estudio de las soluciones tecnológicas.

##### **Descripción:**

Se estudian las soluciones existentes para el sistema integral propuesto, así como cobertura de soporte o herramientas incluidas en la licencia de adquisición del gestor de base de datos.

##### **Objetivos:**

1. Analizar detalladamente cuáles son los requerimientos del sistema, tanto directos como indirectos.
2. Analizar las distintas soluciones profesionales que existen el mercado.
3. Localizar todos los riesgos, como costes, soporte, actualizaciones, herramientas necesarias, etc.

**Subtareas:**

<b>Subtarea</b>	Análisis de los requisitos
<b>Descripción</b>	Estudio de todas las necesidades y requisitos del sistema Global Service Desk.
<b>Objetivos</b>	Conseguir una visión completa de lo que se necesita.

<b>Subtarea</b>	Estudio de soluciones tecnológicas
<b>Descripción</b>	Localizar las distintas soluciones en el mercado que se adapten a los requerimientos propuestos.
<b>Objetivos</b>	Localizar las soluciones que mejor garanticen el servicio a cubrir.

<b>Subtarea</b>	Análisis de riesgos.
<b>Descripción</b>	Estudiar y analizar todos los riesgos directos e indirectos que puedan surgir.
<b>Objetivos</b>	Localizar las mejores soluciones que cubran todos los riesgos detectados.

**Tarea 2: Instalación y configuración del sistema integral de base de datos.****Descripción:**

Se instala y configura la solución integral elegida de base de datos.

**Objetivos:**

1. Analizar y estudiar los mecanismos para la instalación a realizar.
2. Realizar la instalación con los pasos definidos.
3. Configurar el sistema para dar el servicio que se requiere.

**Subtareas:**

<b>Subtarea</b>	Análisis y estudio de los requisitos de instalación.
<b>Descripción</b>	Análisis y estudio de todos los requisitos de instalación y configuración de la tecnología elegida para el sistema Global Service Desk.
<b>Objetivos</b>	Conseguir los requisitos y los pasos para poder instalar y configurar de forma óptima el sistema integral.

<b>Subtarea</b>	Instalación del gestor de base de datos.
<b>Descripción</b>	Seguir los pasos propuestos por IBM para realizar la instalación del gestor de base de datos.
<b>Objetivos</b>	Realizar una instalación completa del sistema de base de datos.

<b>Subtarea</b>	Configuración del sistema de base de datos.
<b>Descripción</b>	Realizar la configuración propuesta por IBM para cubrir las necesidades del sistema presentes y futuras.
<b>Objetivos</b>	Dejar configurado el sistema de base de datos para adaptarse con las garantías necesarias a los retos propuestos que debe cumplir el aplicativo.

### Tarea 3: Creación del modelo de datos y *scripts* necesarios.

#### **Descripción:**

Creación de todo el conjunto de objetos necesario para crear el modelo de datos de la aplicación Global Service Desk.

#### **Objetivos:**

1. Analizar y estudiar el modelo propuesto que cubra todas las funciones que le son requeridas.
2. Estudiar la sintaxis del lenguaje SQL de la tecnología elegida.
3. Creación del código necesario para generar el modelo portable a cualquier entorno.
4. Documentar completamente todo el modelo.

#### **Subtareas:**

<b>Subtarea</b>	Análisis y estudio de todos los objetos necesarios para desarrollar el modelo.
<b>Descripción</b>	Localizar todos los objetos necesarios para que el modelo de datos cumpla con las expectativas propuestas, sea escalable y portable.
<b>Objetivos</b>	Obtener todos los objetos necesarios que conforman el modelo de datos.

<b>Subtarea</b>	Creación de los <i>scripts</i> para generar el modelo de objetos del modelo de datos.
<b>Descripción</b>	Generar el código SQL para crear todos los objetos necesarios que conforman el modelo de datos del aplicativo Global Service Desk.
<b>Objetivos</b>	Creación de <i>scripts</i> completos de todos los objetos que conforman el modelo de datos.

<b>Subtarea</b>	Documentación del modelo
<b>Descripción</b>	Realizar la documentación completa del modelo de datos, metiendo los comentarios necesarios en el código con la sintaxis adecuada.
<b>Objetivos</b>	Dejar el modelo de datos perfectamente documentado para poder adaptarlo y modificarlos en función de los requerimientos futuros.

#### Tarea 4: Ciclo de vida de un *ticket*.

##### **Descripción:**

Generación de todas las fases necesarias para definir y testear el ciclo de vida de un *ticket*.

##### **Objetivos:**

1. Analizar y estudiar las diferentes fases por las que puede atravesar un *ticket*.
2. Definir los estados que puede contener un *ticket* en cada fase y las acciones que se pueden realizar.
3. Creación de los diferentes grupos y ámbitos en el trato de un *ticket*.
4. Testeo de todas las fases de un *ticket*.

##### **Subtareas:**

<b>Subtarea</b>	Análisis y estudio de las diferentes fases de un <i>ticket</i> .
<b>Descripción</b>	Identificar las diferentes fases de un <i>ticket</i> y en que situaciones se pueden originar.
<b>Objetivos</b>	Obtener e identificar todas las fases de un <i>ticket</i> .

<b>Subtarea</b>	Identificar las acciones que modifican las fases de un <i>ticket</i> .
<b>Descripción</b>	Localizar todas las acciones que modifican un <i>ticket</i> en las diferentes fases por las que puede atravesar.
<b>Objetivos</b>	Definir las acciones que modifican un <i>ticket</i> en cada fase.

<b>Subtarea</b>	Testeo del ciclo de vida de un <i>ticket</i> .
<b>Descripción</b>	Comprobar el ciclo de vida de un <i>ticket</i> así como los grupos por los que atraviesa.
<b>Objetivos</b>	Probar el correcto funcionamiento de un <i>ticket</i> así como sus cambios de estados en los diferentes grupos.

#### Tarea 5: Configuración de la solución integral.

##### **Descripción:**

Realizar las configuraciones necesarias para la creación de una solución integral que ofrezca, alta disponibilidad, políticas de *backup* y registro de auditoría.

##### **Objetivos:**

1. Configuración y montaje de la solución de alta disponibilidad (HA-HDR).
2. Configuración y creación de los mecanismos necesarios para definir políticas de *backups*.
3. Configuración de un sistema de auditoría.

##### **Subtareas:**

<b>Subtarea</b>	Configuración y puesta en marcha de un sistema HA.
<b>Descripción</b>	Realizar todos los pasos necesarios para poner en marcha un sistema de alta disponibilidad para el aplicativo Global Service Desk.
<b>Objetivos</b>	Configurar alta disponibilidad en el gestor de base de datos.

<b>Subtarea</b>	Configurar los <i>backups</i> para poder definir políticas de <i>backups</i> .
<b>Descripción</b>	Realizar todos los pasos necesarios para que el gestor de base de datos realice los <i>backups</i> de nivel 0 e incrementales necesarios.
<b>Objetivos</b>	Creación de políticas de <i>backups</i> como solución integral de Global Service Desk.

<b>Subtarea</b>	Configuración de un sistema de auditoría.
<b>Descripción</b>	Dejar registro de todas las acciones realizadas sobre información sensible y por qué perfil son realizadas.
<b>Objetivos</b>	Poner en funcionamiento un sistema de auditoría de base de datos.

Tarea 6: Creación de acciones de contingencia y resolución de problemas.

**Descripción:**

Definir las acciones necesarias para recuperar el sistema ante desastres así como de funcionamiento no deseado del aplicativo Global Service Desk.

**Objetivos:**

1. Definir los fallos que pueden surgir en el sistema.
2. Creación de planes de contingencia para solucionar los fallos del sistema.
3. Resolución de problemas ante errores no deseados del aplicativo Global Service Desk.

**Subtareas:**

<b>Subtarea</b>	Identificar los posibles fallos del sistema.
<b>Descripción</b>	Investigar las posibles situaciones que hagan fallar el sistema, tanto por errores hardware como software.
<b>Objetivos</b>	Localizar los puntos de error del sistema integral Global Service Desk.

<b>Subtarea</b>	Crear planes de contingencia.
<b>Descripción</b>	Definir los planes de contingencia necesarios para solucionar de forma efectiva los errores que puedan aparecer en la solución integral.
<b>Objetivos</b>	Realizar las operativas necesarias para levantar el sistema en el mínimo espacio de tiempo posible.

<b>Subtarea</b>	Solución de errores del aplicativo Global Service Desk.
<b>Descripción</b>	Generar soluciones a los posibles errores de datos que pueda generar el aplicativo Global Service Desk.
<b>Objetivos</b>	Garantizar la estabilidad y consistencia del aplicativo Global Service Desk.

### 1.5 Breve resumen de productos obtenidos

- Software del gestor de base de datos.
- Software de herramientas de *backup* y *restore*.
- Software de herramientas de HA.
- Software de herramientas de auditoría.
- *Scripts* SQL de creación de modelo Global Service Desk.
- Memoria del proyecto.

### 1.6 Breve descripción de los otros capítulos de la memoria

En el Capítulo 1 se indican todas las tareas necesarias y los motivos para poner en marcha este proyecto, así como las decisiones adoptadas y las tareas definidas.

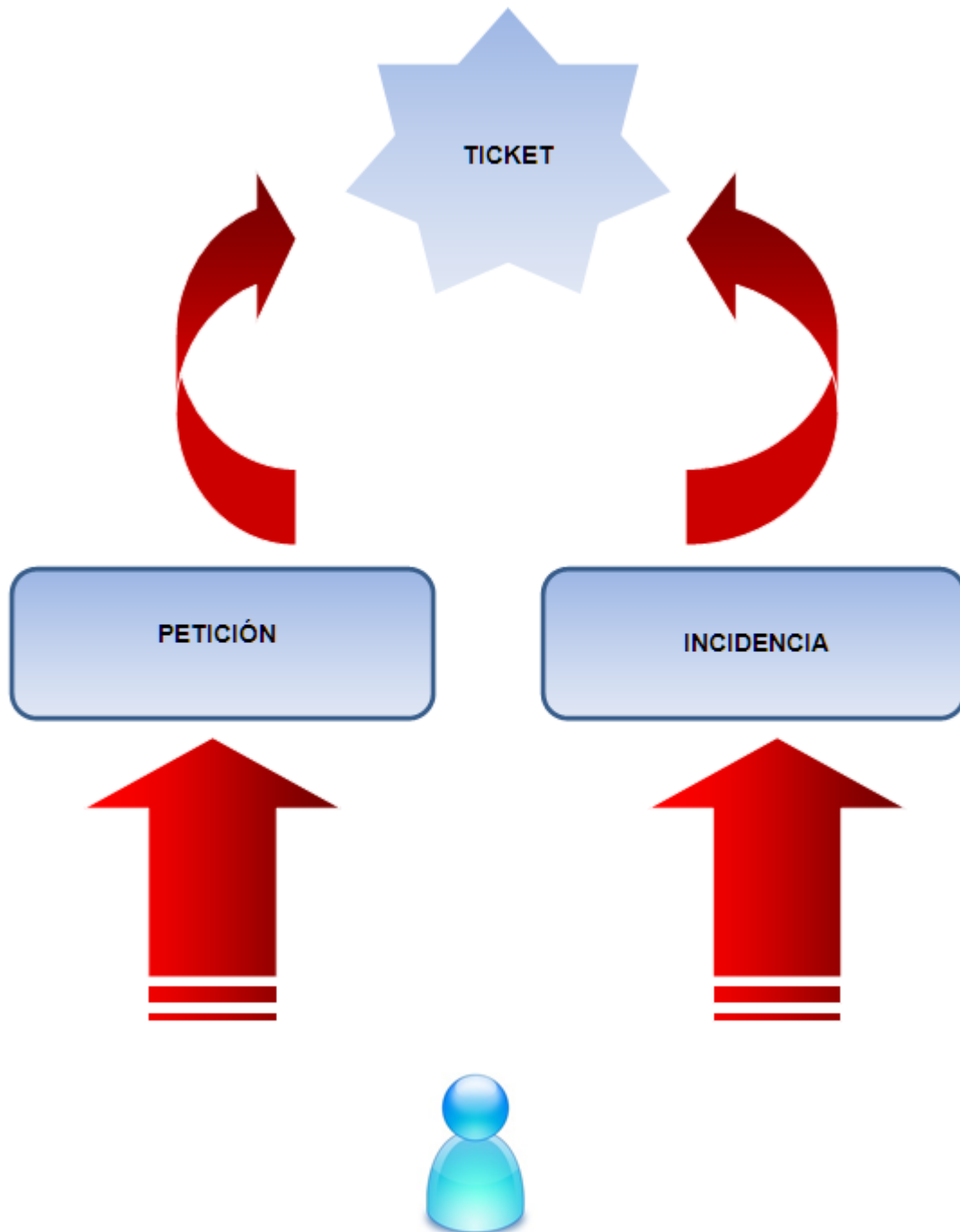
En los capítulos centrales se aborda en detalle todo el análisis, diseño e implantación del proyecto. También se muestran las características que debe cubrir así como las fases por la que atraviesa y los planes de contingencia ante errores del sistema.

En el capítulo de conclusiones se muestra una visión global donde se agrupan los diferentes retos, aciertos y errores surgidos además de sus soluciones.

En los diferentes capítulos de Anexos se detallan todos los aspectos técnicos seguidos para instalar y configurar las diferentes soluciones que engloban el sistema integral Global Service Desk.

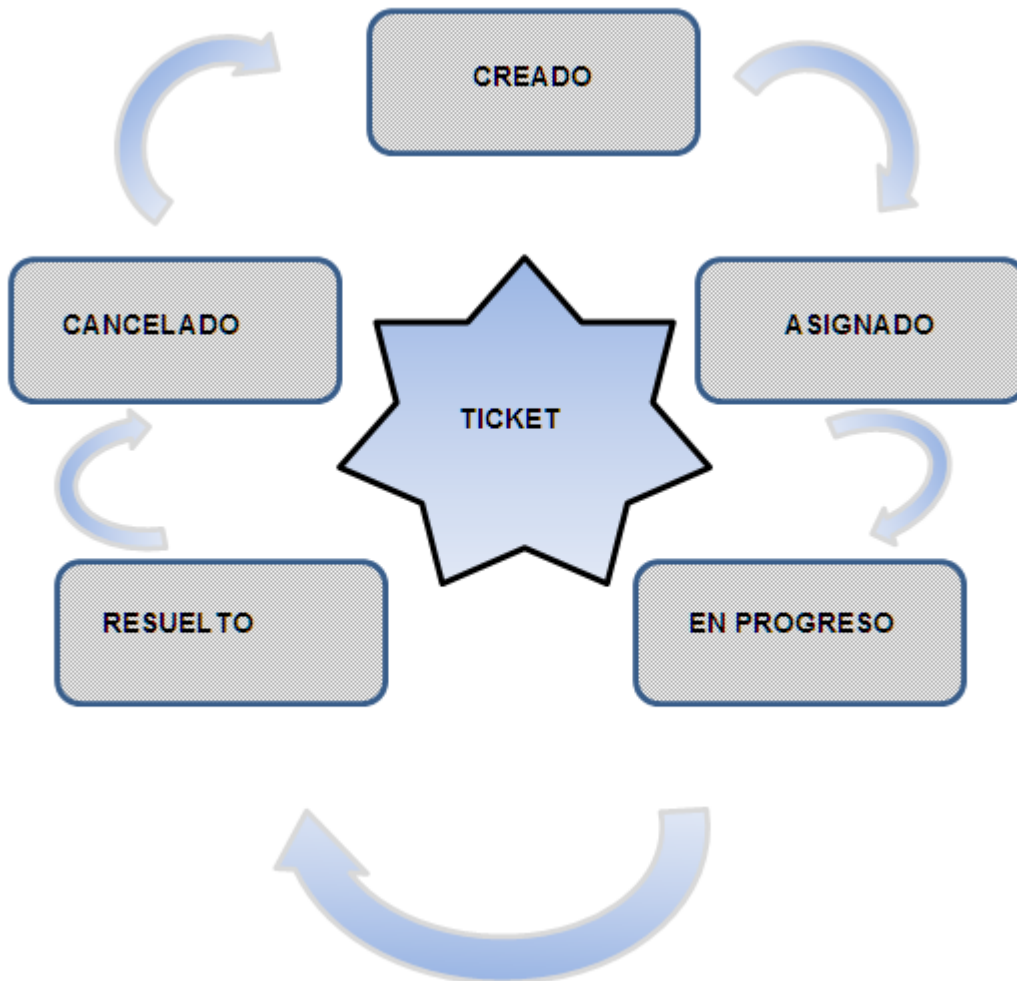
## 2. Requisitos de la aplicación Global Service Desk

Se propone la creación de un sistema de gestión de incidencias y peticiones globales (Global Service Desk), siendo el *ticket* la pieza única de generación de una incidencia o petición:

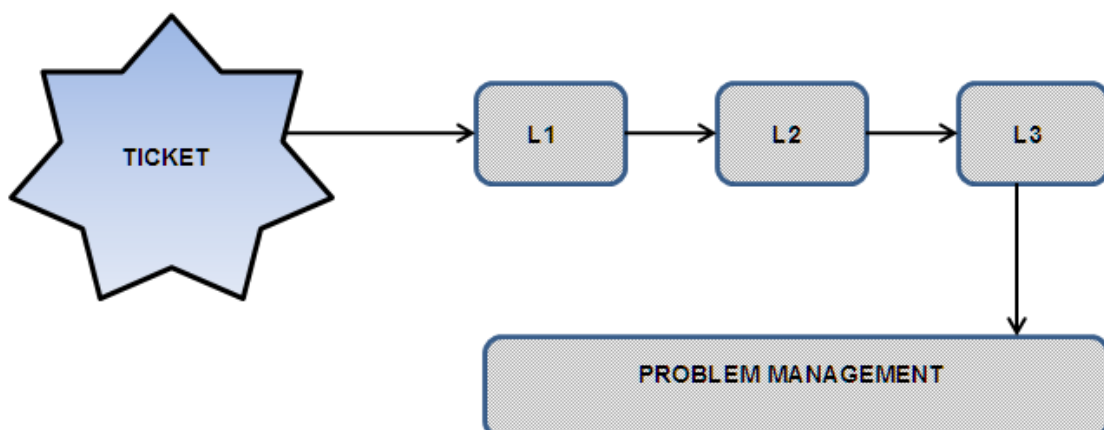




Así mismo, se define el siguiente ciclo de vida para un *ticket*:



Se debe agregar, que el proceso de resolución de un *ticket* comprende el siguiente flujo:



## 2.1 Características a cumplir por el aplicativo Global Service Desk

El sistema Global Service Desk cumple las siguientes características:

- Poder acceder desde cualquier dispositivo (incluso móvil).
- El sistema ha de ser totalmente trazable.
- Se tiene que poder asignar prioridad y tiempos de resolución.
- Debe ser un sistema escalable.
- Soporta sentencias SQL de analíticas sobre los datos almacenados.

## 2.2 Mejorar la concurrencia del aplicativo Global Service Desk

Con el fin de mejorar la concurrencia en la base de datos y evitar problemas de bloqueos y de temidos *deadlocks*, se generan unas pautas a seguir como orientaciones al desarrollo del aplicativo por parte del departamento de base de datos.

Dentro del modelo de datos existen entidades que son susceptibles de originar estos problemas.

La entidad *tickets* es el centro de la aplicación y desde el aplicativo Global Service Desk hay que hacer exhaustivas pruebas de carga para minimizar el impacto de la concurrencia y de los llamados “cuellos de botella”.

Para ello, se dan pautas como:

- Establecer el nivel de aislamiento en DIRTY READ para todos aquellos procesos que no tengan que mantener la integridad del dato.
- Estableciendo solo COMMITED READ en los procesos que así lo requieran.
- Evitar utilizar niveles de aislamiento como REPEAT READ que bloquea todos los registros que se proyectan en una sentencia SELECT.

En el caso de que el aplicativo Global Service Desk se desarrolle en IBM WebSphere Application Server hay que tener especial atención con las versiones antiguas del producto, la versión 7 establece el valor por defecto del nivel de aislamiento de los *datasources* en REPEAT READ. La versión 9 de este producto establece un nivel por defecto menos restrictivo (COMMITED READ).

- Eliminar del sistema *online* propio del aplicativo los procesos *batch*. Estos procesos (normalmente masivos) deben ir en horario fuera del *online* para no ocasionar problemas de concurrencia.

- Utilizar sentencias de espera por bloqueo "SET LOCK MODE TO WAIT NUMERO\_SEGUNDOS", de esta forma cuando se accede a un recurso que permanece bloqueado, la sesión espera un número definido de segundos antes de abortar con un error de recurso bloqueado. Dando tiempo a que el recurso se libere en un entorno con alta concurrencia.
- Generación de las tablas con nivel de bloqueo por defecto en modo registro, *lock mode row*, evitando utilizar bloqueos por defecto en modo página (*lock mode page*).

De esta forma, se asegura que cuando se leen o escriben datos en una tabla solo se bloquea los registros que se utilizan consiguiendo que no se bloquee toda una página entera de datos (Informix en UNIX tiene un tamaño de página de 4Kbytes) y todos los registros que esta contiene.

Información consultada:

"Setting the lock mode to wait"

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.perf.doc/ids\\_prf\\_422.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.perf.doc/ids_prf_422.htm)

"Niveles de aislamiento de IBM Data Server Driver para JDBC y SQLJ"

[https://www.ibm.com/support/knowledgecenter/es/SSEPGG\\_9.5.0/com.ibm.db2.luw.apdv.java.doc/src/tpc/imjcc\\_r0052429.html](https://www.ibm.com/support/knowledgecenter/es/SSEPGG_9.5.0/com.ibm.db2.luw.apdv.java.doc/src/tpc/imjcc_r0052429.html)

"Niveles de aislamiento"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_11.50.0/com.ibm.ddi.doc/ids\\_ddi\\_042.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_11.50.0/com.ibm.ddi.doc/ids_ddi_042.htm)

"DEADLOCK TIMEOUT y SET LOCK MODE"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_11.50.0/com.ibm.ddi.doc/ids\\_ddi\\_043.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_11.50.0/com.ibm.ddi.doc/ids_ddi_043.htm)

### 2.3 Pautas para mejorar el rendimiento de la base de datos

En vista a la creación de una base de datos optimizada y que ofrezca el mejor rendimiento para la aplicación Global Service Desk se definen las siguientes pautas:

- Cálculo de extents:

Se debe asegurar que los datos almacenados por una tabla en disco están lo más contiguos posibles, de esta forma se evitan tiempos de acceso a los datos más altos al tener que localizar el dato en distintos trozos del disco.

Por lo tanto, en función del número de registros estimado para la tabla en un periodo de tiempo, el tamaño generar de los registros y del índice, IBM establece una fórmula para asegurar que se reserva el espacio

suficiente contiguo en disco, que se detalla en el punto 3.3.1 Cálculo de *extents*.

- Creación de índices:

En función de las búsquedas y sentencias que necesite el aplicativo Global Service Desk es necesario optimizarlas y asegurar el acceso por índice a los datos.

Es por ello que se deben crear los índices necesarios (CREATE INDEX) para asegurar este acceso óptimo y evitar los accesos secuenciales no optimizados a la información.

Como consecuencia, instrucciones como "SET EXPLAIN ON" ayudan a visualizar el plan de acceso a los datos por el gestor, mostrando si el coste de las sentencias SQL es demasiado alto o no se accede por índice a los datos.

- Ejecución de estadísticas:

Instrucción como "UPDATE STATISTICS" generan una sincronización de información entre los datos reales en disco y la información que el gestor conoce de ellos, produciendo una actualización de los metadatos del catálogo del gestor.

De esta forma, se asegura que el algoritmo de acceso a la información del gestor siempre utilizará el camino mejor hacia los mismos.

Información consultada:

"ids pagesize and maximum number of extents"

[https://www.ibm.com/developerworks/community/blogs/informix\\_admins\\_blog/entry/ids\\_pagesize\\_and\\_maximum\\_number\\_of\\_extents5?lang=es](https://www.ibm.com/developerworks/community/blogs/informix_admins_blog/entry/ids_pagesize_and_maximum_number_of_extents5?lang=es)

"Actualizar las estadísticas para una rutina SPL"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_12.1.0/com.ibm.udr.doc/ids\\_udr\\_321.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_12.1.0/com.ibm.udr.doc/ids_udr_321.htm)

"La sentencia UPDATE STATISTICS"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_11.70.0/com.ibm.udr.doc/ids\\_udr\\_330.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_11.70.0/com.ibm.udr.doc/ids_udr_330.htm)

"Create and use functional indexes in Informix Dynamic Server"

<https://www.ibm.com/developerworks/data/library/techarticle/dm-0712wilcox/index.html>

"CREATE INDEX statement"

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.sqls.doc/ids\\_sqs\\_0401.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.sqls.doc/ids_sqs_0401.htm)

## 3. Modelo de datos Global Service Desk

### 3.1 Detalle del Modelo de datos

Se detallan todas las entidades definidas para el modelo Global Service Desk:

- **Tickets**

Toda incidencia o petición, tanto de forma manual, por un agente (a través de CHAT o TLF), como automática (vía portal WEB o procesos automáticos). Esto se registra en el campo `ticketType`.

Además, el campo `ticketuserCod` determina dentro de la tabla `Users` al grupo de soporte al que pertenece (tabla `Support`).

Por lo tanto, la aplicación puede determinar en que grupo de soporte está en ese momento su *ticket*.

Asimismo, el campo `ticketType` determina si el *ticket* es del tipo incidencia (INC) o petición (PET).

Se establecen los siguientes campos:

- `ticketNum` (Varchar) → Representa el número de *ticket*, puede contener números y letras.
  - Ejemplo de `ticketNum` → T000000001
- `ticketType` (Varchar) → Tipo de *ticket*, incidencia o petición, puede contener números y letras.
  - Ejemplo de `ticketType` → INC, o PET
- `ticketuserCod` (Varchar) → código de usuario, se utiliza el correo del usuario, puede contener números y letras (si aparece NULL es que se ha generado automáticamente por el sistema).
  - Ejemplo de `userCod` → [pereponsXXX@corp\\_mail.com](mailto:pereponsXXX@corp_mail.com)
- `ticketCreate` (DateTime) → Fecha y hora de creación del *ticket*.
  - Ejemplo `ticketCreate` → 20180129 14:05
- `ticketPriority` (Varchar) → Prioridad que se establece al *ticket*, puede contener números y letras.
  - Ejemplo de `ticketPriority` → P3
- `ticketService` (Varchar) → Código de servicio de un *ticket*, puede contener números y letras.
  - Ejemplo de `ticketService` → S00001
- `ticketChannel` (Varchar) → Canal de apertura del *ticket*, puede contener números y letras.
  - Ejemplo de `ticketChannel` → Chat
- `ticketCycle` (Varchar) → Almacena los grupos por los que pasa un *ticket*, puede contener números y letras.
  - Ejemplo de `ticketCycle` → L1

- ticketDescription (Lvarchar) → contiene la descripción de una incidencia o petición, se define como LONG VARCHAR y puede contener números y letras
  - Ejemplo de ticketDescription → Descripción del *ticket*
- ticketmessageCod (Vchar) → Representa el código del mensaje, puede contener números y letras.
  - Ejemplo de messageCod → MS0000456
- ticketDocument CLOB → Se utiliza para almacenar anexos si son necesarios para abrir el *ticket* (Word, Excel, PDF; etc).
  - Ejemplo de ticketDocument → Word, Excel, PDF; etc.

Integridad referencial:

Contiene una clave primaria por el código del *ticket* (ticketNum), admitiendo números y letras:

```
create unique index ix_Tickets on Tickets (ticketNum) using btree in
cycprod_idx1;
```

```
alter table Tickets add constraint primary key (ticketNum) constraint pk_Tickets
;
```

En consecuencia, solo los usuarios registrados podrán generar un *ticket* por lo que se establece una relación de dependencia entre el ticketuserCod y el Código del usuario de la tabla Users.

Es por ello, que no permite abrir un *ticket* si no está el usuario registrado en la tabla Users.

```
ALTER TABLE Tickets ADD CONSTRAINT (FOREIGN KEY (ticketuserCod)
REFERENCES Users (userCod) CONSTRAINT fk_Tickets_Users);
```

- **Users**

Representa todos los usuarios registrados en el sistema, solo estos usuarios podrán abrir un *ticket*.

El control de los usuarios se realiza mediante el correo corporativo ya que todos los usuarios de la empresa lo tienen.

Mediante un formulario del aplicativo Global Service Desk e introduciendo el correo electrónico del usuario, el aplicativo enviará un correo al usuario que debe confirmar ese registro.

Aunque el usuario se inserta en la tabla Users para almacenar los datos por si son necesarios (por ejemplo para campañas de captación), el campo userSubscribe será nulo si los usuarios no lo han confirmado por el enlace enviado por correo electrónico.

Solo en el momento en que se confirme quedará registrada la fecha de la subscripción al sistema.

De esta forma, el aplicativo cuando se crea un *ticket* debe verificar si el campo userSubscribe es distinto de NULL.

Además, no se permite el registro de ningún usuario que no pertenezca a una sede registrada en la tabla Centers, es decir el campo usercenterCod debe tener un valor válido y registrado en la tabla Centers.

Mediante el campo userType se registra si el usuario es un agente o Automático.

Se establecen los siguientes campos:

- userCod (Varchar) → código de usuario, se utiliza el correo del usuario, puede contener números y letras.
  - Ejemplo de userCod → pereponsXXX@corp\_mail.com
- userReg (DateTime) → Fecha y hora de cuando se registro el usuario.
  - Ejemplo de userReg → 20180201 13:06
- userUnsubscribe (DateTime) → Fecha de baja en el sistema.
  - Ejemplo de userUnsubscribe → NULL
- userType (Varchar) → Tipo de usuario, agente, automático, etc.
  - Ejemplo de userType → Agente
- userSubscribe (DateTime) → Fecha de alta en el sistema.
  - Ejemplo de userSubscribe → 20180201 13:06
- usercenterCod (Varchar) → código de la sede de la compañía, puede contener números y letras.
  - Ejemplo de centerCod → SD00001

- usersupportCod (Varchar) → Código del grupo de soporte, puede contener números y letras.
  - Ejemplo de supportCod → G00001

Integridad referencial:

Contiene una clave primaria por el código del usuario (userCod), que es el correo corporativo que todos los usuarios tienen:

```
create unique index ix_Users on Users (userCod) using btree in cycprod_idx1;
alter table Users add constraint primary key (userCod) constraint pk_Users ;
```

Establece una relación de dependencia con la tabla Centers de tal forma que los usuarios solo darán soporte a las sedes dadas de alta, asegurando el servicio adecuado a los usuarios de cada sede.

Por lo tanto, no se permite el registro de ningún usuario que no pertenezca a una sede registrada en la tabla Centers

```
ALTER TABLE Users ADD CONSTRAINT (FOREIGN KEY (usercenterCod)
REFERENCES Centers (centerCod) CONSTRAINT fk_Users_Centers);
ALTER TABLE Users ADD CONSTRAINT (FOREIGN KEY (usersupportCod)
REFERENCES Support (supportCod) CONSTRAINT fk_Users_Support);
```

- **Centers**

Representa las sedes de la compañía, los usuarios solo darán soporte a las sedes registradas en esta tabla.

Se establecen los siguientes campos:

- centerCod (Varchar)→ código de la sede de la compañía, puede contener números y letras.
  - Ejemplo de centerCod → SD00001
- centerCountry (Varchar)→ País de la sede de la compañía, puede contener números y letras (en principio solo caracteres).
  - Ejemplo de centerCountry → Turquía
- centerAddress (Varchar)→ Dirección de la sede de la compañía, puede contener números y letras.
  - Ejemplo de centerAddress → 52 Victory Street Izmir
- centerTypology (Varchar)→ Tipología de la sede de la compañía, puede contener números y letras (en principio solo caracteres).
  - Ejemplo de centerTypology → Centro de producción



### Integridad referencial:

Contiene una clave primaria por el código del centro (centerCod):

```
create unique index ix_Centers on Centers (centerCod) using btree in
cycprod_idx1;

alter table Centers add constraint primary key (centerCod) constraint
pk_Centers ;
```

- **Services**

Contiene el catálogo de servicios de la compañía tanto para incidencias como para peticiones.

Además cada registro en la tabla Services tiene asignada una prioridad (que de momento son: P1, P2, P3 y P4) y un tiempo de resolución (serviceTimeRes).

Se establecen los siguientes campos:

- serviceCod (Varchar)→ Representa el código de servicio de la compañía, puede contener números y letras.
  - Ejemplo de serviceCod → S00001
- serviceDescription (Varchar)→ Representa el código de servicio, puede contener números y letras.
  - Ejemplo de serviceDescription → SAP Egipto
- serviceGroup (Varchar)→ Grupo al que pertenece el servicio, puede contener números y letras.
  - Ejemplo de serviceGroup → Aplicaciones
- servicePriority (Varchar)→ Prioridad del grupo al que pertenece el servicio, puede contener números y letras.
  - Ejemplo de servicePriority → P2
- serviceTimeRes (Varchar)→ Tiempo de resolución del servicio, puede contener números y letras.
  - Ejemplo de serviceTimeRes → 24h

### Integridad referencial:

Contiene una clave primaria por el código del centro (centerCod):

```
create unique index ix_Services on Services (serviceCod) using btree in
cycprod_idx1;

alter table Services add constraint primary key (serviceCod) constraint
pk_Services ;
```

- **State**

Registra todos los cambios de estado por los que atraviesan los *tickets*, permitiendo almacenar información trazable sobre en qué momento ha cambiado de estado un *ticket* (*stateChangeDate*), quién lo ha tratado (*userCod*), por lo estados que ha pasado un *ticket* (*stateNew*), etc.

Se establecen los siguientes campos:

- *stateCodState* (serial) → Guarda el código de la operación para su registro en el sistema.
  - Ejemplo de *stateCodState* → 1000
- *stateTicketNum* (Varchar) → Representa el número de *ticket*, puede contener números y letras.
  - Ejemplo de *stateTicketNum* → T000000001
- *stateChangeDate* (DateTime) → Fecha y hora del cambio de estado del *ticket*.
  - Ejemplo *stateChangeDate* → 20170615 12:01
- *stateUserCod* (Varchar) → Agente que realiza el cambio.
  - Ejemplo *stateUserCod* → silviagomezYYY@corp\_mail.com
- *stateNew* (Varchar) → Estado nuevo en el que se queda un *ticket* después del cambio.
  - Ejemplo *stateNew* → Resuelto

Integridad referencial:

Contiene una clave primaria por el código de la operación (*stateCodState*), se crea seriada para que lo autogenera la base de datos.

```
create unique index ix_State on State (stateCodState) using btree in  
cycprod_idx1;
```

```
alter table State add constraint primary key (stateCodState) constraint pk_State  
;
```

Establece una relación con la tabla Tickets de tal forma que solo los *tickets* contenidos en la tabla Tickets pueden almacenar su estado en la tabla State.

```
ALTER TABLE State ADD CONSTRAINT (FOREIGN KEY (stateTicketNum)  
REFERENCES Tickets (ticketNum) CONSTRAINT fk_State_Tickets);
```

- **Messages**

Registra todos los mensajes que se originan en el sistema, incorporando el requisito del servicio de mensajera que informará al usuario de cualquier cambio en el estado de su *ticket*.

De esta forma, se almacenan todos los mensajes enviados y recibidos así como información de la hora del mensaje, quién lo recibe, el estado del mismo, etc.

Se establecen los siguientes campos:

- messageCod (Varchar) → Representa el código del mensaje, se permiten números y letras.
  - Ejemplo de messageCod → MS0000456
- messageTime (DateTime) → Fecha y hora del mensaje, puede contener números y letras.
  - Ejemplo de messageTime → 20170215 14:05
- messageReceiver (Varchar) → Receptor del mensaje, puede contener números y letras.
  - Ejemplo de messageReceiver → pacoperezYYY@corp\_mail.com
- messageState (Varchar) → Estado del mensaje, puede contener números y letras (en principio solo caracteres).
  - Ejemplo de messageState → Recibido
- messageTicketNum (Varchar) → Representa el número de *ticket*, puede contener números y letras.
  - Ejemplo de messageTicketNum → T000000001

Integridad referencial:

Contiene una clave primaria por el código del mensaje (messageCod)

```
create unique index ix_Messages on Messages (messageCod) using btree in  
cycprod_idx1;
```

```
alter table Messages add constraint primary key (messageCod) constraint  
pk_Messages ;
```

Establece una relación con la tabla Tickets de tal forma que la tabla Messages almacena todos los mensajes que contiene cada *ticket*.

```
ALTER TABLE Messages ADD CONSTRAINT (FOREIGN KEY  
(messageTicketNum) REFERENCES Tickets (ticketNum) CONSTRAINT  
fk_Messages_Tickets);
```

- **Support**

Almacena los grupos de soporte.

Se establecen los siguientes campos:

- supportCod (Varchar) → Código del grupo de soporte, puede contener números y letras.
  - Ejemplo de supportCod → G00001

- supportName (Varchar) → Nombre del grupo de soporte, puede contener números y letras.
  - Ejemplo de supportName → Soporte Inicial
- supportType (Varchar) → Tipo de soporte, puede contener números y letras.
  - Ejemplo de supportType → L1
- supportuserCod (Varchar) → código de usuario, se utiliza el correo del usuario, puede contener números y letras.
  - Ejemplo de userCod → pereponsXXX@corp\_mail.com

Integridad referencial:

Contiene una clave primaria por el código de soporte (supportCod)

```
create unique index ix_Services on Services (serviceCod) using btree in
cycprod_idx1;
```

```
alter table Services add constraint primary key (serviceCod) constraint
pk_Services ;
```

• **Log\_procedures**

Esta tabla pertenece a la administración de la aplicación y registra todas las ejecuciones de los procedimientos almacenados que componen la lógica de la aplicación.

Se establecen los siguientes campos:

- procCod (Serial) → Código autogenerated del registro de la operación realizada.
  - Ejemplo de procCod → 1000
- procName (Varchar) → Registra el nombre del procedimiento.
  - Ejemplo de procName → Alta\_Usuario
- procParamIn (Lvarchar) → Parámetros de entrada separados por “//”.
  - Ejemplo de procParamIn → User8@corp\_mail.com//2018-05-24  
19:59:32.963//2018-05-24  
19:59:32.963//Administrador//2018-05-24  
19:59:32.963//SD00001//G00001
- procParamOut (Lvarchar) → Parámetros de salida (RSP).
  - Ejemplo de procParamOut → Error -111 en: Users , para más detalle compruebe el código ISAM mostrado
- procReg (DateTime) → Registra la fecha y hora de la ejecución del procedimiento.
  - Ejemplo de procReg → 2018-05-24 19:59:32.963

### Integridad referencial:

Contiene una clave primaria por el código de la operación (procCod):

```
create unique index ix_Log_procedures on Log_procedures (procCod) using  
btree in cycprod_idx1;  
  
alter table Log_procedures add constraint primary key (procCod) constraint  
pk_Log_procedures ;
```

- **timeSollnc**

Esta tabla pertenece al repositorio estadístico de la aplicación y registra la condición:

- Dado un mes cualquiera, tiempo medio de resolución de incidencias por cada tipo de prioridad.

### Se establecen los siguientes campos:

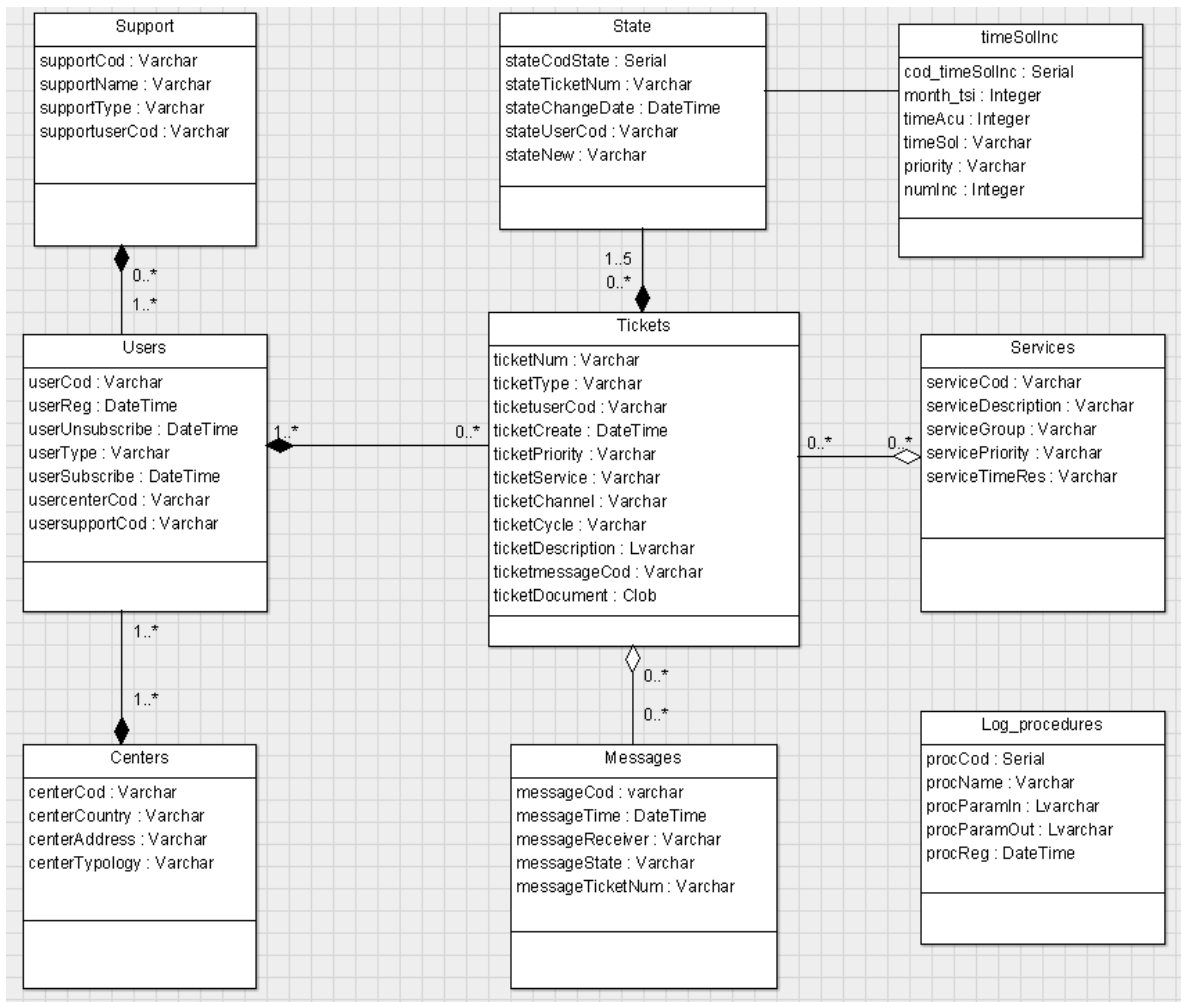
- cod\_timeSollnc (Serial) → Código autogenerated del registro de la operación realizada.
  - Ejemplo de procCod → 1000
- month\_tsi (Integer) → Registra el mes de la media.
  - Ejemplo de month\_tsi → 6
- timeAcu (Integer) → Registra el acumulado del tiempo de resolución de las incidencias (en minutos).
  - Ejemplo de timeAcu → 3157
- timeSol (Varchar) → Registra la media de tiempo de resolución acumulado / número de incidencias (en minutos).
  - Ejemplo de timeSol → 3480
- priority (Varchar) → Registra la prioridad para la media.
  - Ejemplo de priority → P1
- numInc (Integer) → Registra el número de incidencias por mes.
  - Ejemplo de numInc → 2

### Integridad referencial:

Contiene una clave primaria por el código de la operación (cod\_timeSollnc):

```
reate unique index ix_timeSollnc on timeSollnc (cod_timeSollnc) using btree in  
cycprod_idx1;  
  
alter table timeSollnc add constraint primary key (cod_timeSollnc) constraint  
pk_timeSollnc;
```

### 3.2 Diagrama UML del modelo de Global Service Desk



Información consultada:

"Qué es el lenguaje unificado de modelado (UML)"

<https://www.lucidchart.com/pages/es/qu%C3%A9-es-el-lenguaje-unificado-de-modelado-uml>

"Diagrama de clases"

[https://es.wikipedia.org/wiki/Diagrama\\_de\\_clases](https://es.wikipedia.org/wiki/Diagrama_de_clases)

### 3.3 Creación de prototipo

Una vez diseñado el aplicativo se procede a la implantación del modelo en la base de datos.

Mediante sintaxis SQL general y características propias de Informix se crea el modelo y sus relaciones de dependencias, estableciendo integridad referencial.

Se establece un modelo base de mil incidencias para reservar espacio contiguo en disco (*extents*) de cada tabla que se procede a detallar en el siguiente punto.

### 3.3.1 Cálculo de *extents*.

Mediante el cálculo de los *extents* se reserva espacio contiguo en disco para mejorar los accesos a la información contenida en las tablas de la base de datos.

De esta forma, la información es más fácil de localizar y se lee de forma secuencial en disco, esta característica influye mucho en el rendimiento del aplicativo y, con ello, en las operaciones sobre las tablas de la base de datos.

El fabricante del producto Informix (IBM) establece la siguiente fórmula para el cálculo de *extents* en base a un cálculo estimado de registros, siendo lo habitual calcular los *extents* en función de un crecimiento anual.

Según las siguientes fórmulas:

#### Estimating Extent Size of Attached Index

- Index extent size = (index\_key\_size / table\_row\_size) \* table\_extent\_size
- Index next extent size = (index\_key\_size/table\_row\_size)\* table\_next\_extent\_size

#### Estimating Extent Size of Detached Index

- Detached Index extent size = ( (index\_key\_size + 9) / table\_row\_size) \* table\_extent\_size

*For example, suppose you have the following values:*

*index\_key\_size = 8*

*bytes*

*table\_row\_size = 33*

*bytes*

*table\_extent\_size = 150 \* 2-kilobyte page*

*index\_key\_size is the total widths of the indexed column or columns.*

*table\_row\_size is the sum of all the columns in the row.*

*table\_extent\_size is the value that you specify in the EXTENT SIZE keyword*

*of the CREATE TABLE statement.*

*The above formula calculates the extent size as*

follows:

$$\begin{aligned} \text{Detached Index extent size} &= (8 + 9) / 33 * 150 * 2 - \\ &\text{kilobyte page} \\ &= (17/33) * 300 \\ &\text{kilobytes} \\ &= 154 \text{ kilobytes} \end{aligned}$$

Siendo el NEXT EXTENTS aproximadamente un 20% DEL EXTENTS\_PRINCIPAL.

El no tener los *extents* bien dimensionados ocasiona que la tabla se encuentre fragmentada en el disco, lo que provoca que el acceso a la información no esté optimizado, al buscar la información contenida en las tablas en espacios repartidos por los discos (aunque los nuevos armarios y robot de discos minimizan este impacto) influyendo en el rendimiento.

Por lo tanto, es necesario calcular el primer *extents* (*first extent*) de cada tabla, así como el siguiente *extent* que el gestor necesitará (*next extent*), operación básica de un administrador de base de datos.

Es por ello, que se crea un archivo Excel en el que se alojan las fórmulas con el fin de crear una plantilla fácil de utilizar y que contribuye a otorgar un valor añadido al proyecto.

Indicando el número de registros estimado que se preveen para un año, junto con el Row Size e Index Size devuelto por el gestor (el comando dbschema o en el catálogo de sysmaster se obtiene esta información de estos parámetros):

Portapapeles		Fuente		Alineación		Número					
H5		= (REDONDEAR.MAS(D5/(TRUNCAR(4068/(F5+4)))))*4									
1	A	B	C	D	E	F	G	H	I	J	K
2	CALCULAR EXTENTS										
3											
4	TABLA	ACTUAL	MEDIO(6m)	LARGO (12m)	Row Size	Index Size	Extent Size Data (KB)	Extent Size Index (KB)	First Extent (Data+Idx)	Next Extent (20%)	
5				1000	2230	30	4000	54	4054	812	
6	TABLA DE PRUEBA										

Portapapeles		Fuente		Alineación		Número					
I5		= REDONDEAR.MAS(((G5)/F5)*H5;0)									
1	A	B	C	D	E	F	G	H	I	J	K
2	CALCULAR EXTENTS										
3											
4	TABLA	ACTUAL	MEDIO(6m)	LARGO (12m)	Row Size	Index Size	Extent Size Data (KB)	Extent Size Index (KB)	First Extent (Data+Idx)	Next Extent (20%)	
5				1000	2230	30	4000	54	4054	812	
6	TABLA DE PRUEBA										
7											



Portapapeles		Fuente		Alineación		Número				
J5		=H5+I5								
A	B	C	D	E	F	G	H	I	J	K
<b>CALCULAR EXTENTS</b>										
<b>TABLA</b>	ACTUAL	MEDIO(6m)	LARGO (12m)	Row Size	Index Size	Extent Size Data (KB)	Extent Size Index (KB)	<b>First Extent (Data+Idx)</b>	<b>Next Extent (20%)</b>	
<b>TABLA DE PRUEBA</b>			1000	2230	30	4000	54	<b>4054</b>	<b>812</b>	

Portapapeles		Fuente		Alineación		Número				
K5		=MULTIPLO.SUPERIOR(MAX((J5*0,2);4);4)								
A	B	C	D	E	F	G	H	I	J	K
<b>CALCULAR EXTENTS</b>										
<b>TABLA</b>	ACTUAL	MEDIO(6m)	LARGO (12m)	Row Size	Index Size	Extent Size Data (KB)	Extent Size Index (KB)	<b>First Extent (Data+Idx)</b>	<b>Next Extent (20%)</b>	
<b>TABLA DE PRUEBA</b>			1000	2230	30	4000	54	<b>4054</b>	<b>812</b>	

Por lo tanto, el cálculo de *extents* del prototipo del modelo queda de la siguiente forma:

<b>CALCULAR EXTENTS</b>										
TABLA	ACTUAL	MEDIO(6m)	LARGO (12m)	Row Size	Index Size	Extent Size Data (KB)	Extent Size Index (KB)	First Extent (Data+Idx)	Next Extent (20%)	
<b>Centers</b>			1000	204	56	212	59	<b>271</b>	<b>56</b>	
<b>Services</b>			1000	255	56	268	59	<b>327</b>	<b>68</b>	
<b>State</b>			1000	163	56	168	58	<b>226</b>	<b>48</b>	
<b>Messages</b>			1000	163	56	168	58	<b>226</b>	<b>48</b>	
<b>Support</b>			1000	204	56	212	59	<b>271</b>	<b>56</b>	
<b>Users</b>			1000	234	168	236	170	<b>406</b>	<b>84</b>	
<b>Tickets</b>			1000	2421	224	4000	371	<b>4371</b>	<b>876</b>	

Se envía el Excel creado anexo a la memoria con el nombre:

Estimacion\_EXTENTS\_Tablas.xls

Información consultada:

"DB Import"

[http://kb.deister.net/index.php/DB\\_Import](http://kb.deister.net/index.php/DB_Import)

"ids pagesize and maximum number of extents"

[https://www.ibm.com/developerworks/community/blogs/informix\\_admins\\_blog/entry/ids\\_pagesize\\_and\\_maximum\\_number\\_of\\_extents5?lang=es](https://www.ibm.com/developerworks/community/blogs/informix_admins_blog/entry/ids_pagesize_and_maximum_number_of_extents5?lang=es)

"Extent sizes for tables in a dbspace"

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_11.50.0/com.ibm.perf.doc/ids\\_prf\\_311.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_11.50.0/com.ibm.perf.doc/ids_prf_311.htm)

### 3.3.2 Modelo de usuarios y permisos.

Para realizar las pruebas de funcionamiento del modelo es necesario crear usuarios en el sistema (Sistema Operativo de las máquinas AIX), así como la asignación de los permisos necesarios.

Se crean los usuarios de pruebas USU1\_GSD y USU2\_GSD, asignando los permisos de conexión a la base de datos, así como los siguientes sobre el modelo:

```
-- PERMISOS DE CONEXION A LA BASE DE DATOS GSD

GRANT CONNECT TO USU1_GSD;
GRANT CONNECT TO USU2_GSD;

-- PERMISOS DE SELECT,INSERT,DELETE,UPDATE SOBRE EL MODELO
AL USUARIO USU1_GSD;

GRANT SELECT,INSERT,DELETE,UPDATE ON Tickets TO USU1_GSD AS
GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON Users TO USU1_GSD AS
GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON Centers TO USU1_GSD AS
GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON Services TO USU1_GSD AS
GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON State TO USU1_GSD AS
GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON Messages TO USU1_GSD
AS GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON Support TO USU1_GSD AS
GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON Log_procedures TO
USU1_GSD AS GSD;

-- PERMISOS DE SELECT,INSERT,DELETE,UPDATE SOBRE EL MODELO
AL USUARIO USU2_GSD;

GRANT SELECT,INSERT,DELETE,UPDATE ON Tickets TO USU2_GSD AS
GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON Users TO USU2_GSD AS
GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON Centers TO USU2_GSD AS
GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON Services TO USU2_GSD AS
GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON State TO USU2_GSD AS
GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON Messages TO USU2_GSD
AS GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON Support TO USU2_GSD AS
GSD;
GRANT SELECT,INSERT,DELETE,UPDATE ON Log_procedures TO
USU2_GSD AS GSD;
```

Información consultada:

"Informix Guide to SQL"

[http://www.cs.uoi.gr/~pitoura/informix-doc/Informix\\_Guide\\_to\\_SQL-Syntax.pdf](http://www.cs.uoi.gr/~pitoura/informix-doc/Informix_Guide_to_SQL-Syntax.pdf)

"Sintaxis de Informix SQL"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_11.50.0/com.ibm.sqlnode.doc/sql.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_11.50.0/com.ibm.sqlnode.doc/sql.htm)

### 3.3.3 Creación de la base de datos Global Service Desk.

Los *scripts* de creación del modelo engloban:

- Creación de la base de datos GSD.
- Creación de índices, primary key y foreign key en el dbspace de índices cycprod\_idx1.
- Tablas en el dbspace de datos cycprod\_dat1.
- Relaciones de integridad referencial.
- Permisos de conexión y de utilización del modelo.
- Procedimientos de gestión del modelo, conteniendo toda la lógica.
- Se incorporan los *scripts* de creación de todo el modelo para el aplicativo Global Service Desk.

Se documenta el *log* de ejecución de la estructura y datos del modelo en:

ANEXO C: Generación de *scripts* SQL del modelo.

Los *scripts* de generación de la estructura y de los datos del modelo GSD (sin el modelo del repositorio que tiene sus propios *scripts* documentados en el apartado del repositorio estadístico de la memoria) son:

PROTOTIPO\_ESTRUCTURA.sql

PROTOTIPO\_DATOS.sql

Para su ejecución se utiliza el intérprete de SQL, DBACCESS incorporado con Informix.

Información consultada:

"Informix Administrators Blog"

[https://www.ibm.com/developerworks/community/blogs/informix\\_admins\\_blog/entry/ids\\_page\\_size\\_and\\_maximum\\_number\\_of\\_extents5?lang=es](https://www.ibm.com/developerworks/community/blogs/informix_admins_blog/entry/ids_page_size_and_maximum_number_of_extents5?lang=es)

"How to find the number of extents used by a table using oncheck"

<http://www-01.ibm.com/support/docview.wss?uid=swg21153986>

"Extent sizes for tables in a dbspace"

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_11.50.0/com.ibm.perf.doc/ids\\_prf\\_311.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_11.50.0/com.ibm.perf.doc/ids_prf_311.htm)

### 3.3.4 Integridad del modelo de base de datos.

Se establece la siguiente integridad referencial en el modelo de base de datos con el fin de que cuando se crea un *ticket* originado por una incidencia o petición contenga toda la relación necesaria para que el modelo mantenga la integridad de la información.

Para la creación de las claves primarias se ha tenido en cuenta el código identificativo del elemento.

De tal forma que la clave primaria (que es un índice único dentro de la base de datos) se apoye en un índice único creado para tal fin y alojado en el dbspace de índices (cycprod\_idx1), de esta forma se crea el modelo configurado para mejorar la administración y organizar adecuadamente la arquitectura de la instancia GSD (Global Service Desk):

```
create unique index ix_Centers on Centers (centerCod) using btree in
cycprod_idx1;

alter table Centers add constraint primary key (centerCod) constraint
pk_Centers ;

create unique index ix_Services on Services (serviceCod) using btree in
cycprod_idx1;

alter table Services add constraint primary key (serviceCod) constraint
pk_Services ;

create unique index ix_State on State (stateCodState) using btree in
cycprod_idx1;

alter table State add constraint primary key (stateCodState) constraint pk_State
;

create unique index ix_Messages on Messages (messageCod) using btree in
cycprod_idx1;

alter table Messages add constraint primary key (messageCod) constraint
pk_Messages ;

create unique index ix_Support on Support (supportCod) using btree in
cycprod_idx1;

alter table Support add constraint primary key (supportCod) constraint
pk_Support ;

create unique index ix_Users on Users (userCod) using btree in cycprod_idx1;
```

```

alter table Users add constraint primary key (userCod) constraint pk_Users ;

create unique index ix_Tickets on Tickets (ticketNum) using btree in
cycprod_idx1;

alter table Tickets add constraint primary key (ticketNum) constraint pk_Tickets
;
create unique index ix_Log_procedures on Log_procedures (procCod) using
btree in cycprod_idx1;

alter table Log_procedures add constraint primary key (procCod) constraint
pk_Log_procedures ;

create unique index ix_timeSollnc on timeSollnc (cod_timeSollnc) using btree
in cycprod_idx1;

alter table timeSollnc add constraint primary key (cod_timeSollnc) constraint
pk_timeSollnc;

```

La integridad referencial se establece en función de los campos de la tabla hija que se establecen con la clave primaria de la tabla padre, de la siguiente forma:

```

ALTER TABLE Users ADD CONSTRAINT (FOREIGN KEY (usercenterCod)
REFERENCES Centers (centerCod) CONSTRAINT fk_Users_Centers);
ALTER TABLE Users ADD CONSTRAINT (FOREIGN KEY (usersupportCod)
REFERENCES Support (supportCod) CONSTRAINT fk_Users_Support);

ALTER TABLE State ADD CONSTRAINT (FOREIGN KEY (stateTicketNum)
REFERENCES Tickets (ticketNum) CONSTRAINT fk_State_Tickets);

ALTER TABLE Tickets ADD CONSTRAINT (FOREIGN KEY (ticketService)
REFERENCES Services (serviceCod) CONSTRAINT fk_Tickets_Services);
ALTER TABLE Messages ADD CONSTRAINT (FOREIGN KEY (messageTicketNum)
REFERENCES Tickets (ticketNum) CONSTRAINT
fk_Messages_Tickets);
ALTER TABLE Tickets ADD CONSTRAINT (FOREIGN KEY (ticketuserCod)
REFERENCES Users (userCod) CONSTRAINT fk_Tickets_Users);

```

Es por ello que el modelo de entidades:

- Users → Se relaciona con las tablas Centers y Support, donde cada usuario debe estar alojado en un centro y al soporte asignado.
- Tickets → Se relaciona con las tablas Services y Users, para cada incidencia o petición que se genere y guarde información de todo el modelo necesario.
- State → Se relaciona con las tablas Tickets, para almacenar los diferentes estados por los que atraviesa un *ticket*.

- Message → Se relaciona con las tablas Tickets, para almacenar los diferentes mensajes que contiene un *ticket*.

De esta forma, los *tickets* almacenan sus estados en la tabla State, se le asignan los usuarios de soporte (Users) y se establecen los mensajes definidos en el *ticket* en la tabla Message así como el servicio al que está fijado (Services).

## 4. Funcionamiento del prototipo

En los siguientes puntos se detalla cómo funciona el prototipo y cómo debe interactuar la aplicación con la base de datos.

La forma de interactuar con la base de datos es mediante procedimientos almacenados, por lo que se generan procedimientos para todas las operaciones de CONSULTAS-ALTAS-BAJAS-MODIFICACIONES.

Cualquier interacción (que no sea de administración, que se permitirá mediante sentencias SQL contra la base de datos) tanto de forma manual en la apertura de un *ticket* como automática se hace mediante procedimientos almacenados.

Los procedimientos controlan las excepciones y disponen de mensajes de error para todas ellas, incluso mensajes definidos a medida, mediante controles del tipo:

```

ON EXCEPTION SET error_num, error_isam, error_txt

IF error_isam = 0 THEN

    LET RSP = 'OK';

ELSE

    IF error_isam = -100 THEN

        LET RSP = '-100 El usuario ya existe';

    ELSE

        LET RSP = 'Error ' || error_isam || ' en: ' || error_txt || ' , para más
detalle compruebe el codigo ISAM mostrado ';

    END IF;

END IF;

END EXCEPTION WITH RESUME

```

En la gestión de errores se capturan tanto el SQLCODE como el código ISAM, gestionando principalmente el código ISAM ya que es un error que ofrece un nivel más elevado de detalle, siendo el SQLCODE un error de información más genérica.

Todos los procedimientos retornan un OK si la ejecución resultó correcta, de lo contrario se muestra el código de error a medida o devuelto por la base de datos y el objeto donde se ha producido, dando con ello toda la información posible, ya que Informix no permite almacenar directamente el mensaje del código de error.

Además, se insertan en la tabla Log\_procedures todas las ejecuciones de los procedimientos para guardar un registro completo, guardando la siguiente información:

- Código autogenerated por la base de datos (serial).
- Nombre del procedimiento.
- Parámetros de entrada separados por "//".
- Parámetros de salida.
- Fecha y hora de la ejecución.

A continuación, se detallan las pruebas realizadas y el funcionamiento del modelo.

Anexo a la memoria se envían los *scripts* tanto de la creación de la estructura del modelo como de los datos de prueba del funcionamiento del prototipo (los *scripts* del repositorio están documentados en el apartado a tal efecto) en los siguientes *scripts*:

```
PROTOTIPO_ESTRUCTURA.sql
PROTOTIPO_DATOS.sql
PROTOTIPO_PROCEDURES_ESTADOS_TICKET.sql
PROTOTIPO_PROCEDURES_MENSAJES_TICKET.sql
PROTOTIPO_PROCEDURES_TICKET.sql
PROTOTIPO_PROCEDURES_USUARIO.sql
```

Por lo tanto, el modelo se crea de la siguiente forma y en el siguiente orden:

```
dbaccess -e sysmaster@prod_tcp PROTOTIPO_ESTRUCTURA.sql >
PROTOTIPO_ESTRUCTURA.log 2>&1

dbaccess -e GSD@prod_tcp PROTOTIPO_DATOS.sql >
PROTOTIPO_DATOS.log 2>&1

dbaccess -e GSD@prod_tcp
PROTOTIPO_PROCEDURES_ESTADOS_TICKET.sql >
PROTOTIPO_PROCEDURES_ESTADOS_TICKET.log 2>&1
```

```
dbaccess          -e          GSD@prod_tcp
PROTOTIPO_PROCEDURES_MENSAJES_TICKET.sql >
PROTOTIPO_PROCEDURES_MENSAJES_TICKET.log 2>&1

dbaccess -e GSD@prod_tcp PROTOTIPO_PROCEDURES_TICKET.sql >
PROTOTIPO_PROCEDURES_TICKET.log 2>&1

dbaccess -e GSD@prod_tcp PROTOTIPO_PROCEDURES_USUARIO.sql >
PROTOTIPO_PROCEDURES_USUARIO.log 2>&1
```

Información consultada:

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.sqls.doc/ids\\_sqs\\_1351.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.sqls.doc/ids_sqs_1351.htm)  
<http://www.oninit.com/manual/informix/100/sqls/sqls946.htm>  
<http://www.oninit.com/manual/informix/english/docs/dbdk/is40/sqls/0311.html>  
<http://www.pacs.tju.edu/informix/answers/english/docs/dbdk/infoshelf/sqlt/14.fm12.html>  
<https://stackoverflow.com/questions/15746756/limitation-of-error-code-on-exception-in-stored-procedure>  
<https://stackoverflow.com/questions/4825362/informix-stored-procedure-return-values>  
[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.sqls.doc/ids\\_sqs\\_0884.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.sqls.doc/ids_sqs_0884.htm)  
<http://www.oninit.com/manual/informix/english/docs/dbdk/is40/jdbc/data5.html>

## 4.1 Gestión de usuarios

Se procede a continuación a detallar el ciclo de funcionamiento de la gestión de usuarios y su control de errores, mediante procedimientos almacenados se controlan todas las operaciones.

Todos los procedimientos de gestión de usuarios se generan en el *script*:

PROTOTIPO\_PROCEDURES\_USUARIO.sql

### 4.1.1 Consulta de usuarios

Mediante el procedimiento "GSD".Consulta\_Usuario se consulta un usuario en el sistema introduciendo el código del usuario, de la siguiente forma:

- PARAMETROS DE ENTRADA → CODIGO\_USUARIO
- PARAMETROS DE SALIDA → Todos los campos de la tabla Users (incluido el código de retorno RSP).



```
echo "  
  
execute procedure Consulta_Usuario('User8@corp_mail.com');  
  
"|dbaccess -e GSD@prod_tcp
```

### RESULTADO:

```
rsp          OK  
usercod_out  User8@corp_mail.com  
userreg_out  2018-05-25 13:21:16.753  
userunsubscribe_o+ 2018-05-25 13:21:16.753  
usertype_out  CLIENTE  
usersubscribe_out 2018-05-25 13:21:16.753  
usercentercod_out SD00001  
usersupportcod_out G00001
```

Este procedimiento, además del código RSP, devuelve los resultados de la consulta.

Si el usuario no existe el procedimiento devuelve (error controlado):

```
rsp          NO EXISTE EL USUARIO PRUEBA@corp_mail.com en la tabla  
Users  
usercod_out  
userreg_out  
userunsubscribe_o+  
usertype_out  
usersubscribe_out  
usercentercod_out  
usersupportcod_out
```

Si la tabla no existe o no está accesible por cualquier motivo (como por bloqueos, error -143) devuelve el código de error:

```
Error -143 En el objeto Users, para más detalle compruebe el código ISAM  
mostrado
```

Se registra la operación en la tabla Log\_procedures:

```
proccod      47  
procname     Consulta_Usuario  
procparamin  User8@corp_mail.com  
procparamout OK  
procreg      2018-05-25 14:40:16.004  
  
proccod      48  
procname     Consulta_Usuario
```

```
proccod      7
procname     Alta_Usuario
procparamin  PRUEBA@corp_mail.com
procparamout NO EXISTE EL USUARIO PRUEBA@corp_mail.com en la
             tabla Users
procreg      2018-05-25 14:40:30.886
```

#### 4.1.2 Alta de usuarios

Mediante el procedimiento "GSD".Alta\_Usuario se inserta un usuario en el sistema, de la siguiente forma:

- PARAMETROS DE ENTRADA → Todos los campos de la tabla Users.
- PARAMETROS DE SALIDA → RSP (RESULTADO DE LA OPERACIÓN).

```
echo "
execute                                     procedure
Alta_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'Administrador',C
CURRENT,'SD00001','G00001');
"|dbaccess -e GSD@prod_tcp
```

#### RESULTADO:

```
OK
```

Si el usuario ya existe y se vuelve a insertar el procedimiento devuelve (error controlado):

```
Error -100 El usuario ya existe
```

Si la tabla no existe o no está accesible por cualquier motivo (como por bloqueos, error -143) devuelve el código de error:

```
Error -143 En el objeto Users, para más detalle compruebe el código ISAM
mostrado
```

Se registra la operación en la tabla Log\_procedures:

```
proccod      7
procname     Alta_Usuario
procparamin  User8@corp_mail.com//2018-05-24 21:03:04.139//2018-05-24
21:03:04
             .139//Administrador//2018-05-24 21:03:04.139//SD00001//G00001
procparamout -100 El usuario ya existe
procreg      2018-05-24 21:03:04.139
```

### 4.1.3 Baja de usuarios

Mediante el procedimiento "GSD".Baja\_Usuario se borra un usuario en el sistema, de la siguiente forma y pasando al procedimiento código del usuario:

- PARAMETROS DE ENTRADA→Código de usuario.
- PARAMETROS DE SALIDA→ RSP (RESULTADO DE LA OPERACIÓN).

```
echo "  
execute procedure Baja_Usuario('User8@corp_mail.com');  
"|dbaccess -e GSD@prod_tcp
```

#### RESULTADO:

```
OK
```

Se comprueba que el usuario exista, de no ser así el procedimiento devuelve (error controlado) y no realiza el *delete*:

```
NO EXISTE EL USUARIO User8@corp_mail.com en la tabla Users
```

Esta comprobación se realiza de la siguiente forma:

```
LET contador = 0;  
  
SELECT COUNT(*)  
INTO contador  
from Users where userCod = userCod_1;  
  
IF contador = 0 THEN  
  
    LET RSP = ' NO EXISTE EL USUARIO ' || userCod_1 || ' en la tabla  
Users';  
  
ELSE  
  
    delete from Users where userCod = userCod_1;  
    LET RSP = 'OK';  
  
END IF
```

Si la tabla no existe o no está accesible por cualquier motivo (como por bloqueos, error -143) devuelve el código de error:

```
Error -143 En el objeto Users, para más detalle compruebe el codigo ISAM  
mostrado
```

Se registra la operación en la tabla Log\_procedures:

```
proccod    14
procname   Baja_Usuario
procparamin User8@corp_mail.com
procparamout OK
procreg    2018-05-24 21:33:30.948

proccod    15
procname   Baja_Usuario
procparamin User8@corp_mail.com
procparamout NO EXISTE EL USUARIO User8@corp_mail.com en la tabla
Users
procreg    2018-05-24 21:33:35.947
```

#### 4.1.4 Actualización de usuarios

Mediante el procedimiento "GSD".Actualiza\_Usuario se actualiza un usuario en el sistema, de la siguiente forma y pasando al procedimiento los parámetros totales del usuario:

- PARAMETROS DE ENTRADA → Todos los campos de la tabla Users.
- PARAMETROS DE SALIDA → RSP (RESULTADO DE LA OPERACIÓN).

```
echo "
execute                                     procedure
Actualiza_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'CLIENTE',C
URRENT,'SD00001','G00001');
"|dbaccess -e GSD@prod_tcp
```

RESULTADO:

OK

Se comprueba que el usuario exista y que el código de Centro y de soporte sea válido, de no ser así el procedimiento devuelve (error controlado) y no realiza la actualización, se realiza el control de errores de la siguiente forma:

```
LET contador = 0;
SELECT COUNT(*)
INTO contador
from Users where userCod = userCod_1;
```

```

IF contador = 0 THEN

    LET RSP = ' NO EXISTE EL USUARIO ' || userCod_1 || ' en la tabla
Users';

ELSE

    SELECT COUNT(*)
    INTO contador2
    from Users where usercenterCod = usercenterCod_1;

    IF contador2 = 0 THEN

        LET RSP = ' NO EXISTE EL CODIGO DE CENTRO ' || userCod_1 || ' en
la tabla Centers ';

        ELSE

            SELECT COUNT(*)
            INTO contador3
            from Users where usersupportCod = usersupportCod_1;

            IF contador3 = 0 THEN

                LET RSP = ' NO EXISTE EL CODIGO DE SOPORTE ' ||
usersupportCod_1 || ' en la tabla Support';

                ELSE

                    update Users
                    set
userCod=userCod_1,userReg=userReg_1,userUnsubscribe=userUnsubscribe_
1,userType=userType_1,userSubscribe=userSubscribe_1,usercenterCod=user
centerCod_1,usersupportCod=usersupportCod_1
                    where userCod = userCod_1;

                    LET RSP = 'OK';

                END IF

            END IF

        END IF

    END IF

```

Por lo tanto, si introducimos un usuario no existente se devuelve:

**NO EXISTE EL USUARIO USUSUS@corp\_mail.com en la tabla Users**

Si se introduce un código de centro que no existe:

NO EXISTE EL CODIGO DE CENTRO User8@corp\_mail.com en la tabla Centers

Si se introduce un código de centro que no existe o no es válido:

NO EXISTE EL CODIGO DE CENTRO SDSDSBSD en la tabla Centers

Si se introduce un código de soporte que no existe o no es válido:

NO EXISTE EL CODIGO DE SOPORTE 1234 en la tabla Support

Si la tabla no existe o no está accesible por cualquier motivo (como por bloqueos, error -143) devuelve el código de error:

Error -143 En el objeto Users, para más detalle compruebe el código ISAM mostrado

Se registra la operación en la tabla Log\_procedures:

```
proccod    30
procname   Actualiza_Usuario
procparamin User8@corp_mail.com//2018-05-25 13:21:16.753//2018-05-25
13:21:16
           .753//CLIENTE//2018-05-25 13:21:16.753//SD00001//G00001
procparamout OK
procreg    2018-05-25 13:21:16.753

proccod    31
procname   Actualiza_Usuario
procparamin USUSUS@corp_mail.com//2018-05-25 13:27:56.356//2018-05-
25 13:27:5
           6.356//CLIENTE//2018-05-25 13:27:56.356//SD00001//G00001
procparamout NO EXISTE EL USUARIO USUSUS@corp_mail.com en la
tabla Users
procreg    2018-05-25 13:27:56.356

proccod    32
procname   Actualiza_Usuario
procparamin USUSUS@corp_mail.com//2018-05-25 13:29:27.172//2018-05-
25 13:29:2
           7.172//CLIENTE//2018-05-25 13:29:27.172//SDSDSDSD//G00001
procparamout NO EXISTE EL USUARIO USUSUS@corp_mail.com en la
tabla Users
procreg    2018-05-25 13:29:27.172

proccod    33
procname   Actualiza_Usuario
procparamin User8@corp_mail.com//2018-05-25 13:29:41.524//2018-05-25
13:29:41
```



- PARAMETROS DE ENTRADA→ Número de Ticket(NUMERO\_TICKET)
- PARAMETROS DE SALIDA→ Todos los campos de la tabla Ticket (incluido del código de retorno RSP).

```
echo "
execute procedure Consulta_Ticket('T000000001');
"|dbaccess -e GSD@prod_tcp
```

### RESULTADO:

```
rsp          OK
numero_ticket  T000000001
tipo_ticket    INC
codigo_usuario  User1@corp_mail.com
fecha_creacion_ti+ 2018-05-23 16:55:26.456
prioridad_ticket  P4
servicio_ticket  SE0000242
canal_ticket    WEB
ciclo_ticket    T000000001
descripcion_ticket Solicitar permisos de acceso a la aplicaci3n GSD del tipo
                  consulta para los usuarios del departamento de Riesgos.
codigo_mensaje  MS0000456
documento_asociado
```

Este procedimiento además del código RSP devuelve los resultados de la consulta.

Si el *ticket* no existe el procedimiento devuelve (error controlado):

```
rsp          NO EXISTE EL Ticket T000 en la tabla Tickets
numero_ticket
tipo_ticket
codigo_usuario
fecha_creacion_ti+
prioridad_ticket
servicio_ticket
canal_ticket
ciclo_ticket
descripcion_ticket
codigo_mensaje
documento_asociado
```

Si la tabla no existe o no está accesible por cualquier motivo (como por bloqueos, error -143) devuelve el código de error:





FILETOCLOB('/backups\_bbdd/tmp/GSD/PruebaClob.txt', 'server') o utilizar 'client' si el archivo reside en el cliente.

```
echo "  
  
insert into Tickets values ('T0000000011', 'INC', 'User2@corp_mail.com',  
CURRENT, 'P1' , 'SE0000678','WEB', 'T000000004', 'Caida de aplicación',  
'MS0000898',FILETOCLOB('/backups_bbdd/tmp/GSD/PruebaClob.txt',  
'server'));  
1 row(s) inserted.  
  
execute procedure Consulta_Ticket('T0000000011');  
  
rsp          OK  
numero_ticket    T0000000011  
tipo_ticket      INC  
codigo_usuario   User2@corp_mail.com  
fecha_creacion_ti+ 2018-05-27 10:53:29.148  
prioridad_ticket P1  
servicio_ticket  SE0000678  
canal_ticket     WEB  
ciclo_ticket     T000000004  
descripcion_ticket Caida de aplicación  
codigo_mensaje   MS0000898  
documento_asociado  
ppppppppp (texto del documento /backups_bbdd/tmp/GSD/PruebaClob.txt)  
  
"|dbaccess -e GSD@prod_tcp
```

Al ser un punto a definir y establecer a gestionar por el aplicativo para esta fase del prototipo se realizan las inserciones con NULL de los documentos anexos.

Si el *ticket* ya existe y se vuelve a insertar el procedimiento devuelve (error controlado):

```
error -100 El Ticket ya existe
```

Si la tabla no existe o no está accesible por cualquier motivo (como por bloqueos, error -143) devuelve el código de error:

```
Error -143 En el objeto Tickets, para más detalle compruebe el codigo ISAM  
mostrado
```

Se registra la operación en la tabla Log\_procedures:

```
proccod    108  
procname   Alta_Ticket  
procparamin      T0000000010//INC//User2@corp_mail.com//2018-05-27  
10:33:43.987//P
```





La actualización de *tickets* en el campo CLOB (ticketDocument) ha de realizarse con funciones FILETOCLOB documentadas en los *scripts* y comentada en el punto de alta de un *ticket*, para el prototipo utilizamos NULL.

Se comprueba que el *ticket* exista y que el código de Servicio y de usuario sea válido, de no ser así el procedimiento devuelve (error controlado) y no realiza la actualización, se realiza el control de errores de la siguiente forma:

```
LET contador = 0;

SELECT COUNT(*)
INTO contador
from Tickets where ticketNum = ticketNum_1;

IF contador = 0 THEN

    LET RSP = ' NO EXISTE EL Ticket ' || ticketNum_1 || ' en la tabla
Tickets';

ELSE

    SELECT COUNT(*)
    INTO contador2
    from Services where serviceCod = ticketService_1;

    IF contador2 = 0 THEN

        LET RSP = ' NO EXISTE EL CODIGO DE SERVICIO ' || ticketService_1
|| ' en la tabla Services ';

    ELSE

        SELECT COUNT(*)
        INTO contador3
        from Users where userCod = ticketuserCod_1;

        IF contador3 = 0 THEN

            LET RSP = ' NO EXISTE EL CODIGO DE USUARIO ' ||
ticketuserCod_1 || ' en la tabla Users';

        ELSE

            -----
            -----
            -- Se realiza la inserción con los parámetros enviados en la tabla
de Tickets.
```

```

-- Para las inserciones del campo CLOB (ticketDocument) se tiene
que utilizar la función:
-- FILETOCLOB('/backups_bbdd/tmp/GSD/PruebaClob.txt',
'server') o utilizar 'client' si el archivo reside en el cliente.
-- Al ser un punto a gestionar por el aplicativo para el prototipo se
utiliza NULL.
-----
-----

```

```

update Tickets
set
ticketNum=ticketNum_1,ticketType=ticketType_1,ticketuserCod=ticketuserCod_
1,ticketCreate=ticketCreate_1,ticketPriority=ticketPriority_1,ticketService=ticket
Service_1,ticketChannel=ticketChannel_1,ticketCycle=ticketCycle_1,ticketDescr
iption=ticketDescription_1,ticketmessageCod=ticketmessageCod_1,ticketDocu
ment=ticketDocument_1
where ticketNum = ticketNum1;

LET RSP = 'OK';

END IF

END IF

END IF

```

Por lo tanto, si introducimos un Servicio no existente se devuelve:

**NO EXISTE EL CODIGO DE SERVICIO SESESESE en la tabla Services**

Si se introduce un código de usuario que no existe:

**NO EXISTE EL CODIGO DE USUARIO NOVALIDO@corp\_mail.com en la tabla Users**

Si se introduce un código de *ticket* que no existe o no es válido:

**NO EXISTE EL Ticket T000000020 en la tabla Tickets**

Si la tabla no existe o no está accesible por cualquier motivo (como por bloqueos, error -143) devuelve el código de error:

**Error -143 En el objeto Tickets, para más detalle compruebe el codigo ISAM mostrado**

Se registra la operación en la tabla Log\_procedures:

proccod	127
procname	Actualiza_Ticket

```

proccod      128
procname     Actualiza_Ticket
proccod      128
procname     Actualiza_Ticket
proccod      129
procname     Actualiza_Ticket
proccod      130
procname     Actualiza_Ticket

```

T000000011//INC//User2@corp\_mail.com//2018-05-27  
 12:09:50.184//P  
 1//SE0000678//WEB//T000000004//Se actualiza el ticket - EN  
 MARCHA  
 -//MS0000898  
 OK  
 2018-05-27 12:09:50.184

T000000020//INC//User2@corp\_mail.com//2018-05-27  
 12:10:24.543//P  
 1//SE0000678//WEB//T000000004//Se actualiza el ticket - EN  
 MARCHA  
 -//MS0000898  
 NO EXISTE EL Ticket T000000020 en la tabla Tickets  
 2018-05-27 12:10:24.543

T000000011//INC//User2@corp\_mail.com//2018-05-27  
 12:10:49.195//P  
 1//SESESESE//WEB//T000000004//Se actualiza el ticket - EN  
 MARCHA  
 -//MS0000898  
 NO EXISTE EL CODIGO DE SERVICIO SESESESE en la tabla  
 Services  
 2018-05-27 12:10:49.195

T000000011//INC//NO\_VALIDO@corp\_mail.com//2018-05-27  
 12:11:35.53  
 3//P1//SE0000678//WEB//T000000004//Se actualiza el ticket - EN MA  
 RCHA -//MS0000898  
 NO EXISTE EL CODIGO DE USUARIO  
 NO\_VALIDO@corp\_mail.com en la tab  
 la Users  
 2018-05-27 12:11:35.533

### 4.3 Gestión de estados de los *tickets*

Cuando un *ticket* es dado de alta o se modifica a cualquier estado la aplicación Global Service Desk inserta o modifica los estados de un *ticket* en tabla de estados (State).

De esta forma, en la tabla State se registran todos los estados por los que ha pasado un *ticket*.

Toda la gestión de los estados de los *tickets* se realiza mediante procedimientos almacenados en la base de datos con control de errores.

Se definen los procedimientos de gestión de los estados de un *ticket* en el *script*:

PROTOTIPO\_PROCEDURES\_ESTADOS\_TICKET.sql

#### 4.3.1 Consulta del estado de un *ticket*

Mediante el procedimiento "GSD".Consulta\_Estado\_Ticket se consulta el estado de un *ticket* en el sistema introduciendo el código del *ticket*, de la siguiente forma:

- PARAMETROS DE ENTRADA → Número de Ticket (codigo\_ticket)
- PARAMETROS DE SALIDA → Todos los campos de la tabla State (incluyendo el código de retorno RSP).

```
echo "  
execute procedure Consulta_Estado_Ticket('T000000001');  
"|dbaccess -e GSD@prod_tcp
```

#### RESULTADO:

```
rsp          OK  
codigo_estado      8  
codigo_ticket      T000000001  
fecha_cambio_esta+ 2018-05-23 16:55:26.457  
codigo_usuario     User1@corp_mail.com  
nuevo_estado       Creado  
  
rsp          OK  
codigo_estado      9  
codigo_ticket      T000000001  
fecha_cambio_esta+ 2018-05-23 16:55:26.458  
codigo_usuario     User1@corp_mail.com  
nuevo_estado       Asignado  
  
rsp          OK  
codigo_estado      10  
codigo_ticket      T000000001  
fecha_cambio_esta+ 2018-05-23 16:55:26.458
```



```
codigo_usuario    User1@corp_mail.com
nuevo_estado     En progreso

rsp              OK
codigo_estado    11
codigo_ticket     T000000001
fecha_cambio_esta+ 2018-05-23 16:55:26.458
codigo_usuario    User1@corp_mail.com
nuevo_estado     Cancelado
```

Este procedimiento además del código RSP devuelve los resultados de la consulta sobre la tabla State.

Si el *ticket* no existe el procedimiento devuelve (error controlado):

```
rsp              NO EXISTE EL Ticket T000000001111 en la tabla State
codigo_estado
codigo_ticket
fecha_cambio_esta+
codigo_usuario
nuevo_estado
```

Si la tabla no existe o no está accesible por cualquier motivo (como por bloqueos, error -143) devuelve el código de error:

```
Error -143 En el objeto State, para más detalle compruebe el codigo ISAM
mostrado
```

Se registra la operación en la tabla Log\_procedures:

```
proccod    145
procname    Consulta_CODIGO_TICKET
procparamin T000000001
procparamout OK
procreg     2018-05-28 13:24:46.579

proccod    146
procname    Consulta_CODIGO_TICKET
procparamin T000000001111
procparamout NO EXISTE EL Ticket T000000001111 en la tabla State
procreg     2018-05-28 13:25:00.782
```

#### 4.3.2 Alta del estado de un *ticket*

Mediante el procedimiento "GSD".Alta\_Estado\_Ticket se inserta un estado de un *ticket* en el sistema, de la siguiente forma:

- PARAMETROS DE ENTRADA → Todos los campos de la tabla Tickets (Menos el código interno del estado que es un serial automático).



```
-----  
-----  
-- Se llama al procedimiento estadístico timeSollnc si el estado es resuelto:  
-----  
-----
```

```
IF stateNew = 'Resuelto' THEN  
  
    CALL    timeSollnc    ('Alta',stateTicketNum    ,stateChangeDate    ,  
stateUserCod , stateNew);  
  
END IF
```

### 4.3.3 Baja de un estado de un *ticket*

Mediante el procedimiento "GSD".Baja\_ Estado\_Ticket se borra el estado de un ticket en el sistema, de la siguiente forma y pasando al procedimiento código del ticket:

- PARAMETROS DE ENTRADA → CODIGO DEL TICKET Y ESTADO A BORRAR.
- PARAMETROS DE SALIDA → RSP (RESULTADO DE LA OPERACIÓN).

```
echo "  
  
execute procedure Baja_Estado_Ticket('T000000001','PRUEBA_NUEVO');  
  
"|dbaccess -e GSD@prod_tcp
```

RESULTADO:

```
OK
```

Se comprueba que el *ticket* y el estado exista, de no ser así el procedimiento devuelve (error controlado) y no realiza el *delete*:

```
NO EXISTE EL ESTADO DEL TICKET T000000001 en la tabla State
```

Esta comprobación se realiza de la siguiente forma:

```
LET contador = 0;  
  
SELECT COUNT(*)  
INTO contador  
from State where stateTicketNum = stateTicketNum_1 and stateNew =  
stateNew_1;  
  
IF contador = 0 THEN
```

```

        LET RSP = ' NO EXISTE EL ESTADO DEL TICKET ' ||
stateTicketNum_1 || ' en la tabla State';

ELSE

        delete from State where stateTicketNum = stateTicketNum_1 and
stateNew = stateNew_1;
        LET RSP = 'OK';

END IF

```

Si la tabla no existe o no está accesible por cualquier motivo (como por bloqueos, error -143) devuelve el código de error:

Error -143 En el objeto State, para más detalle compruebe el código ISAM mostrado

Se registra la operación en la tabla Log\_procedures:

```

proccod    164
procname   Baja_estado_Ticket
procparamin T000000001//PRUEBA_NUEVO
procparamout OK
procreg    2018-05-29 11:20:49.071

proccod    165
procname   Baja_estado_Ticket
procparamin T000000001//PRUEBA_NUEVO
procparamout NO EXISTE EL ESTADO DEL TICKET T000000001 en la tabla
State
procreg    2018-05-29 11:21:07.505

```

Se realiza la llamada al repositorio estadístico de la siguiente forma:

```

-----
-----
-- Se llama al procedimiento estadístico timeSollnc si el estado es
resuelto:
-----
-----

IF stateNew_1 = 'Resuelto' THEN

        SELECT statechangedate
        INTO stateChangeDate_1
        from State
        where stateTicketNum = stateTicketNum_1 and stateNew =
stateNew_1;

```

```

SELECT stateusercod
INTO stateUserCod_1
from State
where stateTicketNum = stateTicketNum_1 and stateNew =
stateNew_1;

CALL timeSolInc ('Baja',stateTicketNum_1 ,stateChangeDate_1 ,
stateUserCod_1 , stateNew_1);

END IF

```

#### 4.3.4 Actualización del estado de un *ticket*

Mediante el procedimiento "GSD". Actualiza\_Estado\_Tickets se actualiza el estado de un *ticket* en el sistema, de la siguiente forma y pasando al procedimiento los parámetros totales del usuario:

- PARAMETROS DE ENTRADA: Todos los campos de la tabla State (INCLUIDO EL CAMPO SERIAL AUTOGENERADO, para actualizar solo un estado del ticket).
- PARAMETROS DE SALIDA: RSP (RESULTADO DE LA OPERACIÓN).

```

echo "

execute                                     procedure
Actualiza_Estado_Tickets(329,'T000000001111',CURRENT,'User1@corp_mail.
com','ESTADO_ACTUALIZADO');

"|dbaccess -e GSD@prod_tcp

```

RESULTADO:

OK

Se comprueba que el *ticket* exista en la tabla State, de no ser así el procedimiento devuelve (error controlado) y no realiza la actualización, se realiza el control de errores de la siguiente forma:

```

LET contador = 0;

SELECT COUNT(*)
INTO contador
from State where stateTicketNum = stateTicketNum_1;

IF contador = 0 THEN

```



Se llama al procedimiento estadístico de la siguiente forma:

```
LET contador = 0;

SELECT COUNT(*)
INTO contador
from State where stateTicketNum = stateTicketNum_1 and statecodstate =
statecodstate_1;

IF contador = 0 THEN

    LET RSP = ' NO EXISTE EL ESTADO DEL TICKET ' ||
stateTicketNum_1 || ' en la tabla State';

ELSE

    -----
    -----
    -- Se guarda el estado de la tabla State para comprobaciones
estadísticas.
    -----
    -----

    SELECT statenew
    INTO COMPROBAR_ESTADO
    from State
    where stateTicketNum = stateTicketNum_1 and statecodstate =
statecodstate_1;

    -----
    -----
    -- Se Actualiza la tabla State con los datos facilitados.
    -----
    -----

    update State
    set
stateTicketNum=stateTicketNum_1,stateChangeDate=stateChangeDate_1,stat
eUserCod=stateUserCod_1,stateNew=stateNew_1
    where stateTicketNum = stateTicketNum_1 and statecodstate =
statecodstate_1;

    LET RSP = 'OK';

    -----
    -----
    -- Se llama al procedimiento estadístico timeSollnc si el estado es
resuelto o si se modifica el estado del Resuelto a otro:
```

```

-- Si se quita un estado 'Resuelto' hay que borrarlo de la tabla de
estadística.
-----
-----

IF COMPROBAR_ESTADO = 'Resuelto' AND stateNew_1 <> 'Resuelto'
THEN

CALL timeSolInc ('Actualizar_Del',stateTicketNum_1
,stateChangeDate_1 , stateUserCod_1 , stateNew_1);

ELSE

IF COMPROBAR_ESTADO = 'Resuelto' AND stateNew_1 =
'Resuelto' THEN

-----
-----
-- NO hacemos nada, ya que no se modifica el estado de la tabla,
se pueden modificar otros parámetros.
-----
-----

ELSE

CALL timeSolInc ('Actualizar_Add',stateTicketNum_1
,stateChangeDate_1 , stateUserCod_1 , stateNew_1);

END IF

END IF

END IF

```

#### 4.4 Gestión de mensajes de los *tickets*

Los mensajes de los *tickets* gestionan las operaciones de comunicación entre usuarios, clientes y usuarios de soporte. La tabla Messages es la encargada de registrarlos.

Se definen los procedimientos de gestión de los mensajes de un *ticket* en el *script*:

PROTOTIPO\_PROCEDURES\_MENSAJES\_TICKET.sql



#### 4.4.1 Consultar los mensajes de un *ticket*

Mediante el procedimiento "GSD".Consulta\_Mensajes\_Ticket se consulta todos los mensajes de un *ticket* en el sistema introduciendo el código del *ticket*, de la siguiente forma:

- PARAMETROS DE ENTRADA → CODIGO\_TICKET
- PARAMETROS DE SALIDA → Resultado de la operación (RSP) y todos los campos de la tabla Messages.

```
echo "  
  
execute procedure Consulta_Mensajes_Ticket('T000000001');  
  
"|dbaccess -e GSD@prod_tcp
```

#### RESULTADO:

```
rsp      OK  
codigo_mensaje MS0000456  
fecha_mensaje 2018-05-23 16:55:26.460  
usuario_mensaje User1@corp_mail.com  
estado_mensaje Recibido  
codigo_ticket  T000000001  
  
rsp      OK  
codigo_mensaje MS0000457  
fecha_mensaje 2018-05-23 16:55:26.460  
usuario_mensaje User1@corp_mail.com  
estado_mensaje Recibido  
codigo_ticket  T000000001  
  
rsp      OK  
codigo_mensaje MS0000458  
fecha_mensaje 2018-05-23 16:55:26.460  
usuario_mensaje User1@corp_mail.com  
estado_mensaje Enviado  
codigo_ticket  T000000001  
  
rsp      OK  
codigo_mensaje MS0000459  
fecha_mensaje 2018-05-23 16:55:26.461  
usuario_mensaje User1@corp_mail.com  
estado_mensaje Enviado  
codigo_ticket  T000000001
```

Este procedimiento además del código RSP devuelve los resultados de la consulta sobre la tabla Message.



## RESULTADO:

OK

Si el *ticket* no existe el procedimiento devuelve (error controlado):

error -111 El Ticket al que asociar el mensaje no existe

Si la tabla no existe o no está accesible por cualquier motivo (como por bloqueos, error -143) devuelve el código de error:

Error -143 En el objeto Messages, para más detalle compruebe el código ISAM mostrado

Se registra la operación en la tabla Log\_procedures:

```
proccod    176
procname   Alta_Mensajes_Ticket
proparamin                               MS0000555//2018-05-29
12:35:41.441//User1@corp_mail.com//PRUEBA_A
        LTA_MENSAJE//T000000001
proparamout OK
procreg    2018-05-29 12:35:41.441

proccod    177
procname   Alta_Mensajes_Ticket
proparamin                               MS0000555//2018-05-29
12:36:25.793//User1@corp_mail.com//PRUEBA_A
        LTA_MENSAJE//T00000000111
proparamout error -111 El Ticket al que asociar el mensaje no existe
procreg    2018-05-29 12:36:25.793
```

### **4.3.3 Baja de un mensaje de un *ticket***

Mediante el procedimiento "GSD".Baja\_Estado\_Ticket se borra el mensaje de un *ticket* en el sistema, de la siguiente forma y pasando al procedimiento código del *ticket* y del mensaje:

- PARAMETROS DE ENTRADA → CODIGO DEL TICKET Y CODIGO DEL MENSAJE.
- PARAMETROS DE SALIDA → RSP (RESULTADO DE LA OPERACIÓN).

```
echo "  
  
execute procedure Baja_Estado_Ticket('T000000001','MS0000555');  
  
"|dbaccess -e GSD@prod_tcp
```

### RESULTADO:

OK

Se comprueba que el *ticket* y el mensaje exista, de no ser así el procedimiento devuelve (error controlado) y no realiza el *delete*:

NO EXISTE EL ESTADO DEL TICKET T000000001 en la tabla Message

Esta comprobación se realiza de la siguiente forma:

```
LET contador = 0;  
  
SELECT COUNT(*)  
INTO contador  
from Messages where messageTicketNum = messageTicketNum_1 and  
messageCod = messageCod_1;  
  
IF contador = 0 THEN  
  
    LET RSP = ' NO EXISTE EL ESTADO DEL TICKET ' ||  
messageTicketNum_1 || ' en la tabla Message';  
  
ELSE  
  
    delete from Messages where messageTicketNum =  
messageTicketNum_1 and messageCod = messageCod_1;  
    LET RSP = 'OK';  
  
END IF
```

Si la tabla no existe o no está accesible por cualquier motivo (como por bloqueos, error -143) devuelve el código de error:

Error -143 En el objeto Message, para más detalle compruebe el código ISAM mostrado

Se registra la operación en la tabla Log\_procedures:

```
proccod    180  
procname   Baja_Mensaje_Ticket  
procparamin T000000001//MS0000555  
procparamout OK
```

```

procreg    2018-05-29 12:52:00.360

proccod    181
procname    Baja_Mensaje_Ticket
procparamin T000000001//MS0000555
procparamout NO EXISTE EL ESTADO DEL TICKET T000000001 en la tabla
Message
procreg    2018-05-29 12:52:14.872

```

#### 4.3.4 Actualización del mensaje de un *ticket*

Mediante el procedimiento "GSD".Actualiza\_Mensajes\_Tickets se actualizan los mensajes de un *ticket* en el sistema, de la siguiente forma y pasando al procedimiento los parámetros totales del usuario:

- PARAMETROS DE ENTRADA: Todos los campos de la tabla Message (incluido el código del mensaje).
- PARAMETROS DE SALIDA: RSP (RESULTADO DE LA OPERACIÓN).

```

echo "

execute                                     procedure
Actualiza_Mensajes_Tickets('MS0000555888',CURRENT,'User1@corp_mail.co
m','ACTUALIZAMOS_MENSAJE_22','T000000001');

"|dbaccess -e GSD@prod_tcp

```

#### RESULTADO:

```
OK
```

Se comprueba que el *ticket* y el código del mensaje exista en la tabla Messages, de no ser así el procedimiento devuelve (error controlado) y no realiza la actualización, se realiza el control de errores de la siguiente forma:

```

LET contador = 0;

SELECT COUNT(*)
INTO contador
from Messages where messageTicketNum = messageTicketNum_1 and
messageCod = messageCod_1;

IF contador = 0 THEN

    LET RSP = ' NO EXISTE EL Mensaje DEL TICKET ' ||
messageTicketNum_1 || ' en la tabla Messages';

ELSE

```

```

        update Messages
        set
messageCod=messageCod_1,messageTime=messageTime_1,messageReceiv
er=messageReceiver_1,messageState=messageState_1,messageTicketNum=
messageTicketNum_1
        where messageTicketNum = messageTicketNum_1 and
messageCod = messageCod_1;

        LET RSP = 'OK';

END IF

```

Por lo tanto, si se introduce un código de *ticket* o de mensaje que no figura en la tabla Messages se devuelve el error controlado:

**NO EXISTE EL Mensaje DEL TICKET T000000001 en la tabla Messages**

Si la tabla no existe o no está accesible por cualquier motivo (como por bloqueos, error -143) devuelve el código de error:

**Error -143 En el objeto Messages, para más detalle compruebe el código ISAM mostrado**

Se registra la operación en la tabla Log\_procedures:

```

proccod    188
procname   Actualiza_Mensajes_Tickets
procparamin                MS0000555//2018-05-29
13:19:19.364//User1@corp_mail.com//ACTUALIZ
        AMOS_MENSAJE_22//T000000001
procparamout OK
procreg    2018-05-29 13:19:19.364

proccod    189
procname   Actualiza_Mensajes_Tickets
procparamin                MS0000555888//2018-05-29
13:19:44.405//User1@corp_mail.com//ACTUA
        LIZAMOS_MENSAJE_22//T000000001
procparamout NO EXISTE EL Mensaje DEL TICKET T000000001 en la tabla
State
procreg    2018-05-29 13:19:44.405

```

## 5. Repositorio estadístico

Mediante el repositorio estadístico se generan resultados de estadísticas sin utilizar en ninguna parte del proceso funciones agregadas (sum, avg, etc.).

Para conseguir esto, se guardan en tablas del repositorio estadístico los valores necesarios para dar respuesta a las estadísticas planteadas.

Además, se cargan los valores iniciales en las tablas del repositorio y se gestionan las operaciones automáticamente desde los procedimientos necesarios que deben desencadenar las llamadas a los procedimientos del repositorio.

Por lo tanto, los *scripts* del repositorio son los siguientes y se deben ejecutar en el mismo orden para una versión inicial:

```
PROTOTIPO_ESTRUCTURA_ESTADISTICO.sql  
PROTOTIPO_DATOS_ESTADISTICO.sql  
PROTOTIPO_PROCEDURES_ESTADISTICO.sql
```

De la siguiente forma:

```
dbaccess          -e          sysmaster@prod_tcp  
PROTOTIPO_ESTRUCTURA_ESTADISTICO.sql          >  
PROTOTIPO_ESTRUCTURA_ESTADISTICO.log 2>&1  
  
dbaccess -e GSD@prod_tcp PROTOTIPO_DATOS_ESTADISTICO.sql >  
PROTOTIPO_DATOS_ESTADISTICO.log 2>&1  
  
dbaccess          -e          GSD@prod_tcp  
PROTOTIPO_PROCEDURES_ESTADISTICO.sql          >  
PROTOTIPO_PROCEDURES_ESTADISTICO.log 2>&1
```

El resultado de las ejecuciones se documenta en:

## **ANEXO H: Generación de scripts SQL del repositorio estadístico**

La forma de generar los datos del repositorio es mediante la lógica necesaria en los procedimientos del mismo (PROTOTIPO\_PROCEDURES\_ESTADISTICO.sql).

Estos procedimientos se llaman (con llamadas CALL) desde los procedimientos de ALTA\_BAJA\_MODIFICACIONES que gestionan el modelo.

Se descarta la llamada a los procedimientos del repositorio mediante disparadores (*triggers*) ya que supone un punto más de fallo en el modelo.

Es innecesario al gestionarse íntegramente los cambios y consultas en el modelo mediante procedimientos, también para las generaciones automáticas de *tickets* e incluso los procesos *batch* que se ejecuten en *offline*.

Todos los procesos de modificación de datos (incluido *batch* nocturno) del modelo deben llamar a los procedimientos encargados de ello.

También, se descarta la utilización de *triggers* para facilitar las labores de administración fuera del proceso del repositorio estadístico, como cargas masivas, borrados, etc., que llevarán su propio ciclo e incluso procedimientos de nivelación del entorno.

De esta forma, se optimiza el modelo ya que solo se llaman a los procedimientos del repositorio estadístico cuando es necesario, evitando que siempre se produzca la llamada (como los *triggers*) y solo cuando se cumple una determinada condición se produce una llamada (se produce un CALL al procedimiento estadístico bajo condiciones definidas).

### 5.1 Dado un mes cualquiera, tiempo medio de resolución de incidencias por cada tipo de prioridad

Esta consulta estadística se genera solo cuando el estado de un *ticket* se actualiza a “Resuelto”, esto supone un registro en la tabla State y supone que el *ticket* ha sido resuelto.

Solo se genera el evento en la tabla timeSollnc si es una incidencia, es decir si el campo ticketType de la tabla Tickets es ‘INC’.

Esta comprobación, aunque se puede realizar en los procedimientos de cambios de estado y reduce las llamadas al procedimiento estadístico, se elige ponerla en éste para reducir y simplificar el código de los demás procedimientos (habría que ver el impacto con concurrencia real).

Por lo tanto, se realiza la llamada al procedimiento estadístico "GSD".timeSollnc que hace las comprobaciones finales con los parámetros recibidos:

```
IF ticketType_sta = 'INC' THEN
    IF acc_timeSollnc = 'Baja' OR acc_timeSollnc = 'Actualizar_Del' THEN
        -----
        -----
        -- Se Decrementa el contador de incidencias si es una llamada de
        estado 'Resuelto' borrado.
        -----
        -----
        UPDATE timeSollnc
        SET numInc=numInc - 1
        WHERE priority = priority_1 and month_tsi = month_1;
        -----
        -----
```



```

-- Se inserta en la tabla de log de procedimientos, el nombre del
procedimiento, parámetros de entrada y de salida, hora de la ejecución.
-----
-----

INSERT INTO Log_procedures values (0, 'timeSollnc', 'SE
DECREMENTA LA INCIDENCIA',RSP,CURRENT);

ELSE

UPDATE timeSollnc
SET
timeAcu=FINAL_ACUMULADO,timeSol=HORA_MEDIA,numInc=numInc_1
WHERE priority = priority_1 and month_tsi = month_1;

-----
-----

-- Se inserta en la tabla de log de procedimientos, el nombre del
procedimiento, parámetros de entrada y de salida, hora de la ejecución.
-----
-----

INSERT INTO Log_procedures values (0, 'timeSollnc',
cod_timeSollnc_1||'/'||month_1||'/'||FINAL_ACUMULADO||'/'||HORA_MEDIA||'/'
'priority_1||'/'||numInc_1,RSP,CURRENT);

END IF

END IF

```

Los procedimientos de gestión del modelo Global Service Desk que llaman a este procedimiento estadístico son:

- "GSD".Alta\_Estado\_Ticket;
- "GSD".Actualiza\_Estado\_Tickets;
- "GSD".Baja\_Estado\_Ticket;

Las condiciones de llamada al procedimiento estadístico desde estos procedimientos de gestión queda documentada en los apartados relativos a los mismos.

El procedimiento timeSollnc tiene los siguientes parámetros:

- PARAMETROS DE ENTRADA → Todos los campos de la tabla State (MENOS EL CAMPO SERIAL AUTOGENERADO).
- PARAMETROS DE SALIDA → El código RSP (RESULTADO DE LA OPERACIÓN) que se registra en la tabla Log\_procedures.

Se controlan las excepciones:

```
ON EXCEPTION SET error_num, error_isam, error_txt
IF error_isam <> 0 THEN
    LET RSP = 'Error ' || error_isam || ' en: ' || error_txt || ' , para más detalle
    compruebe el codigo ISAM mostrado ';
END IF;
END EXCEPTION WITH RESUME
```

Se obtienen los meses con funciones MONTH:

```
LET month_1 = MONTH(stateChangeDate_1);
```

Se calcula la media de resolución de una incidencia, en función de la diferencia de la fecha de creación y de resolución de la incidencia.

Se acumula el tiempo de resolución de la incidencia más el acumulado en la tabla del repositorio.

La media corresponde a la acumulación del tiempo de resolución de la incidencia dividido por el número de incidencias, parámetro que se va incrementando según se van resolviendo las incidencias o decreciendo si se cambia el estado de Resuelto a otro distinto (por ejemplo, porque se quiera reabrir una incidencia):

```
LET HORA_DIA_NEW = LEFT(HORA_NEW,10);
LET HORA_HORA_NEW = SUBSTR(HORA_NEW,11,2);
LET HORA_MINUTE_NEW = SUBSTR(HORA_NEW,14,2);

LET FINAL_NEW = (HORA_DIA_NEW*1440) + (HORA_HORA_NEW*60) +
(HORA_MINUTE_NEW);
LET FINAL_ACUMULADO = FINAL_NEW + timeAcu_1;

LET HORA_MEDIA = FINAL_ACUMULADO / numInc_1;
```

De tal forma que, cuando se actualiza o se crea un estado nuevo de un *ticket*, se actualiza la tabla del repositorio estadístico, de la siguiente forma:

```
echo "
```

```
execute Alta_estado_ticket('T000000004',CURRENT,'User1@corp_mail.com','Resuelto');
procedure
Alta_estado_ticket('T000000004',CURRENT,'User1@corp_mail.com','Resuelto');
";
"|dbaccess -e GSD@prod_tcp
```

El resultado es el siguiente en la tabla del repositorio (timeSollnc) devolviendo la media en MINUTOS :

```
echo "

select * from timeSollnc where priority = 'P1' and month_tsi = 6;

"|dbaccess -e GSD@prod_tcp

cod_timesolinc 54
month_tsi      6
timesol      3157
priority       P1
numinc         1

1 row(s) retrieved.
```

### Responder a la pregunta

**Dado un mes cualquiera, tiempo medio de resolución de incidencias por cada tipo de prioridad.**

Basta con hacer una *select* por el mes a la tabla del repositorio estadístico, de la siguiente forma:

```
echo "

--select * from timeSollnc where priority = 'P1' and month_tsi = 6;
select * from timeSollnc where month_tsi = 6;

"|dbaccess -e GSD@prod_tcp

cod_timesolinc 6
month_tsi     6
```

timeacu 4530  
**timesol 215.7142857142857143**  
**priority P1**  
numinc 21

cod\_timesolinc 18  
**month\_tsi 6**  
timeacu 0  
**timesol 0**  
**priority P2**  
numinc 0

cod\_timesolinc 30  
**month\_tsi 6**  
timeacu 0  
**timesol 0**  
**priority P3**  
numinc 0

cod\_timesolinc 42  
**month\_tsi 6**  
timeacu 0  
**timesol 0**  
**priority P4**  
numinc 0

## 6. Conclusiones

Con este proyecto se pretende ofrecer al mercado una solución completa y suficiente que garantice una aplicación de gestión de incidencias así como todo lo necesario para gestionar los datos almacenados cumpliendo con las normativas actuales de seguridad.

De esta forma, se ofrece una solución profesional que se diferencia de la competencia y ofrece un valor añadido al cliente, asegurando que este no tiene que realizar más implantaciones, reduciendo significativamente los costes.

Por este motivo, se destinan especiales recursos a la instalación y configuración de un gestor de base de datos con replicación ofreciendo con ello alta disponibilidad, un sistema completo de auditoría, así como también un sistema de *backup* y restauración de copias de seguridad.

En consecuencia, se ofrece una solución completa y escalable a cualquier volumen en un entorno profesional, esto ha supuesto adaptaciones en la planificación inicial que ha obligado a realizar los ajustes que se detallan a continuación.

### 6.1. Ajustes de planificación

Debido a la configuración completa de la solución, se observa que se deben ajustar como procesos iniciales y antes de la instalación del modelo de datos todos los aspectos de soporte a la solución, estos son:

- Instalación y configuración del gestor de base de datos.
- Propuestas de optimización del sistema de base de datos.
- Propuestas de mejora de rendimiento en entorno con concurrencia real.
- Instalación y configuración de la solución de alta disponibilidad.
- Instalación y configuración de un sistema de copias de seguridad.
- Instalación y configuración de un sistema de auditoría.

Esto, persigue el fin de ofrecer una solución real testeada, siendo primordial probar el modelo de Global Service Desk en un sistema final configurado.










Debido a esto, se intenta conseguir el objetivo de evaluar el rendimiento del sistema y optimizarlo, dando pautas reales para ello, en función de los resultados de pruebas de carga y de las pautas dadas por el fabricante (IBM).

Por lo tanto, la planificación real obliga a ajustes con el fin de adaptarla a la planificación inicial propuesta.

Además, en cada una de las fases se embebe la resolución de problemas, ya que surgen en todas las fases y es una parte fundamental.

### Planificación inicial del proyecto:

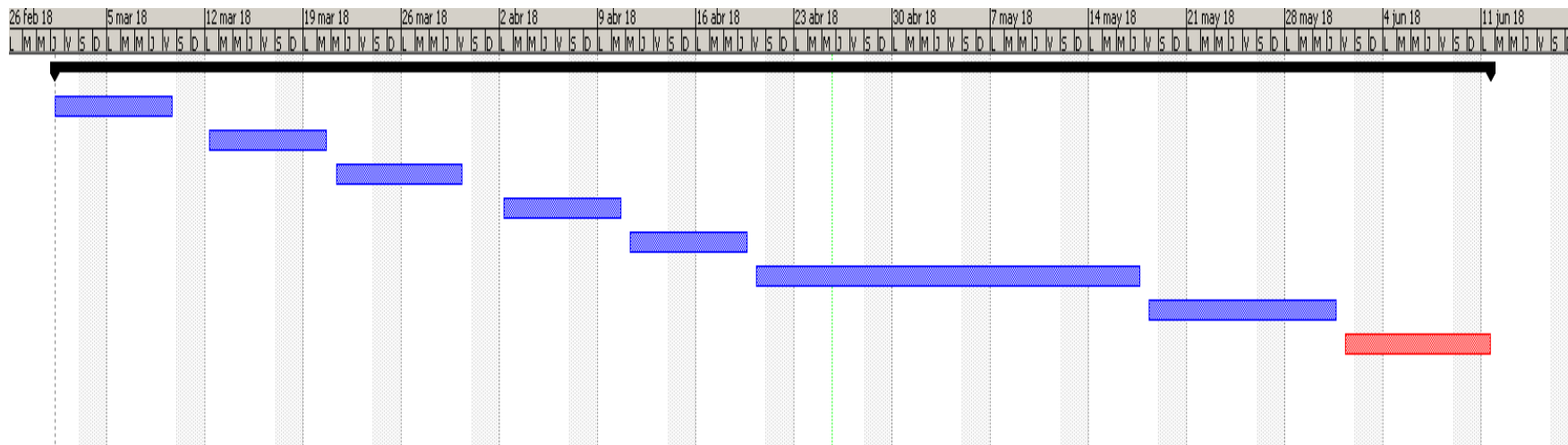
En el que se dividen las tareas de implementación del proyecto en función de metodologías como Zielorientierte Projektplanung (ZOPP), metodología alemana que permite orientar el proyecto en función de los objetivos a alcanzar.

		Nombre	Duración	Inicio	Terminado
1		<input type="checkbox"/> <b>Global Service Desk</b>	<b>73 days</b>	<b>1/03/18 8:00</b>	<b>11/06/18 17:00</b>
2		Instalación del Gestor de Base de datos .	7 days	1/03/18 8:00	9/03/18 17:00
3		Configuración del Gestor de Base de datos.	7 days	10/03/18 8:00	20/03/18 17:00
4		Diseño, Modelado y Preparación de scripts del modelo	20 days	21/03/18 8:00	17/04/18 17:00
5		Pruebas del modelo (ciclo de vida de un ticket).	7 days	18/04/18 8:00	26/04/18 17:00
6		Políticas de backup, estadísticas y seguridad de la base de datos.	7 days	27/04/18 8:00	7/05/18 17:00
7		Configuración del sistema en alta disponibilidad	7 days	8/05/18 7:00	16/05/18 17:00
8		Configuración del sistema de auditoría	7 days	17/05/18 7:00	25/05/18 17:00
9		Plan de contingencia.	7 days	26/05/18 7:00	5/06/18 17:00
10		Resolución de problemas.	4 days	6/06/18 7:00	11/06/18 17:00

Replanificación real del proyecto:

Se observa que para la realización de pruebas reales sobre el modelo resulta vital disponer del entorno real configurado.

	📅	Nombre	Duracion	Inicio	Terminado
1		☑Global Service Desk	73 days	1/03/18 8:00	11/06/18 17:00
2		Instalación del Gestor de Base de datos .	7 days	1/03/18 8:00	9/03/18 17:00
3	📅	Configuración del Gestor de Base de datos.	7 days	10/03/18 8:00	20/03/18 17:00
4	📅	Políticas de backup, estadísticas y seguridad de la base de datos.	7 days	21/03/18 9:00	30/03/18 9:00
5	📅	Configuración del sistema en alta disponibilidad	7 days	31/03/18 7:00	10/04/18 17:00
6	📅	Configuración del sistema de auditoría	7 days	11/04/18 7:00	19/04/18 17:00
7	📅	Diseño, Modelado y Preparación de scripts del modelo	20 days	20/04/18 7:00	17/05/18 17:00
8	📅	Pruebas del modelo (ciclo de vida de un ticket).	10 days	18/05/18 8:00	31/05/18 17:00
9	📅	Plan de contingencia.	7 days	1/06/18 7:00	11/06/18 17:00



## 6.2. Soluciones adoptadas para el sistema Global Service Desk

A continuación se detallan las soluciones adoptadas en los diferentes Anexos.

### 6.2.1. Instalación de Informix

Debido a los requisitos necesarios para ofrecer una solución robusta, estable y profesional, se opta por un sistema gestor de base de datos con alta solidez y experiencia en el mercado para ofrecer las soluciones presentes y también futuras al sistema integral Global Service Desk, permitiendo su escalabilidad.

Es por ello, que se elige Informix como gestor de base de datos y en concreto la versión 11.70.FC8 que ya está suficientemente testada por clientes así como totalmente dentro de soporte de IBM y sin fecha de fin del mismo actualmente:

[← View lifecycle details for other products.](#)

---

Sort results by: [Relevance](#) | [Product name](#) | [End of support date](#)

[View details](#) [Uncheck all](#)

Show  entries Search:

<a href="#">View</a>	<a href="#">Product name</a> <small>(** indicates comments or exception)</small>	<a href="#">Version/ Rel./Mod.</a>	<a href="#">Policy type</a>	<a href="#">Product ID</a>	<a href="#">General availability</a>	<a href="#">End of Support</a>
<input type="checkbox"/>	<a href="#">Informix Choice Edition</a>	11.70.x	E	5725-B41	<a href="#">2010-11-02</a>	
<input type="checkbox"/>	<a href="#">Informix Growth Edition</a>	11.70.x	E	5725-A40	<a href="#">2010-11-02</a>	
<input type="checkbox"/>	<a href="#">Informix Express Edition</a>	11.70.0	E	5724-Z04	2010-10-25	
<input type="checkbox"/>	<a href="#">Informix Ultimate Edition</a>	11.70.x	E	5725-A39	<a href="#">2010-11-02</a>	
<input type="checkbox"/>	<a href="#">Informix Growth Warehouse Edition</a>	11.70.0	E	5725-F59	<a href="#">2011-11-18</a>	
<input type="checkbox"/>	<a href="#">Informix Ultimate Warehouse Edition</a>	11.70.x	E	5725-D14	<a href="#">2011-04-07</a>	

Previous  Next

### Información consultada:

“Software lifecycle”

<https://www-01.ibm.com/software/support/lifecycleapp/PLCSearch.wss?q=11.70&ibm-search=Search>

Aunque existen soluciones profesionales en el mercado como DB2, Oracle, SQL Server, etc. se opta por Informix ya que la solución de alta disponibilidad es muy conocida, fácil de implementar y muy robusta. Además el precio está en la misma línea de la competencia siendo la solución HA y la auditoría herramientas contenidas en la licencia del producto.

También se evita instalar la última versión de Informix que no ofrece las garantías de tiempo de solidez en el mercado. Las siglas de la versión 11.70.FC8 significan:



- F → versión de 64 bits ("U" si es de 32 bits).
- C8 → Supone un nivel estable de parches, a este nivel avanzado de parches se han solucionado muchos de los errores reportados por los clientes de Informix en todo el mundo, a partir de la versión 11,70 FC1.

En consecuencia, la instalación se hace de acuerdo al manual de administración de Informix facilitado por IBM y en concreto con la información de su WEB.

Así mismo, la instalación se realiza sobre un Sistema Operativo UNIX dando mayor robustez al sistema con máquinas AIX propias de IBM.

En este punto se sigue el patrón de ofrecer una solución con productos conocidos en el mercado y que generan prestigio. Siendo este punto fundamental para vender este tipo de soluciones.

Con esto, la instalación se realizará con los *scripts* propios del instalador y por línea de comandos, documentando exhaustivamente todo el proceso en el ANEXO A.

Información consultada:

"Instalación de Informix en Windows"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_11.70.0/com.ibm.igmsw.doc/WIN\\_wrapper.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_11.70.0/com.ibm.igmsw.doc/WIN_wrapper.htm)

"Instalación de Informix"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_11.50.0/com.ibm.4ql\\_inst\\_all.doc/fgl\\_ing\\_010.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_11.50.0/com.ibm.4ql_inst_all.doc/fgl_ing_010.htm)

"Installing Informix"

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.inst.doc/inst.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.inst.doc/inst.htm)

## 6.2.2. Configuración de Informix

Éste es un aspecto muy importante ya que la configuración del gestor de base de datos permite obtener soluciones óptimas de rendimiento en función de las especificaciones claras en el modelo a implementar como son el guardar una traza de todo el recorrido de un *ticket*.

Además, IBM proporciona unos aspectos fundamentales en sus manuales de administración y administración avanzada para sentar las bases de un sistema configurado para alcanzar un rendimiento óptimo y con vistas a la escalabilidad del sistema.

Así mismo, en el modelo Global Service Desk hay que almacenar información propia de un *data warehouse* ya que no hay contemplado borrado (que se definirá en procesos *batch* e incluso como intervenciones programadas), por lo

que hay que alojar los datos en tablas de acceso *online* que contienen registros históricos.

Además el gestor ha de estar configurado para ofrecer soltura y rapidez en todo el ciclo de vida de las peticiones, ya que se contemplan un número indeterminado de usuarios accediendo desde cualquier dispositivo, incluso móvil.

Las tareas de configuración comprenden:

- Configuración de la arquitectura de la instancia y la base de datos donde se alojarán los datos, como los discos en modo RAW, espacios de almacenamiento separados para el dbspace root, espacios temporales para ordenaciones en memoria, separar datos e índices en dbspaces diferentes, etc.
- Parámetros de configuración propios del número de usuarios, memoria, tamaño de *logs* de transacciones (*logical logs*), volcado de datos a disco mediante operaciones *flush*, *checkpoints*, etc.
- Al ser un sistema en producción y que además se va a configurar en alta disponibilidad (HA) hay que hacer backup de los registros lógicos de transacciones (LOGICAL LOGS), para lo que se ha definido un gestor de almacenamiento (TSM también de IBM), indicando al gestor las variables DSM necesarias.
- También, se definen TERMACP Y TERMINFO por si los clientes que se conectan utilizan sistemas V y 4.3 BSD

En consecuencia, en el ANEXO B se detallan todos los pasos técnicos empleados para la configuración del sistema.

Información consultada:

"Guía de instalación"

[https://publib.boulder.ibm.com/tividd/td/tec/SC32-0823-00/es\\_ES/HTML/INSTmst65.htm](https://publib.boulder.ibm.com/tividd/td/tec/SC32-0823-00/es_ES/HTML/INSTmst65.htm)

"Configuración para Informix"

[https://www.ibm.com/support/knowledgecenter/es/SSZJPZ\\_9.1.0/com.ibm.swg.im.iis.conn.drs.doc/topics/DRS010.html](https://www.ibm.com/support/knowledgecenter/es/SSZJPZ_9.1.0/com.ibm.swg.im.iis.conn.drs.doc/topics/DRS010.html)

"Configuración manual de Informix"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_11.70.0/com.ibm.igmsw.doc/ids\\_cw\\_012x.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_11.70.0/com.ibm.igmsw.doc/ids_cw_012x.htm)

"IBM® Tivoli Storage Manager"

[https://www.ibm.com/support/knowledgecenter/en/SSGSG7/landing/welcome\\_ssgsq7.html](https://www.ibm.com/support/knowledgecenter/en/SSGSG7/landing/welcome_ssgsq7.html)

"Using OnBar and ontape utility"

<https://www.ibm.com/developerworks/tivoli/library/t-infoback-tsam/index.html>

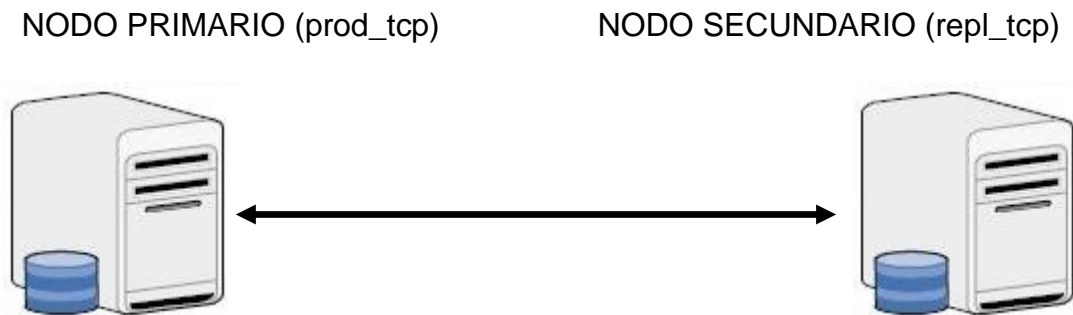
### 6.2.3. Instalación y configuración de HDR (HA) de Informix

La solución HA de Informix (IBM HDR) es una solución muy conocida en el mercado, así como su robustez y su sencilla administración.

Esto, junto a que la solución viene incluida en la licencia del producto hace muy atractivo Informix como tecnología a implantar.

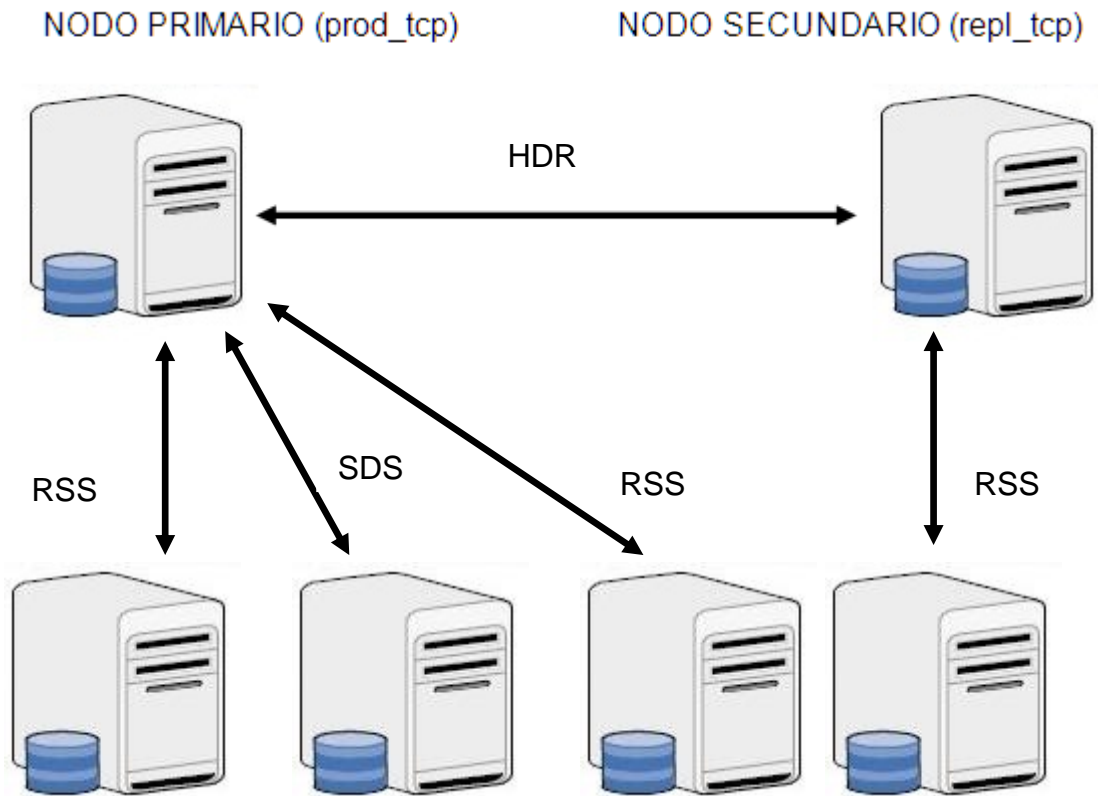
Para ello, se establecen dos máquinas que serán las reales que darán servicio a todo el sistema Global Service Desk, con sistema operativo AIX y con las instancias:

- Prod\_tcp → nodo primario
- Repl\_tcp → nodo secundario.



Al establecer un sistema HA se asegura un nivel de disponibilidad alto del sistema, manteniendo el servicio ante una caída de uno de los nodos mediante la toma de control del sistema por el otro nodo con una mínima intervención humana.

Además, todos los sistemas de replicación de Informix (HDR, ER, SD, RS, etc.) son totalmente compatibles entre sí y adjuntos con la licencia del producto, de esta forma se puede establecer la red de replicación necesaria para ofrecer las soluciones de escalabilidad que el mercado necesite.



Para configurar y activar la replicación basta con realizar una restauración del nodo primario en el nodo secundario. Desde la máquina secundaria donde va a residir la réplica HDR (se ha configurado como repl\_tcp) y apuntando a la máquina primaria (se ha configurado como prod\_tcp).

Se utiliza el comando facilitado por IBM ontape (para realizar copias de seguridad) con las opciones STUDIO para que los datos se envíen sin almacenarlos en un disco o cinta.

De tal forma, que todos los aspectos técnicos quedan documentados en el ANEXO D.

Información consultada:

"Sistema de réplica de alta disponibilidad"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_12.1.0/com.ibm.erep.doc/ids\\_erp\\_154.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_12.1.0/com.ibm.erep.doc/ids_erp_154.htm)

"Novedades de Enterprise Replication para Informix, versión 12.10"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_12.1.0/com.ibm.erep.doc/ids\\_erp\\_602.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_12.1.0/com.ibm.erep.doc/ids_erp_602.htm)

"Starting HDR for the First Time"

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.admin.doc/ids\\_admin\\_0931.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.admin.doc/ids_admin_0931.htm)

#### 6.2.4. Instalación y configuración de auditoría de Informix

Hoy en día, la instalación de cualquier producto y sobre todo de soluciones integrales exige un nivel de seguridad máximo que ofrezca la posibilidad de monitorizar la actividad de cualquier usuario.

Además, la generación de ficheros de auditoría es la mejor forma de registrar toda la actividad de los usuarios y procedimientos en la base de datos, generando un *log* de todas las acciones sobre ella.

Es por ello que el sistema Global Service Desk debe cumplir con normativas como la LOPD, así como ser capaz de ofrecer información del uso y acceso que se ofrece a los recursos, tanto por seguridad como por facilitar información a cualquier auditoría.

En esta línea, la herramienta de auditoría de Informix está incluida en la licencia del gestor, por lo que ya está instalada y no supone incremento de costes del producto.

En consecuencia, se debe configurar y arrancar para que genere los registros de las instrucciones a auditar, previamente definidas las máscaras en función del nivel exigido de seguridad.

La auditoría de Informix ofrece monitorización detallada a nivel de usuario y a nivel de objetos de la base de datos, incluso de conjuntos de registros dentro de una tabla, permitiendo de una forma sencilla y mediante simples comandos registrar toda la actividad definida.

Mediante el comando *onaudit* se gestiona la auditoría de Informix, en concreto para crear máscaras de auditoría para todos los usuarios y que se guarde registro de todas las operaciones; DELETE, INSERT, UPDATE, SELECT se realiza mediante comandos *onaudit*.

Así mismo, cuantos más eventos y más amplia sea la monitorización, más recursos exige la máquina donde se aloja el motor de base de datos. Ya que monitorizar instrucciones como SELECT repercute negativamente en el rendimiento del sistema.

La solución de auditoría registra los eventos en el siguiente formato, en el que se documentan:

- Hora del suceso.
- Máquina desde donde se realiza la petición.
- Nombre de la instancia.
- Usuario que provoca el evento.
- Tipo de evento que ha realizado.
- Base de datos donde se ha realizado el suceso.
- Identificador de la tabla.

- Identificador del registro.

```
ONLN|2018-05-07
13:13:10.445|acheron|34734222|prod_tcp|USU1_GSD|0:INRW:GSD:498:1153
4786:260::
ONLN|2018-05-07
13:13:10.535|acheron|2949412|prod_tcp|USU1_GSD|0:INRW:GSD:8292:1154
0635:10857793::
ONLN|2018-05-07
13:13:10.625|acheron|1049050|prod_tcp|USU1_GSD|0:INRW:GSD:10088:115
38418:151708930::
ONLN|2018-05-07
13:13:10.704|acheron|7209292|prod_tcp|USU1_GSD|0:INRW:GSD:8292:1154
0635:10857985::
ONLN|2018-05-07
13:13:10.869|acheron|35127484|prod_tcp|USU1_GSD|0:INRW:GSD:8292:115
40635:10857986::
ONLN|2018-05-07
13:13:10.883|acheron|35127484|prod_tcp|USU1_GSD|0:UPRW:GSD:133:1153
4421:228343823:11534421:228343823::
ONLN|2018-05-07
13:13:10.887|acheron|35127484|prod_tcp|USU1_GSD|0:UPRW:GSD:16317:11
540073:611121153:11540073:611121153::
ONLN|2018-05-07
13:13:10.890|acheron|35127484|prod_tcp|USU1_GSD|0:INRW:GSD:14236:11
540863:1221126::
ONLN|2018-05-07
13:13:10.998|acheron|7602560|prod_tcp|USU1_GSD|0:UPRW:GSD:14236:115
40863:1187850:11540863:1187850::
ONLN|2018-05-07
13:13:11.013|acheron|35127484|prod_tcp|USU1_GSD|0:DLRW:GSD:529:1153
4817:258::
```

En consecuencia, todos los aspectos técnicos quedan perfectamente documentados en el ANEXO E.

Información consultada:

"El programa de utilidad 'onaudit': configuración de la auditoría"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_12.1.0/com.ibm.sec.doc/ids\\_au\\_100.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_12.1.0/com.ibm.sec.doc/ids_au_100.htm)

"Códigos y campos de sucesos de auditoría"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_12.1.0/com.ibm.sec.doc/ids\\_au\\_104.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_12.1.0/com.ibm.sec.doc/ids_au_104.htm)

"Análisis de auditoría"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_12.1.0/com.ibm.sec.doc/ids\\_au\\_075.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_12.1.0/com.ibm.sec.doc/ids_au_075.htm)

### 6.3. Situación futura deseada

Con el objetivo de ser una solución no solo académica si no vendible profesionalmente en el mercado como una solución actual y completa que integra elementos definitorios, se presenta la siguiente situación futura ideal.

- Se desearía realizar baterías de pruebas reales en entornos con concurrencia real, para localizar fallos y problemas los más cercanos a la realidad.
- Realizar baterías de análisis en clientes para que sean ellos los que expongan las soluciones que desean y los puntos de mejoras que más se adapten al mundo real.
- Testeo de diferentes prototipos que más se ajustan a los clientes, además de generar evolutivos suficientemente testeados.
- Disponer de más tiempo para realizar más pruebas sobre los procedimientos de gestión del modelo GSD y detectar situaciones de mejora.
- Disponer de más tiempo para implementar en su totalidad el repositorio estadístico y sugerir más métricas que incluso se puedan cruzar con aplicativos de BI.

En consecuencia, se optimizaría la solución y se completaría el modelo para que cubra un abanico amplio de necesidades reales de negocio, aumentando con ello la capacidad de venta frente a la competencia.

En esta línea, también sería deseable integrar el sistema con soluciones ERP e incluso BI para que mediante la información estructurada que se obtenga de Global Service Desk analizar parámetros de negocio y mejoras frente a la competencia.

## 7. Glosario

**BBDD:** Bases de datos.

**OFF-LINE:** Fuera de línea, gestor de BBDD no operativo.

**Horario OFF-LINE:** Fuera del horario laboral.

**ON-LINE:** En línea, gestor de BBDD operativo

**Horario ON-LINE:** Dentro del horario laboral

**OLTP:** Procesamiento de Transacciones En Línea

**IDS:** Informix Dynamic Server

**HA:** High availability, alta disponibilidad.

**HDR:** Alta disponibilidad por replicación de BBDD de Informix

**GSD:** Global Service Desk

## 8. Bibliografía/Webgrafía

*"IBM Informix high availability and scalability"*

[https://www.ibm.com/developerworks/data/roadmaps/roadmap\\_ifx\\_hacluster.html](https://www.ibm.com/developerworks/data/roadmaps/roadmap_ifx_hacluster.html)

*"Configuration of IBM Informix high-availability support for Java clients"*

[https://www.ibm.com/support/knowledgecenter/en/SSEPGG\\_9.7.0/com.ibm.db2.luw.apdv.java.doc/src/tpc/imjcc\\_c0056168.html](https://www.ibm.com/support/knowledgecenter/en/SSEPGG_9.7.0/com.ibm.db2.luw.apdv.java.doc/src/tpc/imjcc_c0056168.html)

*"Informix"*

<https://es.wikipedia.org/wiki/Informix>

*"Establecimiento del nivel de auditoría"*

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_12.1.0/com.ibm.sec.doc/ids\\_au\\_057.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_12.1.0/com.ibm.sec.doc/ids_au_057.htm)

*"Máscaras de auditoría"*

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_12.1.0/com.ibm.sec.doc/ids\\_au\\_004.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_12.1.0/com.ibm.sec.doc/ids_au_004.htm)

*"Usuario Azul"*

<http://es.gofreedownload.net/free-icon/vista-icon/blue-user-110995/#.WsJHINRuZhE>

*"Product overview"*

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.po.doc/po.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.po.doc/po.htm)

*"Powerful IBM Informix ODBC driver"*

<https://www.progress.com/odbc/ibm-informix>

*"Guide to SQL: Syntax"*

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.sqls.doc/sqls.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.sqls.doc/sqls.htm)

*"Sintaxis de Informix SQL"*

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_11.50.0/com.ibm.sqlnode.doc/sql.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_11.50.0/com.ibm.sqlnode.doc/sql.htm)

*"Setting the lock mode to wait"*

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.perf.doc/ids\\_prf\\_422.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.perf.doc/ids_prf_422.htm)

*"Niveles de aislamiento de IBM Data Server Driver para JDBC y SQLJ"*

[https://www.ibm.com/support/knowledgecenter/es/SSEPGG\\_9.5.0/com.ibm.db2.luw.apdv.java.doc/src/tpc/imjcc\\_r0052429.html](https://www.ibm.com/support/knowledgecenter/es/SSEPGG_9.5.0/com.ibm.db2.luw.apdv.java.doc/src/tpc/imjcc_r0052429.html)

*"Niveles de aislamiento"*

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_11.50.0/com.ibm.ddi.doc/ids\\_ddi\\_042.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_11.50.0/com.ibm.ddi.doc/ids_ddi_042.htm)



"DEADLOCK TIMEOUT y SET LOCK MODE"

[https://www.ibm.com/support/knowledgecenter/es/SSGU8G\\_11.50.0/com.ibm.ddi.doc/ids\\_ddi\\_043.htm](https://www.ibm.com/support/knowledgecenter/es/SSGU8G_11.50.0/com.ibm.ddi.doc/ids_ddi_043.htm)

"Qué es el lenguaje unificado de modelado (UML)"

<https://www.lucidchart.com/pages/es/qu%C3%A9-es-el-lenguaje-unificado-de-modelado-uml>

"Lenguaje unificado de modelado"

[https://es.wikipedia.org/wiki/Lenguaje\\_unificado\\_de\\_modelado](https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado)

"Modelo de clases"

<https://users.dcc.uchile.cl/~psalinas/uml/modelo.html>

"LOCK MODE Clause"

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.sqls.doc/ids\\_sqs\\_0083.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.sqls.doc/ids_sqs_0083.htm)

"Extents"

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.admin.doc/ids\\_admin\\_0485.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.admin.doc/ids_admin_0485.htm)

"Extent sizes for tables in a dbspace"

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_11.50.0/com.ibm.perf.doc/ids\\_prf\\_311.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_11.50.0/com.ibm.perf.doc/ids_prf_311.htm)

"Informix Administrators Blog"

[https://www.ibm.com/developerworks/community/blogs/informix\\_admins\\_blog/entry/ids\\_pagesize\\_and\\_maximum\\_number\\_of\\_extents5?lang=es](https://www.ibm.com/developerworks/community/blogs/informix_admins_blog/entry/ids_pagesize_and_maximum_number_of_extents5?lang=es)

"How to find the number of extents used by a table using oncheck"

<http://www-01.ibm.com/support/docview.wss?uid=swg21153986>

"Extent sizes for tables in a dbspace"

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_11.50.0/com.ibm.perf.doc/ids\\_prf\\_311.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_11.50.0/com.ibm.perf.doc/ids_prf_311.htm)

"Tuning Informix SQL"

<https://www.ibm.com/developerworks/data/library/techarticle/dm-0409fan/index.html>

"Examples and Analysis"

<https://www.ibm.com/developerworks/data/zones/informix/library/techarticle/0502fan/0502fan.html>

"Storage optimization methods"

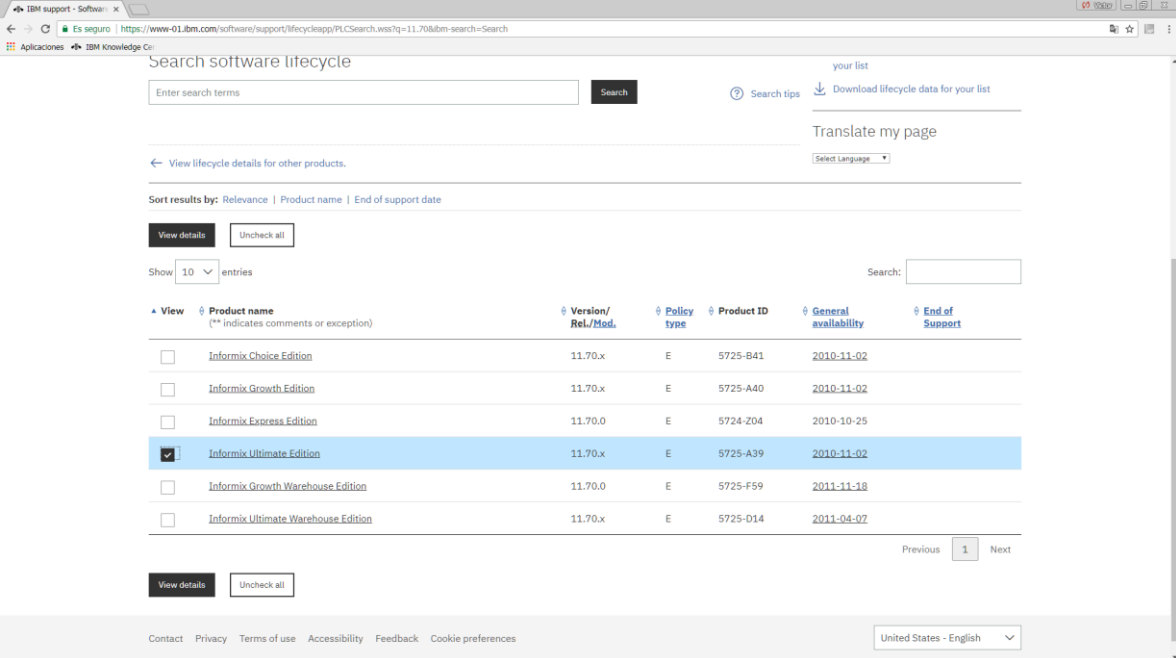
[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.admin.doc/ids\\_admin\\_1421.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.admin.doc/ids_admin_1421.htm)

## 9. ANEXO A: Instalación de Informix

Debido a los requisitos necesarios para cumplimentar el proyecto y descritos en el apartado anterior, se opta por un sistema gestor de base de datos con alta solidez y experiencia en el mercado para ofrecer soluciones presentes y futuras al Global Service Desk.

Es por ello, que se elige Informix como gestor de base de datos y en concreto la versión 11.70.FC8 que ya está suficientemente testada por clientes, así como totalmente dentro de soporte de IBM y sin fecha de fin del mismo actualmente.

Se muestra una captura de pantalla del ciclo de vida de Informix Dynamic Server (Ultimate Edition) versión 11.70.FC8 en la web de IBM:



The screenshot shows the IBM Support website interface for searching software lifecycle information. The search results are displayed in a table with the following columns: Product name, Version/Rel./Mod., Policy type, Product ID, General availability, and End of Support. The 'Informix Ultimate Edition' row is selected and highlighted in blue.

View	Product name (* indicates comments or exception)	Version/ Rel./Mod.	Policy type	Product ID	General availability	End of Support
<input type="checkbox"/>	Informix Choice Edition	11.70.x	E	5725-B41	2010-11-02	
<input type="checkbox"/>	Informix Growth Edition	11.70.x	E	5725-A40	2010-11-02	
<input type="checkbox"/>	Informix Express Edition	11.70.0	E	5724-Z04	2010-10-25	
<input checked="" type="checkbox"/>	Informix Ultimate Edition	11.70.x	E	5725-A39	2010-11-02	
<input type="checkbox"/>	Informix Growth Warehouse Edition	11.70.0	E	5725-F59	2011-11-18	
<input type="checkbox"/>	Informix Ultimate Warehouse Edition	11.70.x	E	5725-D14	2011-04-07	

The screenshot shows the IBM Support website for Informix Ultimate Edition 11.70.x. The page includes a navigation bar with the IBM logo, a search bar, and a 'Marketplace' button. The main content area is titled 'Software lifecycle' and 'Informix Ultimate Edition 11.70.x'. It contains several sections: 'Announcement letter dates are US only...', 'Subscribe to software support lifecycle updates', 'Download software support lifecycle data', 'View lifecycle details for other products', 'Informix Ultimate Edition' (Version: 11.70.x, License type: JPLA, PID: 5725-A39), 'Lifecycle dates, announcement letters and other information' (ES: 02-Nov-2010, 210-367, EQM: 12-Dec-2017, 917-138, Last updated: 18-Sep-2017), 'My product list' (1 product is in your list, Manage your product list, Subscribe to the lifecycle news feed for your list, Download lifecycle data for your list), and 'Translate my page'. The footer contains links for 'Contact', 'Privacy', 'Terms of use', 'Accessibility', 'Feedback', 'Cookie preferences', and a language selector set to 'United States - English'.

Además, evitamos instalar la última versión de Informix que no ofrece las garantías de tiempo de solidez en el mercado.

Conviene subrayar que las tareas de instalación se harán de acuerdo al manual de administración de Informix facilitado por IBM y en concreto con la información de su WEB:

[https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.inst.doc/inst.htm](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.inst.doc/inst.htm)

Es por esto que, la instalación se realizará sobre un Sistema Operativo AIX dando mayor robustez al sistema con máquinas UNIX propias de IBM.

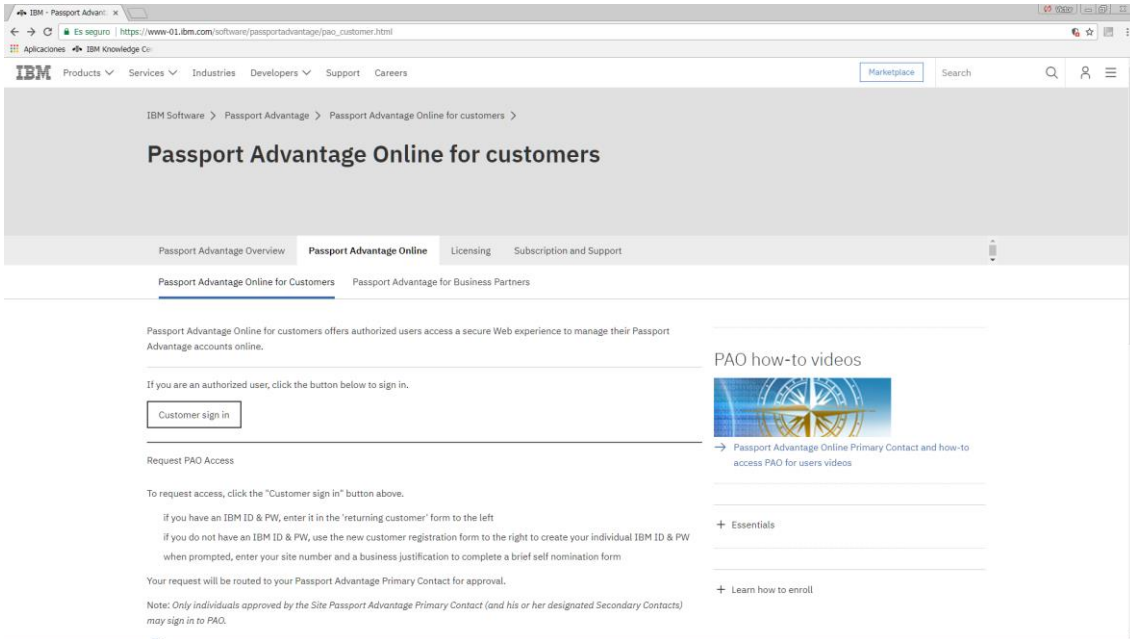
Así mismo, la instalación se realizará con los *scripts* propios del instalador y por línea de comandos, documentando exhaustivamente todo el proceso.

Por lo tanto, descargamos el fichero de instalación para Unix sobre una máquina AIX con Sistema Operativo versión 7.1, siguiendo los siguientes pasos:

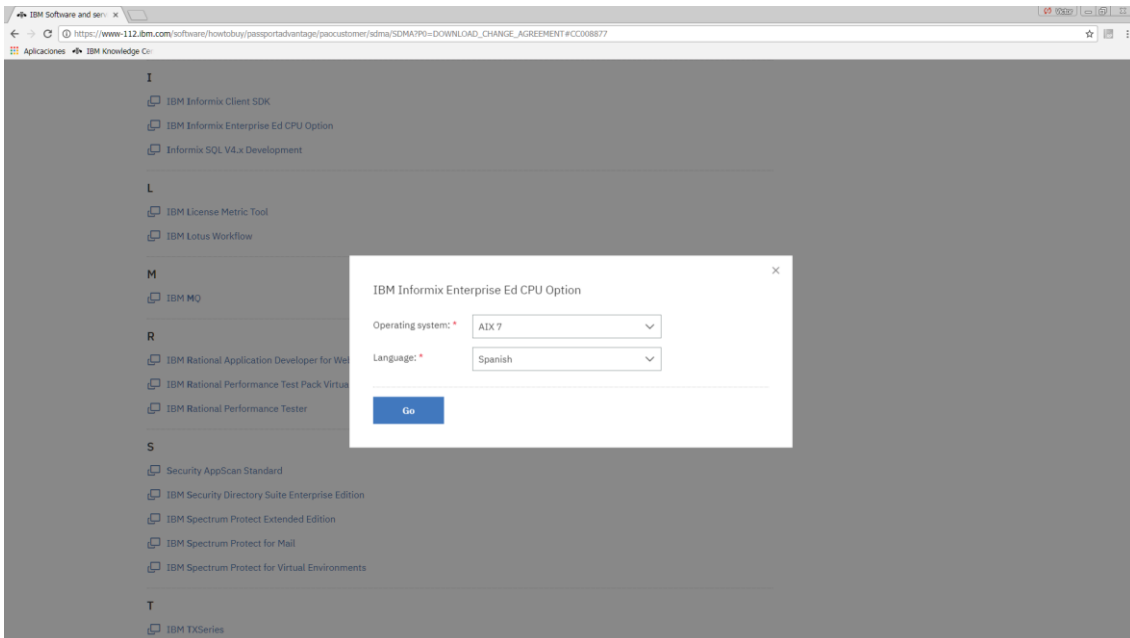
## 1.1 PASOS DE INSTALACIÓN:

### 1.1.1 **Descargar el fichero de instalación:**

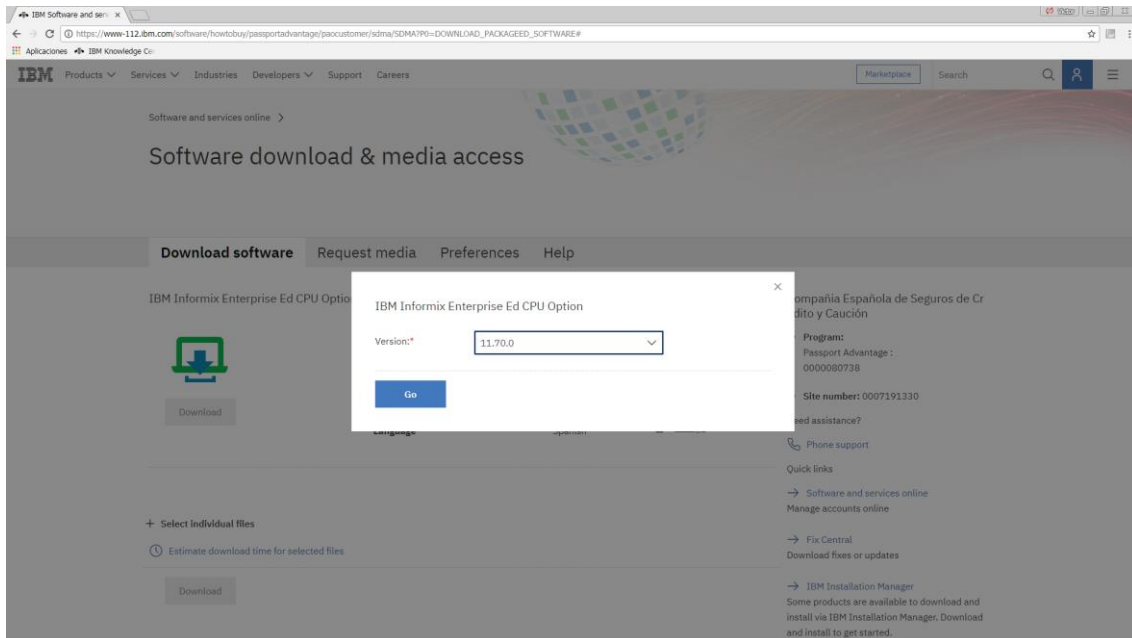
Se descarga el fichero de instalación de IDS (Ultimate Edition Versión 11.70 FC8) del Passport Advantage de IBM:



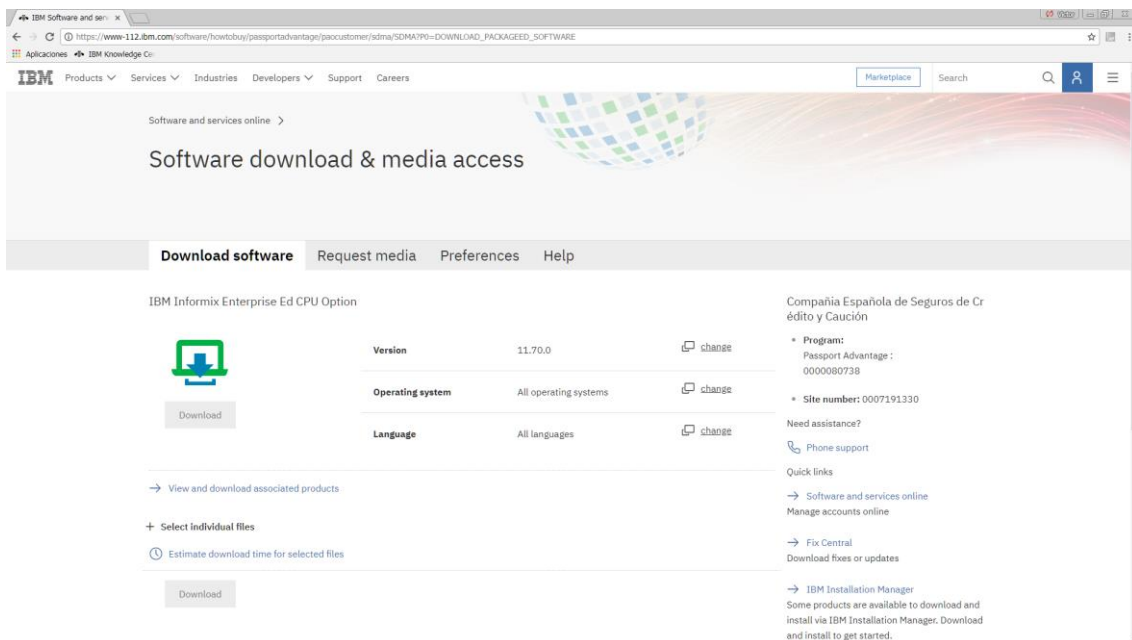
Descargamos el producto seleccionando el SO de la máquina donde se instalará:



Indicamos la versión que se va a descargar:



Se procede a descargar el producto:



### 1.1.2 Creación de usuario, lv y filesystem necesario:

Creamos el usuario informix perteneciente al grupo informix que será el administrador de la instancia:

```
acheron:informix:[]:/usr/local/informix>id
uid=101(informix) gid=101(informix) groups=1(staff)
```

Comprobamos el VG de AIX donde tenemos espacio y donde vamos a crear el lv que alojará el file system dedicado para la instalación del producto:

```

acheron:informix:[]:/usr/local/informix>lsvg ifx104vg
VOLUME GROUP:          ifx104vg          VG IDENTIFIER:
00f6fdcc00004c000000014be44038c1
VG STATE:              active            PP SIZE:      8 megabyte(s)
VG PERMISSION:         read/write        TOTAL PPs:    25537 (204296
megabytes)
MAX LVs:               256              FREE PPs:     192 (1536 megabytes)
LVs:                   11               USED PPs:     25345 (202760 megabytes)
OPEN LVs:              11              QUORUM:       4 (Enabled)
TOTAL PVs:              7              VG DESCRIPTORS: 7
STALE PVs:              0              STALE PPs:    0
ACTIVE PVs:             7              AUTO ON:      yes
MAX PPs per VG:        32768            MAX PVs:      1024
LTG size (Dynamic):    256 kilobyte(s)  AUTO SYNC:    no
HOT SPARE:              no              BB POLICY:    relocatable
MIRROR POOL STRICT:   off
PV RESTRICTION:        none             INFINITE RETRY: no
DISK BLOCK SIZE:       512              CRITICAL VG:  no
FS SYNC OPTION:        no

```

Creamos el lv a través del menú de AIX smitty indicándole que va a ser de 3 Gbytes, en este caso 384 páginas de 8 Mbytes que es el tamaño del VG seleccionado, de tipo jfs2:

```

Add a Logical
Volume
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]
Logical volume NAME          [informixlv]
* VOLUME GROUP name          ifx104vg
* Number of LOGICAL PARTITIONS [384]
#
PHYSICAL VOLUME names       []
+
Logical volume TYPE          [jfs2]
+
POSITION on physical volume  middle
+
RANGE of physical volumes    minimum
+
MAXIMUM NUMBER of PHYSICAL VOLUMES []
#
to use for allocation

```

```

Number of COPIES of each logical                                1
+
  partition
Mirror Write Consistency?                                     active
+
Allocate each logical partition copy                          yes
+
  on a SEPARATE physical volume?
RELOCATE the logical volume during                            yes
+
  reorganization?
Logical volume LABEL                                         []
MAXIMUM NUMBER of LOGICAL PARTITIONS                          [512]
#
Enable BAD BLOCK relocation?                                  yes
+
SCHEDULING POLICY for writing/reading                          parallel
+
  logical partition copies
Enable WRITE VERIFY?                                         no
+
File containing ALLOCATION MAP                                  []
Stripe Size?                                                 [Not Striped]
+
Serialize IO?                                                no
+
Mirror          Pool          for          First          Copy
+
Mirror          Pool          for          Second         Copy
+
Mirror          Pool          for          Third          Copy
+
Infinite Retry Option                                       no

```

Creamos el File System con las opciones de montaje automático para asegurar que siempre esté montado aunque se reinicie la máquina:

```

Add an Enhanced
Journaled File System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]
* LOGICAL VOLUME name                                         informixlv
+
* MOUNT POINT                                                 [/usr/local/informix]
Mount AUTOMATICALLY at system restart?                       yes
+

```

```

PERMISSIONS                                read/write
+
Mount OPTIONS                               []
+
Block Size (bytes)                          4096
+
Logical          Volume          for          Log
+
Inline Log size (MBytes)                    []
#
Extended          Attribute          Format
+
ENABLE Quota Management?                    no
+
Enable EFS?                                  no
+
Allow internal snapshots?                    no
+
Mount GROUP          []

```

Se monta el file system creado y se dan permisos al usuario informix:

```

acheron:informix:[]:/usr/local/informix>mount /usr/local/informix
acheron:informix:[]:/usr/local/informix>chown informix:informix /usr/local/informix

```

### 1.1.3. Instalación de IDS 11.70 FC8

Conviene crear un directorio para alojar los ficheros que se descomprimen y que, una vez instalado, se pueda borrar para liberar espacio y dejar una instalación limpia:

```

acheron:informix:[]:/usr/local/informix>mkdir instalar
acheron:informix:[]:/usr/local/informix>cd instalar
acheron:informix:[]:/usr/local/informix/instalar>

```

Descomprimos el fichero como usuario informix:

```

acheron:informix:[]:/usr/local/informix/instalar>ls -l
total 1042152
-rw-r--r--      1  informix  informix    533575680  Feb  18  2014
IUE_11.70.FC8_AIX_64_ML.tar
acheron:informix:[]:/backups_bbdd/IFXDESK/instalar>tar -xvf
IUE_11.70.FC8_AIX_64_ML.tar

```



Exportamos las variables de entorno relativas al directorio de informix donde vamos a instalar (INFORMIXDIR) y el nombre de la instancia (INFORMIXSERVER), a partir de este paso siempre como usuario root:

```
acheron:root:[]:/backups_bbdd/IFXDESK/instalar>export  
INFORMIXDIR=/usr/local/informix  
acheron:root:[]:/backups_bbdd/IFXDESK/instalar>export  
INFORMIXSERVER=prod_tcp
```

Ejecutamos el instalador:

```
acheron:root:[]:/backups_bbdd/IFXDESK/instalar>./ids_install
```

Arranca el instalador:

```
Preparing to install...  
Extracting the JRE from the installer archive...  
Unpacking the JRE...  
Extracting the installation resources from the installer archive...  
Configuring the installer for this system's environment...  
  
Launching installer...  
  
=====
```

IBM Informix Software Bundle	(created with InstallAnywhere)
------------------------------	--------------------------------

```
-----  
  
Preparing CONSOLE Mode Installation...  
  
=====
```

Getting started with IBM Informix Software Bundle
---

```
-----
```

Empezamos a instalar el producto:

```
=====
=====
Installation Goals
-----

What do you want to accomplish?

->1- Install products and features
   2- Extract the product files (-DLEGACY option)

Seleccionamos la opción complete:

->1- Typical
   2- Custom
```

Elegimos todas las opciones por defecto, con el ODBC, CSDK (cliente de Informix), GLS (lenguajes) y IBM Informix database server (servidor de base de datos):

```
=====
=====
Choose Product Features
-----

ENTER A COMMA_SEPARATED LIST OF NUMBERS REPRESENTING THE
FEATURES YOU WOULD
LIKE TO SELECT, OR DESELECT. TO VIEW A FEATURE'S DESCRIPTION,
ENTER
'?<NUMBER>'. PRESS <RETURN> WHEN YOU ARE DONE:

  1-  IBM Informix database server
  2-  Base Server
  3-  Extensions and tools
  4-  J/Foundation
  5-  Database extensions
  6-  Conversion and reversion support
  7-  XML publishing
  8-  Demonstration database scripts
  9-  Enterprise Replication
 10-  Data loading utilities
 11-  onunload and onload utilities
 12-  dbload utility
 13-  High-Performance Loader
 14-  Backup and Restore
 15-  archecker utility
 16-  ON-Bar utility
 17-  Informix Storage Manager
 18-  Informix interface to Tivoli Storage Manager
```

- 19-  Administrative utilities
- 20-  Performance monitoring utilities
- 21-  Miscellaneous monitoring utilities
- 22-  Auditing utilities
- 23-  Database import and export utilities
- 24-  IBM Informix Client SDK
- 25-  IBM Informix Object Interface for C++
- 26-  IBM Informix Object Interface for C++ demos
- 27-  IBM Informix ESQL/C
- 28-  IBM Informix ESQL/C demos
- 29-  7.2 application compatibility module
- 30-  IBM Informix LIBDMI for client applications
- 31-  IBM Informix ODBC Driver
- 32-  IBM Informix ODBC Driver demos
- 33-  IBM Informix Connect
- 34-  IBM Informix Object Interface for C++
- 35-  IBM Informix ESQL/C
- 36-  7.2 application compatibility module
- 37-  IBM Informix LIBDMI for client applications
- 38-  IBM Informix ODBC Driver
- 39-  IBM Informix JDBC
- 40-  Global Language Support (GLS)
- 41-  West European and Americas
- 42-  East European and Slavic
- 43-  Chinese
- 44-  Japanese
- 45-  Korean
- 46-  Thai

ENTER THE NUMBER FOR YOUR CHOICE, OR PRESS <ENTER> TO  
ACCEPT THE DEFAULT:: 2

La instancia la creamos por fuera del instalador y procedemos a instalar los productos:

```
=====
=====
Installing...
-----

[=====|=====|=====|=====]
=====]
[-----|-----|-----|-----]

=====
=====
Installation Complete
-----

Congratulations! IBM Informix Software Bundle installation is complete.

Product install status:
IBM Informix 11.70: Successful
IBM Informix Client-SDK: Successful
IBM Informix JDBC Driver: Successful

For more information about using Informix products, see the IBM Informix 11.70
Information Center at
http://publib.boulder.ibm.com/infocenter/idshelp/v117/index.jsp.
```

Como se puede ver, la instalación se ha realizado correctamente, observamos los binarios instalados y borramos el directorio con los ficheros propios de la instalación para liberar espacio y dejar el entorno limpio.

## 10. ANEXO B: Configuración de Informix

Otro punto importante es la configuración del gestor de base de datos, ya que hay especificaciones claras en el modelo a implementar como son el guardar una traza de todo el recorrido de un *ticket*.

También, hay que almacenar información propia de un *data warehouse*, ya que no hay contemplado borrado, por lo que hay que alojar los datos en tablas de acceso *online* que contienen registros históricos.

Además, el gestor ha de estar configurado para ofrecer soltura y rapidez en todo el ciclo de vida de las peticiones, ya que se contemplan un número indeterminado de usuarios accediendo desde cualquier dispositivo, incluso móvil.

Las tareas de configuración comprenden:

- Configuración de la arquitectura de la instancia y la base de datos donde se alojarán los datos, como los discos en modo RAW, espacios de almacenamiento separados para el dbspace root, temporales, separar datos e índices en dbspaces diferentes, etc.
- Parámetros de configuración propios del número de usuarios, memoria, tamaño de *logs* de transacciones (*logical logs*), volcado de *flush* a disco, *checkpoints*, etc.

### 1.1 PASOS PARA CONFIGURAR LA INSTANCIA DE BASE DE DATOS

#### 1.1.1 Creación de fichero de variables:

Para crear una nueva instancia hay que crear un fichero de variables para indicarle al gestor de base de datos cuál es su INFORMIXDIR, su fichero de configuración de conexiones (SQLHOSTS), fichero de configuración de la instancia (ONCONFIG), incluso las variables de entorno del formato de fecha, etc.

De la siguiente forma:

```
acheron:informix:[]:/usr/local/informix>cat variab_prod
export INFORMIXDIR=/usr/local/IDS
export
INFORMIXSQLHOSTS=/usr/local/scripts/BBDD/informix/entorno/sqlhosts.prod
export INFORMIXSERVER=prod_tcp
export ONCONFIG=onconfig.prod
export DBDATE=y4md0
export TERMCAP=$INFORMIXDIR/etc/termcap
export PATH=$INFORMIXDIR/bin:$PATH
```

```

export INFORMIXTERM=terminfo
export IFX_DEF_TABLE_LOCKMODE=ROW

#export DSMI_CONFIG=/usr/tivoli/tsm/client/api/bin64/dsm_informix.opt
#export DSMI_DIR=/usr/tivoli/tsm/client/api/bin64
#export DSMI_LOG=/usr/tivoli/tsm/client/api/bin64

# POR PMR 05664
#export DONTDRAINPOOLS=1

export
LIBPATH=/usr/lib:$COBDIR/lib:$INFORMIXDIR/lib:$INFORMIXDIR/lib/esql:$INFORMIXDIR/lib/tools:$LIBPATH

```

Al ser un sistema en producción y que además se va a configurar en alta disponibilidad (HA) hay que hacer *backup* de los registros lógicos de transacciones (LOGICAL LOGS), para lo que se ha definido un gestor de almacenamiento (TSM), con lo que se le indica al gestor con las variables DSM.

También, se definen TERMACP Y TERMINFO por si los clientes que se conectan con sistemas V y 4.3 BSD

### 1.1.2 Creación de fichero de conexiones (SQLHOST)

Hay que configurar las conexiones disponibles de las que va a disponer la instancia nueva, para ello definimos dentro de la ruta que le indicamos en el fichero de variables:

```

export
INFORMIXSQLHOSTS=/usr/local/scripts/BBDD/informix/entorno/sqlhosts.prod
acheron:informix:[prod_tcp]:/usr/local/informix>cat $INFORMIXSQLHOSTS
#*****
#
# Licensed Material - Property Of IBM
#
# "Restricted Materials of IBM"
#
# IBM Informix Dynamic Server
# (c) Copyright IBM Corporation 1996, 2004 All rights reserved.
#
# Title:   sqlhosts.demo
# Description:
#         Default sqlhosts file for running demos.
#
#*****

# Instancia de Produccion
prod_tcp      onsoctcp      acheron      serv_prod_tcp
repl_tcp      onsoctcp      elcano       1545

```

```
#####  
#####
```

Se definen puertos libres. En el primario (prod\_tcp) se define el puerto:

```
acheron:informix:[prod_tcp]:/usr/local/informix>egrep -i "serv_prod_tcp"  
/etc/services  
serv_prod_tcp 1533/tcp
```

Y en el secundario (repl\_tcp) el 1545.

### 1.1.3 Creación del rootdbspace y su lv en modo RAW

Creamos el lv en modo RAW que contendrá el CHUNK para el ROOTDBSPACE de 2Gbytes, espacio suficiente ya que es conveniente crear los LOGICAL LOGS y el PHYSICAL LOG en dbspace separados y sacarlos del ROOTDBSPACE:

Se crea el LV rootdbs con 256 páginas de 8 Mbytes que es el tamaño de página del VG seleccionado.

```
mklv -y'rootdbs' -t'raw' -e'x' ifx10vg 256
```

Hay que dar los permisos que necesita informix a los LV de la instancia, estos son 660 y usuario y propietario informix:

```
acheron:informix:[prod_tcp]:/usr/local/informix>cd /dev  
acheron:informix:[prod_tcp]:/usr/local/informix>chmod 660 rootdbs  
acheron:informix:[prod_tcp]:/usr/local/informix>chown informix:informix rootdbs
```

Una vez con los permisos adecuados, es conveniente crear un *link* para tener los LV ordenados y tener más movilidad al no acceder directamente a los lv:

```
acheron:informix:[prod_tcp]:/usr/local/informix>mkdir /chk_ifx/prod/  
acheron:informix:[prod_tcp]:/usr/local/informix>ln -s /dev/rootdbs  
/chk_ifx/prod/rootdbs  
acheron:informix:[prod_tcp]:/usr/local/informix>ls -l /chk_ifx/prod/rootdbs  
lrwxrwxrwx 1 informix informix 13 Apr 06 2017 /chk_ifx/prod/rootdbs ->  
/dev/rootdbs
```

#### 1.1.4 Inicializar la instancia

Asignamos los siguientes valores en el fichero de configuración de la instancia (onconfig.prod):

```
ROOTNAME rootdbs
ROOTPATH /chk_ifx/prod/rootdbs
ROOTOFFSET 0
ROOTSIZE 2097152
MIRROR 0
MIRRORPATH
MIRROROFFSET 0

SERVERNUM 0
DBSERVERNAME prod_tcp
```

Los demás valores los dejamos por defecto e inicializamos la instancia:

```
acheron:informix:[prod_tcp]:/usr/local/informix>oninit -iv
```

Lo lanzamos con “-v”, verbose para mostrar información por pantalla.

Respondemos yes (es una operación delicada ya que inicializa toda la memoria y borra toda la información del rootdbspace).

Cuando termina el comando ya tenemos la instancia creada y podemos ver el almacenamiento con onstat -d (en sistemas UNIX el tamaño mostrado se da en páginas de 4Kbytes):

```
IBM Informix Dynamic Server Version 11.70.FC8 -- On-Line (Prim) -- Up 1 days
01:34:53 -- 14642880 Kbytes
```

##### Dbspaces

address	number	flags	fchunk	nchunks	pgsize	flags	owner
70000019623d028	1	0x40001	1	1	4096	N BA	informix

rootdbs

1 active, 2047 maximum

##### Chunks

address	chunk/dbs	offset	size	free	bpages	flags	pathname
70000019623d1d0	1	1 0	524288	502392			PO-B--

/chk\_ifx/prod/rootdbs

1 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are



displayed in terms of "pgsize" of the DBspace to which they belong.

Expanded chunk capacity mode: always

### 1.1.5 Creación del PHYSICAL LOG

Se crea el PHYSICAL LOG y se saca del ROOTDBSPACE, de esta forma nuestra arquitectura es más sencilla de administrar al poder monitorizar de forma aislada el physical log, en él se guarda una imagen previa de las páginas que van a sufrir modificaciones:

Para ello, creamos el dbspace que va a contener el physical log siguiendo los pasos del rootdbspace:

```
acheron:informix:[prod_tcp]:/usr/local/informix>ls -l /chk_ifx/prod/physdbs
lrwxrwxrwx  1 informix informix    13 Apr 06 2017 /chk_ifx/prod/physdbs ->
/dev/rphysdbs
```

```
acheron:informix:[prod_tcp]:/usr/local/informix>onspaces -c -d physdbs -p
/chk_ifx/prod/physdbs -o 0 -s 2129920
```

Una vez que tenemos el dbspace del physical log creado se procede a sacarlo del rootdbspace ajustando el tamaño del physical log para que entre en ese espacio, ya que se reservan por defecto una serie de páginas para operaciones del gestor:

```
acheron:informix:[prod_tcp]:/usr/local/informix>onparams -p -s 2048000 -d
physdbs
```

Ya podemos ver reflejado el cambio en el fichero de configuración (ONCONFIG):

```
PHYSFILE 2048000
PLOG_OVERFLOW_PATH $INFORMIXDIR/tmp
PHYSBUFF 1024

#####
####
# Logical Log Configuration Parameters
#####
####
# LOGFILES    - The number of logical log files
# LOGSIZE     - The size of each logical log, in KB
# DYNAMIC_LOGS - The type of dynamic log allocation.
#             Acceptable values are:
#             2 Automatic. IDS adds a new logical log to the
#             root dbspace when necessary.
#             1 Manual. IDS notifies the DBA to add new logical
#             logs when necessary.
```

```
#          0 Disabled
# LOGBUFF  - The size of the logical log buffer, in KB
```

Se puede ver la estructura en disco con el espacio asignado con el comando "onstat -d":

```
Dbspaces
address      number  flags    fchunk  nchunks  pgsize  flags  owner
name
70000019623d028 1      0x40001  1       1        4096   N BA  informix
rootdbs
700000198cd5e60 2      0x40001  2       1        4096   N BA  informix
physdbs

2 active, 2047 maximum

Chunks
address      chunk/dbs  offset  size    free    bpages  flags  pathname
70000019623d1d0 1      1      0       524288  502392  PO-B--
/chk_ifx/prod/rootdbs
700000198cd9d68 2      2      0       532480  20427   PO-B--
/chk_ifx/prod/physdbs

2 active, 32766 maximum
```

### 1.1.5 Creación de los LOGICAL LOGS

Al igual que con el PHYSICAL LOG, el LOGICAL LOG hay que sacarlo del rootdbspace para generar una arquitectura limpia y gestionar mejor la administración.

Por lo tanto, se crean dos espacios de almacenamiento que van a contener los logical logs, de esta forma nos aseguramos más disponibilidad ante una caída o fallo del disco, separando los lv en discos distintos:

```
acheron:informix:[prod_tcp]:/usr/local/informix>ls -l /chk_ifx/prod/log1dbs
lrwxrwxrwx  1 informix informix      13 Apr 06 2017 /chk_ifx/prod/log1dbs ->
/dev/rlog1dbs
acheron:informix:[prod_tcp]:/usr/local/informix>ls -l /chk_ifx/prod/log2dbs
lrwxrwxrwx  1 informix informix      13 Apr 06 2017 /chk_ifx/prod/log2dbs ->
/dev/rlog2dbs

acheron:informix:[prod_tcp]:/usr/local/informix>onspaces -c -d log1dbs -p
/chk_ifx/prod/log1dbs -o 0 -s 8396800
acheron:informix:[prod_tcp]:/usr/local/informix>onspaces -c -d log2dbs -p
/chk_ifx/prod/log2dbs -o 0 -s 8396800
```

Ahora se crean los logical logs que se estiman en 40 Mbytes cada uno, alcanzando la cifra de 408 entre los dos dbspaces.

En consecuencia, la instancia tiene espacio de transacciones suficientes para un *online* OLTP y además con un colchón prácticamente de 24 horas ante emergencias como no poder hacer *backup* de los logical logs ante una caída de TSM, etc.

Procedemos a crear los logical logs, intercambiándolos entre los dos espacios de almacenamiento y creando un bucle *while* hasta el llenado de los dbspaces:

```

acheron:informix:[prod_tcp]:/usr/local/informix>while true
do
    onparams -a -d log1dbs -s 40960 -y
    onparams -a -d log2dbs -s 40960 -y
done

```

Se pueden borrar los primeros que son necesarios para inicializar la instancia ya que todavía habitan en el rootdbspace:

```

acheron:informix:[prod_tcp]:/usr/local/informix>onparams -d -l 1 -y

```

Comprobamos el resultado con “onstat -l”:

```

acheron:informix:[prod_tcp]:/usr/local/informix>onstat -l|pg

IBM Informix Dynamic Server Version 11.70.FC8 -- On-Line (Prim) -- Up 5 days
12:27:21 -- 14642880 Kbytes

Physical Logging
Buffer bufused bufsize numpages numwrits pages/io
P-2 75 256 6265855 27503 227.82
  phybegin physize phypos phyused %used
  2:53 512000 267721 5078 0.99

Logical Logging
Buffer bufused bufsize numrecs numpages numwrits recs/pages pages/io
L-1 1 256 111913207 6391454 2120207 17.5 3.0
  Subsystem numrecs Log Space used
  OLDRSAM 111737160 18670588264
  SBLOB 130773 426781840
  HA 45274 2514496

address number flags uniqid begin size used %used
70000019623cfa8 1 U-B---- 339462 3:53 10240 10240
100.00
7000001967a0f50 2 U-B---- 339463 4:53 10240 10240
100.00
7000001967a0fb8 3 U-B---- 339464 3:10293 10240 10240
100.00
700000196786f50 4 U-B---- 339465 4:10293 10240 10240
100.00

```

700000196786fb8	5	U-B----	339466	3:20533	10240	10240
100.00						
70000019623d938	6	U-B----	339467	4:20533	10240	10240
100.00						
70000019623d9a0	7	U-B----	339468	3:30773	10240	10240
100.00						
70000019623da08	8	U-B----	339469	4:30773	10240	10240
100.00						
70000019623da70	9	U-B----	339470	3:41013	10240	10240
100.00						
70000019623dad8	10	U-B----	339471	4:41013	10240	10240
100.00						
70000019623db40	11	U-B----	339472	3:51253	10240	10240
100.00						
70000019623dba8	12	U-B----	339473	4:51253	10240	10240
100.00						
70000019623dc10	13	U-B----	339474	3:61493	10240	10240
100.00						
70000019623dc78	14	U-B----	339475	4:61493	10240	10240
100.00						
70000019623dce0	15	U-B----	339476	3:71733	10240	10240
100.00						
70000019623dd48	16	U-B----	339477	4:71733	10240	10240
100.00						
70000019623ddb0	17	U-B----	339478	3:81973	10240	10240
100.00						
70000019623de18	18	U-B----	339479	4:81973	10240	10240
100.00						
.						
.						
.						
700000198cd5c58	404	U-B----	339457	4:2058293	10240	6363
62.14						
700000198cd5cc0	405	U-B----	339458	3:2068533	10240	7813
76.30						
700000198cd5d28	406	U-B----	339459	4:2068533	10240	49
0.48						
700000198cd5d90	407	U-B----	339460	3:2078773	10240	10240
100.00						
700000198cd5df8	408	U-B----	339461	4:2078773	10240	10240
100.00						
408 active, 408 total						

Se puede ver la estructura en disco con el espacio asignado con el comando “onstat -d”:

```

Dbspaces
address      number  flags    fchunk  nchunks  pgsize  flags  owner
name
70000019623d028 1      0x40001  1       1        4096   N BA   informix
rootdbs
700000198cd5e60 2      0x40001  2       1        4096   N BA   informix
physdbs
700000198cd6d18 3      0x40001  3       1        4096   N BA   informix
log1dbs
700000198cd7028 4      0x40001  4       1        4096   N BA   informix
log2dbs

4 active, 2047 maximum

Chunks
address      chunk/dbs  offset  size    free    bpages  flags  pathname
70000019623d1d0 1      1      0      524288  502392  PO-B--
/chk_ifx/prod/rootdbs
700000198cd9d68 2      2      0      532480  20427   PO-B--
/chk_ifx/prod/physdbs
700000198cda028 3      3      0      2099200 10187   PO-B--
/chk_ifx/prod/log1dbs
700000198cda228 4      4      0      2099200 10187   PO-B--
/chk_ifx/prod/log2dbs

4 active, 32766 maximum

```

### 1.1.6 Creación de espacios temporales

Debido a las operaciones necesarias estimadas que ocuparán los espacios temporales por ejemplo, sentencias SQL con cláusulas “order by”), se estima un espacio temporal de 5 dbspaces de 5 Gbytes cada uno para atender a este tipo de peticiones y mejorar el rendimiento. Por lo tanto, se crea de la siguiente forma:

```

acheron:informix:[prod_tcp]:/usr/local/informix>onspaces -c -d temp1dbs -t -p
/chk_ifx/prod/temp1dbs -o 0 -s 5242880
acheron:informix:[prod_tcp]:/usr/local/informix>onspaces -c -d temp2dbs -t -p
/chk_ifx/prod/temp2dbs -o 0 -s 5242880
acheron:informix:[prod_tcp]:/usr/local/informix>onspaces -c -d temp3dbs -t -p
/chk_ifx/prod/temp3dbs -o 0 -s 5242880
acheron:informix:[prod_tcp]:/usr/local/informix>onspaces -c -d temp4dbs -t -p
/chk_ifx/prod/temp4dbs -o 0 -s 5242880
acheron:informix:[prod_tcp]:/usr/local/informix>onspaces -c -d temp5dbs -t -p
/chk_ifx/prod/temp5dbs -o 0 -s 5242880

```

Se puede ver la estructura en disco con el espacio asignado con el comando "onstat -d":

```

Dbspaces
address      number  flags    fchunk  nchunks  pgsize  flags  owner
name
70000019623d028 1      0x40001  1       1        4096   N BA   informix
rootdbs
700000198cd5e60 2      0x40001  2       1        4096   N BA   informix
physdbs
700000198cd6d18 3      0x40001  3       1        4096   N BA   informix
log1dbs
700000198cd7028 4      0x40001  4       1        4096   N BA   informix
log2dbs
700000198cd71d0 5      0x42001  5       1        4096   N TBA  informix
temp1dbs
700000198cd7378 6      0x42001  6       1        4096   N TBA  informix
temp2dbs
700000198cd7520 7      0x42001  7       1        4096   N TBA  informix
temp3dbs
700000198cd76c8 8      0x42001  8       1        4096   N TBA  informix
temp4dbs
700000198cd7870 9      0x42001  9       1        4096   N TBA  informix
temp5dbs

9 active, 2047 maximum

Chunks
address      chunk/dbs  offset  size    free    bpages  flags  pathname
70000019623d1d0 1          1      0      524288  502392  PO-B--
/chk_ifx/prod/rootdbs
700000198cd9d68 2          2      0      532480  20427   PO-B--
/chk_ifx/prod/physdbs
700000198cda028 3          3      0      2099200 10187   PO-B--
/chk_ifx/prod/log1dbs
700000198cda228 4          4      0      2099200 10187   PO-B--
/chk_ifx/prod/log2dbs
700000198cda428 5          5      0      1310720 1310317 PO-B--
/chk_ifx/prod/temp1dbs
700000198cda628 6          6      0      1310720 1310189 PO-B--
/chk_ifx/prod/temp2dbs
700000198cda828 7          7      0      1310720 1310317 PO-B--
/chk_ifx/prod/temp3dbs
700000198cdaa28 8          8      0      1310720 1310317 PO-B--
/chk_ifx/prod/temp4dbs
700000198cdac28 9          9      0      1310720 1310317 PO-B--
/chk_ifx/prod/temp5dbs

9 active, 32766 maximum

```

### 1.1.7 Creación del espacio de datos y de índices

Se crea un espacio de almacenamiento para contener todo el modelo de objetos de Global Service Desk.

Empezamos creando un espacio de almacenamiento propio de los tipos de datos *blob* o *clob*, ya que se guardará información relativa a documentos PDF, Word, Excel, etc.

Se establece el espacio en 40 Gbytes creando el dbspace blobsdbs, dejando el tamaño de página por defecto (ya que no existe información suficiente para hacer un estudio del tamaño de cada documento y fijar una media de tamaño de página):

```
acheron:informix:[prod_tcp]:/usr/local/informix>onspaces -c -S blobsdbs -p /chk_ifx/prod/blobsdbs -o 0 -s 41943040
```

Debido a que contendrá tanto datos propios del *online* como datos almacenados de históricos sobre los que se harán analítica de datos se crea el dbspace cycprod\_dat1 de 550 Gbytes:

```
acheron:informix:[prod_tcp]:/usr/local/informix>onspaces -c -d cycprod_dat1 -p /chk_ifx/prod/cycprod_dat -o 0 -s 576716800
```

Además, se establece la separación de datos e índices en dbspaces distintos con el fin de lograr un mejor rendimiento y una mejora de cara a la administración del gestor.

Con lo que se crea el dbspace de índices cycprod\_idx1 de 100Gbytes, espacio suficiente para almacenar índices como versión inicial.

```
acheron:informix:[prod_tcp]:/usr/local/informix>onspaces -c -d cycprod_idx1 -p /chk_ifx/prod/cycprod_idx1 -o 0 -s 104857600
```

Se puede ver la estructura en disco con el espacio asignado con el comando “onstat -d”:

```

Dbspaces
address      number  flags   fchunk  nchunks  pgsize  flags  owner
name
70000019623d028 1      0x40001  1       1        4096    N BA   informix
rootdbs
700000198cd5e60 2      0x40001  2       1        4096    N BA   informix
physdbs
700000198cd6d18 3      0x40001  3       1        4096    N BA   informix
log1dbs
700000198cd7028 4      0x40001  4       1        4096    N BA   informix
log2dbs
700000198cd71d0 5      0x42001  5       1        4096    N TBA  informix
temp1dbs
700000198cd7378 6      0x42001  6       1        4096    N TBA  informix
temp2dbs
700000198cd7520 7      0x42001  7       1        4096    N TBA  informix
temp3dbs
700000198cd76c8 8      0x42001  8       1        4096    N TBA  informix
temp4dbs
700000198cd7870 9      0x42001  9       1        4096    N TBA  informix
temp5dbs
700000198cd7a18 10     0x48001  10      1        4096    N SBA  informix
blobsdbs
700000198cd7bc0 11     0x40001  11      20       4096    N BA   informix
cycprod_dat1
700000198cd7d68 12     0x40001  12      7        4096    N BA   informix
cycprod_idx1

12 active, 2047 maximum

Chunks
address      chunk/dbs  offset  size   free   bpages  flags  pathname
70000019623d1d0 1      1      0      524288 502392  PO-B--
/chk_ifx/prod/rootdbs
700000198cd9d68 2      2      0      532480 20427  PO-B--
/chk_ifx/prod/physdbs
700000198cda028 3      3      0      2099200 10187  PO-B--
/chk_ifx/prod/log1dbs
700000198cda228 4      4      0      2099200 10187  PO-B--
/chk_ifx/prod/log2dbs
700000198cda428 5      5      0      1310720 1310317  PO-B--
/chk_ifx/prod/temp1dbs
700000198cda628 6      6      0      1310720 1310189  PO-B--
/chk_ifx/prod/temp2dbs

```



700000198cda828	7	7	0	1310720	1310317	PO-B--
/chk_ifx/prod/temp3dbs						
700000198cdaa28	8	8	0	1310720	1310317	PO-B--
/chk_ifx/prod/temp4dbs						
700000198cdac28	9	9	0	1310720	1310317	PO-B--
/chk_ifx/prod/temp5dbs						
700000198cdae28	10	10	0	10485760	3742853	9437131
POSB-- /chk_ifx/prod/lobsdbs						
Metadata	1048576	739498	1048576			
700000198cdb028	11	11	0	144179200	8046	PO-B--
/chk_ifx/prod/cycprod_dat1						
700000198cdb228	12	12	0	26214400	21453	PO-B--
/chk_ifx/prod/cycprod_idx1						
12 active, 32766 maximum						

### 1.1.8 Configurar el fichero ONCONFIG de la instancia

Se modifican los parámetros por defecto en función de los requerimientos, por lo tanto hay que modificar memoria, listas de bloqueos, conexiones máximas permitidas, etc.

#### 1.1.8.1 Límite de consumo de logical logs por transacción

Se modifica el límite del porcentaje de logical logs que una transacción puede consumir antes de hacer rollback, establecemos un umbral probado por la documentación de IBM:

```
#####
####
# LTXHWM      - The percentage of the logical logs that can be
#              filled before a transaction is determined to be a
#              long transaction and is rolled back
# LTXEHWM     - The percentage of the logical logs that have been
#              filled before the server suspends all other
#              transactions so that the long transaction being
#              rolled back has exclusive use of the logs
#
#####
####

LTXHWM 40
LTXEHWM 45
```

### 1.1.8.2 Configuración de logs del gestor

```
MSGPATH /usr/local/IDS/logs/online_prod.log
```

### 1.1.8.3 Especificación de espacios temporales y de datos especiales

Es necesario indicar al gestor cuáles son los espacios temporales y dónde se almacenan los datos especiales:

```
DBSPACETEMP  
temp1dbs,temp2dbs,temp3dbs,temp4dbs,temp5dbs,temp6dbs  
SBSPACETEMP blobsdbs
```

### 1.1.8.4 Pool de hebras por protocolo

Se definen el *pool* de conexiones máximas por protocolo:

```
# NETTYPE          - The configuration of poll threads  
#                  for a specific protocol. The  
#                  format is:  
#                  NETTYPE <protocol>,<# poll threads>  
#                  ,<number of connections/thread>  
#                  ,(NET|CPU)  
#                  You can include multiple NETTYPE  
#                  entries for multiple protocols.  
NETTYPE soctcp,2,400,NET
```

### 1.1.8.5 Procesadores virtuales

Se habilita el multiprocesador y se establecen 17 procesadores virtuales (se corresponden con los núcleos de los procesadores reales), con afinidad entre ellos para una mejor gestión de las hebras de los procesos.

```
MULTIPROCESSOR 1  
VPCLASS cpu,num=17,aff=(0-17),noage
```

### 1.1.8.6 Tabla de bloqueos

Se crea una tabla de bloqueos de 2.000.000, a partir de aquí se penalizará el rendimiento al adquirir dinámicamente bloques definidos de bloqueos con el consiguiente consumo de segmentos de memoria.

```
LOCKS 2000000
```

### 1.1.8.7 Nivel de bloqueo por tabla

Un aspecto fundamental para mejorar la concurrencia que se prevé, por lo que se define un nivel de bloqueo a nivel de registros y no de página (*page*) que empeora la concurrencia al bloquear todos los registros contenidos en una página:

```
DEF_TABLE_LOCKMODE row
```

### 1.1.8.8 Segmentos de memoria iniciales y dinámicos

Hay que definir la memoria inicial de la instancia (SHMVIRTSIZE), en este caso y en función de las tablas de bloqueos se estima en 8 Gbytes adquiriendo los segmentos necesarios (SHMADD) en 2 Gbytes, intentando siempre minimizar esa operación de adquisición de nuevos segmentos para mejorar el rendimiento.

```
SHMVIRTSIZE 8388608  
SHMADD 2097152
```

### 1.1.8.9 Intervalos de *checkpoints*

Lo importante aquí es que los puntos de sincronización de memoria con el disco duren lo menos posible (IBM dice entre 0 y 1 segundo) para que no ralentice el online:

```
CKPTINTVL 300
```

### 1.1.8.10 Ruta de *backups*

En este punto es importante definir una ruta distinta de /dev/null para que se guarde archivado de los backups y de los logical logs:

```
LTAPEDEV $INFORMIXDIR/tmp/ltape  
TAPEDEV $INFORMIXDIR/tmp/ltape
```

### 1.1.8.11 Rutas de los logs y volcados de error del gestor:

Se modifican las rutas de los *logs* del *online* y de los *backups* del gestor, que se realizaran con *onbar*.

También se le indica la ruta de volcado de memoria y generación de ficheros "af" de información de errores (DUMPDIR):

```
MSGPATH /usr/local/IDS/logs/online_prod.log  
BAR_ACT_LOG $INFORMIXDIR/logs/bar_act_prod.log  
BAR_DEBUG_LOG $INFORMIXDIR/logs/bar_debug_prod.log  
DUMPDIR /backups_bbdd/bkp_prod/prod_shm/dump
```

## 11. ANEXO C: Generación de *scripts* SQL del modelo

Creación de la estructura vacía y los permisos:

```
acheron:informix:[prod_tcp]:/usr/local/informix>
dbaccess -e sysmaster@prod_tcp PROTOTIPO_ESTRUCTURA.sql 2>&1 | tee
-a PROTOTIPO_ESTRUCTURA.log

egrep -i "error" PROTOTIPO_ESTRUCTURA.log

cat PROTOTIPO_ESTRUCTURA.log

Database selected.

-----
-----
-- VICTOR MANUEL MIÑAMBRES CHAMORRO.
-----
-----
-- TFG ----- GLOBAL SERVICE DESK -----
-----
-----
-- Se crea el esquema GSD (Global Service Desk.
-- Se revocan los permisos a public (por defecto) al crear el objeto en base de
datos.
-- Al final del script se dan los permisos a los usuarios de pruebas.
-- No se definen valores por defectos, ya que estos han de estar definidos por
el departamento de desarrollo encargado del aplicativo.
-- Se establecen como piloto 1000 registros al año para el cálculo de extents.
-- Se crea una clave primaria en cada tabla por el código identificador.
-- Se crean las foreing keys correctas que establecen la relación con las tablas
padre.
-----
-----

-- CREACION DE LA BASE DE DATOS GSD

drop database GSD;
Database dropped.

create database GSD in cycprod_dat1;
Database closed.
```

Database created.

--- CREACIÓN DEL MODELO DE DATOS

-- drop table Centers;

```
CREATE TABLE GSD.Centers
(
  centerCod VARCHAR(50),
  centerCountry VARCHAR(50),
  centerAddress VARCHAR(50),
  centerTypology VARCHAR(50)
```

```
) extent size 271 next size 56 lock mode row;
Table created.
```

```
revoke all on Centers from "public" as "informix";
Warning:Privilege not revoked.
```

```
create unique index ix_Centers on Centers (centerCod) using btree in
cycprod_idx1;
Index created.
```

```
alter table Centers add constraint primary key (centerCod) constraint
pk_Centers ;
Table altered.
```

-----  
-----  
-- drop table Services;

```
CREATE TABLE GSD.Services
(
  serviceCod VARCHAR(50),
  serviceDescription VARCHAR(50),
```

```
serviceGroup VARCHAR(50),
servicePriority VARCHAR(50),
serviceTimeRes VARCHAR(50)
```

```
) extent size 327 next size 68 lock mode row;
Table created.
```

```
revoke all on Services from "public" as "informix";
Warning:Privilege not revoked.
```

```
create unique index ix_Services on Services (serviceCod) using btree in
cycprod_idx1;
Index created.
```

```
alter table Services add constraint primary key (serviceCod) constraint
pk_Services ;
Table altered.
```

```
-----
-----
-- drop table Support;
```

```
CREATE TABLE GSD.Support
(
supportCod VARCHAR(50),
supportName VARCHAR(50),
supportType VARCHAR(50),
supportuserCod VARCHAR(50)
```

```
) extent size 271 next size 56 lock mode row;
Table created.
```

```
revoke all on Support from "public" as "informix";
Warning:Privilege not revoked.
```

```
create unique index ix_Support on Support (supportCod) using btree in
cycprod_idx1;
Index created.
```

```
alter table Support add constraint primary key (supportCod) constraint
pk_Support ;
Table altered.
```

```
-----
-----
-- drop table Users;
```

```
CREATE TABLE GSD.Users
(
  userCod VARCHAR(50),
  userReg DATETIME YEAR TO FRACTION(3),
  userUnsubscribe DATETIME YEAR TO FRACTION(3),
  userType VARCHAR(50),
  userSubscribe DATETIME YEAR TO FRACTION(3),
  usercenterCod VARCHAR(50),
  usersupportCod VARCHAR(50)
```

```
) extent size 406 next size 84 lock mode row;
Table created.
```

```
revoke all on Users from "public" as "informix";
Warning:Privilege not revoked.
```

```
create unique index ix_Users on Users (userCod) using btree in cycprod_idx1;
Index created.
```

```
alter table Users add constraint primary key (userCod) constraint pk_Users ;
Table altered.
```

```
ALTER TABLE Users ADD CONSTRAINT (FOREIGN KEY (usercenterCod)
REFERENCES Centers (centerCod) CONSTRAINT fk_Users_Centers);
Table altered.
```

```
ALTER TABLE Users ADD CONSTRAINT (FOREIGN KEY (usersupportCod)
REFERENCES Support (supportCod) CONSTRAINT fk_Users_Support);
```

Table altered.

-----  
-----  
  
-- drop table Tickets;

```
CREATE TABLE GSD.Tickets
(
  ticketNum VARCHAR(50),
  ticketType VARCHAR(50),
  ticketuserCod VARCHAR(50),
  ticketCreate DATETIME YEAR TO FRACTION(3),
  ticketPriority VARCHAR(50),
  ticketService VARCHAR(50),
  ticketChannel VARCHAR(50),
  ticketCycle VARCHAR(50),
  ticketDescription LVARCHAR(2000),
  ticketmessageCod VARCHAR(50),
  ticketDocument CLOB
```

```
) extent size 4371 next size 876 lock mode row;
Table created.
```

```
revoke all on Tickets from "public" as "informix";
Warning:Privilege not revoked.
```

```
create unique index ix_Tickets on Tickets (ticketNum) using btree in
cycprod_idx1;
Index created.
```

```
alter table Tickets add constraint primary key (ticketNum) constraint pk_Tickets
;
Table altered.
```

```
ALTER TABLE Tickets ADD CONSTRAINT (FOREIGN KEY (ticketService)
REFERENCES Services (serviceCod) CONSTRAINT fk_Tickets_Services);
Table altered.
```



```
ALTER TABLE Tickets ADD CONSTRAINT (FOREIGN KEY (ticketuserCod)
REFERENCES Users (userCod) CONSTRAINT fk_Tickets_Users);
Table altered.
```

```
-----
-----
-- drop table State;
```

```
CREATE TABLE GSD.State
(
  stateCodState serial,
  stateTicketNum VARCHAR(50),
  stateChangeDate DATETIME YEAR TO FRACTION(3),
  stateUserCod VARCHAR(50),
  stateNew VARCHAR(50)
```

```
) extent size 226 next size 48 lock mode row;
Table created.
```

```
revoke all on State from "public" as "informix";
Warning:Privilege not revoked.
```

```
create unique index ix_State on State (stateCodState) using btree in
cycprod_idx1;
Index created.
```

```
alter table State add constraint primary key (stateCodState) constraint pk_State
;
Table altered.
```

```
ALTER TABLE State ADD CONSTRAINT (FOREIGN KEY (stateTicketNum)
REFERENCES Tickets (ticketNum) CONSTRAINT fk_State_Tickets);
Table altered.
```

```
-----
-----
-- drop table Messages;
```

```

CREATE TABLE GSD.Messages
(
  messageCod VARCHAR(50),
  messageTime DATETIME YEAR TO FRACTION(3),
  messageReceiver VARCHAR(50),
  messageState VARCHAR(50),
  messageTicketNum VARCHAR(50)
) extent size 226 next size 48 lock mode row;
Table created.

revoke all on Messages from "public" as "informix";
Warning:Privilege not revoked.

create unique index ix_Messages on Messages (messageCod) using btree in
cycprod_idx1;
Index created.

alter table Messages add constraint primary key (messageCod) constraint
pk_Messages ;
Table altered.

ALTER TABLE Messages ADD CONSTRAINT (FOREIGN KEY
(messageTicketNum) REFERENCES Tickets (ticketNum) CONSTRAINT
fk_Messages_Tickets);
Table altered.

-----
--      Tabla de registro de los procedimientos ejecutados.
-----

-- drop table Log_procedures;

CREATE TABLE GSD.Log_procedures
(
  procCod serial,
  procName VARCHAR(100),
  procParamIn LVARCHAR(2000),

```

```
procParamOut LVARCHAR(2000),  
procReg DATETIME YEAR TO FRACTION(3)
```

```
) extent size 226 next size 48 lock mode row;  
Table created.
```

```
revoke all on Log_procedures from "public" as "informix";  
Warning:Privilege not revoked.
```

```
create unique index ix_Log_procedures on Log_procedures (procCod) using  
btree in cycprod_idx1;  
Index created.
```

```
alter table Log_procedures add constraint primary key (procCod) constraint  
pk_Log_procedures ;  
Table altered.
```

```
-----  
-----  
-----  
-----
```

```
-- PERMISOS NECESARIOS DEL PROTOTIPO.
```

```
-- PERMISOS DE CONEXION A LA BASE DE DATOS GSD
```

```
GRANT CONNECT TO USU1_GSD;  
Permission granted.
```

```
GRANT CONNECT TO USU2_GSD;  
Permission granted.
```

```
-- PERMISOS DE SELECT,INSERT,DELETE,UPDATE SOBRE EL MODELO  
AL USUARIO USU1_GSD;
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Tickets TO USU1_GSD AS  
GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Users TO USU1_GSD AS GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Centers TO USU1_GSD AS GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Services TO USU1_GSD AS GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON State TO USU1_GSD AS GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Messages TO USU1_GSD AS GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Support TO USU1_GSD AS GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Log_procedures TO USU1_GSD AS GSD;  
Permission granted.
```

```
-- PERMISOS DE SELECT,INSERT,DELETE,UPDATE SOBRE EL MODELO AL USUARIO USU2_GSD;
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Tickets TO USU2_GSD AS GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Users TO USU2_GSD AS GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Centers TO USU2_GSD AS  
GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Services TO USU2_GSD AS  
GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON State TO USU2_GSD AS  
GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Messages TO USU2_GSD  
AS GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Support TO USU2_GSD AS  
GSD;  
Permission granted.
```

```
GRANT SELECT,INSERT,DELETE,UPDATE ON Log_procedures TO  
USU2_GSD AS GSD;  
Permission granted.
```

-----  
-----

Database closed.

También, se puede comprobar que la base de datos está creada con el comando de Informix ONMONITOR:

Press ESC to return to the Status Menu.  
Use arrow keys to move the cursor.

### DATABASES

Database Name	Owner	When In Dbspace	Log Created	Status
sysutils	informix	rootdbs	04/28/2015	U
sysmaster	informix	rootdbs	04/28/2015	U
sysuser	informix	rootdbs	04/28/2015	U
gsd	informix	cycprod_dat1	06/06/2018	B
sysadmin	informix	rootdbs	04/28/2015	U

Total # of databases -> 6

Inserción de los datos de prueba iniciales del modelo:

```
dbaccess -e GSD@prod_tcp PROTOTIPO_DATOS.sql 2>&1| tee -a  
PROTOTIPO_DATOS.log
```

```
Database selected.
```

```
-----  
-----  
-- VICTOR MANUEL MIÑAMBRES CHAMORRO.  
-----  
-----
```

```
-----  
-- TFG ----- GLOBAL SERVICE DESK -----  
-----  
-----
```

```
-----  
-- Se insertan los datos para crear un prototipo de pruebas.  
-----  
-----  
-----
```

```
-----  
--delete from Support;  
--delete from Centers;  
--delete from Users;  
--delete from State;  
--delete from Messages;  
--delete from Services;  
--delete from Tickets;
```

```
-- Se insertan los grupos de soporte y los usuarios fuera de soporte (que son  
los clientes).
```

```
insert into Support values ('0','Usuario fuera de soporte','0','');  
1 row(s) inserted.
```

```
insert into Support values ('G00001','Soporte nivel  
1','L1','Soporte_Nivel1@corp_mail.com');  
1 row(s) inserted.
```

```
insert into Support values ('G00002','Soporte nivel  
2','L2','Soporte_Nivel2@corp_mail.com');  
1 row(s) inserted.
```

```
insert into Support values ('G00003','Soporte nivel  
3','L3','Soporte_Nivel3@corp_mail.com');  
1 row(s) inserted.
```

```
insert into Support values ('G00004','Soporte nivel 4 (Problem  
Management)','L4','Soporte_Problem_Management@corp_mail.com');
```

1 row(s) inserted.

-----  
-----  
-----  
-----

-- Se insertan los centros.

```
insert into Centers values ('SD00001','Turquía','52 Victory Street Izmir','Centro de Producción');
```

1 row(s) inserted.

```
insert into Centers values ('SD00002','España','Paseo de la Castellana, 6, 4ºA, 28046, Madrid','Centro de Logística');
```

1 row(s) inserted.

```
insert into Centers values ('SD00003','España','Doctor Blanco Soler, 4, 5ºB, 14004, Córdoba','Centro de Distribucción');
```

1 row(s) inserted.

-----  
-----  
-----  
-----

-- Se insertan los usuarios de pruebas.

```
insert into Users values ('User1@corp_mail.com',CURRENT,CURRENT,'Administrador',CURRENT,'SD00001','G00001');
```

1 row(s) inserted.

```
insert into Users values ('User2@corp_mail.com',CURRENT,CURRENT,'Cliente',CURRENT,'SD00003','0');
```

1 row(s) inserted.

```
insert into Users values ('User3@corp_mail.com',CURRENT,CURRENT,'Administrador',NULL,'SD00001','G00001');
```

1 row(s) inserted.



```
insert          into          Users          values
('User4@corp_mail.com',CURRENT,CURRENT,'Cliente',NULL,'SD00003','0');
1 row(s) inserted.
```

```
-----
-----
-----
-----
```

```
-- Se insertan los servicios disponibles.
insert into Services values ('SE0000242','Servicio de soporte al
usuario','Soporte Nivel 1','P4', '4 días');
1 row(s) inserted.
```

```
insert into Services values ('SE0056987','Servicio de soporte al nivel 2','Soporte
Nivel 2','P3', '2 días');
1 row(s) inserted.
```

```
insert into Services values ('SE0056423','Servicio de soporte al nivel 3','Soporte
Nivel 3','P2', '1 día');
1 row(s) inserted.
```

```
insert into Services values ('SE0000678','Servicio de soporte externo','Soporte
Problem Management','P1', '1 hora');
1 row(s) inserted.
```

```
-----
-----
-----
-----
```

```
-- Se insertan los tickets de prueba.
insert into Tickets values ('T000000001', 'INC', 'User1@corp_mail.com',
CURRENT, 'P4' , 'SE0000242','WEB', 'T000000001', 'Solicitar permisos de
acceso a la aplicación GSD del tipo consulta para los usuarios del
departamento de Riesgos.', 'MS0000456',NULL);
1 row(s) inserted.
```

```
insert into Tickets values ('T000000002', 'PET', 'User2@corp_mail.com',
CURRENT, 'P3' , 'SE0056987','CHAT', 'T000000002', 'Cambio de contraseña
para todos los usuarios del departamento de Marketing', 'MS0000598',NULL);
1 row(s) inserted.
```

```
insert into Tickets values ('T000000003', 'INC', 'User1@corp_mail.com',
CURRENT, 'P2' , 'SE0056423','TLF', 'T000000003', 'Problemas al acceder a la
tabla TI de la base de datos con código de error -143', 'MS0000022',NULL);
1 row(s) inserted.
```

```
insert into Tickets values ('T000000004', 'INC', 'User2@corp_mail.com',
CURRENT, 'P1' , 'SE0000678','WEB', 'T000000004', 'Caida de aplicación',
'MS0000898',NULL);
1 row(s) inserted.
```

```
-----
-----
-----
-----
```

-- Se insertan los estados por los que puede pasar un Ticket.

```
insert          into          State          values
(0,'T000000001',CURRENT,'User1@corp_mail.com','Creado');
1 row(s) inserted.
```

```
insert          into          State          values
(0,'T000000001',CURRENT,'User1@corp_mail.com','Asignado');
1 row(s) inserted.
```

```
insert into State values (0,'T000000001',CURRENT,'User1@corp_mail.com','En
progreso');
1 row(s) inserted.
```

```
insert          into          State          values
(0,'T000000001',CURRENT,'User1@corp_mail.com','Cancelado');
1 row(s) inserted.
```

```
insert          into          State          values
(0,'T000000001',CURRENT,'User1@corp_mail.com','Resuelto');
1 row(s) inserted.
```

```
insert          into          State          values
(0,'T000000002',CURRENT,'User2@corp_mail.com','Creado');
1 row(s) inserted.
```

```
insert          into          State          values
(0,'T000000002',CURRENT,'User2@corp_mail.com','Asignado');
1 row(s) inserted.
```

```
insert into State values (0,'T000000002',CURRENT,'User2@corp_mail.com','En
progreso');
1 row(s) inserted.
```

```
insert          into          State          values
(0,'T000000002',CURRENT,'User2@corp_mail.com','Resuelto');
1 row(s) inserted.
```

```
insert          into          State          values
(0,'T000000002',CURRENT,'User2@corp_mail.com','Cancelado');
1 row(s) inserted.
```

```
insert          into          State          values
(0,'T000000003',CURRENT,'User1@corp_mail.com','Creado');
1 row(s) inserted.
```

```
insert          into          State          values
(0,'T000000004',CURRENT,'User2@corp_mail.com','Creado');
1 row(s) inserted.
```

```
insert          into          State          values
(0,'T000000004',CURRENT,'User2@corp_mail.com','Asignado');
1 row(s) inserted.
```

```
insert into State values (0,'T000000004',CURRENT,'User2@corp_mail.com','En
progreso');
1 row(s) inserted.
```

```
-----
-----
-----
-----
```

```
-- Se insertan los mensajes de un Ticket.
```

```
insert          into          Messages          values
('MS0000456',CURRENT,'User1 @corp_mail.com','Recibido','T000000001');
1 row(s) inserted.
```

```
insert          into          Messages          values
('MS0000457',CURRENT,'User1 @corp_mail.com','Recibido','T000000001');
1 row(s) inserted.
```

```
insert          into          Messages          values
('MS0000458',CURRENT,'User1 @corp_mail.com','Enviado','T000000001');
1 row(s) inserted.
```

```
insert          into          Messages          values
('MS0000459',CURRENT,'User1 @corp_mail.com','Enviado','T000000001');
1 row(s) inserted.
```

```
insert          into          Messages          values
('MS0000598',CURRENT,'User2 @corp_mail.com','Enviado','T000000002');
1 row(s) inserted.
```

```
insert          into          Messages          values
('MS00003216',CURRENT,'Soporte_Nivel1 @corp_mail.com','Enviado','T00000
0003');
1 row(s) inserted.
```

```
insert          into          Messages          values
('MS0000022',CURRENT,'Soporte_Nivel2 @corp_mail.com','Recibido','T000000
003');
1 row(s) inserted.
```

```
insert          into          Messages          values
('MS0000898',CURRENT,'Soporte_Nivel3 @corp_mail.com','Recibido','T000000
004');
1 row(s) inserted.
```

```
-----
-----
-----
-----
```

Database closed.

Se pueden ver las tablas creadas en Informix en la base de datos GSD:

```
INFO FOR TABLE >>
Choose a table with the Arrow Keys, or enter a name, then press Return.

----- gsd@prod_tcp ----- Press CTRL-W for Help -----

'gsd'.centers
'gsd'.log_procedu+
'gsd'.messages
'gsd'.services
'gsd'.state
```

```
'gsd'.support
```

```
'gsd'.tickets
```

```
'gsd'.timesolinc
```

```
'gsd'.users
```

### Creación de los procedimientos almacenados de gestión de usuarios

```
dbaccess -e GSD@prod_tcp PROTOTIPO_PROCEDURES_USUARIO.sql 2>&1| tee -a PROTOTIPO_PROCEDURES_USUARIO.log
```

```
egrep -i "Routine created." PROTOTIPO_PROCEDURES_USUARIO.log|wc -l  
4
```

### Creación de los procedimientos almacenados de gestión de *tickets*

```
dbaccess -e GSD@prod_tcp PROTOTIPO_PROCEDURES_TICKET.sql 2>&1| tee -a PROTOTIPO_PROCEDURES_TICKET.log
```

```
egrep -i "Routine created." PROTOTIPO_PROCEDURES_TICKET.log|wc -l  
4
```

### Creación de los procedimientos almacenados de gestión de estados de los *tickets*

```
dbaccess -e GSD@prod_tcp  
PROTOTIPO_PROCEDURES_ESTADOS_TICKET.sql 2>&1| tee -a  
PROTOTIPO_PROCEDURES_ESTADOS_TICKET.log
```

```
egrep -i "Routine created."  
PROTOTIPO_PROCEDURES_ESTADOS_TICKET.log|wc -l  
4
```

### Creación de los procedimientos almacenados de gestión de mensajes de los *tickets*

```
dbaccess -e GSD@prod_tcp  
PROTOTIPO_PROCEDURES_MENSAJES_TICKET.sql 2>&1| tee -a  
PROTOTIPO_PROCEDURES_MENSAJES_TICKET.log
```

```
egrep -i "Routine created."
PROTOTIPO_PROCEDURES_MENSAJES_TICKET.log|wc -l
4
```

También, se comprueba con una SELECT al catálogo de Informix que los procedimientos están creados:

```
Database selected.

select procname from sysprocedures where owner = 'GSD'

procname consulta_usuario
procname alta_usuario
procname baja_usuario
procname actualiza_usuario
procname consulta_ticket
procname alta_ticket
procname baja_ticket
procname actualiza_ticket
procname consulta_estado_ticket
procname alta_estado_ticket
procname baja_estado_ticket
procname actualiza_estado_tickets
procname consulta_mensajes_ticket
procname alta_mensajes_ticket
procname actualiza_mensajes_tickets
procname timesolinc

16 row(s) retrieved.
```

Database closed.

## 12. ANEXO D: Instalación y configuración de HDR (HA)

Desde la máquina secundaria donde va a residir la réplica HDR (se ha configurado como repl\_tcp) y apuntando a la máquina primaria (se ha configurado como prod\_tcp).

Se utiliza el comando facilitado por IBM ontape (para realizar copias de seguridad) con las opciones STDIO para que los datos se envíen sin almacenarlos en un disco o cinta.

De la siguiente forma, se realiza un backup en el nodo primario que se restaura inmediatamente en el nodo secundario:

### 1. Configuración del nodo Secundario

El primer paso es realizar la configuración del nodo secundario.

Por lo tanto, se configura el fichero de variables del nodo secundario:

```
elcano:informix:[]:/usr/local/informix>. ./variab_repl
elcano:informix:[repl_tcp]:/usr/local/informix>cat variab_repl
export INFORMIXDIR=/usr/local/IDS
export INFORMIXSQLHOSTS=$INFORMIXDIR/etc/sqlhosts.repl
export INFORMIXSERVER=repl_tcp
export ONCONFIG=onconfig.repl
export DBDATE=y4md0
export TERMCAP=$INFORMIXDIR/etc/termcap
export INFORMIXTERM=terminfo
export PATH=$INFORMIXDIR/bin:$PATH
export
LIBPATH=/usr/lib:$COBDIR/lib:$INFORMIXDIR/lib:$INFORMIXDIR/lib/esql:$INFORMIXDIR/lib/tools:$LIBPATH
```

Se configuran los datos de conexión al nodo primario en el fichero de conexiones de informix SQLHOSTS:

```
elcano:informix:[repl_tcp]:/usr/local/informix>cat $INFORMIXSQLHOSTS
#*****
#
# Licensed Material - Property Of IBM
```



```

#
# "Restricted Materials of IBM"
#
# IBM Informix Dynamic Server
# (c) Copyright IBM Corporation 1996, 2004 All rights reserved.
#
# Title:   sqlhosts.demo
# Description:
#         Default sqlhosts file for running demos.
#
#*****
prod_tcp      onsoctcp     acheron      1533
repl_tcp      onsoctcp     elcano       1545
desa_tcp      onsoctcp     cucutras     1545

```

## 2. Lanzamiento de Backup/Restore

Lanzamos el comando `ontape` con las opciones señaladas desde el nodo secundario (`repl_tcp`), los datos viajarán mediante SSH:

```

elcano:informix:[repl_tcp]:/usr/local/informix>ssh acheron ".
/usr/local/informix/variab_prod; ontape -s -L 0 -t STDIO" | ontape -p -t STDIO

```

## 3. Establecimiento de nodos primario y secundario

Desde el nodo primario se le indica qué nodo es el secundario (previamente ya se ha configurado en el `SQLHOSTS` en el ANEXO A):

```

acheron:informix:[prod_tcp]:/usr/local/informix>onmode -d primary repl_tcp

```

En el nodo secundario hay que indicar cuál es el nodo primario:

```

elcano:informix:[repl_tcp]:/usr/local/informix>onmode -d secondary prod_tcp

```

#### 4. Comprobamos el estado correcto de la replicación

Una vez establecidos los pasos anteriores se comprueba el estado correcto de la replicación tanto en el nodo primario como en el secundario con la opción especial "onstat -g dri".

Desde el nodo primario:

```
IBM Informix Dynamic Server Version 11.70.FC8 -- On-Line (Prim) -- Up 4 days
16:26:24 -- 14642880 Kbytes
```

```
Data Replication at 7000001988e5028:
```

Type	State	Paired server	Last DR CKPT (id/pg)	Supports
Proxy Writes				
primary	on	repl_tcp	340985 / 1346	NA

```
DRINTERVAL 10
DRTIMEOUT 30
DRAUTO 0
DRLOSTFOUND /usr/local/IDS/etc/dr.lostfound
DRIDXAUTO 1
ENCRYPT_HDR 0
Backlog 2
Last Send 2018/04/02 17:56:05
Last Receive 2018/04/02 17:56:05
Last Ping 2018/04/02 17:56:03
Last log page applied(log id,page): 340985,4103
```

Desde el nodo secundario:

```
elcano:informix:[repl_tcp]:/usr/local/informix>onstat -g dri
```

```
IBM Informix Dynamic Server Version 11.70.FC8 -- Read-Only (Sec) -- Up 4
days 16:20:01 -- 11870320 Kbytes
```

```
Data Replication at 7000000f85aa028:
```

Type	State	Paired server	Last DR CKPT (id/pg)	Supports
Proxy Writes				
HDR Secondary	on	prod_tcp	340985 / 1346	N

```
DRINTERVAL 10
DRTIMEOUT 30
DRAUTO 0
DRLOSTFOUND /usr/local/IDS/etc/dr.lostfound
DRIDXAUTO 1
ENCRYPT_HDR 0
Backlog 0
Last Send 2018/04/02 17:53:30
Last Receive 2018/04/02 17:53:30
Last Ping 2018/04/02 17:53:02
Last log page applied(log id,page): 0,0
```

También se observa en el fichero de log del gestor (online.log) en ambos nodos que la réplica se ha establecido correctamente, enviando la información contenida en los logs.

Desde el nodo primario:

```
acheron:informix:[prod_tcp]:/usr/local/informix>onstat -m
```

```
01:35:08 DR: Sending log 5463 (current), size 10240 pages, 0.17 percent used
01:35:09 DR: Sending Logical Logs Completed
01:35:10 DR: Primary server operational
```

Desde el nodo secundario:

```
elcano:informix:[repl_tcp]:/usr/local/informix> onstat -m
```

```
01:33:43 Starting Log Position - 340697 0x2f02b8
01:33:46 Started processing open transactions on secondary during startup
01:33:46 Finished processing open transactions on secondary during startup.
01:33:46 Checkpoint Completed: duration was 0 seconds.
01:33:46 Thu Mar 29 - loguniq 340697, logpos 0x2f20c0, timestamp:
0x606d47f7 Interval: 784874

01:33:46 Maximum server connections 0
01:33:46 Checkpoint Statistics - Avg. Txn Block Time 0.000, # Txns blocked 0,
Plog used 6, Llog used 0

01:33:48 B-tree scanners disabled.
01:33:48 DR: HDR secondary server operational
```

## 13. ANEXO E: Instalación y configuración de auditoría

La herramienta de auditoría (así como la de HA) vienen con la licencia del gestor, por lo que ya está instalada.

En consecuencia, se debe configurar y arrancar para que genere los registros de las instrucciones a auditar.

Con el comando “onaudit” se gestiona la auditoría de Informix, en concreto para crear una auditoría (se llaman máscaras) para todos los usuarios y que se guarde registro de todas las operaciones;DELETE,INSERT,UPDATE, se realiza mediante:

```
acheron:informix:[prod_tcp]:/usr/local/informix>onaudit -a -u _default -e DLRW
```

Para activar la auditoría:

```
acheron:informix:[prod_tcp]:/usr/local/informix>onaudit -l 1
```

Las instrucciones SELECT no se monitorizan debido a que todo esto repercute en el rendimiento del sistema, de ser necesario para algún usuario o grupo en concreto se puede añadir con el mismo comando anterior.

Por lo tanto, se crea un *script* en Unix para ejecutar la configuración y arrancar la auditoría de una forma sencilla y con las siguientes opciones importantes:

```
acheron:informix:[prod_tcp]:/usr/local/informix>cat
/usr/local/scripts/BBDD/informix/SCRIPTS_DBA/arranca_auditoria_prod_tcp.ks
h

#!/bin/ksh
# Script para arrancar la auditoria de la instancia prod_tcp (nodo primario).
Llamar el script después de arrancar la instancia.
#
/usr/local/scripts/BBDD/informix/SCRIPTS_DBA/arranca_auditoria_prod_tcp.ks
h

export
LOG=/usr/local/scripts/BBDD/informix/SCRIPTS_DBA/LOGS/arranca_auditoria
_prod_tcp.log

./usr/local/informix/variab_prod
```

```
echo "\n\n`date +%D %H:%M:%S`\n\n" >> $LOG 2>&1
```

```
#Añadimos máscara a los usuarios.
```

```
onaudit -a -u _default -e DLRW >> $LOG 2>&1
```

```
#Añadimos máscara a Informix.
```

```
onaudit -a -u informix -e DNCK >> $LOG 2>&1
```

```
#Activamos auditoría a 1.
```

```
onaudit -l 1 >> $LOG 2>&1
```

```
#Definimos un path para los logs.
```

```
onaudit -p /backups_bbdd/prod_tcp/audit >> $LOG 2>&1
```

```
#El servidor procesa el hilo y señala el error en el registro de mensajes.
```

```
onaudit -e 0 >> $LOG 2>&1
```

```
#Configuramos el fichero de auditoría al máximo permitido 2 Gbytes.
```

```
onaudit -s 2000000000 >> $LOG 2>&1
```

Con la siguiente opción “onaudit -c” se puede comprobar que la auditoría está funcionando y las opciones que se han configurado (ADTMODE = 1, indica que la auditoría está arrancada):

```
Onaudit -- Audit Subsystem Configuration Utility
```

```
Current audit system configuration:
```

```
ADTMODE = 1
```

```
ADTERR = 0
```

```
ADTPATH = /backups_bbdd/prod_tcp/audit
```

```
ADTSIZE = 2000000000
```

```
Audit file = 75
```

```
ADTROWS = 0
```

Se puede observar en la ruta definida “/backups\_bbdd/prod\_tcp/audit” que los ficheros de auditoría se generan en cuanto se produce cualquier evento definido a monitorizar:

```
acheron:informix:[]:/usr/local/informix>cd /backups_bbdd/prod_tcp/audit
```

```
acheron:informix:[]:/backups_bbdd/prod_tcp/audit>ls -ltr
```

```
-rw-rw---- 1 informix informix 1999999956 Apr 03 01:23 prod_shm.74
```

```
-rw-rw---- 1 informix informix 206118819 Apr 03 13:49 prod_shm.75
```

# 14. ANEXO F: Comandos de monitorización de Informix

Visualización del estado ON-LINE del gestor:

```
acheron:informix:[prod_tcp]:/usr/local/informix>onstat -  
IBM Informix Dynamic Server Version 11.70.FC8 -- On-Line (Prim) -- Up  
10:41:15 -- 14642880 Kbytes
```

Ver fichero de logs del gestor (online.log):

```
acheron:informix:[prod_tcp]:/usr/local/informix>onstat -m  
IBM Informix Dynamic Server Version 11.70.FC8 -- On-Line (Prim) -- Up  
10:43:04 -- 14642880 Kbytes  
  
Message Log File: /usr/local/IDS/logs/online_prod.log  
11:59:53 Checkpoint Statistics - Avg. Txn Block Time 0.000, # Txns blocked 1,  
Plog used 9235, Llog used 5939  
  
12:01:12 Logical Log 341390 Complete, timestamp: 0x89232aa0.  
12:01:13 Logical Log 341390 - Backup Started  
12:01:15 Logical Log 341390 - Backup Completed  
12:04:54 Checkpoint Completed: duration was 1 seconds.  
12:04:54 Thu Apr 5 - loguniq 341391, logpos 0x11694f0, timestamp:  
0x892cf9d3 Interval: 787463
```

Información de sesiones conectadas a las bases de datos (identificador de la hebra de sesión, número de bloqueos, número de lecturas, escrituras, etc.):

```
acheron:informix:[prod_tcp]:/usr/local/informix>onstat -u  
IBM Informix Dynamic Server Version 11.70.FC8 -- On-Line (Prim) -- Up  
10:50:15 -- 14642880 Kbytes  
  
Userthreads  
address      flags  sessid  user  tty  wait      tout locks nreads  
nwrites  
700000196745028 ---P--D 1      informix -    0      0  0  43238  
68143  
700000196745860 ---P--F 0      informix -    0      0  0  41700  
700000196746098 ---P--F 0      informix -    0      0  0  75989  
7000001967468d0 ---P--F 0      informix -    0      0  0  53181  
700000196747108 ---P--F 0      informix -    0      0  0  81991  
700000196747940 ---P--F 0      informix -    0      0  0  72817  
700000196748178 ---P--F 0      informix -    0      0  0  78850
```

```

7000001967489b0 ---P--F 0      informix -      0      0 0 0
202462
7000001967491e8 ---P--F 0      informix -      0      0 0 0
226581

```

Información de sesiones conectadas a las bases de datos (máquina origen, usuario, memoria utilizada, etc.):

```

acheron:informix:[prod_tcp]:/usr/local/informix>onstat -g ses

IBM Informix Dynamic Server Version 11.70.FC8 -- On-Line (Prim) -- Up
10:44:51 -- 14642880 Kbytes

session
id  user  tty  pid  hostname threads  total  used  dynamic  explain
23297 informix - 0 - 0 16384 12840 off
246  informix - 11600016 elcano 1 57344 45424 off
241  informix - 0 - 1 49152 41792 off
240  informix - 0 - 1 49152 42736 off
76   informix - 0 - 1 475136 361608 off
75   informix - 0 - 1 487424 363656 off
74   informix - 0 - 1 577536 452976 off
8    informix - 0 - 0 16384 12840 off
6    informix - 0 - 0 16384 14432 off
5    informix - 0 - 0 16384 14432 off

```

Información de sesiones conectadas a las bases de datos (tipo de instrucciones que se están lanzando):

```

acheron:informix:[prod_tcp]:/usr/local/informix>onstat -g sql

IBM Informix Dynamic Server Version 11.70.FC8 -- On-Line (Prim) -- Up
10:47:32 -- 14642880 Kbytes

Sess  SQL      Current      Iso Lock      SQL ISAM F.E.
Id    Stmt type Database      Lvl Mode      ERR ERR Vers Explain
23386 SELECT   sysmaster    DR Not Wait 0 0 9.24 Off
23381 SELECT   cycprod     CR Wait 2 0 0 9.28 Off
23380 -        cycprod     CR Wait 2 0 0 9.28 Off
23358 -        cycprod     CR Wait 2 0 0 9.28 Off
23357 SELECT   cycprod     CR Wait 2 0 0 9.28 Off
23356 SELECT   cycprod     CR Wait 2 0 0 9.28 Off

```

Comprobación del estado de los *backups* (Time Stamp de la fecha de realización, etc):

```

acheron:informix:[prod_tcp]:/usr/local/informix>onstat -g arc

IBM Informix Dynamic Server Version 11.70.FC8 -- On-Line (Prim) -- Up
10:52:42 -- 14642880 Kbytes

num  DBSpace          Q Size Q Len Buffer partnum  size  scanner

Dbspaces - Archive Status
name      number level date      log      log-position
rootdbs   1    0   04/05/2018.04:30 341363   0xe018
physdbs   2    0   04/05/2018.05:21 341363   0x64a018
log1dbs   3    0   04/05/2018.05:21 341363   0x616018
log2dbs   4    0   04/05/2018.05:21 341363   0x631018
blobsdbs  10   0   04/05/2018.05:13 341363   0x49a018
cycprod_dat1  11  0   04/05/2018.04:30 341363   0x29018
cycprod_idx1  12  0   04/05/2018.04:30 341363   0x29018

```

Estado general de la memoria del gestor (número de *deadlocks*, búsquedas secuenciales sin índice, espera por bloqueos, etc):

```

acheron:informix:[prod_tcp]:/usr/local/informix>onstat -p

IBM Informix Dynamic Server Version 11.70.FC8 -- On-Line (Prim) -- Up
10:58:32 -- 14642880 Kbytes

Profile
dskreads  pagreads  bufreads  %cached dskwrits  pagwrits  bufwrits
%cached

367516686  635542658  15684570791  97.66  1947713  4869738
1331019283 99.90

isamtot  open    start  read  write  rewrite  delete  commit  rollbk
8240624048 163698958 176354268 5820035366 621784651 351694
659488 156029 2319

gp_read  gp_write  gp_rewrt  gp_del  gp_alloc  gp_free  gp_curs
5627    10    1690    4    0    0    2

ovlock  ovuserthread  ovbuff  usercpu  syscpu  numckpts  flushes
0    0

ixda-RA  idx-RA  da-RA  logrec-RA  RA-pgsused  lchwaits
0    147947010 257590308 2    274215290 13666065

```

Comprobar las hebras de sesión que mantienen objetos abiertos:



```

acheron:informix:[prod_tcp]:/usr/local/informix>onstat -g opn

IBM Informix Dynamic Server Version 11.70.FC8 -- On-Line (Prim) -- Up 205
days 23:08:36 -- 8875712 Kbytes
tid rstcb          isfd op_mode  op_flags  partnum  ucount ocount lockmode
29 0x070000019254cb70 79 0x00060000 0x00000403 0x00100012 1 0
0 0

32049220 0x070000019254ec50 0 0x00000400 0x00000317 0x00b0059b 2
2 0 1
32049220 0x070000019254ec50 1 0x00000002 0x00000003 0x00b0059b 2
2 0 0

68 0x070000019254fcc0 0 0x00000400 0x00000297 0x001001a6 2 2
0 61
68 0x070000019254fcc0 1 0x00000002 0x00000003 0x001001a6 2 2
0 0

```

**70000019254cb70** → corresponde a la hebra de sesión que mantiene el recurso bloqueado (en este caso es una sesión de Informix):

```

acheron:informix:[prod_tcp]:/usr/local/informix>onstat -u|grep -i
70000019254cb70
70000019254cb70 Y--P--D 8 informix - 7000001949d3420 0 0
133530 0

```

Estado general de la replicación (*online*, último *logical log* enviado, etc.):

```

acheron:informix:[prod_tcp]:/usr/local/informix>onstat -g dri

IBM Informix Dynamic Server Version 11.70.FC8 -- On-Line (Prim) -- Up
11:00:23 -- 14642880 Kbytes

Data Replication at 7000001988e5028:
Type      State      Paired server      Last DR CKPT (id/pg)  Supports
Proxy Writes
primary   on         repl_tcp           341394 / 4897        NA

DRINTERVAL 10
DRTIMEOUT  30
DRAUTO     0
DRLOSTFOUND /usr/local/IDS/etc/dr.lostfound
DRIDXAUTO  1
ENCRYPT_HDR 0
Backlog    159
Last Send  2018/04/05 12:31:07

```

Last Receive 2018/04/05 12:31:07  
Last Ping 2018/04/05 12:30:54  
Last log page applied(log id,page): 341394,6238

Ayuda de los comandos de monitorización:

```
acheron:informix:[prod_tcp]:/usr/local/informix>onstat -?
usage: onstat [ -abBcCdDfFgGhjkLLmOpPRstTuxXz ] [ -i ] [ -r [<seconds>] ]
      [ -o [<outfile>] ] [ <infile> ]

-- Print this help text
<infile>
    Read shared memory information from specified dump file
-a Interpreted as onstat -mcuxskbPFhRtdGfLLpO; onstat -g all; onstat -XC
-b Print buffers
-B Print all buffers
-c Print configuration file
-C Print b-tree scanner requests
prof Print profile information for the system and scanner threads
hot Print hot list index keys
part Print all partitions with index statistics
clean Print information about all partitions cleaned and need to be
range Print savings in pages processed with range scanning
map Print current alicec bitmap for all indexes being cleaned
alice Print efficiency of alicec cleaning method
all Print all onstat -C options
-d [update]
    Print spaces and chunks
    update - Ask server to update BLOB chunk statistics
-D Print spaces and detailed chunk stats
-f Print dataskip status
-F Print page flushers

-g <cmd> MT COMMAND or ENTERPRISE REPLICATION COMMAND
(see below)

MT COMMANDS:
act Print active threads
afr <pool name|session id>
    Print allocated pool fragments
all Print all MT information
ath Print all threads
bfr <blk pool address>
    Print allocated block pool blocks for <blk pool address>
buf Print profile information related to buffer pools.
ckp Print checkpoint statistics
csm Print Connection Manager statistics
con Print conditions with waiters
cpu Print CPU info for all threads
dbc Print dbScheduler/dbWorker thread info
```

ddr Print DDR log post processing information  
 defragment  
     Print the status of defragmentation commands given  
 dic Print dictionary cache information  
 dis Print a list of database servers and the status of each  
 dll Print dynamic library statistics  
 dmp <address> <length>  
     Dump <length> bytes of shared memory starting at <address>  
 dri [ sta | msg | ckpt | idx ]  
     Print data replication information  
 dsc Print a list of distribution cache information  
 dsk Storage manager information  
 env [ all | [<session-id>] ] [<variable-name>[,<variable-name>...]]  
     Display environment variable settings.  
 ffr <pool name|session id>  
     Print free pool fragments  
 glo Print MT global information  
 his [<ntraces>]  
     Prints SQL statement tracing information for <ntraces>  
         no <ntraces> = Complete output from trace buffer  
 imc Print information about connected MaxConnect instances  
 iob Print big buffer usage by IO VP class  
 iof Print disk IO statistics by chunk/file  
 iog Print AIO global information  
 iov Print disk IO statistics by vp  
 ipl Print index page logging status  
 lap Print light append information  
 Imm Print Low Memory Manager information  
 lmx Print all locked mutexes  
 lsc Print Light Scan information  
 mem [<pool name>|<session id>]  
     Print pool statistics.  
 mgm Print Memory Grant Manager information  
 nbm Print block map for non-resident segments  
 nsc [<client id>]  
     Print net shared memory status  
 nsd Print net shared memory data  
 nss [<session id>]  
     Print net shared memory status  
 ntd Print net dispatch information  
 ntm Print net message information  
 ntt Print net user thread access times  
 ntu Print net user thread profile information  
 opn [<tid>]  
     Print open tables  
 plk Print partition lock profiles  
 pos Print /INFORMIXDIR/etc/.infos.DBSERVERNAME file  
 ppf [<partition number> | 0]  
     Print partition profiles  
 ppd [<partition number> | 0]

Print partition compression dictionary information  
 pqs [<session id>]  
 Print statistics for an active query  
 prc Print information about SPL routine cache  
 cluster [verbose]  
 Print cluster information      laq Print recovery queue statistics  
 probe Print query probing data (workload analysis)  
 proxy [all | [<proxy id> [<txn id> [<op num>]]] ]  
 Print updatable secondary related information  
 qst Print queue statistics  
 rbm Print block map for resident segment  
 rea Print ready threads  
 rss [verbose | log | <RSS Srv name>] Print RSS server related  
 information  
 rwm print Read/Write Mutex lists  
 sch Print VP scheduler statistics  
 scn RSAM scan info  
 sds [verbose | <SDS server name>] Print SDS related information  
 seg Print memory segment statistics  
 ses [<session id>]  
 Print session information  
 sle Print all sleeping threads  
 smb Print smart-large-object usage  
 smx [ses] Print smx related information  
 spi Print spin locks with long spins  
 sql [<session id>]  
 Print SQL information  
 sqh [<sql heap address>]  
 Print sql heap for <sql heap address>  
     no <sql heap address> = summary for all sql heaps  
 src <pattern> <mask>  
 Search memory for <pattern>, where <pattern>==(memory&<mask>)  
 ssc [pool|all]  
 Prints ssc pool summary, or statement cache summary and  
 entries, including key only entries (all)  
 stk <tid>  
 Dump the stack of a specified thread  
 stm [<session id>]  
 Prints all prepared statements approximate memory usage in a  
 session  
 stq [<session id>]  
 Print stream queue information  
 sts Print max and current stack sizes  
 tgp Print generic page thread profiles  
 tpf [<tid> | 0]  
 Print thread profiles  
 ufr <pool name|session id>  
 Print pool usage breakdown  
 vpcache Print CPU VP memory block cache statistics  
 wai Print waiting threads

wmx Print all mutexes with waiters  
wst Print thread wait statistics

#### ENTERPRISE REPLICATION COMMANDS:

cat [scope | replname ]  
Print Enterprise Replication global catalog information  
cdr Print Enterprise Replication statistics  
cdr config [parameter\_name] [long]  
cdr config CDR\_ENV [variable\_name] [long]  
Print Enterprise Replication configuration information  
Not specifying a parameter will display the name of and  
information about all available parameters.  
dtd Print statistics for the Enterprise Replication delete table  
cleaner  
dss [ UDR | UDRx ]  
Print statistics about data sync threads and user-defined data  
types  
grp [ A|E|Ex|G|L|Lx|M|Mz|P|pager|R|S|SI|Sx|T|UDR|UDRx ]  
Print statistics about the Enterprise Replication grouper  
nif [ all | sites | serverid | sum ]  
Print statistics about the Enterprise Replication network  
interface  
que Print statistics for the Enterprise Replication high-level queues  
rcv [serverid]  
Print statistics about the Enterprise Replication receive manager  
rep [replname]  
Print events that are in the queue for the schedule manager  
rqm [ ACKQ | CNTRLQ | RECVQ | SENDQ | SYNCQ | SBSPACES | FULL  
| BRIEF |  
VERBOSE ]  
Print statistics of the Enterprise Replication low-level queues  
sync Print the Enterprise Replication synchronization status

#### CHANGED DATA CAPTURE COMMANDS:

In commands that take an optional <sessid> parameter, omitting the  
CDC session id or using the value 0 applies the command to all CDC  
sessions.

cdc [<sessid>] [long]  
Print Change Data Capture information.  
cdc [<sessid>] config [long]  
Print Change Data Capture configuration parameters for  
session(s).  
cdc [<sessid>] bufm [long]  
Print Change Data Capture buffer manager information for  
session(s). Use long option to display information about  
individual allocated buffers.  
cdc [<sessid>] table [<full-table-name>] [long]  
Print Change Data Capture captured table information. Use

dbname:owner.tabname format for <full-table-name> parameter.  
Omit <full-table-name> parameter to display information about  
all captured tables.

- G Print global transaction ids
- h Print buffer hash chain info
- i Interactive mode
- j Print interactive status of the active onpload process
- k Print locks
- l Print logging
- L Print distribution of available locks on the lock free lists
- m Print message log
- o Put shared memory into specified file (default: onstat.out)
- O Print Optical Subsystem memory and staging cache information
- p Print profile
- P Print partition buffer summary
- r Repeat options every <seconds> seconds (default: 5)
- R Print LRU queues
- s Print latches
- t Print TBLspaces
- T Print tablespace information
- u Print user threads
- x Print transactions
- X Print entire list of sharers and waiters for buffers
- z Zero profile counts

## 15. ANEXO G: Batería de pruebas realizadas

Se documentan las baterías de pruebas realizadas para poder probar el funcionamiento del modelo.

### 1. Inserción de estado de un *ticket* 'Resuelto'

La tabla de estados de un *ticket* (*state*) contiene el ciclo de estados para diversos números de *tickets*, en concreto realizamos las pruebas con el *ticket* T000000004:

Estados registrados en la tabla State para el *ticket* T000000004:

```
Database selected.

select * from State where stateticketnum = 'T000000004';

statecodstate 128
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod User2@corp_mail.com
statenew Creado

statecodstate 129
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod User2@corp_mail.com
statenew Asignado

statecodstate 130
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.849
stateusercod User2@corp_mail.com
statenew En progreso

3 row(s) retrieved.

Database closed.
```

Estado de la tabla timeSollnc:

```
Database selected.

select * from timeSollnc where priority = 'P1' and month_tsi = 6;
```

```
cod_timesolinc 6
month_tsi      6
timeacu        0
timesol        0
priority       P1
numinc         0
```

1 row(s) retrieved.

Database closed.

Se inserta el estado 'Resuelto' mediante su procedimiento "Alta\_Estado\_Ticket".

Se hacen las pruebas con intervalo de dos días para sacar un volumen alto de minutos en la tabla:

HORA DE CREACIÓN DE LA INCIDENCIA = 2018-06-03 12:51:19.848

HORA DE RESOLUCIÓN DE LA INCIDENCIA = 2018-06-05 17:29:01.476

POR LO TANTO, EL TIEMPO DE RESOLUCIÓN = 3157 minutos (aproximadamente 2 días y 4,61 horas).

Por lo tanto, esta acción supone:

- Inserción del nuevo estado del *ticket* en la tabla State.
- Inserción de las métricas en la tabla timeSolInc del registro estadístico.
- Registro de la operación realizada en la tabla Log\_procedures.

#### RESULTADO DE TODAS LAS OPERACIONES DESENCADENADAS:

Database select.

```
execute                                     procedure
Alta_Estado_Ticket('T000000004',CURRENT,'User1@corp_mail.com','Resuelto
');
```

(expression) OK

1 row(s) retrieved.



```
--execute procedure
Actualiza_Estado_Tickets('T000000004',CURRENT,'User1@corp_mail.com','Resuelto');
select * from State where stateticketnum = 'T000000004';
```

```
statecodstate 128
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod User2@corp_mail.com
statenew Creado
```

```
statecodstate 129
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod User2@corp_mail.com
statenew Asignado
```

```
statecodstate 130
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.849
stateusercod User2@corp_mail.com
statenew En progreso
```

```
statecodstate 234
stateticketnum T000000004
statechangedate 2018-06-05 17:29:01.476
stateusercod User1@corp_mail.com
statenew Resuelto
```

4 row(s) retrieved.

```
select * from timeSolInc where priority = 'P1' and month_tsi = 6;
```

```
cod_timesolinc 6
month_tsi 6
timeacu 3157
timesol 3157.00000000000000000000
priority P1
numinc 1
```

1 row(s) retrieved.

```
select * from Log_procedures;
```

```
proccod 469
```

```

procname    Alta_Estado_Ticket
procparamin                                T000000004//2018-06-05
17:29:01.476//User1@corp_mail.com//Resuelto
o
procparamout OK
procreg    2018-06-05 17:29:01.476

proccod    470
procname    timeSolInc
procparamin 0//6//3157//3157.0000000000000000//P1//1
procparamout OK
procreg    2018-06-05 17:29:01.476

2 row(s) retrieved.

Database closed.

```

Si se borra el estado 'Resuelto' (mediante un *delete* sin utilizar el procedimiento de borrado para hacer pruebas estadísticas) y se vuelve a insertar:

- Se incrementa el número de peticiones con prioridad P1.
- Se incrementa el tiempo acumulado.
- La media varía al haber 2 incidencias en vez de 1, se divide el acumulado total entre 2.

```

execute                                         procedure
Alta_Estado_Ticket('T000000004',CURRENT,'User1@corp_mail.com','Resuelto
');

(expression) OK

1 row(s) retrieved.

--execute                                         procedure
Actualiza_Estado_Tickets('T000000004',CURRENT,'User1@corp_mail.com','R
esuelto');
select * from State where stateticketnum = 'T000000004';

statecodstate 128
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod   User2@corp_mail.com
statenew       Creado

statecodstate 129
stateticketnum T000000004

```



## 2. Borrado del estado de un ticket con condición 'Resuelto'

Al borrar el estado de un *ticket* con condición 'Resuelto' esta acción desencadena:

- Borrado del estado del *ticket* 'Resuelto' en la tabla State.
- Decrementar el número de peticiones con prioridad en la tabla timeSollnc.

Por lo tanto:

### Estado inicial de la tabla timeSollnc:

```
Database selected.
```

```
select * from timeSollnc where priority = 'P1' and month_tsi = 6;
```

```
cod_timesolinc 6
month_tsi      6
timeacu        15856
timesol        3171.2000000000000000
priority       P1
numinc         5
```

```
1 row(s) retrieved.
```

```
Database closed.
```

### Estado inicial de la tabla State:

```
Database selected.
```

```
select * from State where stateticketnum = 'T000000004';
```

```
statecodstate  128
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod   User2@corp_mail.com
statenew       Creado
```

```
statecodstate  129
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod   User2@corp_mail.com
```

```
statenew      Asignado

statecodstate 130
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.849
stateusercod   User2@corp_mail.com
statenew      En progreso

statecodstate 238
stateticketnum T000000004
statechangedate 2018-06-05 18:00:23.132
stateusercod   User1@corp_mail.com
statenew      Resuelto
```

4 row(s) retrieved.

Database closed.

Al borrar el estado con el procedimiento para ello Baja\_Estado\_Ticket:

```
execute procedure Baja_Estado_Ticket('T000000004','Resuelto');
```

(expression) OK

1 row(s) retrieved.

```
--execute procedure
Alta_Estado_Ticket('T000000004',CURRENT,'User1@corp_mail.com','Resuelto');
```

```
--execute procedure
Actualiza_Estado_Tickets('T000000004',CURRENT,'User1@corp_mail.com','Resuelto');
```

```
select * from State where stateticketnum = 'T000000004';
```

```
statecodstate 128
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod   User2@corp_mail.com
statenew      Creado
```



- Actualización del estado del *ticket* 'Resuelto' en la tabla State.
- Incrementar el número de peticiones con prioridad en la tabla timeSollnc, SOLO SI EL ESTADO ANTERIOR NO ES RESUELTO.

Por lo tanto:

Estado inicial de la tabla timeSollnc:

```
Database selected.

select * from timeSollnc where priority = 'P1' and month_tsi = 6;

cod_timesolinc 6
month_tsi      6
timeacu        16543
timesol        870.6842105263157895
priority       P1
numinc         19

1 row(s) retrieved.

Database closed.
```

Estado inicial de la tabla State:

```
Database selected.

select * from State where stateticketnum = 'T000000004';

statecodstate  128
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod   User2@corp_mail.com
statenew       Creado

statecodstate  129
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod   User2@corp_mail.com
statenew       Asignado

statecodstate  259
stateticketnum T000000004
```

```
statechangedate 2018-06-06 10:05:24.442
stateusercod    User1@corp_mail.com
statenew        Resuelto
4 row(s) retrieved.
```

Database closed.

Al actualizar el estado a 'Resuelto' y si ya está en ese estado (por ejemplo si se actualiza cualquier otro campo, no se toca la tabla de estadística):

Database selected.

```
select * from timeSollnc where priority = 'P1' and month_tsi = 6;
```

```
cod_timesolinc 6
month_tsi      6
timeacu        16543
timesol        870.6842105263157895
priority       P1
numinc         19
```

1 row(s) retrieved.

Database closed.

Y el resultado en la tabla State y el registro de procedimientos verifican que no se ha tocado la tabla del repositorio, ni se ha llamado al procedimiento que la actualiza:

```
execute                                     procedure
Actualiza_Estado_Tickets(259,'T000000004',CURRENT,'User1@corp_mail.co
m','Resuelto');
```

(expression) OK

1 row(s) retrieved.



```
select * from State where stateticketnum = 'T000000004';
```

```
statecodstate 128
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod User2@corp_mail.com
statenew Creado
```

```
statecodstate 129
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod User2@corp_mail.com
statenew Asignado
```

```
statecodstate 259
stateticketnum T000000004
statechangedate 2018-06-06 10:05:24.442
stateusercod User1@corp_mail.com
statenew Resuelto
```

3 row(s) retrieved.

```
select * from timeSollnc where priority = 'P1' and month_tsi = 6;
```

```
cod_timesolinc 6
month_tsi 6
timeacu 16543
timesol 870.6842105263157895
priority P1
numinc 19
```

1 row(s) retrieved.

```
select * from Log_procedures;
```

```
proccod 598
procname Actualiza_Estado_Tickets
procparamin 259//T000000004//2018-06-06
10:05:24.442//User1@corp_mail.com//Re
suelto
procparamout OK
procreg 2018-06-06 10:05:24.442
```

1 row(s) retrieved.

SI SE CAMBIA EL ESTADO DE RESUELTO A OTRO DISTINTO:

- Hay que decrementar el número de incidencias por la prioridad indicada en la tabla del repositorio.

Estado de la tabla State:

```
statecodstate 128
staticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod User2@corp_mail.com
statenew Creado
```

```
statecodstate 129
staticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod User2@corp_mail.com
statenew Asignado
```

```
statecodstate 259
staticketnum T000000004
statechangedate 2018-06-06 10:05:24.442
stateusercod User1@corp_mail.com
statenew Resuelto
```

Estado de la tabla del repositorio:

Database selected.

```
select * from timeSollnc where priority = 'P1' and month_tsi = 6;
```

```
cod_timesolinc 6
month_tsi 6
timeacu 16543
timesol 870.6842105263157895
priority P1
numinc 19
```

1 row(s) retrieved.

Database closed.

Después de la acción:

Se observa como se registra el decremento del número de incidencias en la tabla del repositorio y como se registra la acción en la tabla de control del repositorio (timeSollnc):

```
execute Actualiza_Estado_Tickets(259,'T000000004',CURRENT,'User1@corp_mail.com','En Progreso'); procedure
```

(expression) OK

1 row(s) retrieved.

```
select * from State where stateticketnum = 'T000000004';
```

```
statecodstate 128
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod User2@corp_mail.com
statenew Creado
```

```
statecodstate 129
stateticketnum T000000004
statechangedate 2018-06-03 12:51:19.848
stateusercod User2@corp_mail.com
statenew Asignado
```

```
statecodstate 259
stateticketnum T000000004
statechangedate 2018-06-06 10:14:43.246
stateusercod User1@corp_mail.com
statenew En Progreso
```

3 row(s) retrieved.

```
select * from timeSollnc where priority = 'P1' and month_tsi = 6;
```

```
cod_timesolinc 6
month_tsi 6
timeacu 16543
timesol 870.6842105263157895
priority P1
numinc 18
```





#### 4. Gestión de usuarios:

##### Consulta de usuarios:

Se comprueba que no existe el usuario User8:

```
Database selected.

execute procedure Consulta_Usuario('User8@corp_mail.com');

rsp          NO EXISTE EL USUARIO User8@corp_mail.com en la tabla
Users
codigo_usuario
fecha_registro_us+
fecha_baja_usuario
tipo_usuario
fecha_subscripcion
codigo_centro
codigo_soporte

1 row(s) retrieved.

--execute                                     procedure
Alta_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'Administrador',C
URRENT,'SD00001','G00001');
--execute procedure Baja_Usuario('User8@corp_mail.com');
--execute                                     procedure
Actualiza_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'CLIENTE',C
URRENT,'SD00001','G00001');

Database closed.
```

Se da de alta el usuario en el sistema:

```
Database selected.

--execute procedure Consulta_Usuario('User8@corp_mail.com');
execute                                     procedure
Alta_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'Administrador',C
URRENT,'SD00001','G00001');
```

(expression) OK

1 row(s) retrieved.

```
--execute procedure Baja_Usuario('User8@corp_mail.com');
--execute procedure
Actualiza_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'CLIENTE',C
URRENT,'SD00001','G00001');
```

Database closed.

Se vuelve a comprobar que se ha dado de alta el usuario [User8@corp\\_mail.com](mailto:User8@corp_mail.com) correctamente en el sistema:

Database selected.

```
execute procedure Consulta_Usuario('User8@corp_mail.com');
```

```
rsp      OK
codigo_usuario      User8@corp_mail.com
fecha_registro_us+  2018-06-11 16:21:35.714
fecha_baja_usuario  2018-06-11 16:21:35.714
tipo_usuario        Administrador
fecha_subscripcion  2018-06-11 16:21:35.714
codigo_centro       SD00001
codigo_soporte      G00001
```

1 row(s) retrieved.

```
--execute procedure
Alta_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'Administrador',C
URRENT,'SD00001','G00001');
--execute procedure Baja_Usuario('User8@corp_mail.com');
--execute procedure
Actualiza_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'CLIENTE',C
URRENT,'SD00001','G00001');
```

Database closed.

Se borra el usuario en el sistema:

Database selected.

```
--execute procedure Consulta_Usuario('User8@corp_mail.com');  
--execute procedure  
Alta_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'Administrador',C  
URRENT,'SD00001','G00001');  
execute procedure Baja_Usuario('User8@corp_mail.com');
```

(expression) OK

1 row(s) retrieved.

```
--execute procedure  
Actualiza_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'CLIENTE',C  
URRENT,'SD00001','G00001');
```

Database closed.

Se comprueba que se ha borrado correctamente en el sistema:

Database selected.

```
execute procedure Consulta_Usuario('User8@corp_mail.com');
```

```
rsp          NO EXISTE EL USUARIO User8@corp_mail.com en la tabla  
Users  
codigo_usuario  
fecha_registro_us+  
fecha_baja_usuario  
tipo_usuario  
fecha_subscripcion  
codigo_centro  
codigo_soporte
```

1 row(s) retrieved.



```

--execute                                     procedure
Alta_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'Administrador',C
URRENT,'SD00001','G00001');
--execute procedure Baja_Usuario('User8@corp_mail.com');
--execute                                     procedure
Actualiza_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'CLIENTE',C
URRENT,'SD00001','G00001');

```

Database closed.

Se procede a actualizar el usuario que no es un cliente ('No Cliente') y como no existe en el sistema genera el siguiente error controlado:

Database selected.

```

--execute procedure Consulta_Usuario('User8@corp_mail.com');
--execute                                     procedure
Alta_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'Administrador',C
URRENT,'SD00001','G00001');
--execute procedure Baja_Usuario('User8@corp_mail.com');
execute                                     procedure
Actualiza_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'NO
CLIENTE',CURRENT,'SD00001','G00001');

```

(expression) NO EXISTE EL USUARIO User8@corp\_mail.com en la tabla Users

1 row(s) retrieved.

Database closed.

Se vuelve a dar de alta en el sistema y ahora si deja actualizar el estado, esta es la secuencia completa con la comprobación incluida tanto de la inserción como de la actualización:

Database selected.

```
--execute procedure Consulta_Usuario('User8@corp_mail.com');
execute                                     procedure
Alta_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'Administrador',C
URRENT,'SD00001','G00001');
```

(expression) OK

1 row(s) retrieved.

```
execute procedure Consulta_Usuario('User8@corp_mail.com');
```

```
rsp      OK
codigo_usuario      User8@corp_mail.com
fecha_registro_us+  2018-06-11 16:30:16.259
fecha_baja_usuario  2018-06-11 16:30:16.259
tipo_usuario        Administrador
fecha_subscripcion  2018-06-11 16:30:16.259
codigo_centro       SD00001
codigo_soporte      G00001
```

1 row(s) retrieved.

```
--execute procedure Baja_Usuario('User8@corp_mail.com');
execute                                     procedure
Actualiza_Usuario('User8@corp_mail.com',CURRENT,CURRENT,'NO
CLIENTE',CURRENT,'SD00001','G00001');
```

(expression) OK

1 row(s) retrieved.

```
execute procedure Consulta_Usuario('User8@corp_mail.com');
```

```
rsp      OK
codigo_usuario      User8@corp_mail.com
fecha_registro_us+  2018-06-11 16:30:16.262
fecha_baja_usuario  2018-06-11 16:30:16.262
tipo_usuario        NO CLIENTE
fecha_subscripcion  2018-06-11 16:30:16.262
codigo_centro       SD00001
codigo_soporte      G00001
```

1 row(s) retrieved.

Database closed.

Todas las operaciones quedan registradas en la tabla log\_procedures:

```
proccod    1
procname   Consulta_Usuario
procparamin User8@corp_mail.com
procparamout NO EXISTE EL USUARIO User8@corp_mail.com en la tabla
Users
procreg    2018-06-11 16:19:33.678

proccod    2
procname   Alta_Usuario
procparamin User8@corp_mail.com//2018-06-11 16:21:35.714//2018-06-11
16:21:35
           .714//Administrador//2018-06-11 16:21:35.714//SD00001//G00001
procparamout OK
procreg    2018-06-11 16:21:35.714

proccod    3
procname   Consulta_Usuario
procparamin User8@corp_mail.com
procparamout OK
procreg    2018-06-11 16:23:12.246

proccod    4
procname   Baja_Usuario
procparamin User8@corp_mail.com
procparamout OK
procreg    2018-06-11 16:24:21.535

proccod    5
procname   Consulta_Usuario
procparamin User8@corp_mail.com
procparamout NO EXISTE EL USUARIO User8@corp_mail.com en la tabla
Users
procreg    2018-06-11 16:24:59.809

proccod    6
procname   Actualiza_Usuario
procparamin User8@corp_mail.com//2018-06-11 16:26:45.686//2018-06-11
16:26:45
```

```
.686//NO CLIENTE//2018-06-11 16:26:45.686//SD00001//G00001
procparamout NO EXISTE EL USUARIO User8@corp_mail.com en la tabla
Users
procreg 2018-06-11 16:26:45.686

proccod 7
procname Consulta_Usuario
procparamin User8@corp_mail.com
procparamout NO EXISTE EL USUARIO User8@corp_mail.com en la tabla
Users
procreg 2018-06-11 16:27:21.779

proccod 8
procname Consulta_Usuario
procparamin User8@corp_mail.com
procparamout NO EXISTE EL USUARIO User8@corp_mail.com en la tabla
Users
procreg 2018-06-11 16:27:47.195

proccod 9
procname Actualiza_Usuario
procparamin User8@corp_mail.com//2018-06-11 16:27:57.323//2018-06-11
16:27:57
.323//NO CLIENTE//2018-06-11 16:27:57.323//SD00001//G00001
procparamout NO EXISTE EL USUARIO User8@corp_mail.com en la tabla
Users
procreg 2018-06-11 16:27:57.323

proccod 10
procname Alta_Usuario
procparamin User8@corp_mail.com//2018-06-11 16:30:16.259//2018-06-11
16:30:16
.259//Administrador//2018-06-11 16:30:16.259//SD00001//G00001
procparamout OK
procreg 2018-06-11 16:30:16.259

proccod 11
procname Consulta_Usuario
procparamin User8@corp_mail.com
procparamout OK
procreg 2018-06-11 16:30:16.260

proccod 12
procname Actualiza_Usuario
procparamin User8@corp_mail.com//2018-06-11 16:30:16.262//2018-06-11
16:30:16
.262//NO CLIENTE//2018-06-11 16:30:16.262//SD00001//G00001
procparamout OK
procreg 2018-06-11 16:30:16.262
```

```
proccod      13
procname     Consulta_Usuario
procparamin  User8@corp_mail.com
procparamout OK
procreg      2018-06-11 16:30:16.263
```

13 row(s) retrieved.

## 5. Gestión de Tickets:

Se comprueba que el ticket 'T000000001' existe:

Database selected.

```
execute procedure Consulta_Ticket('T000000001');
```

```
rsp          OK
numero_ticket      T000000001
tipo_ticket        INC
codigo_usuario     User1@corp_mail.com
fecha_creacion_ti+ 2018-06-06 18:03:02.759
prioridad_ticket   P4
servicio_ticket    SE0000242
canal_ticket       WEB
ciclo_ticket       T000000001
descripcion_ticket Solicitar permisos de acceso a la aplicación GSD del tipo
                  consulta para los usuarios del departamento de Riesgos.
codigo_mensaje     MS0000456
documento_asociado
```

1 row(s) retrieved.

```
--execute procedure Alta_Ticket('T0000000010', 'INC', 'User2@corp_mail.com',
CURRENT, 'P1' , 'SE0000678','WEB', 'T000000004', 'Caida de aplicación',
'MS0000898',NULL);
--execute procedure Baja_Ticket('T000000008');
--execute          procedure          Actualiza_Ticket('T0000000011',          'INC',
'User2@corp_mail.com', CURRENT, 'P1' , 'SESESESE','WEB', 'T000000004',
'Se actualiza el ticket - EN MARCHA -', 'MS0000898',NULL);
```

Database closed.

Se comprueba otro ticket que no existe (T00000000111):

```
Database selected.

execute procedure Consulta_Ticket('T00000000111');

rsp          NO EXISTE EL Ticket T00000000111 en la tabla Tickets
numero_ticket
tipo_ticket
codigo_usuario
fecha_creacion_ti+
prioridad_ticket
servicio_ticket
canal_ticket
ciclo_ticket
descripcion_ticket
codigo_mensaje
documento_asociado

1 row(s) retrieved.

--execute procedure Alta_Ticket('T0000000010', 'INC', 'User2@corp_mail.com',
CURRENT, 'P1' , 'SE0000678','WEB', 'T000000004', 'Caida de aplicación',
'MS0000898',NULL);
--execute procedure Baja_Ticket('T000000008');
--execute      procedure      Actualiza_Ticket('T0000000011',      'INC',
'User2@corp_mail.com', CURRENT, 'P1' , 'SESESESE','WEB', 'T000000004',
'Se actualiza el ticket - EN MARCHA -', 'MS0000898',NULL);
```

Se da de alta el ticket 'T00000000111' y se comprueba que ahora ya existe:

```
Database selected.

execute procedure Alta_Ticket('T00000000111', 'INC', 'User2@corp_mail.com',
CURRENT, 'P1' , 'SE0000678','WEB', 'T000000004', 'Caida de aplicación',
'MS0000898',NULL);

(expression) OK

1 row(s) retrieved.
```

```
execute procedure Consulta_Ticket('T00000000111');
```

```
rsp          OK
numero_ticket T00000000111
tipo_ticket   INC
codigo_usuario User2@corp_mail.com
fecha_creacion_ti+ 2018-06-11 16:46:29.834
prioridad_ticket P1
servicio_ticket SE0000678
canal_ticket   WEB
ciclo_ticket   T000000004
descripcion_ticket Caída de aplicación
codigo_mensaje MS0000898
documento_asociado
```

1 row(s) retrieved.

```
--execute procedure Baja_Ticket('T000000008');
--execute          procedure Actualiza_Ticket('T00000000111',          'INC',
'User2@corp_mail.com', CURRENT, 'P1' , 'SESESESE','WEB', 'T000000004',
'Se actualiza el ticket - EN MARCHA -', 'MS0000898',NULL);
```

Database closed.

Se borra el ticket 'T00000000111' y se comprueba su borrado en el Sistema:

Database selected.

```
--execute          procedure Alta_Ticket('T00000000111',          'INC',
'User2@corp_mail.com', CURRENT, 'P1' , 'SE0000678','WEB', 'T000000004',
'Caída de aplicación', 'MS0000898',NULL);
--execute procedure Consulta_Ticket('T00000000111');
execute procedure Baja_Ticket('T00000000111');
```

(expression) OK

1 row(s) retrieved.

```
execute procedure Consulta_Ticket('T00000000111');
```

```
rsp          NO EXISTE EL Ticket T00000000111 en la tabla Tickets
numero_ticket
tipo_ticket
codigo_usuario
fecha_creacion_ti+
prioridad_ticket
servicio_ticket
canal_ticket
ciclo_ticket
descripcion_ticket
codigo_mensaje
documento_asociado
```

1 row(s) retrieved.

```
--execute      procedure      Actualiza_Ticket('T00000000111',      'INC',
'User2@corp_mail.com', CURRENT, 'P1' , 'SESESESE','WEB', 'T000000004',
'Se actualiza el ticket - EN MARCHA -', 'MS0000898',NULL);
```

Database closed.

Se intenta actualizar el ticket que ya no está en el sistema (T00000000111) para que devuelva un error controlado:

Database selected.

```
--execute      procedure      Alta_Ticket('T00000000111',      'INC',
'User2@corp_mail.com', CURRENT, 'P1' , 'SE0000678','WEB', 'T000000004',
'Caida de aplicación', 'MS0000898',NULL);
--execute procedure Consulta_Ticket('T00000000111');
--execute procedure Baja_Ticket('T00000000111');
--execute procedure Consulta_Ticket('T00000000111');
execute      procedure      Actualiza_Ticket('T00000000111',      'INC',
'User2@corp_mail.com', CURRENT, 'P1' , 'SESESESE','WEB', 'T000000004',
'Se actualiza el ticket - EN MARCHA -', 'MS0000898',NULL);
```

(expression) NO EXISTE EL Ticket T00000000111 en la tabla Tickets

1 row(s) retrieved.



Database closed.

Se vuelve a dar de alta y se actualiza su servicio a uno incorrecto para devolver un error controlado, se verifica todo el ciclo:

Database selected.

```
execute procedure Alta_Ticket('T00000000111', 'INC', 'User2@corp_mail.com',  
CURRENT, 'P1' , 'SE0000678','WEB', 'T000000004', 'Caida de aplicación',  
'MS0000898',NULL);
```

(expression) OK

1 row(s) retrieved.

```
execute procedure Consulta_Ticket('T00000000111');
```

```
rsp          OK  
numero_ticket    T00000000111  
tipo_ticket      INC  
codigo_usuario   User2@corp_mail.com  
fecha_creacion_ti+ 2018-06-11 16:51:25.227  
prioridad_ticket P1  
servicio_ticket  SE0000678  
canal_ticket     WEB  
ciclo_ticket     T000000004  
descripcion_ticket Caida de aplicación  
codigo_mensaje   MS0000898  
documento_asociado
```

1 row(s) retrieved.

```
--execute procedure Baja_Ticket('T00000000111');
```

```
execute          procedure      Actualiza_Ticket('T00000000111',          'INC',  
'User2@corp_mail.com', CURRENT, 'P1' , 'SESESESE','WEB', 'T000000004',  
'Se actualiza el ticket - EN MARCHA -', 'MS0000898',NULL);
```

```
(expression) NO EXISTE EL CODIGO DE SERVICIO SESESESE en la tabla
Services
```

```
1 row(s) retrieved.
```

```
execute procedure Consulta_Ticket('T00000000111');
```

```
rsp          OK
numero_ticket T00000000111
tipo_ticket   INC
codigo_usuario User2@corp_mail.com
fecha_creacion_ti+ 2018-06-11 16:51:25.227
prioridad_ticket P1
servicio_ticket SE0000678
canal_ticket   WEB
ciclo_ticket   T000000004
descripcion_ticket Caída de aplicación
codigo_mensaje MS0000898
documento_asociado
```

```
1 row(s) retrieved.
```

```
Database closed.
```

Se actualiza el tipo de ticket abierto por TLF (antes WEB) con código de servicio correcto, se verifica todo el ciclo:

```
Database selected.
```

```
execute          procedure      Actualiza_Ticket('T00000000111',      'INC',
'User2@corp_mail.com', CURRENT, 'P1' , 'SE0000678','TLF', 'T000000004',
'Se actualiza el ticket - EN MARCHA -', 'MS0000898',NULL);
```

```
(expression) OK
```

```
1 row(s) retrieved.
```

```
execute procedure Consulta_Ticket('T00000000111');
```

```
rsp          OK
numero_ticket T00000000111
tipo_ticket   INC
codigo_usuario User2@corp_mail.com
fecha_creacion_ti+ 2018-06-11 16:53:55.453
prioridad_ticket P1
servicio_ticket SE0000678
canal_ticket   TLF
ciclo_ticket   T000000004
descripcion_ticket Se actualiza el ticket - EN MARCHA -
codigo_mensaje MS0000898
documento_asociado
```

1 row(s) retrieved.

Database closed.

Se registran todas las operaciones en la tabla log\_procedures:

Database selected.

```
--execute      procedure      Alta_Ticket('T00000000111',      'INC',
'User2@corp_mail.com', CURRENT, 'P1' , 'SE0000678','WEB', 'T000000004',
'Caida de aplicación', 'MS0000898',NULL);
--execute procedure Consulta_Ticket('T00000000111');
--execute procedure Baja_Ticket('T00000000111');
```

```
--execute      procedure      Actualiza_Ticket('T00000000111',      'INC',
'User2@corp_mail.com', CURRENT, 'P1' , 'SE0000678','TLF', 'T000000004',
'Se actualiza el ticket - EN MARCHA -', 'MS0000898',NULL);
--execute procedure Consulta_Ticket('T00000000111');
```

```
select * from log_procedures;
```

```
proccod      14
procname     Consulta_CODIGO_TICKET
procparamin  T0000000011
procparamout NO EXISTE EL Ticket T0000000011 en la tabla State
```

procreg 2018-06-11 16:35:25.568

proccod 15  
procname Consulta\_CODIGO\_TICKET  
proparamin T000000001  
proparamout OK  
procreg 2018-06-11 16:36:06.945

proccod 16  
procname Alta\_Estado\_Ticket  
proparamin T0000000011//2018-06-11  
16:37:50.484//User1@corp\_mail.com//PRUEBA  
\_NUEVO  
proparamout error -111 El ticket asociado al estado no existe  
procreg 2018-06-11 16:37:50.484

proccod 17  
procname Consulta\_CODIGO\_TICKET  
proparamin T000000001  
proparamout OK  
procreg 2018-06-11 16:37:50.534

proccod 18  
procname Alta\_Estado\_Ticket  
proparamin T0000000011//2018-06-11  
16:40:17.168//User1@corp\_mail.com//Creado  
proparamout error -111 El ticket asociado al estado no existe  
procreg 2018-06-11 16:40:17.168

proccod 19  
procname Consulta\_CODIGO\_TICKET  
proparamin T000000001  
proparamout OK  
procreg 2018-06-11 16:40:17.170

proccod 20  
procname Consulta\_Ticket  
proparamin T000000001  
proparamout OK  
procreg 2018-06-11 16:43:34.357

proccod 21  
procname Consulta\_Ticket  
proparamin T00000000111  
proparamout NO EXISTE EL Ticket T00000000111 en la tabla Tickets  
procreg 2018-06-11 16:44:53.113

proccod 22  
procname Alta\_Ticket

proccod 23  
procname Consulta\_Ticket  
proccod 23  
procname Consulta\_Ticket  
proccod 24  
procname Baja\_Ticket  
proccod 25  
procname Consulta\_Ticket  
proccod 26  
procname Actualiza\_Ticket  
proccod 27  
procname Alta\_Ticket  
proccod 28  
procname Consulta\_Ticket  
proccod 29  
procname Actualiza\_Ticket

T00000000111//INC//User2@corp\_mail.com//2018-06-11  
16:46:29.834//  
P1//SE0000678//WEB//T000000004//Caida de aplicación//MS0000898

proccod 23  
procname Consulta\_Ticket  
proccod 23  
procname Consulta\_Ticket  
proccod 24  
procname Baja\_Ticket  
proccod 25  
procname Consulta\_Ticket  
proccod 26  
procname Actualiza\_Ticket  
proccod 27  
procname Alta\_Ticket  
proccod 28  
procname Consulta\_Ticket  
proccod 29  
procname Actualiza\_Ticket

T00000000111//INC//User2@corp\_mail.com//2018-06-11  
16:49:46.337//  
P1//SESESESE//WEB//T000000004//Se actualiza el ticket - EN  
MARCHA  
-//MS0000898

NO EXISTE EL Ticket T00000000111 en la tabla Tickets

NO EXISTE EL Ticket T00000000111 en la tabla Tickets

NO EXISTE EL Ticket T00000000111 en la tabla Tickets

```

proccod      30
procname     Consulta_Ticket
proccod      31
procname     Actualiza_Ticket
proccod      32
procname     Consulta_Ticket
19 row(s) retrieved.

```

## 6. Gestión de Estados de Tickets:

Se consultan los estados de un Ticket que no existe:

Database selected.

```
execute procedure Consulta_Estado_Ticket('T0000000022');
```

```

rsp          NO EXISTE EL Ticket T0000000022 en la tabla State
codigo_estado
codigo_ticket
fecha_cambio_esta+
codigo_usuario
nuevo_estado

```

1 row(s) retrieved.

```
--execute                                     procedure
Alta_Estado_Ticket('T000000001111',CURRENT,'User1@corp_mail.com','PRU
EBA_NUEVO');
--execute procedure Baja_Estado_Ticket('T000000001','PRUEBA_NUEVO');
--execute                                     procedure
Actualiza_Estado_Tickets(329,'T000000001111',CURRENT,'User1@corp_mail.
com','ESTADO_ACTUALIZADO');
```

Database closed.

Se consultan los estados de un ticket que si existe:

Database selected.

```
execute procedure Consulta_Estado_Ticket('T000000001');
```

```
rsp          OK
codigo_estado  1
codigo_ticket  T000000001
fecha_cambio_esta+ 2018-06-06 18:03:02.762
codigo_usuario  User1@corp_mail.com
nuevo_estado    Creado
```

```
rsp          OK
codigo_estado  2
codigo_ticket  T000000001
fecha_cambio_esta+ 2018-06-06 18:03:02.763
codigo_usuario  User1@corp_mail.com
nuevo_estado    Asignado
```

```
rsp          OK
codigo_estado  3
codigo_ticket  T000000001
fecha_cambio_esta+ 2018-06-06 18:03:02.763
codigo_usuario  User1@corp_mail.com
nuevo_estado    En progreso
```

```
rsp          OK
codigo_estado  4
codigo_ticket  T000000001
fecha_cambio_esta+ 2018-06-06 18:03:02.764
```

```
codigo_usuario    User1@corp_mail.com
nuevo_estado     Cancelado

rsp              OK
codigo_estado     5
codigo_ticket     T000000001
fecha_cambio_esta+ 2018-06-06 18:03:02.764
codigo_usuario    User1@corp_mail.com
nuevo_estado     Resuelto
```

5 row(s) retrieved.

```
--execute                                     procedure
Alta_estado_Ticket('T000000001111',CURRENT,'User1@corp_mail.com','PRU
EBA_NUEVO');
--execute procedure Baja_estado_Ticket('T000000001','PRUEBA_NUEVO');
--execute                                     procedure
Actualiza_estado_Tickets(329,'T000000001111',CURRENT,'User1@corp_mail.
com','ESTADO_ACTUALIZADO');
```

Database closed.

Se da de alta en el sistema un estado de un ticket que no existe para salida de error controlado:

Database selected.

```
--execute procedure Consulta_estado_Ticket('T000000001');
execute                                     procedure
Alta_estado_Ticket('T000000001111',CURRENT,'User1@corp_mail.com','Cerr
ado');
```

(expression) error -111 El ticket asociado al estado no existe

1 row(s) retrieved.

```
--execute procedure Baja_estado_Ticket('T000000001','PRUEBA_NUEVO');
--execute                                     procedure
Actualiza_estado_Tickets(329,'T000000001111',CURRENT,'User1@corp_mail.
com','ESTADO_ACTUALIZADO');
```



Se da de alta en el sistema un estado de un ticket que existe y se comprueba:

Database selected.

```
execute procedure
Alta_estado_Ticket('T000000004',CURRENT,'User1@corp_mail.com','Cerrado'
);
```

(expression) OK

1 row(s) retrieved.

```
execute procedure Consulta_estado_Ticket('T000000004');
```

```
rsp          OK
codigo_estado      12
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-06 18:03:02.768
codigo_usuario     User2@corp_mail.com
nuevo_estado       Creado
```

```
rsp          OK
codigo_estado      13
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-06 18:03:02.768
codigo_usuario     User2@corp_mail.com
nuevo_estado       Asignado
```

```
rsp          OK
codigo_estado      14
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-06 18:03:02.769
codigo_usuario     User2@corp_mail.com
nuevo_estado       En progreso
```

```
rsp          OK
codigo_estado      19
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-11 17:03:05.498
codigo_usuario     User1@corp_mail.com
nuevo_estado       Cerrado
```

5 row(s) retrieved.

```
--execute procedure Baja_Estado_Ticket('T000000004','Cerrado');
--execute                                     procedure
Actualiza_Estado_Tickets(329,'T000000001111',CURRENT,'User1@corp_mail.
com','ESTADO_ACTUALIZADO');

--execute procedure Consulta_Estado_Ticket('T000000001');
```

Database closed.

Se actualiza el estado del ticket pero este no existe, se observa la salida correcta del error controlado:

Database selected.

```
--execute                                     procedure
Alta_Estado_Ticket('T000000004',CURRENT,'User1@corp_mail.com','Cerrado'
);
--execute procedure Consulta_Estado_Ticket('T000000004');

--execute procedure Baja_Estado_Ticket('T000000004','Cerrado');
execute                                     procedure
Actualiza_Estado_Tickets(30,'T000000004',CURRENT,'User1@corp_mail.com'
,'ESTADO_ACTUALIZADO');
```

(expression) NO EXISTE EL ESTADO DEL TICKET T000000004 en la tabla State

1 row(s) retrieved.

```
--execute procedure Consulta_Estado_Ticket('T000000001');
```

Database closed.

Se actualiza el estado del ticket correctamente a 'Cancelado', el estado existe, se comprueba:

```
Database selected.

--execute                                     procedure
Alta_estado_Ticket('T000000004',CURRENT,'User1@corp_mail.com','Cerrado'
);
--execute procedure Consulta_estado_Ticket('T000000004');

--execute procedure Baja_estado_Ticket('T000000004','Cerrado');
execute                                     procedure
Actualiza_estado_Tickets(19,'T000000004',CURRENT,'User1@corp_mail.com'
,'Cancelado');

(expression) OK

1 row(s) retrieved.

execute procedure Consulta_estado_Ticket('T000000004');

rsp      OK
codigo_estado      12
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-06 18:03:02.768
codigo_usuario     User2@corp_mail.com
nuevo_estado       Creado

rsp      OK
codigo_estado      13
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-06 18:03:02.768
codigo_usuario     User2@corp_mail.com
nuevo_estado       Asignado

rsp      OK
codigo_estado      14
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-06 18:03:02.769
codigo_usuario     User2@corp_mail.com
nuevo_estado       En progreso

rsp      OK
```

```
codigo_estado      18
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-11 17:01:14.832
codigo_usuario     User1@corp_mail.com
nuevo_estado       Cerrado
```

```
rsp                OK
codigo_estado      19
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-11 17:08:06.931
codigo_usuario     User1@corp_mail.com
nuevo_estado       Cancelado
```

5 row(s) retrieved.

Database closed.

Se borra el estado de un Ticket y se comprueba todo el ciclo tanto de inserción como de borrado:

Database selected.

```
execute                                                     procedure
Alta_Estado_Ticket('T000000004',CURRENT,'User1@corp_mail.com','Resuelto
');
```

(expression) OK

1 row(s) retrieved.

```
execute procedure Consulta_Estado_Ticket('T000000004');
```

```
rsp                OK
codigo_estado      20
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-11 17:47:03.253
codigo_usuario     User2@corp_mail.com
nuevo_estado       Creado
```

```
rsp                OK
```

```
codigo_estado      21
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-11 17:47:03.254
codigo_usuario     User2@corp_mail.com
nuevo_estado      Asignado
```

```
rsp                OK
codigo_estado      22
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-11 17:47:03.255
codigo_usuario     User2@corp_mail.com
nuevo_estado      En progreso
```

```
rsp                OK
codigo_estado      24
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-11 17:53:44.438
codigo_usuario     User1@corp_mail.com
nuevo_estado      Resuelto
```

4 row(s) retrieved.

```
--execute procedure Consulta_Estado_Ticket('T000000004');
--execute procedure Baja_Estado_Ticket('T000000004','Cerrado');
--execute procedure Actualiza_Estado_Tickets(19,'T000000004',CURRENT,'User1@corp_mail.com',
,'Cancelado');
execute procedure Baja_Estado_Ticket('T000000004','Resuelto');
```

(expression) OK

1 row(s) retrieved.

```
execute procedure Consulta_Estado_Ticket('T000000004');
```

```
rsp                OK
codigo_estado      20
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-11 17:47:03.253
codigo_usuario     User2@corp_mail.com
nuevo_estado      Creado
```

```
rsp                OK
codigo_estado      21
codigo_ticket      T000000004
fecha_cambio_esta+ 2018-06-11 17:47:03.254
```

```
codigo_usuario    User2@corp_mail.com
nuevo_estado     Asignado

rsp              OK
codigo_estado     22
codigo_ticket     T000000004
fecha_cambio_esta+ 2018-06-11 17:47:03.255
codigo_usuario    User2@corp_mail.com
nuevo_estado     En progreso
```

3 row(s) retrieved.

Database closed.

Toda la operación se registra en la tabla Log\_procedures y se comprueba su comportamiento estadístico:

```
echo "
select * from log_procedures;
"|dbaccess -e GSD

proccod    88
procname   timeSolInc
proccod    89
procname   Baja_estado_Ticket
proccod    90
procname   Consulta_CODIGO_TICKET
```

## 7. Gestión de Mensajes de Tickets:

Se consultan los mensajes de un Ticket:

```
Database selected.
```

```
execute procedure Consulta_Mensajes_Ticket('T000000001');
```

```
rsp      OK
codigo_mensaje MS0000456
fecha_mensaje 2018-06-06 18:03:02.769
usuario_mensaje User1@corp_mail.com
estado_mensaje Recibido
codigo_ticket  T000000001
```

```
rsp      OK
codigo_mensaje MS0000457
fecha_mensaje 2018-06-06 18:03:02.770
usuario_mensaje User1@corp_mail.com
estado_mensaje Recibido
codigo_ticket  T000000001
```

```
rsp      OK
codigo_mensaje MS0000458
fecha_mensaje 2018-06-06 18:03:02.771
usuario_mensaje User1@corp_mail.com
estado_mensaje Enviado
codigo_ticket  T000000001
```

```
rsp      OK
codigo_mensaje MS0000459
fecha_mensaje 2018-06-06 18:03:02.771
usuario_mensaje User1@corp_mail.com
estado_mensaje Enviado
codigo_ticket  T000000001
```

```
4 row(s) retrieved.
```

```
--execute procedure
Alta_Mensajes_Ticket('MS0000555',CURRENT,'User1@corp_mail.com','PRUE
BA_ALTA_MENSAJE','T000000001');
```

```
--execute procedure Baja_Estado_Ticket('T000000001','MS0000555');
--execute procedure
Actualiza_Mensajes_Tickets('MS0000555888',CURRENT,'User1@corp_mail.co
m','ACTUALIZAMOS_MENSAJE_22','T000000001');
```

Database closed.

Si el ticket no existe:

```
Database selected.
```

```
execute procedure Consulta_Mensajes_Ticket('T000000001111');
```

```
rsp          NO EXISTE MENSAJES PARA T000000001111 en la tabla
Messages
codigo_mensaje
fecha_mensaje
usuario_mensaje
estado_mensaje
codigo_ticket
```

1 row(s) retrieved.

```
--execute procedure
Alta_Mensajes_Ticket('MS0000555',CURRENT,'User1@corp_mail.com','PRUE
BA_ALTA_MENSAJE','T000000001');
--execute procedure Baja_Estado_Ticket('T000000001','MS0000555');
--execute procedure
Actualiza_Mensajes_Tickets('MS0000555888',CURRENT,'User1@corp_mail.co
m','ACTUALIZAMOS_MENSAJE_22','T000000001');
```

Database closed.

Se procede a dar de alta un mensaje en el sistema a un ticket:

```
Database selected.
```



```
execute procedure
Alta_Mensajes_Ticket('MS0000555',CURRENT,'User1@corp_mail.com','PRUE
BA_ALTA_MENSAJE','T000000001');
```

(expression) OK

1 row(s) retrieved.

```
execute procedure Consulta_Mensajes_Ticket('T000000001');
```

```
rsp      OK
codigo_mensaje MS0000456
fecha_mensaje 2018-06-06 18:03:02.769
usuario_mensaje User1@corp_mail.com
estado_mensaje Recibido
codigo_ticket  T000000001
```

```
rsp      OK
codigo_mensaje MS0000457
fecha_mensaje 2018-06-06 18:03:02.770
usuario_mensaje User1@corp_mail.com
estado_mensaje Recibido
codigo_ticket  T000000001
```

```
rsp      OK
codigo_mensaje MS0000458
fecha_mensaje 2018-06-06 18:03:02.771
usuario_mensaje User1@corp_mail.com
estado_mensaje Enviado
codigo_ticket  T000000001
```

```
rsp      OK
codigo_mensaje MS0000459
fecha_mensaje 2018-06-06 18:03:02.771
usuario_mensaje User1@corp_mail.com
estado_mensaje Enviado
codigo_ticket  T000000001
```

```
rsp      OK
codigo_mensaje MS0000555
fecha_mensaje 2018-06-11 17:24:52.654
usuario_mensaje User1@corp_mail.com
estado_mensaje PRUEBA_ALTA_MENSAJE
codigo_ticket  T000000001
```

5 row(s) retrieved.

```
--execute procedure Baja_Estado_Ticket('T000000001','MS0000555');
--execute procedure
Actualiza_Mensajes_Tickets('MS0000555888',CURRENT,'User1@corp_mail.co
m','ACTUALIZAMOS_MENSAJE_22','T000000001');
```

Database closed.

Se actualiza el mensaje dado de alta y se comprueba:

Database selected.

```
execute procedure
Alta_Mensajes_Ticket('MS0000555',CURRENT,'User1@corp_mail.com','PRUE
BA_ALTA_MENSAJE','T000000001');
```

(expression) OK

1 row(s) retrieved.

```
--execute procedure Consulta_Mensajes_Ticket('T000000001');
--execute procedure Baja_Estado_Ticket('T000000001','MS0000555');
execute procedure
Actualiza_Mensajes_Tickets('MS0000555',CURRENT,'User1@corp_mail.com','
ACTUALIZAMOS_MENSAJE_22','T000000001');
```

(expression) OK

1 row(s) retrieved.

```
execute procedure Consulta_Mensajes_Ticket('T000000001');
```

```
rsp      OK
codigo_mensaje  MS0000456
fecha_mensaje   2018-06-06 18:03:02.769
usuario_mensaje User1@corp_mail.com
estado_mensaje  Recibido
```

```
codigo_ticket T000000001
```

```
rsp OK
codigo_mensaje MS0000457
fecha_mensaje 2018-06-06 18:03:02.770
usuario_mensaje User1@corp_mail.com
estado_mensaje Recibido
codigo_ticket T000000001
```

```
rsp OK
codigo_mensaje MS0000458
fecha_mensaje 2018-06-06 18:03:02.771
usuario_mensaje User1@corp_mail.com
estado_mensaje Enviado
codigo_ticket T000000001
```

```
rsp OK
codigo_mensaje MS0000459
fecha_mensaje 2018-06-06 18:03:02.771
usuario_mensaje User1@corp_mail.com
estado_mensaje Enviado
codigo_ticket T000000001
```

```
rsp OK
codigo_mensaje MS0000555
fecha_mensaje 2018-06-11 17:26:51.372
usuario_mensaje User1@corp_mail.com
estado_mensaje ACTUALIZAMOS_MENSAJE_22
codigo_ticket T000000001
```

5 row(s) retrieved.

```
--execute procedure Baja_Estado_Ticket('T000000001','MS0000555');
```

Database closed.

Se borra el mensaje actualizado:

```
execute procedure Baja_Estado_Ticket('T000000001','MS0000555');
```

```
(expression) OK
```

1 row(s) retrieved.

```
execute procedure Consulta_Mensajes_Ticket('T000000001');
```

```
rsp          OK
codigo_mensaje MS0000456
fecha_mensaje 2018-06-06 18:03:02.769
usuario_mensaje User1@corp_mail.com
estado_mensaje Recibido
codigo_ticket  T000000001
```

```
rsp          OK
codigo_mensaje MS0000457
fecha_mensaje 2018-06-06 18:03:02.770
usuario_mensaje User1@corp_mail.com
estado_mensaje Recibido
codigo_ticket  T000000001
```

```
rsp          OK
codigo_mensaje MS0000458
fecha_mensaje 2018-06-06 18:03:02.771
usuario_mensaje User1@corp_mail.com
estado_mensaje Enviado
codigo_ticket  T000000001
```

```
rsp          OK
codigo_mensaje MS0000459
fecha_mensaje 2018-06-06 18:03:02.771
usuario_mensaje User1@corp_mail.com
estado_mensaje Enviado
codigo_ticket  T000000001
```

4 row(s) retrieved.

Database closed.

Si se vuelve a intentar borrar el mensaje ya no existe y se devuelve error controlado:

```
--execute procedure
Alta_Mensajes_Ticket('MS0000555',CURRENT,'User1@corp_mail.com','PRUE
BA_ALTA_MENSAJE','T000000001');
```

```
--execute procedure Consulta_Mensajes_Ticket('T000000001');
--execute procedure Baja_Estado_Ticket('T000000001','MS0000555');
--execute procedure
Actualiza_Mensajes_Tickets('MS0000555',CURRENT,'User1@corp_mail.com',
ACTUALIZAMOS_MENSAJE_22','T000000001');
```

```
execute procedure Baja_Estado_Ticket('T000000001','MS0000555');
```

(expression) NO EXISTE EL ESTADO DEL TICKET T000000001 en la tabla Message

1 row(s) retrieved.

```
execute procedure Consulta_Mensajes_Ticket('T000000001');
```

```
rsp      OK
codigo_mensaje MS0000456
fecha_mensaje 2018-06-06 18:03:02.769
usuario_mensaje User1@corp_mail.com
estado_mensaje Recibido
codigo_ticket  T000000001
```

```
rsp      OK
codigo_mensaje MS0000457
fecha_mensaje 2018-06-06 18:03:02.770
usuario_mensaje User1@corp_mail.com
estado_mensaje Recibido
codigo_ticket  T000000001
```

```
rsp      OK
codigo_mensaje MS0000458
fecha_mensaje 2018-06-06 18:03:02.771
usuario_mensaje User1@corp_mail.com
estado_mensaje Enviado
codigo_ticket  T000000001
```

```
rsp      OK
codigo_mensaje MS0000459
fecha_mensaje 2018-06-06 18:03:02.771
usuario_mensaje User1@corp_mail.com
estado_mensaje Enviado
codigo_ticket  T000000001
```

4 row(s) retrieved.

Database closed.

La tabla Log\_procedures registra todas las operaciones:

```
proccod    61
procname   Baja_Mensaje_Ticket
procparamin T000000001//Asignado
procparamout NO EXISTE EL ESTADO DEL TICKET T000000001 en la tabla
Message
procreg    2018-06-11 17:16:56.390

proccod    62
procname   Consulta_CODIGO_TICKET
procparamin T000000001
procparamout OK
procreg    2018-06-11 17:16:56.392

proccod    63
procname   Consulta_CODIGO_TICKET
procparamin T000000001
procparamout OK
procreg    2018-06-11 17:20:32.329

proccod    64
procname   Baja_Mensaje_Ticket
procparamin T000000004//Resuelto
procparamout NO EXISTE EL ESTADO DEL TICKET T000000004 en la tabla
Message
procreg    2018-06-11 17:20:45.103

proccod    65
procname   Consulta_CODIGO_TICKET
procparamin T000000001
procparamout OK
procreg    2018-06-11 17:20:45.105

proccod    66
procname   Consulta_Mensajes_Ticket
procparamin T000000001
procparamout OK
procreg    2018-06-11 17:23:07.055

proccod    67
procname   Consulta_Mensajes_Ticket
procparamin T000000001111
procparamout NO EXISTE MENSAJES PARA T000000001111 en la tabla
Messages
```

procreg 2018-06-11 17:23:57.737  
 proccod 68  
 procname Alta\_Mensajes\_Ticket  
 proparamin MS0000555//2018-06-11  
 17:24:52.654//User1@corp\_mail.com//PRUEBA\_A  
 LTA\_MENSAJE//T000000001  
 proparamout OK  
 procreg 2018-06-11 17:24:52.654  
 proccod 69  
 procname Consulta\_Mensajes\_Ticket  
 proparamin T000000001  
 proparamout OK  
 procreg 2018-06-11 17:24:52.656  
 proccod 70  
 procname Actualiza\_Mensajes\_Tickets  
 proparamin MS0000555//2018-06-11  
 17:26:16.847//User1@corp\_mail.com//ACTUALIZ  
 AMOS\_MENSAJE\_22//T000000001  
 proparamout OK  
 procreg 2018-06-11 17:26:16.847  
 proccod 71  
 procname Baja\_Mensaje\_Ticket  
 proparamin T000000001//MS0000555  
 proparamout OK  
 procreg 2018-06-11 17:26:16.849  
 proccod 72  
 procname Alta\_Mensajes\_Ticket  
 proparamin MS0000555//2018-06-11  
 17:26:51.370//User1@corp\_mail.com//PRUEBA\_A  
 LTA\_MENSAJE//T000000001  
 proparamout OK  
 procreg 2018-06-11 17:26:51.370  
 proccod 73  
 procname Actualiza\_Mensajes\_Tickets  
 proparamin MS0000555//2018-06-11  
 17:26:51.372//User1@corp\_mail.com//ACTUALIZ  
 AMOS\_MENSAJE\_22//T000000001  
 proparamout OK  
 procreg 2018-06-11 17:26:51.372  
 proccod 74  
 procname Consulta\_Mensajes\_Ticket  
 proparamin T000000001  
 proparamout OK

```

procreg    2018-06-11 17:26:51.373

proccod    75
procname   Baja_Mensaje_Ticket
procparamin T000000001//MS0000555
procparamout OK
procreg    2018-06-11 17:27:49.052

proccod    76
procname   Consulta_Mensajes_Ticket
procparamin T000000001
procparamout OK
procreg    2018-06-11 17:27:49.053

proccod    77
procname   Baja_Mensaje_Ticket
procparamin T000000001//MS0000555
procparamout NO EXISTE EL ESTADO DEL TICKET T000000001 en la tabla
Message
procreg    2018-06-11 17:28:43.630

proccod    78
procname   Consulta_Mensajes_Ticket
procparamin T000000001
procparamout OK
procreg    2018-06-11 17:28:43.632

65 row(s) retrieved.

```

## 16. ANEXO H: Generación de scripts SQL del repositorio estadístico

### Creación de la estructura del repositorio estadístico

```

dbaccess -e GSD@prod_tcp PROTOTIPO_ESTRUCTURA_ESTADISTICO.sql
2>&1| tee -a PROTOTIPO_ESTRUCTURA_ESTADISTICO.log

-----
-----
-- VICTOR MANUEL MIÑAMBRES CHAMORRO.
-----
-----
-- TFG ----- GLOBAL SERVICE DESK -----
-----

```



```
-----  
-----  
-- Se crea el esquema para el repositorio estadístico.  
-----  
-----
```

```
drop table timeSollnc;
```

```
CREATE TABLE GSD.timeSollnc
```

```
(  
  cod_timeSollnc SERIAL,  
  month_tsi      INTEGER,  
  timeAcu       INTEGER,  
  timeSol       VARCHAR(50),  
  priority      VARCHAR(50),  
  numInc       INTEGER
```

```
) extent size 200 next size 64 lock mode row;  
Table created.
```

```
revoke all on timeSollnc from "public" as "informix";  
Warning:Privilege not revoked.
```

```
create unique index ix_timeSollnc on timeSollnc (cod_timeSollnc) using btree  
in cycprod_idx1;  
Index created.
```

```
alter table timeSollnc add constraint primary key (cod_timeSollnc) constraint  
pk_timeSollnc;  
Table altered.
```

```
-----  
-----  
Database closed.
```

### Creación de los datos iniciales del repositorio estadístico

```
dbaccess -e GSD@prod_tcp PROTOTIPO_DATOS_ESTADISTICO.sql 2>&1|  
tee -a PROTOTIPO_DATOS_ESTADISTICO.log
```

Database selected.

-----  
-----  
-- VICTOR MANUEL MIÑAMBRES CHAMORRO.  
-----

-----  
-- TFG ----- GLOBAL SERVICE DESK -----  
-----

-----  
-- Se insertan los datos para crear un prototipo de pruebas.  
-----

-----  
--delete from timeSollnc;  
-----

-----STICO-----  
-----

-----  
-- Se insertan los valores iniciales en las tablas de estadística (timeSollnc)

insert into timeSollnc values (0,'1','0','0','P1',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'2','0','0','P1',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'3','0','0','P1',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'4','0','0','P1',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'5','0','0','P1',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'6','0','0','P1',0);  
1 row(s) inserted.

```
insert into timeSollnc values (0,'7','0','0','P1',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'8','0','0','P1',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'9','0','0','P1',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'10','0','0','P1',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'11','0','0','P1',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'12','0','0','P1',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'1','0','0','P2',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'2','0','0','P2',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'3','0','0','P2',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'4','0','0','P2',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'5','0','0','P2',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'6','0','0','P2',0);  
1 row(s) inserted.
```

insert into timeSollnc values (0,'7','0','0','P2',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'8','0','0','P2',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'9','0','0','P2',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'10','0','0','P2',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'11','0','0','P2',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'12','0','0','P2',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'1','0','0','P3',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'2','0','0','P3',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'3','0','0','P3',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'4','0','0','P3',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'5','0','0','P3',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'6','0','0','P3',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'7','0','0','P3',0);

1 row(s) inserted.

insert into timeSollnc values (0,'8','0','0','P3',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'9','0','0','P3',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'10','0','0','P3',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'11','0','0','P3',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'12','0','0','P3',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'1','0','0','P4',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'2','0','0','P4',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'3','0','0','P4',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'4','0','0','P4',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'5','0','0','P4',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'6','0','0','P4',0);  
1 row(s) inserted.

insert into timeSollnc values (0,'7','0','0','P4',0);  
1 row(s) inserted.

```
insert into timeSollnc values (0,'8','0','0','P4',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'9','0','0','P4',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'10','0','0','P4',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'11','0','0','P4',0);  
1 row(s) inserted.
```

```
insert into timeSollnc values (0,'12','0','0','P4',0);  
1 row(s) inserted.
```

```
Database closed.
```

## Creación de los procedimientos del repositorio estadístico

```
dbaccess -e GSD@prod_tcp PROTOTIPO_PROCEDURES_ESTADISTICO.sql  
2>&1| tee -a PROTOTIPO_PROCEDURES_ESTADISTICO.log
```

```
egrep -i "Routine created."  
PROTOTIPO_PROCEDURES_ESTADISTICO.log|wc -l  
1
```