

Proyecto DeGiraTracker

Autora: Lucía Laura Zuleta.

Fecha Creación: 8/8/2025

| Control de versiones | | | |
|----------------------|--------------|---|------------------------|
| Versión | Autor | Modificaciones | Fecha de actualización |
| 1.1 | Lucía Zuleta | Se agrega los puntos de definición de Situación Problemática, Modelo de negocio | 10/08/2025 |
| 1.2 | Lucia Zuleta | <ul style="list-style-type: none">Se quitan FK del listado de tablas para evitar redundancia.Se actualiza el DER con las relaciones que se quitaron. | 11/08/2025 |
| 1.3 | Lucia Zuleta | <ul style="list-style-type: none">Se agregan al proyecto Vistas, SP, Funciones, TriggersSe actualiza el DER | 05/09/2025 |

1. introducción:

Dentro del presente documento se proporcionará la solución de la base de datos del proyecto “DeGiraTracker”, una herramienta de gestión de tickets orientada a registrar, asignar y dar seguimiento a solicitudes o incidencias en múltiples proyectos.

2. Objetivo:

Desarrollar e implementar el sistema “DeGiraTracker”, una herramienta de gestión de tickets inspirada en las funcionalidades de Jira, que permita registrar, asignar y dar seguimiento a solicitudes o incidencias, contemplando usuarios, roles, información esencial de cada ticket y reglas de configuración. El objetivo es centralizar la información, facilitar la resolución eficiente de problemas, generar reportes estratégicos (como tiempos de resolución, productividad por persona y tipos de tickets más frecuentes) y garantizar que ciertas acciones se realicen únicamente bajo condiciones predefinidas, mejorando así la trazabilidad y la toma de decisiones.

3. Situación Problemática:

Actualmente la gestión de tickets y seguimiento de tareas se realiza de forma fragmentada, es decir utilizando planillas tipo Excel, mails y herramientas que no se encuentran integradas. Esto provoca pérdida de información, duplicidad, dificultad para hacer seguimiento en tiempo real y limita la trazabilidad de las acciones que realizan los usuarios. Además no existe un control eficiente de permisos y roles, lo que limita la seguridad y dificulta asignar responsabilidades claras.

La implementación de una base de datos de tipo relacional permite centralizar toda la información garantizando la integridad de los datos, optimizar la asignación de tareas y facilitar la generación de

reportes estratégicos, reduciendo significativamente los tiempos de respuesta y mejorando la coordinación entre distintos equipos de trabajo.

4. Modelo de negocio:

La organización que utiliza esta solución puede ser una empresa de prestación de servicios tecnológicos que desarrolla y gestiona múltiples proyectos de forma simultánea para clientes internos y externos. Su estructura se compone de diferentes áreas funcionales como soporte técnico, desarrollo de software, gestión de proyectos y administración, cada una con usuarios que desempeñan roles específicos y niveles de permisos diferenciados.

La operación diaria requiere la recepción, registro, asignación y seguimiento de tickets que representan solicitudes, incidencias o tareas. Cada ticket está asociado a un proyecto, cuenta con un responsable asignado y un estado de avance, y puede involucrar la interacción de varios equipos. Actualmente, la coordinación entre áreas se dificulta debido a la dispersión de la información y la falta de un sistema centralizado.

El modelo de negocio propuesto centraliza la gestión de tickets en una base de datos relacional que permite:

- ✓ Registrar y clasificar solicitudes según tipo, prioridad y proyecto.
- ✓ Asignar responsables y establecer fechas de resolución.
- ✓ Mantener un historial de cambios y comunicaciones.
- ✓ Generar métricas y reportes para la toma de decisiones.

De esta forma, la organización optimiza la utilización de recursos, mejora la trazabilidad de sus procesos y asegura una comunicación eficiente entre los equipos involucrados.

5. Listado de tablas:

A continuación, se listan todas las tablas que forman parte del proyecto. Tener en cuenta que el orden del listado es el mismo que el orden de ejecución con fin de poder identificarlas de forma ágil en el script.

Nota: Las tablas con encabezado azul están definidas como tablas intermedias.

| Tabla Proyectos: Dentro de esta tabla se definiran los proyectos que existen en la herramienta sobre los cuales se puede trabajar. Esta tabla es importante ya que al generar un ticket podremos saber a que proyecto pertenece | | | | | | |
|--|---------------|----------------|-------------|-------------|--------|--|
| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | Índice | Descripción |
| id_proyecto | int | si | si | no | N/A | Identificador del proyecto |
| nombre_proyecto | varchar (100) | si | no | no | N/A | Nombre que se visualizará en el ticket |
| cliente | varchar (150) | si | no | no | N/A | Persona/empresa owner del proyecto |

| | | | | | | |
|--------------|----------|----|----|----|-----|--|
| fecha_inicio | datetime | si | no | no | N/A | Fecha en la que inició el proyecto |
| activo | booleano | si | no | no | N/A | Indica si el proyecto se encuentra activo o dado de baja |

| Tabla Usuarios: En esta tabla se guardarán los usuarios de la aplicación y sus credenciales de acceso | | | | | | |
|--|--------------|----------------|-------------|-------------|--------|---|
| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign Key | Índice | Descripción |
| id_usuario | INT | SI | SI | NO | N/A | id de cada usuario registrado |
| usuario | VARCHAR (30) | SI | NO | si | N/A | Usuario con el cual se va a loguear la persona en la aplicación |
| contrasenia | VARCHAR (60) | SI | NO | NO | N/A | contraseña del usuario (deberia se un dato encriptado) |
| activo | booleano | si | no | no | N/A | Este campo nos indica si el usuario se encuentra activo y puede utilizar la aplicación o no. Con este campo podremos hacer una baja logica, lo que nos permitirá tener registro de los usuarios que en algun momento accedieron a la aplicación, permitiendo no solo un control, sino tambien tener información sobre acciones que haya realizado |
| fecha_creacion | DATETIME | SI | NO | NO | N/A | fecha de creacion del usuario |
| ultima_modificacion | DETETIME | NO | NO | NO | N/A | fecha de la ultima modificacion del usuario |

| Tabla Roles: Dentro de esta tabla se encontrarán los roles definidos en la aplicación para asignar a los usuarios dependiendo de las necesidades que tengan para trabajar. | | | | | | |
|---|--------------|----------------|-------------|-------------|---------|-----------------------|
| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | Indices | Descripción |
| id_rol | int | si | si | no | N/A | Identificador del rol |

| | | | | | | |
|-------------|-----------------|----|----|----|-----|---|
| nombre_rol | varchar (50) | si | no | no | N/A | Nombre del rol |
| detalle_rol | varchar(300) | no | no | no | N/A | Descripción del rol, se puede agregar un detalle más específico sobre el rol |
| activo | Booleano | si | no | no | N/A | Nos indica si el rol se encuentra disponible en la aplicación. Este campo sirve para poder realizar una baja lógica en vez de física, en caso de que no sea requerido utilizarlo más. Por defecto este campo deberá contener un CERO (0). Esto también evitará que al hacer una baja lógica se tengan que quitar todas las relaciones con los permisos existentes |

Tabla Permisos: Dentro de esta tabla se encontraran los permisos existentes en la aplicación, creación, modificación, accesos a distintas funcionalidades etc. La finalidad es que estos permisos se puedan agrupar en un rol, para al momento de generar un nuevo usuario y asignarle el rol, que automáticamente cuente con los permisos necesarios en vez de asignarlos uno por uno

| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | | Descripción |
|---------------------|------------------|----------------|-------------|-------------|-----|---|
| id_permiso | int | si | Si | no | N/A | identificador del permiso |
| nombre_permiso | varchar (100) | si | no | no | N/A | Nombre del permiso |
| url | varchar (300) | si | no | no | N/A | url que dirige al código donde se encuentra programado el permiso |
| activo | booleano | si | no | no | N/A | Este campo indica si el permiso se encuentra activo o no. Por defecto deberá estar activo. |
| fecha_creacion | datetime | si | no | no | N/A | Fecha de creación del permiso |
| modificado_por | int | no | no | si | N/A | Usuario que realizó modificaciones sobre el permiso, si las tuviera. Se relaciona con la tabla usuarios |
| ultima_modificacion | datetime | no | no | no | N/A | Fecha de la última modificación realizada sobre el permiso |

Tabla Prioridades: Dentro de esta tabla se encontrarán definidas las distintas prioridades que puede tener un ticket

| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | Índice | Descripción |
|------------------|--------------|----------------|-------------|-------------|--------|------------------------------|
| id_prioridad | int | si | si | no | N/A | Identificado de la prioridad |
| nombre_prioridad | varchar (30) | si | no | no | N/A | Nombre de la prioridad |

Tabla Tipos incidencias: Dentro de esta tabla se encontrarán definidos los tipos de incidencias que puede tener un ticket, por ejemplo si es una consulta, si es un requerimiento, si es soporte, etc

| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | Índice | Descripción |
|-------------------------|---------------|----------------|-------------|-------------|--------|--|
| id_tipo_incidencia | int | si | si | no | N/A | Identificador del tipo de incidencia |
| nombre_tipo_incidencia | varchar (30) | si | no | no | N/A | Nombre del tipo de incidencia. |
| detalle_tipo_incidencia | varchar (300) | no | no | no | N/A | Más información sobre el tipo de incidencia. |

Tabla Estados: Dentro de esta tabla se encontrarán definidos los estados existentes por lo cuales puede pasar un ticket

| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | Índice | Descripción |
|----------------|--------------|----------------|-------------|-------------|--------|---|
| id_estado | int | si | si | no | N/A | Identificador del estado |
| nombre_estado | varchar (50) | si | no | no | N/A | Nombre del estado (El que aparecerá en la aplicación) |
| detalle_estado | varchar(300) | no | no | no | N/A | Información adicional sobre el estado |

Tabla Tipos de cambio: Dentro de esta tabla se registrarán los tipos de cambio que es posible hacer en un ticket, por ejemplo cambiar la persona asignada, cambiar el estado, etc

| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | Indice | Descripción |
|--------------------|---------------|----------------|-------------|-------------|--------|---------------------------------------|
| id_tipo_cambio | int | si | si | no | N/A | Identificador del tipo de cambio |
| nombre_cambio | varchar (50) | si | no | no | N/A | Nombre del cambio |
| descripcion_cambio | varchar (300) | no | no | no | N/A | Información adicional sobre el cambio |

Tabla Detalle personas: En esta tabla se guardará la información adicional de cada uno de los usuarios de la aplicación

| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | Índice | Descripción |
|--------------------|--------------|----------------|-------------|-------------|--------|---------------|
| id_detalle_persona | int | si | si | no | N/A | identificador |

| | | | | | | |
|------------|--------------|----|----|----|-----|--|
| id_usuario | int | si | no | si | N/A | se relaciona con la tabla USUARIOS, y se referencia a que usuario corresponde la informacion |
| nombre | varchar (30) | si | no | no | N/A | Nombre del usuario |
| apellido | varchar (30) | si | no | no | N/A | Apellido del usuario |
| mail | varchar (50) | si | no | no | N/A | Mail del usuario |

| Tabla usuarios_x_rol: Es una tabla intermedia que relaciona los usuarios con los roles que tendrá cada uno | | | | | | |
|--|--------------|----------------|-------------|-------------|---|--|
| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | Índice | Descripción |
| id_usuario_x_rol | int | si | si | no | N/A | Identificador de la relacion de usuario y rol asignado |
| id_usuario | int | si | no | si | Nombre Índice: idx_usuarios_rols (Es indice compuesto de id_usuario y id_rol) | Identificador del usuario que contendrá los roles |
| id_rol | int | si | no | si | Nombre indice: idx_usuarios_rols (Es indice compuesto de id_usuario y id_rol) | Identificador del rol que tendrá cada usuario |

| Tabla roles_x_permisos: Esta es una tabla intermedia, donde se definirá la relacion entre los roles cargados en la tabla "Roles" y los permisos cargados en la tabla "Permisos" | | | | | | |
|---|--------------|----------------|-------------|-------------|--------|---|
| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | Índice | Descripción |
| id_rol_x_permiso | int | si | si | no | N/A | Identificador de la relacion de rol y permiso |

| | | | | | | |
|------------|-----|----|----|----|---|--|
| id_rol | int | si | no | si | Nombre indice: idx_rols_permisos (Es indice compuesto de id_rol, id_permiso) | Id del rol al cual se le van a cargar los permisos |
| id_permiso | int | si | no | si | Nombre indice: idx_rols_permisos (Es indice compuesto de id_rol, id_permiso) | id del permiso que se asignará a determinado rol |

| Tabla Reglas SLA: Dentro de esta tabla se encontrarán definidas las reglas de SLA, tiempos de resolucion. Dependiendo del proyecto al que pertenezca, el tipo de ticket que sea, y la prioridad que tenga, se va a definir el maximo tiempo de resolución que debería tener el ticket. Esto servirá principalmente para sacar reportes, verificar los tiempos reales de resolución, para aplicar las estrategias necesarias y poder hacer el cumplimiento correcto | | | | | | |
|--|------------------|-------------------|----------------|----------------|--------|--|
| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | Índice | Descripción |
| id_regla_sla | int | si | si | no | N/A | Identificador de la regla |
| id_proyecto | int | si | no | si | N/A | Identificador del proyecto al cual se va a asignar la regla |
| id_prioridad | int | si | no | si | N/A | Identificador de la prioridad que deberá tener el ticket para que la regla se cumpla |
| id_tipo_incidencia | int | si | no | si | N/A | Identificador del tipo de ticket que deberá ser para que la regla se cumpla |
| tiempo_resolucion | decimal(3, 2) | si | no | no | N/A | Tiempo maximo que deberá llevar la resolución, en base al proyecto, la prioridad del ticket, y el tipo de ticket |
| activo | boolean | si | no | no | N/A | Este campo indica si la regla se puede utilizar o no, la finalidad es evitar el borrado físico para no tener inconsistencias. |

| Tabla: Tickets: Dentro de esta tabla se guardaran todos los tickets existentes en la aplicación. | | | | | | |
|---|-----------------|-------------------|----------------|----------------|--------|-------------|
| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | Índice | Descripción |

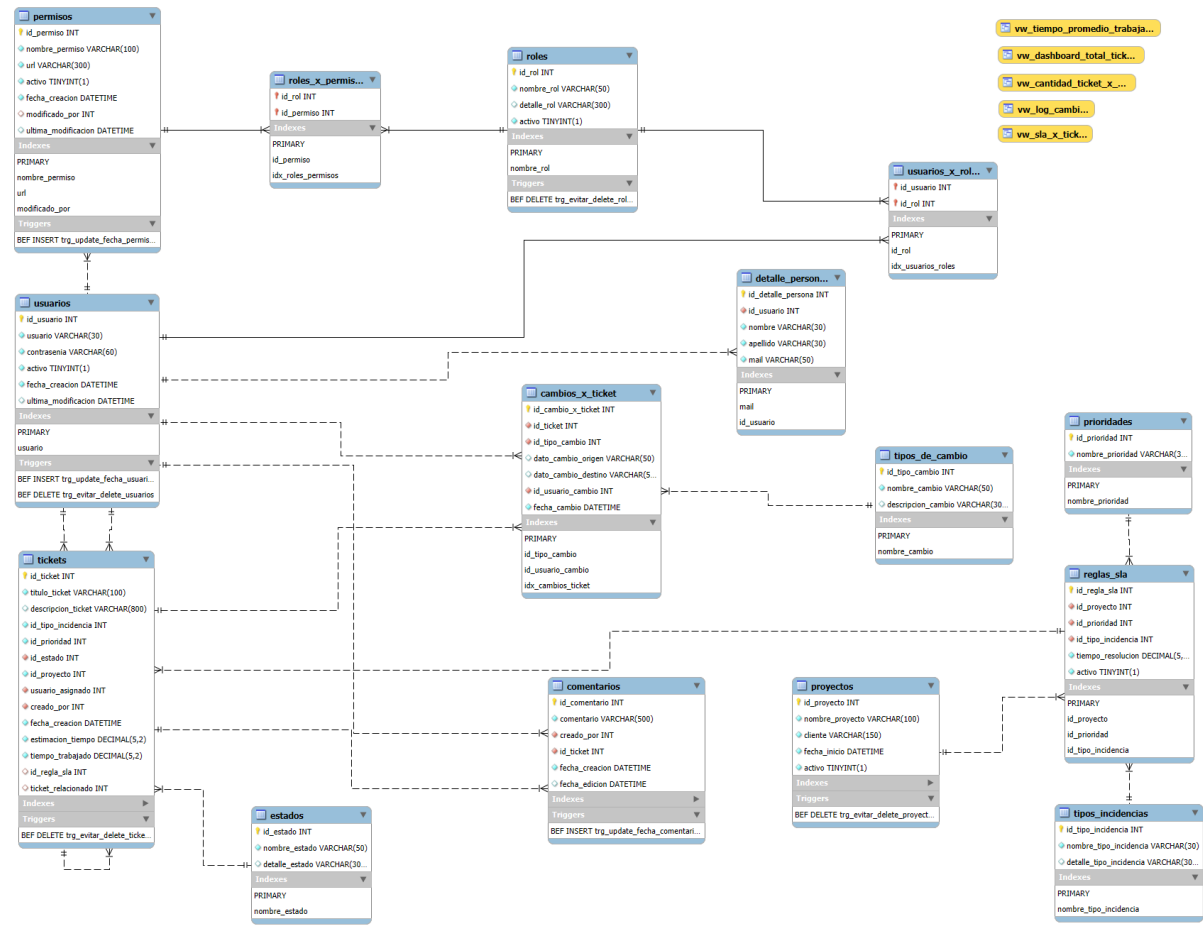
| | | | | | | |
|--------------------|---------------|----|----|----|---|---|
| id_ticket | int | si | si | no | N/A | Identificador del ticket |
| titulo_ticket | varchar (100) | si | no | no | N/A | Titulo que el usuario le ingresa al ticket |
| descripcion_ticket | varchar (800) | no | no | no | N/A | Descripción del ticket |
| id_tipo_incidencia | int | si | no | no | Nombre Índice: idx_tickets_tipo_incidencia | Id del tipo de incidencia. Nos indicará de que tipo es |
| id_prioridad | int | si | no | no | Nombre Índice: idx_tickets_prioridad | Id de la prioridad, el ticket tendrá esta prioridad |
| id_estado | int | si | no | si | Nombre Índice: idx_tickets_estado | identificador del estado que tiene actualmente el ticket |
| id_proyecto | int | si | no | no | Nombre Índice: idx_tickets_proyecto | identificador del proyecto al que pertenece el ticket |
| usuario_asignado | int | si | no | si | Nombre Índice: idx_tickets_usuario_asignado | Se relaciona con la tabla usuarios. Nos indica quien es la persona que tiene que trabajar sobre el ticket |
| creado_por | int | si | no | si | N/A | Id del usuario que generó el ticket |
| fecha_creacion | datetime | si | no | no | N/A | Fecha en la cual se creó el ticket |
| estimacion_tiempo | decimal (3,2) | no | no | no | N/A | Cantidad de tiempo que se estima que va a llevar la resolución del ticket |
| tiempo_trabajado | decimal (3,2) | no | no | no | N/A | Tiempo real que llevó la resolución del ticket. (Servirá para hacer comparar con lo establecido en los SLA) |
| id_regla_sla | int | no | no | si | N/A | Id de la regla de SLA que aplica para determinado ticket. |
| ticket_relacionado | int | no | no | si | N/A | En caso de que esté relacionado con otro ticket, en este campo se pondrá el id del ticket relacionado. |

Tabla Comentarios: Dentro de esta tabla se guardarán los comentarios realizados en los tickets

| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | Índice | Descripción |
|----------------|---------------|----------------|-------------|-------------|---------------------------------------|---|
| id_comentario | int | si | si | no | N/A | Identificador del comentario |
| comentario | varchar (500) | si | no | no | N/A | Comentario ingresado en el ticket |
| creado_por | int | si | no | si | N/A | Usuario que creó el comentario |
| id_ticket | int | si | no | si | Nombre Índice: idx_comentarios_ticket | Id del ticket al cual pertenece el comentario |
| fecha_creacion | datetime | si | no | no | N/A | Fecha en la que se creó el comentario |
| fecha_edicion | datetime | no | no | no | N/A | Fecha en la que se editó el comentario |

| Tabla cambios_x_ticket: Dentro de esta tabla se guardará el historico de cambios los tickets | | | | | | |
|--|--------------|----------------|-------------|-------------|-----------------------------------|---|
| Campo | Tipo de dato | Es obligatorio | Primary key | Foreign key | | Descripción |
| id_cambio_x_ticket | int | si | si | no | N/A | Identificador del cambio |
| id_ticket | int | si | no | si | Nombre Índice: idx_cambios_ticket | Ticket en el que se realizó el cambio (Se relaciona con la tabla tickets) |
| id_tipo_cambio | int | si | no | si | N/A | Id del tipo de cambio realizado (Se relaciona con la tabla Tipos_de_cambio) |
| dato_cambio_origen | varchar (50) | no | no | no | N/A | En caso de que sea una actualización de información, se registrará el dato que fue modificado |
| dato_cambio_destino | varchar (50) | no | no | no | N/A | Dato que fue insertado al realizar el cambio |
| id_usuario_cambio | int | si | no | si | N/A | Usuario que realizó el cambio |
| fecha_cambio | datetime | si | no | no | N/A | Fecha en la que se realizó el cambio |

6. Diagrama de entidad relacion (DER):



7. Definición de vistas.

Dentro del proyecto se generaron cinco vistas.

- **vw_cantidad_ticket_x_usr:** Esta vista nos indica la cantidad de tickets que tiene asignado cada usuario en todos los estados, es decir que se incluyen los tickets que están pendientes, en curso, y finalizados.
Tablas Intervinientes: tickets
- **vw_tiempo_promedio_trabajado:** Esta vista nos indica el tiempo promedio de trabajo sobre cada uno de los tickets existentes. Incluye los tickets en todos los estados existentes.
Tablas Intervinientes: tickets
- **vw_log_cambios:** Esta vista genera un log completo sobre los cambios en los tickets, dado que según el DER tenemos seteada la información de los cambios en tablas separadas (cambios por tickets, tipos de cambios, usuarios, proyectos, etc.). Esta vista nos permite consultar de forma completa el histórico de cambios sobre los tickets agrupando todos los datos juntos.
Tablas Intervinientes: tickets, cambios_x_ticket, tipos_de_cambio, tipos_incidencias, usuarios, proyectos.
- **vw_sla_x_ticket:** Dentro de esta vista se puede visualizar cada uno de los tickets existentes junto con la regla de SLA configurada que le corresponde, el tiempo real trabajado sobre el ticket, el tiempo de resolución que le corresponde según el SLA y si cumple con la regla o no. La finalidad de esta vista, es obtener el cumplimiento de forma rápida para poder hacer un mejor seguimiento sobre los tiempos de resolución.

Tablas Intervinientes: tickets, reglas_sla, tipos_incidencias, prioridades, estados, proyectos, usuarios.

- **vw_dashboard_total_tickets:** Esta vista representa un tablero (Ref. Tablero de Kanban), donde se muestra el título de cada ticket en la columna del estado en el que se encuentra actualmente, el tiempo de resolución según la regla de sla que debe cumplir, si cumple con el tiempo del sla y en caso de que no cumpla con el tiempo configurado, muestra la cantidad de horas de diferencia que hay con respecto a la configuración de la regla.

Tablas Intervinientes: tickets, usuarios, reglas_sla, proyectos.

8. Definición de funciones.

Dentro del proyecto se agregaron las siguientes funciones:

- **fn_tiempo_restante_ticket:** Esta función sirve para calcular el tiempo de resolución restante de los tickets con respecto a la regla de SLA configurada. Se ingresa como parámetro el id del ticket.
- **fn_tiempo_promedio_resolucion:** Esta función tiene el objetivo de calcular el tiempo promedio de resolución de los tickets en base al proyecto. Se ingresa como parámetro el Id del proyecto, y se calcula el promedio de tiempo de resolución. Esta función incluye todos los tickets existentes, es decir que no importa en que estado se encuentren actualmente.

9. Definición de Store Procedures.

Dentro del proyecto se agregaron los siguientes SP:

- **sp_cierre_ticket:** Se generó este ticket para facilitar el cambio de estado de los tickets a “Cerrado”, se valida que el ticket afectado deberá encontrarse en el estado “En validación” antes de ser cerrado. En caso de que el ticket no cumpla con el estado solicitado, se mostrará un mensaje indicando que el ticket se encuentra en un estado incorrecto y no se puede cerrar. Recibe como parámetro de entrada el Id del ticket.

Tablas Intervinientes: Tickets

- **sp_cancela_ticket:** Este SP tiene la función de pasar tickets a estado cancelado, en este caso se valida que el ticket no esté en estado “Cerrado”, ya que sería el único estado en el que no se puede cancelar. Recibe como parámetro el Id del ticket.

Tablas Intervinientes: Tickets

- **sp_tickets_vencidos:** Este SP realiza una consulta sobre los tickets que se encuentran vencidos, es decir que el tiempo trabajado superó el tiempo de resolución según lo configurado en las reglas de SLA. Recibe como parámetro el Id del proyecto el cual se quiere consultar, y devuelve el/los registros de la tabla tickets que no cumplieron las reglas configuradas.

Tablas Intervinientes: tickets, reglas_sla

10. Definición de Triggers.

En este caso se generaron dos triggers que fueron implementados en varias tablas, ya que se requería aplicar la misma funcionalidad en varias tablas.

- **trg_update_fecha:** Actualiza la fecha de modificación al hacer un update sobre uno o varios registros. Aplicado en las siguientes tablas: comentarios, permisos, usuarios
- **trg_evitar_delete:** Evita que se borren registros de las siguientes tablas: proyectos, tickets, usuarios, roles.