**SISTEMA DE TICKETS PARA SOPORTE TECNICO DE TEXIN**

Sistema de Software para control y seguimiento de tickets de soporte técnico del área de sistemas de la empresa Texin FR.

Dicho sistema tiene la capacidad de escalar para llevar tickets del área de ventas, compras, Rh y demás áreas que confirmen la empresa.

Dentro del sistema un usuario logeado (Para la primera etapa del proyecto ya que a posterior puede ser un usuario no legeado, considerando algún sistema de identificación y que no existan tickets duplicados), abre un ticket especificando la categoría de su problema, una descripción del mismo, así como alguna imagen o fotografía adjunta para tener un punto de referencia (Opcional).

El ticket levantado se asignará directamente al agente preferente, que en el caso será el encargado del departamento de soporte técnico.

En caso que él no pueda atenderlo podrá reasignarlo a algún otro agente, esto se toma con base a las categorías que seleccione el usuario.

Si el usuario desconoce la categoría para levantar su ticket existirá una categoría de otros, la cual será asignada al administrador de sistema que a su vez podrá solucionarla o bien reasignarla a un agente que encaje en la categoría del problema, actualizará la categoría y el agente asignado.

El usuario una vez levantado el ticket recibirá notificaciones vía e-mail o vía notificación de su app de escritorio o móvil, cada que exista una actualización del ticket, el mismo contará con diferentes estados de vida.

El usuario tendrá un chat abierto con el agente que tenga asignado, esto si la solución es de manera remota, el tipo de solución al problema se podrá designar como remoto o en sitio.

Un usuario podrá tener diferentes tickets siempre y cuando no pertenezcan a la misma categoría. La categoría llamada **otros,** dará una máximo de 5 tickets por usuario y el agente o administrador asignado tendrá la opción de cerrar dicho ticket por duplicidad del problema.

Un usuario tendrá la posibilidad de reabrir un ticket ya cerrado siempre y cuando sea parte del mismo problema que no se solucionó, dicha solicitud llegara al administrador del sistema y al agente que le dio solución, en caso de validar que el problema persiste el ticket tendrá un estado de reabierto.

Cada usuario logeado tendrá un listado de sus tickets creados y las soluciones que se tuvieron, esto para poder consultarlos en caso de presentar un problema similar y poder encontrar la solución por su cuenta, cada usuario llevara una estadística de los tickets creados y las causas.

Se creará una base de conocimiento con las soluciones realizadas por los agentes, tomando la pregunta que realizo el usuario y los pasos que realizo el agente para solucionar el problema, cuando el usuario pueda solucionar su problema con la base de soluciones deberá indicarlo.

Dicha solución debe ser redactada con todo detalle por el agente que le dio solución, en caso de que sea una situación muy especifica el agente podrá optar por no agregarla a la base de soluciones dicha acción será evaluada por el administrador del sistema.

Cuando el usuario cree un ticket de acuerdo con la categoría que selecciono y la descripción inicial del problema se le presentaran las posibles soluciones que se encuentren disponibles en la base de no existir alguna para su problema seleccionara que no existe solución similar y se procederá a levantar el ticket y asignar al agente a resolverlo.

Cada Ticket tendrá un Área que para esta etapa será **Sistemas** (posteriormente se agregarán las demás áreas), seguido de una categoría (Sistema Operativo, Red, Internet, Impresoras, Programas, SAP, Hardware, Otros), dependiendo de las situaciones que presenten los usuarios las categorías se podrán agregar, Fecha y Hora de Apertura, Fecha y hora de primer contacto, Fecha y hora de solución, en caso de aplicar Fecha y hora de Reapertura, Fecha y hora de cierre final, tiempo de solución, estatus (Abierto, Reabierto, Cerrado) con un sub estatus para cada uno (Abierto : en espera de respuesta, en proceso; Reabierto: En espera de reapertura, en proceso; Cerrado: Solucionado por Agente, Sin solución, Solucionado por Base de soluciones Usuario, Solucionado por Base de soluciones Agente).

Cada agente podrá tener uno o más roles al igual el agente podrá levantar tickets con el administrador del sistema o agentes de otras áreas o la suya.

El agente tendrá la capacidad de reasignar un ticket especificando la causa de la reasignación, tendrá dentro de su panel de administración, los tickets abiertos y el estatus de los mismos, dicho agente podrá consultar al igual su base de conocimiento para tomar en cuenta las posibles soluciones a los tickets que tenga abiertos, de encontrar alguna se la presentará al usuario y si el problema se soluciona lo marcara como solución por base de soluciones. Igualmente podrá ver los tickets que tenga cerrados y el tiempo de solución de los mismos, cuando un agente tenga una nueva asignación llegará una notificación a su aplicación móvil o de escritorio, en caso de que un ticket requiera solución en sitio el agente deberá seleccionarlo.

Cada Agente tendrá un tiempo promedio de respuesta a los tickets, la cantidad de tickets que se solucionaron en sitio, la cantidad que se solucionaron vía remota, tiempo promedio de solución. Aportaciones a la base de soluciones.

El administrador de sistema tendrá la capacidad de crear nuevos usuarios, consultar las analíticas de cada agente, cada usuario, podrá asignar roles de agente o usuario y para qué área, podrá eliminar usuarios, tendrá la capacidad de entrar a un panel de agente donde tendrá todas las funciones de un agente y una opción de usuario donde tendrá las opciones de levantar tickets como usuario.

El administrador podrá reasignar tickets y recibirá notificaciones de los tickets que se encuentren activos, el mismo tendrá rol de analista para poder descargar los datos de cada agente en un reporte.

Analista de datos, el analista de datos podrá ver la cantidad de usuarios que levantaron tickets por tiempo determinado. La categoría que más tickets género, los agentes que se encuentran disponibles, las estadísticas de cada agente. Tiempo de respuesta general y tiempo de respuesta por agente y categoría, numero de respuestas en la base de soluciones, número de casos solucionados por agente y por la base de soluciones.

Usuarios registrados y numero de tickets creados por usuario.

Podrá descargar un reporte del uso del sistema, con el número de tickets creados en un tiempo determinado.

Los reportes generados podrán contener número de tickets abiertos, solucionados, sin solución, tiempo de primer contacto, tiempo de solución, tipo de solución, numero de tickets reabiertos, esto por categoría, departamento, usuario, agente.

//

Usuarios solo con correos institucionales.

Patronaje 3 personas

Trazo 2 personas

Bordado

Usuario con opción

Almacén MP 3 personas.

Red

Enfermería 2 personas.

Usuarios diferentes con mismo correo.

@maauad.com

//

**COMPONENTES**

El sistema contara con una aplicación de escritorio, una app móvil y una aplicación web, se piensa que todas las interfaces consuman una API RESTFUL la cual brindara toda la validación, información y funcionalidades, se utilizaran sockets para mantener la comunicación abierta entre el servidor y los clientes, a su vez la comunicación entre clientes.

Base de datos en MySQL la cual contendrá toda la información de los usuarios, tickets y de la que se podrán obtener analíticas.

Sesiones: los usuarios tendrán un username y una contraseña encriptada para poder ingresar al sistema, el método de autenticación será vía JWT, un usuario podrá tener sesión abierta en 2 dispositivos diferentes, ejemplo su app móvil y la aplicación de escritorio. Se utilizará un sistema de autenticación en 2 pasos para permitir el inicio de sesión en otro dispositivo.

Mensajes: El sistema permitirá un chat en tiempo real entre el usuario que levanto el ticket y el agente que le responde, al igual se enviaran notificaciones tanto a las apps como al correo electrónico una vez que exista una actualización del estado del ticket o alguna modificación en el mismo.

Aplicaciones: Las aplicaciones son únicamente como interfaz del usuario y el elegirá con cual interactuar, Desktop App, Web App, Móvil App, los agentes preferentemente deberán iniciar la Desktop App al iniciar su pc.

Reportes: El sistema tendrá una opción de exportación de reportes en formato xls para poder manipularlos en Excel, los únicos que podrán descargar los reportes serán los administradores y analistas de datos.

**ENTIDADES**

* Usuario
  + Consumidor
  + Empleado
    - Analista
    - Administrador
    - Agente
* Tipo de Usuario
* Equipos
* Teléfonos
* Ticket
* Ticket reabierto
* Roles
* Rol-Usuario
* Área
* Categoría
* Base de Soluciones

**Atributos de entidades**

**USUARIOS**

**Usuario**

* Identificador
* Tipo de usuario

**Personal o cliente:**

* Nombre completo
* Nombre de Usuario
* Correo electrónico
* Número de Empleado o No cliente
* Roles (RELACION)
* Contraseña
* Token (Generado desde el Back)
* Rol de Usuario

**Consumidor:**

* Nombre
* correo
* Token (se genera a partir del ticket)
* Ticket (relación)

**Teléfonos**

* Identificador de usuario
* Número telefónico

**Equipo**

* Identificador de usuario
* Número de serie

**ROLES**

**Consumidor CONSULTA DEL USUARIO RELACIONES:**

* Tickets creados
* Tickets abiertos
* Tickets cerrados

**Agente:**

* Área
* Posición del Agente
* Tickets asignados (Consulta bien se podría generar con una vista)
* Tickets abiertos (consulta bien se podría generar con una vista)
* Tickets cerrados (consulta bien se podría generar con una vista)
* Tiempo promedio de respuesta (consulta bien se podría generar con una vista)
* Tickets solucionados (consulta bien se podría generar con una vista)
* Tickets reasignados por el agente (consulta de la tabla Reasignaciones)

**Ticket:**

* Fecha de creación
* Área
* Categoría
* Título del problema
* Descripción del problema
* Prioridad asignada por el usuario (dicha prioridad se puede reasignar por el agente)
* Usuario que lo crea
* Agente asignado
* Conversación (se crea una tabla con todos los mensajes del agente y el usuario pudiendo el administrador mandar mensaje)
* Fecha de cierre
* Status.
* Tipo de Ticket (tipo 1 -ticket con apertura y cierre, tipo 2-ticket con reapertura y cierre final)

**Roles:**

* Nombre del Rol:
  + Agente
  + Administrador
  + Analista

**Categoría:**

* Área
* Nombre
* Descripción

**Área:**

* Nombre
* Descripción

**Base de Soluciones**

* Categoría
* Título de Problema
* Solución
* Agente que lo crea.
* Fecha de creación

**DICCIONARIO DE DATOS**

**Usuario**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| IdUser | Identificador único de usuario este irá relacionado con los tickets y demás funcionalidades dependiendo del tipo de usuario | int | 3 |
| fkUserType | Llave foránea del tipo de usuario. Tipo 1 consumidor  Tipo 2 empleado o cliente | Int | 3 |

**Tipos de Usuario**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| IdUserType | Identificador único del tipo de usuario | int | 3 |
| UserType | Nombre del tipo de usuario por ahora  Empleado  Cliente  Consumidor | Varchar | 10 |

**Consumidor**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| idConsumer | Identificador único del consumidor. | Int | 3 |
| fkUser | Llave foránea relaciona con el id único de usuario | Int | 3 |
| nameConsumer | Nombre completo de la persona. | Varchar | 50 |
| emailConsumer | Correo electrónico de contacto del consumidor que levanta el ticket | Varchar | 40 |

**Empleado**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATRIBUTO** | **DESCRIPCION** | **TIPO DE DATO** | **TAMAÑO** |
| idEmploye | Identificador único del empleado con total relación a su número de empleado | Int | 3 |
| fkUser | Llave foránea relacionada con el id único de usuario | Int | 3 |
| userName | Nombre de usuario para iniciar sesión en el sistema | Varchar | 15 |
| nameEmploye | Nombre completo | Varchar | 50 |
| emailEmploye | Correo electrónico institucional del empleado | Varchar | 40 |
| numberEmploye | Numero de empleado proporcionado por RH | Varchar | 10 |
| passwordEmploye | Contraseña del empleado cifrada en SH2 con una longitud de 256bits | Varchar | 70 |
| fkRole | Identificador del Rol al que puede pertenecer un Agente:  1.- Usuario  2.- Agente  3.- Analista  4.-Administrador  999.- Root  Es una llave de la tabla roles. | Int | 1 |

**Agente**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| IdAgent | Identificador Único del agente | Int | 3 |
| fkEmploye | Llave foránea relacionada con el id único de usuario | Int | 3 |
| fkArea | Llave foránea del área a la que pertenece el agente | Int | 3 |
| principalAgent | Binario para saber si un agente es el principal del área asignada o no | Int | 1 |

**Reasignaciones**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| IdReassignment | Identificador Único de la solicitud de reasignación | Int | 3 |
| fkAgentProvider | Identificador del agente al cual está asignado el ticket | Int | 3 |
| fkTicket | Identificador del ticket a reasignar | Int | 3 |
| fkAgenteReciver | Identificador del agente que recibirá el ticket | Int | 1 |
| status | Estado de aprobado o rechazado para la reasignación del ticket | Varchar | 10 |
| comments | Razones por las que se quiere reasignar el ticket | Text | 100 |
| dateRequest | Fecha de solicitud para la reasignación del agente | Timestamp | N/A |
| dateResponse | Fecha de respuesta a la solicitud | Timestamp | N/A |

**Equipos**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| idEquip | Identificador único del equipo dentro de la base de datos | Int | 3 |
| fkEmploye | Identificador del usuario al que está asignado el equipo | Int | 3 |
| serialNumber | Número de serie del equipo | Varchar | 20 |

**Teléfonos**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| idPhone | Identificador del número de teléfono | Int | 3 |
| fkUser | Idendtificador de usuario al que le pertenece el telefono |  |  |
| countryCode | Código de país para identificar el país del teléfono | Int | 3 |
| numberPhone | Número de teléfono a 10 dígitos sin espacios | Varchar | 10 |
| typeNumber | Identificador de tipo de número telefónico tendrá 2 valores por defecto  Fijo  Celular | Enum | 2 |

**Ticket**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| idTicket | Identificador único del ticket | Int | 5 |
| dateInitial | Fecha de creación del ticket | Timestamp | N/A |
| dateClose | fecha de cierre del ticket | Timestamp | N/A |
| ticketType | Tipo de ticket si es tipo 1 el ticket solo se abrió y cerro por única ocasión si es tipo 2 el ticket es reabierto | Int | 1 |
| fkArea | Llave foránea del Área a la que pertenece el ticket para esta etapa del desarrollo será Sistemas | Int | 3 |
| FkCathegory | Llave foránea de la categoría a la que pertenece el ticket | Int | 3 |
| titleTicket | Título del ticket con el que se identificará el problema | Varchar | 50 |
| description | Descripción del problema lo más específico posible | Text | 150 |
| fkUser | Llave foránea de usuario que crea el ticket | Int | 3 |
| fkAgent | Llave foránea del agente que está asignado al ticket | Int | 3 |
| Priority | Prioridad del ticket dicha prioridad será seleccionada por el usuario, pero el agente es quien determinara si es la correcta o podrá actualizarla | Enum(‘alta’,’Media’,’Baja’) | 3 |
| Status | Campo que determina el estatus del ticket | Enum (Abierto, Solucionado, Cerrado sin Solución, en proceso de reapertura, reabierto.) | N/A |
| Comments | Conversación que será guardada en un archivo json dentro de este campo se almacenará la dirección a dicho archivo. | Varchar | 30 |

**Ticket Reabierto**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| idReoppeing | Identificado único del ticket reabierto | Int | 3 |
| fkTicket | Llave foránea del ticket que será reabierto | Int | 3 |
| fkAgent | Llave foránea del agente que valida la reapertura del ticket | Int | 3 |
| Comments | Razones por las que se solicita la reapertura del ticket | Text | 100 |
| Status | Código de estatus para reaperturar el ticket dependiendo de las razones de reapertura, puede ser aprobado, rechazado | Enum (Aprobado, rechazado) | N/A |
| dateInitial | Fecha en la que se solicita la reapertura y posterior se actualiza si se reabre. | Timestamp | N/A |
| dateClose | Fecha en la que se cierra definitivamente el ticket | Timestamp | N/A |
| fkAdmin | Llave foránea del Administrador que da el segundo VoBo para la reapertura del ticket dependiendo de los comentarios para reapertura. | Int | 3 |
|  |  |  |  |

**Roles**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| idRole | Identificador del rol que puede tener un usuario registrado | Int | 3 |
| nameRole | Nombre del rol que se puede tener | Varchar | 20 |

**Área**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| idArea | Identificador del area que existe dentro de la empresa | Int | 3 |
| nameArea | Nombre del area  Ejemplo:  Sistemas  Factor Humano  Ventas  Compras | Varchar | 30 |

**Categoría**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| idCategory | Identificador de la categoría que puede tener un ticket | Int | 3 |
| nameCategory | Nombre de la categoría | Varchar | 20 |
| Description | Descripción de la categoría para ver los posibles casos en lo que se puede seleccionar  Ejemplo:  Nombre de Categoría: WIFI  Descripción:  Fallos de conexión WIFI  No reconoce la red.  No tengo señal de WIFI | Text | 50 |

**Base de Soluciones**

|  |  |  |  |
| --- | --- | --- | --- |
| ATRIBUTO | DESCRIPCION | TIPO DE DATO | TAMAÑO |
| idSolution | Identificador único para cada solución | Int | 3 |
| problemTitle | Título del problema resuelto | Varchar | 50 |
| Solution | Descripción de la solución con los pasos a seguir | Text | 300 |
| fkAgent | Llave foránea del agente que crea la solución | Int | 3 |
| dateInitial | Fecha de creación de la solución | TimeStamp | 3 |
| dateUpdate | Fecha de actualización de la solución esto si existe alguna modificación. | TimeStamp | N/A |

**Ingreso de Datos**

El ingreso de datos a la base debe ser respetando las relaciones dentro del modelado de datos, a continuacion se detalla como debe ser el ingreso de datos.

1.-

* UserTypes
* Area
* Roles
* Category

2.-

* Users

3.-

* Employes
* Consumers

4.-

* Phones
* Computers
* Agents

5.-

* Tickets
* Basesolution

6.-

* Reassignments
* reoppeningTicket

Siguiendo este orden de llenado es como se podra inicializar el sistema.

**NUEVO CONSUMIDOR**

Para insertar un nuevo consumidor llamamos a un procedimiento almacenado llamado InConsumer el cual recibe como parametros los datos del consumidor crea un nuevo usuario y asigna su Id al consumidor.

DROP PROCEDURE InConsumer;

CREATE DEFINER=`root`@`localhost` PROCEDURE `InConsumer`(IN nameConsumer varchar(50), IN emailConsumer varchar(40))

BEGIN

DECLARE exit HANDLER FOR SQLEXCEPTION

BEGIN

-- ERROR

SELECT "Ha Ocurrido un Error" as ERROR;

ROLLBACK;

END;

START TRANSACTION;

INSERT INTO users (fkUserType) VALUE (1);

INSERT INTO consumers (nameConsumer, emailConsumer, fkUser) values (nameConsumer, emailConsumer ,last\_insert\_id());

SELECT last\_insert\_id() AS IdConsumer;

COMMIT;

END

call InConsumer('Jesus', "jesusach92@gmail.com");

**NUEVO EMPLEADO**

Para insertar un nuevo empleado llamamos a un procedimiento almacenado llamado InEmploye el cual recibe como paramtros los datos del consumidor y crea un nuevo usuario y asigan el id al Consumidor.

drop procedure InEmploye;

DELIMITER $$

CREATE PROCEDURE InEmploye(IN userName varchar(15), IN nameEmploye Varchar(50), IN emailEmploye varchar(40),IN numberEmploye VARCHAR (10), passwordEmploye varchar(70), IN rol int )

begin

declare exit handler for SQLEXCEPTION

BEGIN

-- Error

SELECT "HA OCURRIDO UN ERROR" AS ERROR;

ROLLBACK;

END;

START transaction;

INSERT INTO users (fkUserType) VALUE (2);

INSERT INTO employes (userName, nameEmploye, emailEmploye, numberEmploye, passwordEmploye, fkUser, fkRole) values (userName,nameEmploye, emailEmploye,numberEmploye,passwordEmploye ,last\_insert\_id(), rol);

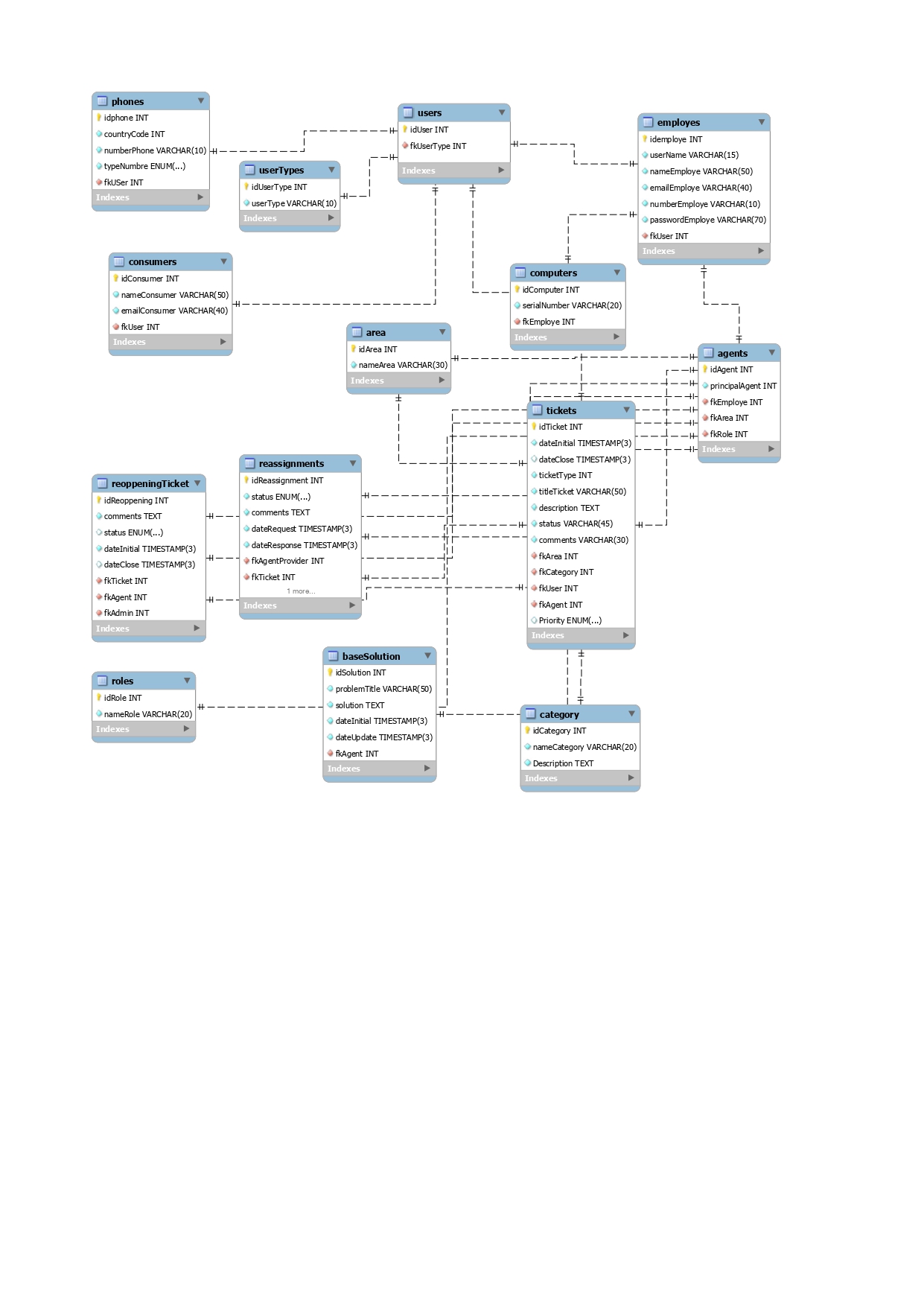
SELECT last\_insert\_id() AS IdEmploye;

COMMIT;

END $$

DELIMITER ;

CALL InEmploye("jesus","Jesus Alberto Castillo","jesus.castillo@maauad.com","t00900","123456") **MODELO ENTIDAD-RELACION**

****

**DESARROLLO DE API**

El Api que se desarrollara del lado del servidor será programada en JAVA, las funcionalidades que tendra serán las siguientes.

Autenticacion via JWT

Envio de informacion del usuario

Carga de modulos de acuerdo a su Rol

Edicion de Perfil

Listar Usuarios

Listar Agentes

Listar Consumers

Procedimiento almacenado para actualizar el nombre o correo de un consumidor.

Procedimiento para actualizar un Empleado

DELIMITER $$

CREATE PROCEDURE UpdateEmploye (IN EmployeId int, IN UnameUser varchar(15), IN UnameEmploye varchar(50), IN UemailEmploye varchar(40), IN UpasswordEmploye varchar(70), IN UfkRole int)

BEGIN

DECLARE msg text;

DECLARE errno int;

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

GET CURRENT DIAGNOSTICS condition 1

msg = message\_text, errno = MYSQL\_ERRNO;

SELECT errno AS NO\_ERROR, msg AS ERRO\_MESSAGE;

ROLLBACK;

END;

START TRANSACTION;

UPDATE employes SET nameUser=UnameUser, nameEmploye=UnameEmploye, emailEmploye=UemailEmploye, passwordEmploye=UpasswordEmploye, fkRole=UfkRole WHERE idemploye=EmployeId;

SELECT \* FROM employes WHERE idemploye=EmployeId;

COMMIT;

END $$

DELIMITER ;

Procedimiento almacenado para guardar un ticket

DELIMITER $$

CREATE PROCEDURE InTicket(IN initialDate timestamp(3),IN subjectTicket varchar(50) ,IN descriptionTicket text, IN areaFk INT UNSIGNED, IN categoryFk INT ,IN userFK INT UNSIGNED,IN agentFk INT UNSIGNED, IN initialPrority INT, IN updateDate timestamp(3))

BEGIN

DECLARE msg TEXT;

DECLARE errno int;

DECLARE exit HANDLER FOR SQLEXCEPTION

BEGIN

GET current diagnostics condition 1

msg = message\_text, errno= mysql\_errno;

-- ERROR

SELECT errno AS NO\_ERROR , msg AS MESSAGE;

ROLLBACK;

END;

START TRANSACTION;

INSERT INTO tickets(dateInitial,subject,description,fkArea,fkCategory,fkUser,fkAgent,Priority,dateUpdate)

VALUES (initialDate,subjectTicket,descriptionTicket,areaFk,categoryFk,userFk,agentFK,initialPrority,updateDate);

SELECT last\_insert\_id() AS NewIdTicket;

COMMIT;

END $$

DELIMITER ;

Proceso almancenado para actualizar solitud de reasigancion de ticket.

DELIMITER $$

CREATE PROCEDURE ReassignmentUpdate(IN comments text ,IN statusIn int unsigned ,in responseDate timestamp(3), IN reassignmentId int)

BEGIN

DECLARE msg TEXT;

DECLARE errno int;

DECLARE idArea int;

DECLARE Agent int;

DECLARE exit HANDLER FOR SQLEXCEPTION

BEGIN

GET current diagnostics condition 1

msg = message\_text, errno= mysql\_errno;

-- ERROR

SELECT errno AS NO\_ERROR , msg AS MESSAGE;

ROLLBACK;

END;

START TRANSACTION;

if statusIn > 0 AND statusIn < 2 then

UPDATE reassignments SET status= statusIn, commentsResponsible=comments, dateResponde=responseDate WHERE idReassignment=reassignmentId;

UPDATE tickets SET dateUpdate=responseDate, fkAgent=(SELECT fkAgentReciver FROM reassignments WHERE idReassigment=reassignmentId) WHERE idTicket=(SELECT fkTicket FROM reassignment WHERE idReassignment=reassignmentId);

ELSE

UPDATE reassignments SET status= statusIn, commentsResponsible=comments, dateResponde=responseDate WHERE idReassignment=reassignmentId;

end if;

COMMIT;

END $$

DELIMITER ;