



Módulo 3. Bases de datos

Sistema de Gestión de Tickets de Incidencias

Gabriel Gómez G.

Contenido

Descripción del Diagrama	2
Entidades Identificadas y sus Atributos	2
Diagrama UML.....	3
Diagrama UML con PlantUML	4
Justificación de las Relaciones.....	5
Conclusiones	5

Descripción del Diagrama

El diagrama UML ilustra las entidades clave del sistema y cómo se relacionan entre sí para gestionar las incidencias reportadas por los clientes. Este modelo permite identificar los elementos fundamentales necesarios para registrar, asignar y dar seguimiento a cada ticket, asegurando una atención eficiente y organizada.

Entidades Identificadas y sus Atributos

1. Cliente

- **idCliente:** Identificador único que distingue a cada cliente.
- **nombre:** Nombre completo del cliente.
- **email:** Correo electrónico de contacto para la comunicación.
- **telefono:** Número telefónico para soporte y seguimiento.

Justificación:

Esta entidad es crucial para asociar cada incidencia a un cliente específico, permitiendo el acceso a sus datos de contacto y facilitando la comunicación y el seguimiento personalizado.

2. Ticket

- **idTicket:** Identificador único del ticket de incidencia.
- **descripcion:** Detalle del problema reportado.
- **categoría:** Clasificación del problema (soporte técnico, facturación, etc.).
- **fechaCreacion:** Fecha (y hora) en la que se generó el ticket.
- **estadoActual:** Estado actual del ticket (abierto, en proceso, cerrado, etc.).

Justificación:

El ticket es el registro central de cada incidencia. Sus atributos permiten documentar de manera precisa el problema, categorizándolo y facilitando el seguimiento de su evolución.

3. Agente

- **idAgente:** Identificador único del agente responsable.
- **nombre:** Nombre del agente encargado.
- **especialidad:** Área de expertise del agente (por ejemplo, soporte técnico o facturación).

Justificación:

Los agentes son los encargados de atender las incidencias. Sus atributos ayudan a asignar el ticket al profesional más adecuado según la especialización, garantizando una resolución eficaz.

4. Interaccion (Historial de Seguimiento)

- **idInteraccion:** Identificador único para cada interacción.
- **fechaHora:** Momento en el que se realizó la interacción.
- **comentario:** Detalle de la actualización o comentario realizado.
- **estadoCambio:** Registro del cambio en el estado del ticket (si aplica).

Justificación:

Esta entidad permite mantener un historial detallado de todas las actualizaciones realizadas en un ticket, documentando la evolución y facilitando la revisión de la atención brindada.

Diagrama UML

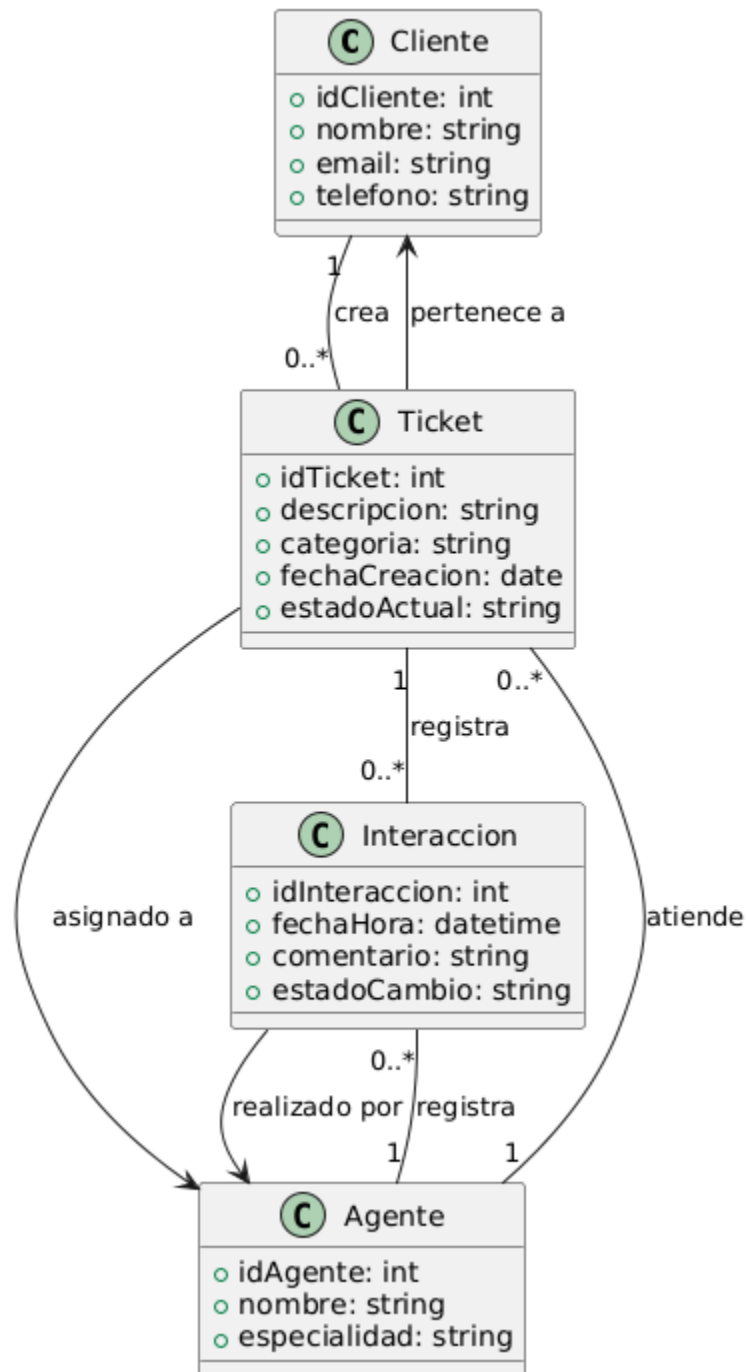


Diagrama UML con PlantUML

@startuml

' Definición de Clases

```
class Cliente {  
    +idCliente: int  
    +nombre: string  
    +email: string  
    +telefono: string  
}
```

```
class Ticket {  
    +idTicket: int  
    +descripcion: string  
    +categoria: string  
    +fechaCreacion: date  
    +estadoActual: string  
}
```

```
class Agente {  
    +idAgente: int  
    +nombre: string  
    +especialidad: string  
}
```

```
class Interaccion {  
    +idInteraccion: int  
  
    +fechaHora: datetime  
    +comentario: string
```

```
    +estadoCambio: string
```

```
}
```

' Relaciones

' Cliente crea Tickets

Cliente "1" -- "0..*" Ticket : crea

' Ticket pertenece a un Cliente

Ticket --> Cliente : pertenece a

' Ticket es asignado a un Agente

Ticket --> Agente : asignado a

Agente "1" -- "0..*" Ticket : atiende

' Ticket tiene un historial de Interacciones

Ticket "1" -- "0..*" Interaccion : registra

' Interaccion realizada por un Agente

Interaccion --> Agente : realizado por

Agente "1" -- "0..*" Interaccion : registra

@enduml

Link de programa:

<https://www.plantuml.com/plantuml/uml/SyfFKj2rKt3CoKnELR1Io4ZDoSa700003+>

Justificación de las Relaciones

- **Cliente — Ticket:** *Un cliente puede generar múltiples tickets.*

Justificación: Permite rastrear todas las incidencias reportadas por un mismo cliente, ofreciendo una visión completa de su historial de problemas y la atención recibida.

- **Ticket — Agente:** *Cada ticket es asignado a un único agente, aunque un agente puede atender varios tickets.*

Justificación: Esta relación asegura que cada incidencia cuente con un responsable que dé seguimiento y gestione la resolución, optimizando la distribución de tareas según la especialidad.

- **Ticket — Interaccion:** *Un ticket puede tener múltiples interacciones, representando cada actualización o comentario realizado.*

Justificación: El historial de interacciones documenta la evolución del ticket, proporcionando transparencia y permitiendo a cualquier agente revisar el progreso del caso de manera rápida y efectiva.

- **Agente — Interaccion:** *Cada interacción es realizada por un agente, y un agente puede realizar múltiples interacciones.*

Justificación: Esta relación permite identificar qué agente realizó cada actualización, facilitando la rendición de cuentas y el análisis del desempeño en la gestión de incidencias.

Conclusiones

- **Decisiones Tomadas:** Se optó por identificar las entidades fundamentales (Cliente, Ticket, Agente, Interacción) que integran el proceso de gestión de incidencias, definiendo atributos esenciales para capturar la información necesaria. Las relaciones fueron establecidas para reflejar con precisión la dinámica de interacción entre los clientes, los tickets y los agentes, lo que facilita tanto el seguimiento de cada incidencia como la asignación de responsabilidades.
- **Desafíos Enfrentados:** Uno de los principales desafíos fue equilibrar el nivel de detalle requerido para un seguimiento exhaustivo de las incidencias con la simplicidad y claridad del modelo UML. Es fundamental que el diagrama sea lo suficientemente detallado para ser implementado en una base de datos relacional sin complicar la estructura, y al mismo tiempo, fácil de comprender para los desarrolladores y agentes del sistema.

- **Resultado Final:** El diagrama resultante ofrece una representación clara y escalable del sistema, permitiendo futuras extensiones (como la incorporación de nuevas categorías o la normalización de algunos atributos) sin perder la integridad del modelo. Esta estructura facilita la implementación de un sistema robusto que garantice una atención de alta calidad y un seguimiento eficiente de cada incidencia.

Este análisis y diseño UML proporcionan una base sólida para el desarrollo e implementación del sistema de gestión de incidencias, asegurando que se cumplan los objetivos de eficiencia y organización en la atención al cliente.