# Sistema de Gestión Integral de Servicios Técnicos y Control de Inventario

## 1. Introducción

El sistema propuesto representa una solución integral para la gestión de servicios técnicos y control de inventario, diseñado para optimizar las operaciones de empresas que ofrecen servicios técnicos especializados. El sistema integra la gestión de usuarios, servicios, órdenes de trabajo, inventario y facturación en una única plataforma cohesiva.

## 2. Arquitectura del Sistema

### 2.1 Gestión de Usuarios

El sistema implementa un modelo jerárquico de usuarios que distingue entre:

 **Clientes**

 Solicitan servicios

 Acceden a su historial de órdenes

 Reciben notificaciones sobre sus servicios

 **Trabajadores**

 Ejecutan servicios técnicos

 Pertenecen a departamentos específicos

 Gestionan órdenes de trabajo

### 2.2 Gestión de Servicios y Departamentos

La estructura departamental se organiza en:

 **Departamentos Especializados**

 Telemática

 Ofimática

 Comunicaciones

 Centro Multiservicio

 **Servicios**

 Categorizados por tipo

 Asociados a departamentos específicos

 Precios base establecidos

### 2.3 Sistema de Órdenes

El módulo de órdenes gestiona el ciclo completo del servicio:

1. **Creación y Programación**

 Registro de fecha y hora

 Asignación de trabajadores

 Especificación de servicios

 **Estados de la Orden**

 Solicitada

 Programada

 En Proceso

 Completada

 Cancelada

 **Seguimiento**

 Registro de actividades

 Notificaciones automáticas

 Control de cambios

### 2.4 Gestión de Inventario

El sistema incluye un módulo completo de control de inventario:

 **Almacén**

 Control de ubicaciones

 Gestión de stock

 Seguimiento de movimientos

 **Piezas**

 Categorización (Servicio/Tienda)

 Control de stock mínimo

 Trazabilidad de uso

 **Integración con Servicios**

 Asociación de piezas a servicios

 Control de consumo

 Alertas de stock

### 2.5 Sistema de Facturación

Gestión automatizada de:

 Generación de facturas

 Cálculo de totales

 Registro de transacciones

### 2.6 Sistema de Notificaciones

Comunicación efectiva mediante:

 Alertas de estado

 Recordatorios

 Confirmaciones

## 3. Características Técnicas

### 3.1 Modularidad

 Componentes independientes

 Interfaces bien definidas

 Fácil mantenimiento

### 3.2 Escalabilidad

 Arquitectura extensible

 Soporte para crecimiento

 Adaptabilidad a cambios

### 3.3 Seguridad

 Autenticación robusta

 Control de acceso por roles

 Trazabilidad de acciones

## 4. Beneficios del Sistema

### 4.1 Operativos

 Automatización de procesos

 Reducción de errores

 Optimización de recursos

### 4.2 Gestión

 Control centralizado

 Reportes detallados

 Toma de decisiones informada

### 4.3 Servicio al Cliente

 Mejor experiencia de usuario

 Transparencia en el servicio

 Comunicación efectiva

### 4.4 Financieros

 Control de costos

 Gestión eficiente de inventario

 Facturación automatizada

## 5. Conclusiones

El sistema propuesto representa una solución completa y robusta para la gestión de servicios técnicos e inventario. Su diseño modular y escalable permite una implementación flexible y adaptable a diferentes necesidades empresariales, mientras que su integración de funcionalidades garantiza una operación eficiente y un servicio de calidad.

**Diagrama Completo**

classDiagram

    class Usuario {

        <<abstract>>

        -idUsuario: int

        -nombre: string

        -email: string

        -contraseña: string

        -tipoUsuario: TipoUsuario

        +registrarUsuario()

        +iniciarSesion()

        +actualizarPerfil()

        +cerrarSesion()

    }

    class TipoUsuario {

        <<enumeration>>

        CLIENTE

        TRABAJADOR

    }

    class Cliente {

        -direccion: string

        -telefono: string

        -historialOrdenes: List

        +solicitarOrden()

        +verHistorial()

    }

    class Trabajador {

        -cargo: string

        -departamento: Departamento

        -salario: double

        +atenderOrden()

        +registrarActividad()

    }

    class Servicio {

        -idServicio: int

        -nombre: string

        -descripcion: string

        -precioBase: double

        -departamento: Departamento

        -tipoServicio: TipoServicio

        +calcularPrecio()

    }

    class TipoServicio {

        <<enumeration>>

        TELEMATICA

        OFIMATICA

        COMUNICACIONES

        MULTISERVICIO

    }

    class Departamento {

        -idDepartamento: int

        -nombre: TipoDepartamento

        -responsable: Trabajador

        -serviciosOfrecidos: List~Servicio~

        +agregarServicio()

        +asignarTrabajador()

    }

    class TipoDepartamento {

        <<enumeration>>

        TELEMATICA

        OFIMATICA

        COMUNICACIONES

        CENTRO\_MULTISERVICIO

    }

    class Almacen {

        -idAlmacen: int

        -nombre: string

        -ubicacion: string

        +agregarPieza()

        +retirarPieza()

        +verificarStock()

        +actualizarInventario()

    }

    class Pieza {

        -idPieza: int

        -codigo: string

        -nombre: string

        -descripcion: string

        -precio: double

        -stock: int

        -stockMinimo: int

        -tipoPieza: TipoPieza

        +actualizarStock()

        +verificarStockMinimo()

    }

    class TipoPieza {

        <<enumeration>>

        SERVICIO

        TIENDA

    }

    class Orden {

        -idOrden: int

        -fechaCreacion: Date

        -fechaProgramada: Date

        -cliente: Cliente

        -entidad: string

        -trabajador: Trabajador

        -descripcion: string

        -serviciosSolicitados: List~ServicioSolicitado~

        -estado: EstadoOrden

        +crearOrden()

        +actualizarEstado()

        +agregarServicio()

        +eliminarServicio()

        +modificarServicio()

        +programarFecha()

        +asignarTrabajador()

    }

    class ServicioSolicitado {

        -servicio: Servicio

        -cantidad: int

        -observaciones: string

        +actualizarCantidad()

        +actualizarObservaciones()

    }

    class EstadoOrden {

        <<enumeration>>

        SOLICITADA

        PROGRAMADA

        EN\_PROCESO

        COMPLETADA

        CANCELADA

    }

    class Factura {

        -idFactura: int

        -fecha: Date

        -orden: Orden

        -subtotal: double

        -total: double

        +calcularTotal()

        +generarFactura()

    }

    class Notificacion {

        -idNotificacion: int

        -tipo: TipoNotificacion

        -mensaje: string

        -fecha: DateTime

        -destinatario: Usuario

        -estado: EstadoNotificacion

        +enviarNotificacion()

        +marcarComoLeida()

    }

    class TipoNotificacion {

        <<enumeration>>

        ORDEN\_CREADA

        ORDEN\_PROGRAMADA

        ORDEN\_EN\_PROCESO

        ORDEN\_COMPLETADA

        ORDEN\_CANCELADA

    }

    class EstadoNotificacion {

        <<enumeration>>

        PENDIENTE

        ENVIADA

        LEIDA

    }

    %% Relaciones

    Usuario <|-- Cliente

    Usuario <|-- Trabajador

    Usuario -- TipoUsuario

    Servicio -- TipoServicio

    Departamento -- TipoDepartamento

    Orden -- EstadoOrden

    Notificacion -- TipoNotificacion

    Notificacion -- EstadoNotificacion

    Cliente "1" -- "\*" Orden: solicita

    Trabajador "1" -- "1" Departamento: pertenece

    Trabajador "1" -- "\*" Orden: atiende

    Departamento "1" -- "\*" Servicio: ofrece

    Orden "1" -- "\*" ServicioSolicitado: contiene

    ServicioSolicitado "\*" -- "1" Servicio: referencia

    Orden "1" -- "1" Factura: genera

    Usuario "1" -- "\*" Notificacion: recibe

    Orden "1" -- "\*" Notificacion: genera

    Almacen "1" -- "\*" Pieza: contiene

    Pieza -- TipoPieza

    ServicioSolicitado "\*" -- "\*" Pieza: utiliza