



**UNIVERSIDAD TECNOLÓGICA DEL PERÚ**

**Curso:**

Desarrollo Web Integrado

**Docente:**

Ing- Alex Fernando Torres Chahuara

**Título del tema:** *Software para la automatización de citas Medicas.*

**Integrantes:**

SOLANO ALVAREZ, ALFREDO MARTIN

U22328252

**Sección:**

37681

Lima, 05 de Junio de 2025

## Índice

1	Descripción del Problema .....	3
2	Objetivos.....	3
2.1	Objetivos Específicos .....	3
3	Diseño de la Aplicación.....	4
3.1	Descripción de los módulos .....	4
3.2	Diseño de las clases.....	5
3.3	Descripción de Módulos .....	6
4	Prototipos .....	27
5	Conclusiones .....	27
6	Recomendaciones.....	27
7	Bibliografía .....	27

## **1 Descripción del Problema**

Actualmente, el proceso de agendar citas médicas en muchos centros de salud es lento y desorganizado. En muchos casos, todo se hace de forma manual o con archivos que no están conectados entre sí, lo que puede generar errores, pérdida de información o que las citas se crucen. Esto no solo afecta a los pacientes, que tienen que esperar más o no reciben atención a tiempo, sino también al personal médico y administrativo, que pierde tiempo tratando de organizarse.

En el caso de este proyecto, se identificó que no hay un sistema que permita controlar de manera ordenada las citas, registrar adecuadamente a los pacientes, guardar el historial de atenciones, ni generar comprobantes de pago de forma automática. Todo esto complica el trabajo diario y puede afectar la calidad del servicio que se brinda.

Por eso, se propone desarrollar un sistema que ayude a organizar mejor estos procesos, para que los usuarios puedan agendar citas fácilmente, los médicos puedan ver su programación y el área administrativa pueda llevar un control claro y seguro de todo.

## **2 Objetivos**

El objetivo principal de este proyecto es desarrollar un sistema que ayude a mejorar la gestión de citas médicas, haciendo que todo el proceso sea más rápido, ordenado y fácil de usar tanto para los pacientes como para el personal de salud.

### **2.1 Objetivos Específicos**

- Permitir el registro de citas médicas de forma rápida, evitando errores y cruces de horarios.
- Registrar los datos de los pacientes y médicos para tenerlos siempre disponibles cuando se necesiten.
- Llevar un control de las atenciones realizadas, incluyendo fecha, hora y observaciones.
- Generar comprobantes de pago de manera automática y con la información correcta.

- Facilitar el acceso a la información desde un solo lugar, mejorando la organización del centro de salud.

### 3 Diseño de la Aplicación

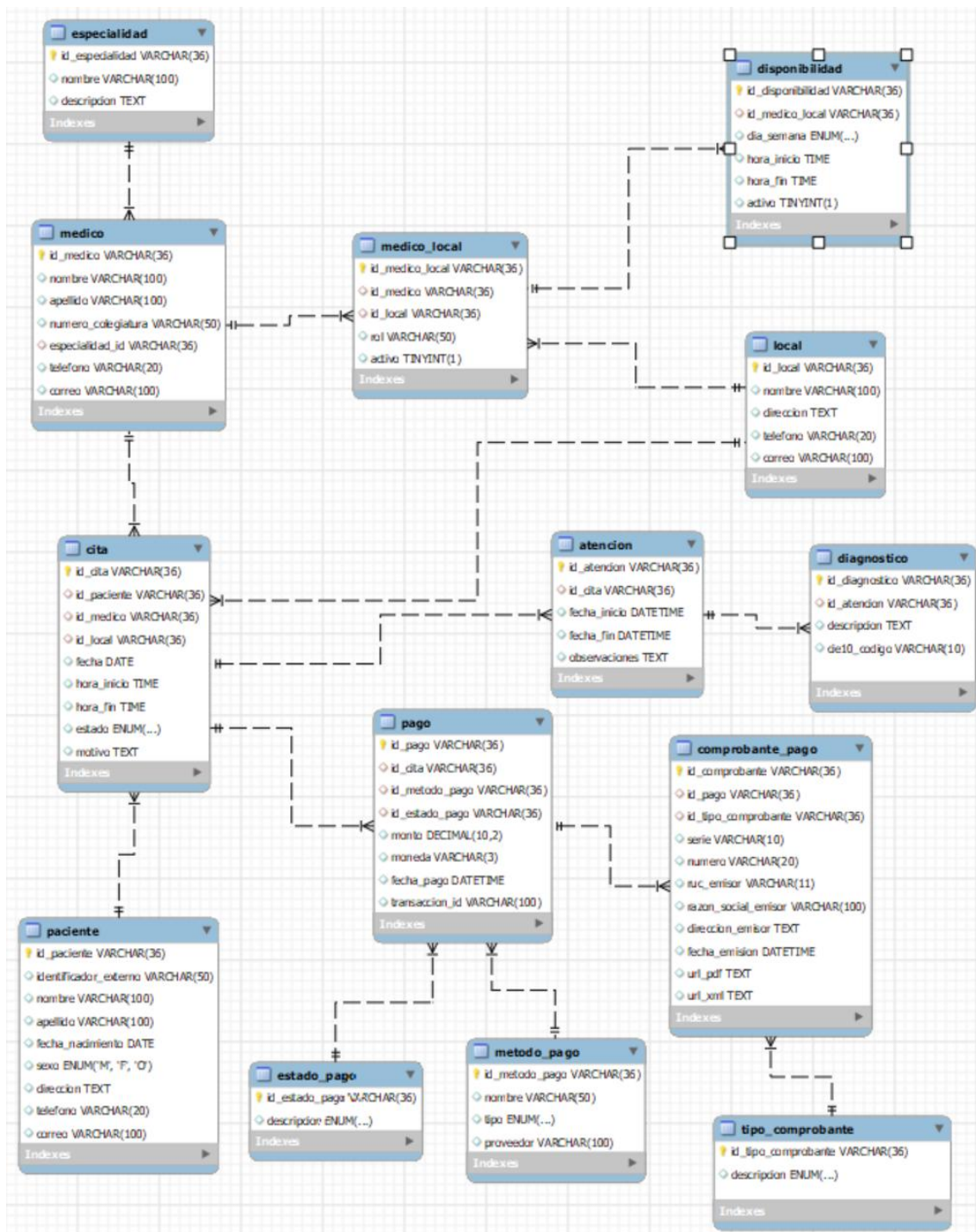
#### 3.1 Descripción de los módulos

#	Nombre del módulo	Funcionalidades del módulo
1	Gestión de Pacientes	Permite registrar nuevos pacientes, actualizar su información y listar los registrados.
2	Gestión de Citas	Permite agendar, modificar o eliminar citas médicas. También asocia la cita a un médico, paciente y local.
3	Atención Médica	Registra el inicio y fin de la atención, y permite agregar observaciones médicas.
4	Gestión de Médicos	Registra datos de médicos, sus especialidades y disponibilidad.
5	Gestión de Comprobantes	Genera comprobantes de pago para cada atención realizada y los vincula con las citas.
6	Métodos y Estados de Pago	Permite gestionar los métodos de pago disponibles y el estado de cada pago (pagado, pendiente, etc.).
7	Gestión de Locales	Registra y administra los locales donde se brindan las atenciones.
8	Diagnósticos	Asocia diagnósticos a las atenciones según la observación médica ingresada.

#	Opción del menú	Descripción del menú
1	Inicio	Página principal del sistema. Muestra accesos directos a los módulos más usados.
2	Pacientes	Permite registrar nuevos pacientes, editar información y ver el listado completo.
3	Citas Médicas	Se usa para agendar citas, consultar por paciente o médico, y hacer cambios o anulaciones.
4	Atenciones	Registro de la atención médica realizada, incluyendo fecha, hora y observaciones.
5	Comprobantes de Pago	Genera comprobantes por las atenciones realizadas y permite asociar los métodos de pago.
6	Médicos	Gestiona la información del personal médico, incluyendo especialidad y horarios.
7	Locales	Muestra y permite administrar los locales donde se realizan las consultas.

8	Métodos de Pago	Permite agregar, modificar o eliminar los métodos de pago aceptados.
9	Diagnósticos	Permite registrar y consultar los diagnósticos médicos por atención.

### 3.2 Diseño de las clases



### 3.3 Descripción de Módulos

Paquete: BusinessEntity

Clase: AtencionBE

Atributos:

Atributo	Descripción
id_atencion	Código único de atención.
id_cita	Código de la cita relacionada.
fecha_inicio	Fecha y hora en que inicia la atención.
fecha_fin	Fecha y hora en que termina la atención.
observaciones	Notas médicas del médico.

Métodos:

Método	Descripción
getId_atencion()	Devuelve el valor de id_atencion.
getId_cita()	Devuelve el valor de id_cita.
getFecha_inicio()	Devuelve el valor de fecha_inicio.
getFecha_fin()	Devuelve el valor de fecha_fin.
getObservaciones()	Devuelve el valor de observaciones.
setId_atencion(valor)	Asigna un valor a id_atencion.
setId_cita(valor)	Asigna un valor a id_cita.
setFecha_inicio(valor)	Asigna un valor a fecha_inicio.
setFecha_fin(valor)	Asigna un valor a fecha_fin.
setObservaciones(valor)	Asigna un valor a observaciones.

Paquete: BusinessEntity

Clase: CitaBE

Atributos:

Atributo	Descripción
id_cita	Identificador único de cita.
descripcion	Descripción o detalle relacionado con cita.

Métodos:

Método	Descripción
getId_cita()	Devuelve el valor de id_cita.
getDescripcion()	Devuelve el valor de descripcion.
setId_cita(valor)	Asigna un valor a id_cita.
setDescripcion(valor)	Asigna un valor a descripcion.

Paquete: BusinessEntity

Clase: ComprobantePagoBE

Atributos:

Atributo	Descripción
id_comprobantepago	Identificador único de comprobantepago.
descripcion	Descripción o detalle relacionado con comprobantepago.

Métodos:

Método	Descripción
getId_comprobantepago()	Devuelve el valor de id_comprobantepago.
getDescripcion()	Devuelve el valor de descripcion.

setId_comprobantepago(valor)	Asigna un valor a id_comprobantepago.
setDescripcion(valor)	Asigna un valor a descripcion.

Paquete: BusinessEntity

Clase: DiagnosticoBE

Atributos:

Atributo	Descripción
id_diagnostico	Identificador único de diagnostico.
descripcion	Descripción o detalle relacionado con diagnostico.

Métodos:

Método	Descripción
getId_diagnostico()	Devuelve el valor de id_diagnostico.
getDescripcion()	Devuelve el valor de descripcion.
setId_diagnostico(valor)	Asigna un valor a id_diagnostico.
setDescripcion(valor)	Asigna un valor a descripcion.

Paquete: BusinessEntity

Clase: DisponibilidadBE

Atributos:

Atributo	Descripción
id_disponibilidad	Identificador único de disponibilidad.
descripcion	Descripción o detalle relacionado con disponibilidad.



Métodos:

Método	Descripción
getId_disponibilidad()	Devuelve el valor de id_disponibilidad.
getDescripcion()	Devuelve el valor de descripcion.
setId_disponibilidad(valor)	Asigna un valor a id_disponibilidad.
setDescripcion(valor)	Asigna un valor a descripcion.

Paquete: BusinessEntity

Clase: EspecialidadBE

Atributos:

Atributo	Descripción
id_especialidad	Identificador único de especialidad.
descripcion	Descripción o detalle relacionado con especialidad.

Métodos:

Método	Descripción
getId_especialidad()	Devuelve el valor de id_especialidad.
getDescripcion()	Devuelve el valor de descripcion.
setId_especialidad(valor)	Asigna un valor a id_especialidad.
setDescripcion(valor)	Asigna un valor a descripcion.

Paquete: BusinessEntity

Clase: EstadoPagoBE

Atributos:

Atributo	Descripción
id_estadopago	Identificador único de estadopago.
descripcion	Descripción o detalle relacionado con estadopago.

Métodos:

Método	Descripción
getId_estadopago()	Devuelve el valor de id_estadopago.
getDescripcion()	Devuelve el valor de descripcion.
setId_estadopago(valor)	Asigna un valor a id_estadopago.
setDescripcion(valor)	Asigna un valor a descripcion.

Paquete: BusinessEntity

Clase: LocalBE

Atributos:

Atributo	Descripción
id_local	Identificador único de local.
descripcion	Descripción o detalle relacionado con local.

Métodos:

Método	Descripción
getId_local()	Devuelve el valor de id_local.
getDescripcion()	Devuelve el valor de descripcion.

setId_local(valor)	Asigna un valor a id_local.
setDescripcion(valor)	Asigna un valor a descripcion.

Paquete: BusinessEntity

Clase: MedicoBE

Atributos:

Atributo	Descripción
id_medico	Identificador único del médico.
nombre	Nombres completos del médico.
especialidad	Especialidad médica del doctor.
disponibilidad	Días y horas en los que puede atender.

Métodos:

Método	Descripción
getId_medico()	Devuelve el valor de id_medico.
getNombre()	Devuelve el valor de nombre.
getEspecialidad()	Devuelve el valor de especialidad.
getDisponibilidad()	Devuelve el valor de disponibilidad.
setId_medico(valor)	Asigna un valor a id_medico.
setNombre(valor)	Asigna un valor a nombre.
setEspecialidad(valor)	Asigna un valor a especialidad.
setDisponibilidad(valor)	Asigna un valor a disponibilidad.

Paquete: BusinessEntity

Clase: MedicoLocalBE

Atributos:

Atributo	Descripción
id_medicolocal	Identificador único de medicolocal.
descripcion	Descripción o detalle relacionado con medicolocal.

Métodos:

Método	Descripción
getId_medicolocal()	Devuelve el valor de id_medicolocal.
getDescripcion()	Devuelve el valor de descripcion.
setId_medicolocal(valor)	Asigna un valor a id_medicolocal.
setDescripcion(valor)	Asigna un valor a descripcion.

Paquete: BusinessEntity

Clase: MetodoPagoBE

Atributos:

Atributo	Descripción
id_metodopago	Identificador único de metodopago.
descripcion	Descripción o detalle relacionado con metodopago.

Métodos:

Método	Descripción
getId_metodopago()	Devuelve el valor de id_metodopago.
getDescripcion()	Devuelve el valor de descripcion.

setId_metodopago(valor)	Asigna un valor a id_metodopago.
setDescripcion(valor)	Asigna un valor a descripcion.

Paquete: BusinessEntity

Clase: PacienteBE

Atributos:

Atributo	Descripción
id_paciente	Identificador único del paciente.
nombre	Nombres completos del paciente.
dni	Documento Nacional de Identidad.
telefono	Número de contacto del paciente.
email	Correo electrónico del paciente.

Métodos:

Método	Descripción
getId_paciente()	Devuelve el valor de id_paciente.
getNombre()	Devuelve el valor de nombre.
getDni()	Devuelve el valor de dni.
getTelefono()	Devuelve el valor de telefono.
getEmail()	Devuelve el valor de email.
setId_paciente(valor)	Asigna un valor a id_paciente.
setNombre(valor)	Asigna un valor a nombre.
setDni(valor)	Asigna un valor a dni.
setTelefono(valor)	Asigna un valor a telefono.
setEmail(valor)	Asigna un valor a email.

Paquete: BusinessEntity

Clase: PagoBE

Atributos:

Atributo	Descripción
id_pago	Identificador único de pago.
descripcion	Descripción o detalle relacionado con pago.

Métodos:

Método	Descripción
getId_pago()	Devuelve el valor de id_pago.
getDescripcion()	Devuelve el valor de descripcion.
setId_pago(valor)	Asigna un valor a id_pago.
setDescripcion(valor)	Asigna un valor a descripcion.

Paquete: BusinessEntity

Clase: TipoComprobanteBE

Atributos:

Atributo	Descripción
id_tipocomprobante	Identificador único de tipocomprobante.
descripcion	Descripción o detalle relacionado con tipocomprobante.

Métodos:

Método	Descripción
getId_tipocomprobante()	Devuelve el valor de id_tipocomprobante.

getDescripcion()	Devuelve el valor de descripcion.
setId_tipocomprobante(valor)	Asigna un valor a id_tipocomprobante.
setDescripcion(valor)	Asigna un valor a descripcion.

Paquete: BusinessLogic

Clase: AtencionBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para atencion

Métodos:

Método	Descripción
validarAtencion()	Valida datos para atencion.
procesarAtencion()	Procesa reglas de negocio de atencion.

Paquete: BusinessLogic

Clase: CitaBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para cita

Métodos:

Método	Descripción
validarCita()	Valida datos para cita.
procesarCita()	Procesa reglas de negocio de cita.

Paquete: BusinessLogic

Clase: ComprobantePagoBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para comprobante pago

Métodos:

Método	Descripción
validarComprobantePago()	Valida datos para comprobantepago.
procesarComprobantePago()	Procesa reglas de negocio de comprobantepago.

Paquete: BusinessLogic

Clase: DiagnosticoBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para diagnostico

Métodos:

Método	Descripción
validarDiagnostico()	Valida datos para diagnostico.
procesarDiagnostico()	Procesa reglas de negocio de diagnostico.

Paquete: BusinessLogic

Clase: DisponibilidadBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para disponibilidad

Métodos:

Método	Descripción
validarDisponibilidad()	Valida datos para disponibilidad.
procesarDisponibilidad()	Procesa reglas de negocio de disponibilidad.

Paquete: BusinessLogic

Clase: EspecialidadBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para especialidad



Métodos:

Método	Descripción
validarEspecialidad()	Valida datos para especialidad.
procesarEspecialidad()	Procesa reglas de negocio de especialidad.

Paquete: BusinessLogic

Clase: EstadoPagoBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para estadopago

Métodos:

Método	Descripción
validarEstadoPago()	Valida datos para estadopago.
procesarEstadoPago()	Procesa reglas de negocio de estadopago.

Paquete: BusinessLogic

Clase: IBaseBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para ibase

Métodos:

Método	Descripción
validarIBase()	Valida datos para ibase.
procesarIBase()	Procesa reglas de negocio de ibase.

Paquete: BusinessLogic

Clase: LocalBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para local

Métodos:

Método	Descripción
validarLocal()	Valida datos para local.
procesarLocal()	Procesa reglas de negocio de local.

Paquete: BusinessLogic

Clase: MedicoBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para medico

Métodos:

Método	Descripción
validarMedico()	Valida datos para medico.
procesarMedico()	Procesa reglas de negocio de medico.

Paquete: BusinessLogic

Clase: MedicoLocalBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para medicolocal

Métodos:

Método	Descripción
validarMedicoLocal()	Valida datos para medicolocal.
procesarMedicoLocal()	Procesa reglas de negocio de medicolocal.

Paquete: BusinessLogic

Clase: MetodoPagoBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para metodopago

Métodos:

Método	Descripción
validarMetodoPago()	Valida datos para metodopago.
procesarMetodoPago()	Procesa reglas de negocio de metodopago.

Paquete: BusinessLogic

Clase: PacienteBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para paciente

Métodos:

Método	Descripción
validarPaciente()	Valida datos para paciente.
procesarPaciente()	Procesa reglas de negocio de paciente.

Paquete: BusinessLogic

Clase: PagoBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para pago

Métodos:

Método	Descripción
validarPago()	Valida datos para pago.
procesarPago()	Procesa reglas de negocio de pago.

Paquete: BusinessLogic

Clase: TipoComprobanteBL

Atributos:

Atributo	Descripción
logica	Método principal de negocio para tipocomprobante

Métodos:

Método	Descripción
validarTipoComprobante()	Valida datos para tipocomprobante.
procesarTipoComprobante()	Procesa reglas de negocio de tipocomprobante.

Paquete: DataAccessObject

Clase: AtencionDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de atencion en base de datos.

Métodos:

Método	Descripción
insertarAtencion()	Agrega un nuevo registro a la tabla atencion.
actualizarAtencion()	Modifica un registro existente de la tabla atencion.
eliminarAtencion()	Elimina un registro de la tabla atencion.
listarAtencion()	Obtiene todos los registros de la tabla atencion.

Paquete: DataAccessObject

Clase: CitaDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de cita en base de datos.

Métodos:

Método	Descripción
insertarCita()	Agrega un nuevo registro a la tabla cita.
actualizarCita()	Modifica un registro existente de la tabla cita.
eliminarCita()	Elimina un registro de la tabla cita.
listarCita()	Obtiene todos los registros de la tabla cita.

Paquete: DataAccessObject

Clase: ComprobantePagoDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de comprobantepago en base de datos.

Métodos:

Método	Descripción
insertarComprobantePago()	Agrega un nuevo registro a la tabla comprobantepago.
actualizarComprobantePago()	Modifica un registro existente de la tabla comprobantepago.
eliminarComprobantePago()	Elimina un registro de la tabla comprobantepago.
listarComprobantePago()	Obtiene todos los registros de la tabla comprobantepago.

Paquete: DataAccessObject

Clase: Conexion

Atributos:

Atributo	Descripción
conexion	Gestión de conexión con la base de datos.

Métodos:

Método	Descripción
abrirConexion()	Establece la conexión con la base de datos.
cerrarConexion()	Cierra la conexión actual con la base de datos.

Paquete: DataAccessObject

Clase: DiagnosticoDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de diagnostico en base de datos.

Métodos:

Método	Descripción
insertarDiagnostico()	Agrega un nuevo registro a la tabla diagnostico.
actualizarDiagnostico()	Modifica un registro existente de la tabla diagnostico.
eliminarDiagnostico()	Elimina un registro de la tabla diagnostico.
listarDiagnostico()	Obtiene todos los registros de la tabla diagnostico.

Paquete: DataAccessObject

Clase: DisponibilidadDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de disponibilidad en base de datos.

Métodos:

Método	Descripción
insertarDisponibilidad()	Agrega un nuevo registro a la tabla disponibilidad.
actualizarDisponibilidad()	Modifica un registro existente de la tabla disponibilidad.
eliminarDisponibilidad()	Elimina un registro de la tabla disponibilidad.
listarDisponibilidad()	Obtiene todos los registros de la tabla disponibilidad.

Paquete: DataAccessObject

Clase: EspecialidadDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de especialidad en base de datos.

Métodos:

Método	Descripción
insertarEspecialidad()	Agrega un nuevo registro a la tabla especialidad.
actualizarEspecialidad()	Modifica un registro existente de la tabla especialidad.
eliminarEspecialidad()	Elimina un registro de la tabla especialidad.
listarEspecialidad()	Obtiene todos los registros de la tabla especialidad.

Paquete: DataAccessObject

Clase: EstadoPagoDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de estadopago en base de datos.

Métodos:

Método	Descripción
insertarEstadoPago()	Agrega un nuevo registro a la tabla estadopago.
actualizarEstadoPago()	Modifica un registro existente de la tabla estadopago.
eliminarEstadoPago()	Elimina un registro de la tabla estadopago.
listarEstadoPago()	Obtiene todos los registros de la tabla estadopago.

Paquete: DataAccessObject

Clase: IBaseDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de ibase en base de datos.

Métodos:

Método	Descripción
insertarIBase()	Agrega un nuevo registro a la tabla ibase.
actualizarIBase()	Modifica un registro existente de la tabla ibase.

eliminarIBase()	Elimina un registro de la tabla ibase.
listarIBase()	Obtiene todos los registros de la tabla ibase.

Paquete: DataAccessObject

Clase: LocalDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de local en base de datos.

Métodos:

Método	Descripción
insertarLocal()	Agrega un nuevo registro a la tabla local.
actualizarLocal()	Modifica un registro existente de la tabla local.
eliminarLocal()	Elimina un registro de la tabla local.
listarLocal()	Obtiene todos los registros de la tabla local.

Paquete: DataAccessObject

Clase: MedicoDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de medico en base de datos.

Métodos:

Método	Descripción
insertarMedico()	Agrega un nuevo registro a la tabla medico.
actualizarMedico()	Modifica un registro existente de la tabla medico.
eliminarMedico()	Elimina un registro de la tabla medico.
listarMedico()	Obtiene todos los registros de la tabla medico.



Paquete: DataAccessObject

Clase: MedicoLocalDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de medicolocal en base de datos.

Métodos:

Método	Descripción
insertarMedicoLocal()	Agrega un nuevo registro a la tabla medicolocal.
actualizarMedicoLocal()	Modifica un registro existente de la tabla medicolocal.
eliminarMedicoLocal()	Elimina un registro de la tabla medicolocal.
listarMedicoLocal()	Obtiene todos los registros de la tabla medicolocal.

Paquete: DataAccessObject

Clase: MetodoPagoDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de metodopago en base de datos.

Métodos:

Método	Descripción
insertarMetodoPago()	Agrega un nuevo registro a la tabla metodopago.
actualizarMetodoPago()	Modifica un registro existente de la tabla metodopago.
eliminarMetodoPago()	Elimina un registro de la tabla metodopago.
listarMetodoPago()	Obtiene todos los registros de la tabla metodopago.

Paquete: DataAccessObject

Clase: PacienteDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de paciente en base de datos.

Métodos:

Método	Descripción
insertarPaciente()	Agrega un nuevo registro a la tabla paciente.
actualizarPaciente()	Modifica un registro existente de la tabla paciente.
eliminarPaciente()	Elimina un registro de la tabla paciente.
listarPaciente()	Obtiene todos los registros de la tabla paciente.

Paquete: DataAccessObject

Clase: PagoDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de pago en base de datos.

Métodos:

Método	Descripción
insertarPago()	Agrega un nuevo registro a la tabla pago.
actualizarPago()	Modifica un registro existente de la tabla pago.
eliminarPago()	Elimina un registro de la tabla pago.
listarPago()	Obtiene todos los registros de la tabla pago.

Paquete: DataAccessObject

Clase: TipoComprobanteDAO

Atributos:

Atributo	Descripción
tabla	Referencia a la tabla de tipocomprobante en base de datos.

Métodos:

Método	Descripción
insertarTipoComprobante()	Agrega un nuevo registro a la tabla tipocomprobante.
actualizarTipoComprobante()	Modifica un registro existente de la tabla tipocomprobante.
eliminarTipoComprobante()	Elimina un registro de la tabla tipocomprobante.
listarTipoComprobante()	Obtiene todos los registros de la tabla tipocomprobante.

#### **4 Prototipos**

#### **5 Conclusiones**

#### **6 Recomendaciones**

#### **7 Bibliografía**