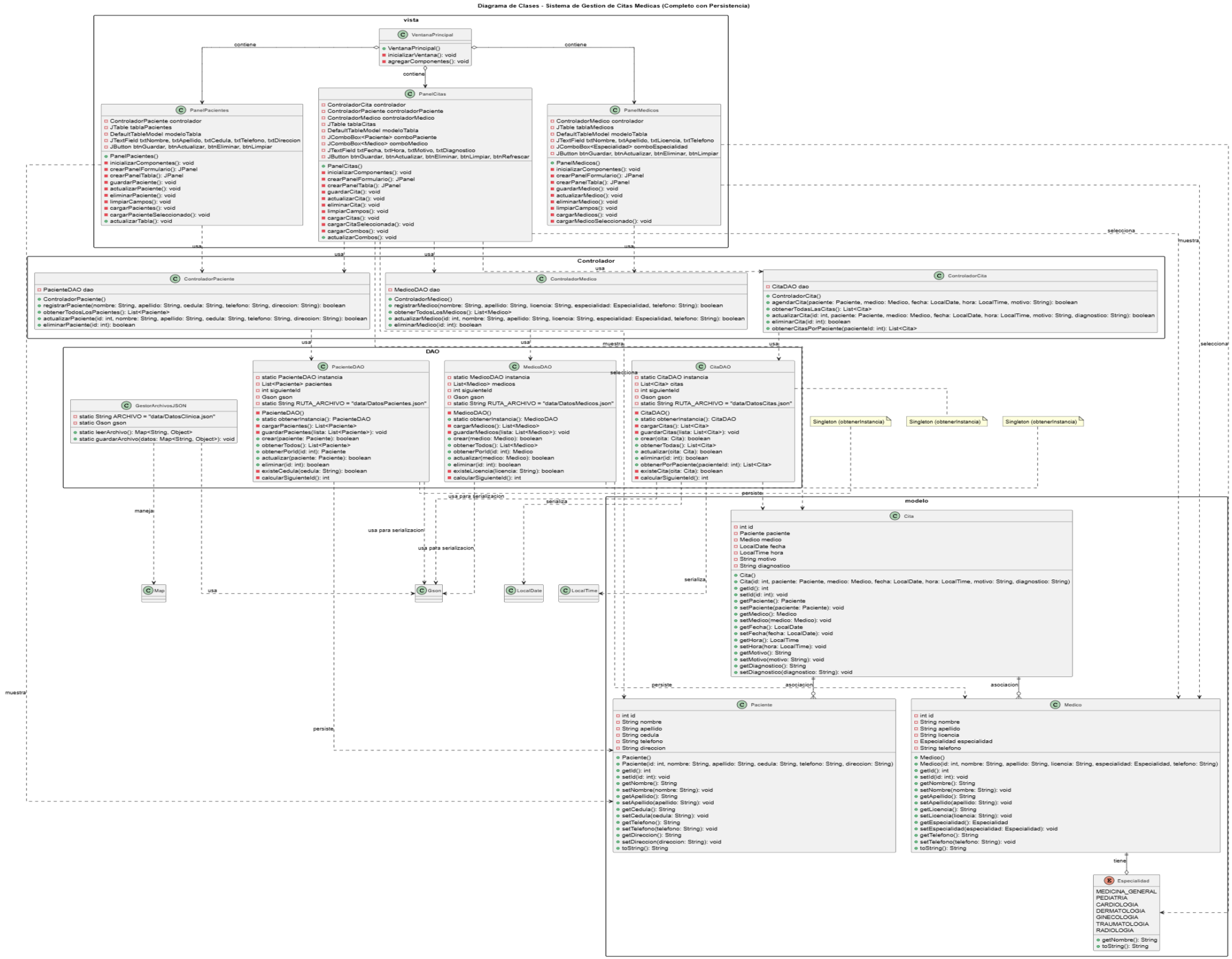
Diagrama de Clases - Sistema de Gestion de Citas Medicas (Completo con Persistencia)
vista
VentanaPrincipal
VentanaPrincipal()
inicializarVentana(): void
agregarComponentes(): void
contiene
contiene
contiene
PanelPacientes
ControladorPaciente controlador
JTable tablaPacientes
DefaultTableModel modeloTabla
JTextField txtNombre, txtApellido, txtCedula, txtTelefono, txtDireccion
JButton btnGuardar, btnActualizar, btnEliminar, btnLimpiar
PanelPacientes()
inicializarComponentes(): void
crearPanelFormulario(): JPanel
crearPanelTabla(): JPanel
guardarPaciente(): void
actualizarPaciente(): void
eliminarPaciente(): void
limpiarCampos(): void
cargarPacientes(): void
cargarPacienteSeleccionado(): void
actualizarTabla(): void
PanelCitas
ControladorCita controlador
ControladorPaciente controladorPaciente
ControladorMedico controladorMedico
JTable tablaCitas
DefaultTableModel modeloTabla
JComboBox<Paciente> comboPaciente
JComboBox<Medico> comboMedico
JTextField txtFecha, txtHora, txtMotivo, txtDiagnostico
JButton btnGuardar, btnActualizar, btnEliminar, btnLimpiar, btnRefrescar
PanelCitas()
inicializarComponentes(): void
crearPanelFormulario(): JPanel
crearPanelTabla(): JPanel
guardarCita(): void
actualizarCita(): void
eliminarCita(): void
limpiarCampos(): void
cargarCitas(): void
cargarCitaSeleccionada(): void
cargarCombos(): void
actualizarCombos(): void
PanelMedicos
ControladorMedico controlador
JTable tablaMedicos
DefaultTableModel modeloTabla
JTextField txtNombre, txtApellido, txtLicencia, txtTelefono
JComboBox<Especialidad> comboEspecialidad
JButton btnGuardar, btnActualizar, btnEliminar, btnLimpiar
PanelMedicos()
inicializarComponentes(): void
crearPanelFormulario(): JPanel
crearPanelTabla(): JPanel
guardarMedico(): void
actualizarMedico(): void
eliminarMedico(): void
limpiarCampos(): void
cargarMedicos(): void
cargarMedicoSeleccionado(): void
usa
usa
usa
usa
selecciona
muestra
Controlador
ControladorPaciente
PacienteDAO dao
ControladorPaciente()
registrarPaciente(nombre: String, apellido: String, cedula: String, telefono: String, direccion: String): boolean
obtenerTodosLosPacientes(): List<Paciente>
actualizarPaciente(id: int, nombre: String, apellido: String, cedula: String, telefono: String, direccion: String): boolean
eliminarPaciente(id: int): boolean
ControladorMedico
MedicoDAO dao
ControladorMedico()
registrarMedico(nombre: String, apellido: String, licencia: String, especialidad: Especialidad, telefono: String): boolean
obtenerTodosLosMedicos(): List<Medico>
actualizarMedico(id: int, nombre: String, apellido: String, licencia: String, especialidad: Especialidad, telefono: String): boolean
eliminarMedico(id: int): boolean
ControladorCita
CitaDAO dao
ControladorCita()
agendarCita(paciente: Paciente, medico: Medico, fecha: LocalDate, hora: LocalTime, motivo: String): boolean
obtenerTodasLasCitas(): List<Cita>
actualizarCita(id: int, paciente: Paciente, medico: Medico, fecha: LocalDate, hora: LocalTime, motivo: String, diagnostico: String): boolean
eliminarCita(id: int): boolean
obtenerCitasPorPaciente(pacienteId: int): List<Cita>
usa
usa
usa
usa
usa
usa
muestra
selecciona
selecciona
DAO
GestorArchivosJSON
static String ARCHIVO = "data/DatosClinica.json"
static Gson gson
static leerArchivo(): Map<String, Object>
static guardarArchivo(datos: Map<String, Object>): void
PacienteDAO
static PacienteDAO instancia
List<Paciente> pacientes
int siguienteId
Gson gson
static String RUTA_ARCHIVO = "data/DatosPacientes.json"
PacienteDAO()
static obtenerInstancia(): PacienteDAO
cargarPacientes(): List<Paciente>
guardarPacientes(lista: List<Paciente>): void
crear(paciente: Paciente): boolean
obtenerTodos(): List<Paciente>
obtenerPorId(id: int): Paciente
actualizar(paciente: Paciente): boolean
eliminar(id: int): boolean
existeCedula(cedula: String): boolean
calcularSiguienteId(): int
MedicoDAO
static MedicoDAO instancia
List<Medico> medicos
int siguienteId
Gson gson
static String RUTA_ARCHIVO = "data/DatosMedicos.json"
MedicoDAO()
static obtenerInstancia(): MedicoDAO
cargarMedicos(): List<Medico>
guardarMedicos(lista: List<Medico>): void
crear(medico: Medico): boolean
obtenerTodos(): List<Medico>
obtenerPorId(id: int): Medico
actualizar(medico: Medico): boolean
eliminar(id: int): boolean
existeLicencia(licencia: String): boolean
calcularSiguienteId(): int
CitaDAO
static CitaDAO instancia
List<Cita> citas
int siguienteId
Gson gson
static String RUTA_ARCHIVO = "data/DatosCitas.json"
CitaDAO()
static obtenerInstancia(): CitaDAO
cargarCitas(): List<Cita>
guardarCitas(lista: List<Cita>): void
crear(cita: Cita): boolean
obtenerTodas(): List<Cita>
actualizar(cita: Cita): boolean
eliminar(id: int): boolean
obtenerPorPaciente(pacienteId: int): List<Cita>
existeCita(cita: Cita): boolean
calcularSiguienteId(): int
Singleton (obtenerInstancia)
Singleton (obtenerInstancia)
Singleton (obtenerInstancia)
usa para serializacion
serializa
usa para serializacion
usa para serializacion
persiste
manejo
usa
serializa
muestra
persiste
Map
Gson
LocalDate
LocalTime
modelo
Cita
int id
Paciente paciente
Medico medico
LocalDate fecha
LocalTime hora
String motivo
String diagnostico
Cita()
Cita(id: int, paciente: Paciente, medico: Medico, fecha: LocalDate, hora: LocalTime, motivo: String, diagnostico: String)
getId(): int
setId(id: int): void
getPaciente(): Paciente
setPaciente(paciente: Paciente): void
getMedico(): Medico
setMedico(medico: Medico): void
getFecha(): LocalDate
setFecha(fecha: LocalDate): void
getHora(): LocalTime
setHora(hora: LocalTime): void
getMotivo(): String
setMotivo(motivo: String): void
getDiagnostico(): String
setDiagnostico(diagnostico: String): void
persiste
asociacion
asociacion
Paciente
int id
String nombre
String apellido
String cedula
String telefono
String direccion
Paciente()
Paciente(id: int, nombre: String, apellido: String, cedula: String, telefono: String, direccion: String)
getId(): int
setId(id: int): void
getNombre(): String
setNombre(nombre: String): void
getApellido(): String
setApellido(apellido: String): void
getCedula(): String
setCedula(cedula: String): void
getTelefono(): String
setTelefono(telefono: String): void
getDireccion(): String
setDireccion(direccion: String): void
toString(): String
Medico
int id
String nombre
String apellido
String licencia
Especialidad especialidad
String telefono
Medico()
Medico(id: int, nombre: String, apellido: String, licencia: String, especialidad: Especialidad, telefono: String)
getId(): int
setId(id: int): void
getNombre(): String
setNombre(nombre: String): void
getApellido(): String
setApellido(apellido: String): void
getLicencia(): String
setLicencia(licencia: String): void
getEspecialidad(): Especialidad
setEspecialidad(especialidad: Especialidad): void
getTelefono(): String
setTelefono(telefono: String): void
toString(): String
tiene
Especialidad
MEDICINA_GENERAL
PEDIATRIA
CARDIOLOGIA
DERMATOLOGIA
GINECOLOGIA
TRAUMATOLOGIA
RADIOLOGIA
getNombre(): String
toString(): String