

Proyecto Final: Sistema de gestión de hospital

Pensamiento computacional

Daniel Felipe Montes -C.C.: 1092852504

Juan Esteban Guerrero -C.C.: 1118366132

José Gabriel Paguay - C.C.: 1087616938

Docente: Jhan Carlos Martinez Ceballos



**UNIVERSIDAD
DEL QUINDÍO**

Universidad de Quindío

Facultad de Ingeniería

Ingeniería de Sistemas y Computación

Programación I

Armenia, Quindío, 2025



1. Abstracción:

¿Qué se solicita finalmente?

R/ Sistema para administración de pacientes, citas médicas y personal médico, enfocado en la optimización de gestión de historiales médicos, programación de consultas y asignación de médicos, mediante una interfaz fácil al manejo.

¿Qué información es relevante dado el problema anterior?

R/ Datos para la creación del usuario (superclase abstracta): nombre, apellido, id, correo, contacto, fecha de nacimiento, contraseña.

Clases hijas (tipos de usuario): médico, paciente, administrador.

Médico: atributos heredados del usuario, especialidad médica, número de licencia.

Paciente: atributos heredados del usuario, grupo sanguíneo, diagnósticos, eps.

Administrador: atributos heredados del usuario.

Datos necesarios para el diagnóstico: cédula del paciente, nombre del paciente, tratamiento, descripción.

Datos necesarios para las citas médicas: id de la cita, fecha, hora, motivo, notas previas, sala.

Especialidades médicas: Pediatría, cardiología, dermatología, neurología, medicina general.

Grupos sanguíneos: A+, A-, B+, B-, AB+, AB-, O+, O-.

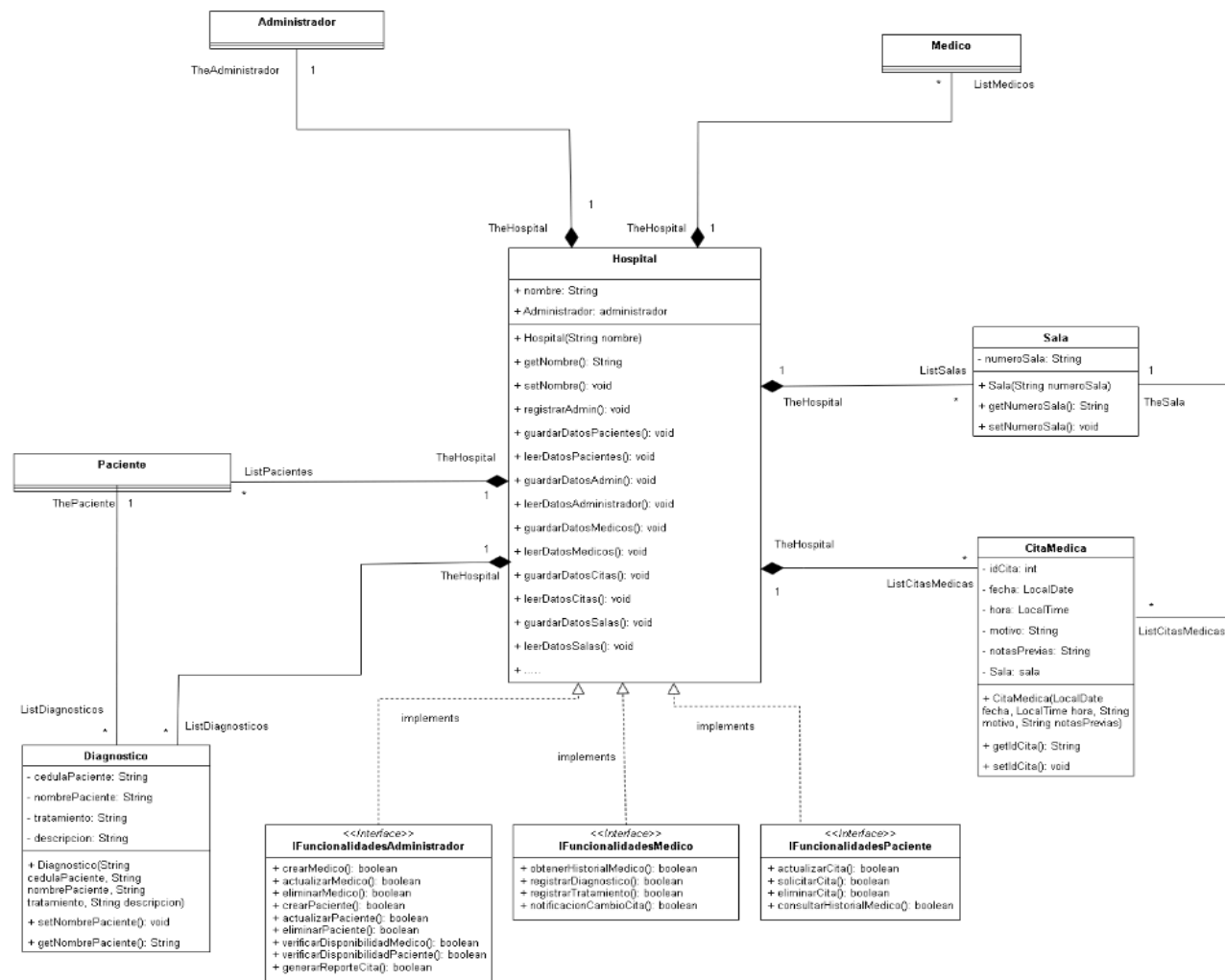
Hospital: nombre, administrador, citas médicas, pacientes, médicos, historiales médicos.

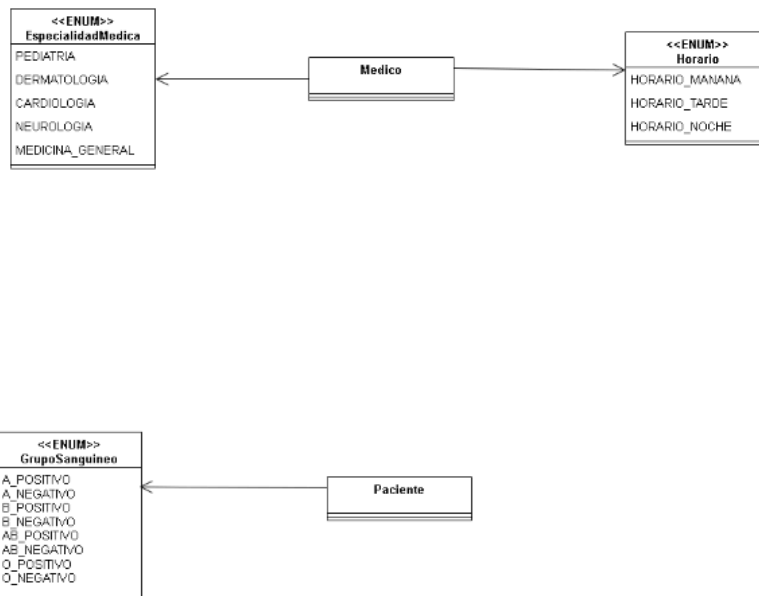
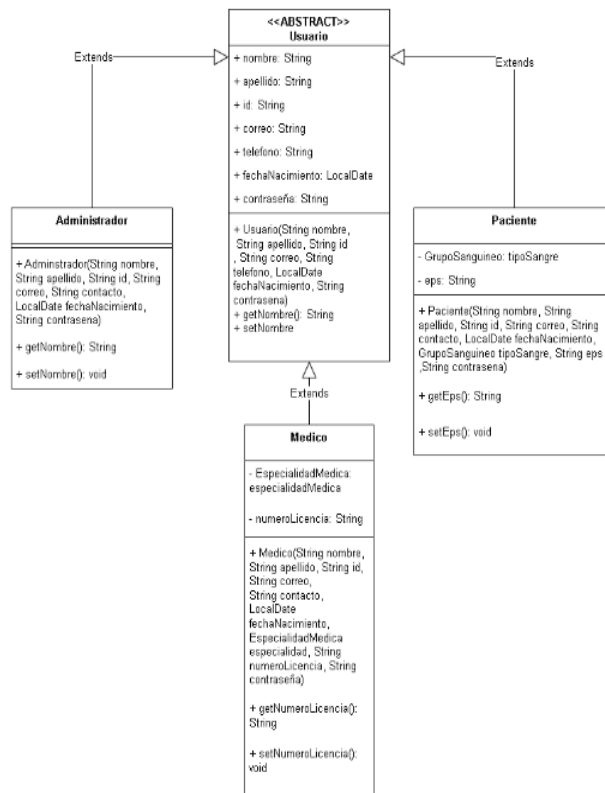
¿Qué funcionalidades se requieren?

Funcionalidades del paciente: actualizar información, solicitar cita, eliminar cita, consultar historial médico.

Funcionalidades del médico: obtener historial médico, registrar diagnóstico, registrar tratamiento, notificación de cambio de cita.

Diagrama UML





2. Descomposición

¿Cómo puedo probar la solución de las funcionalidades en java?

Prueba	Entrada de datos	Salida (Resultados)
assertNotNull (Registrar Administrador)	hospital.administrador	Not null
assertDoesNotThrow (Registrar Paciente)	hospital.registrarPaciente(...)	No lanza excepción
assertDoesNotThrow (Eliminar Paciente)	hospital.eliminarPaciente("123")	No lanza excepción
assertEquals (Actualizar Paciente)	hospital.listPacientes.getFirst().getNombre()	"Nuevo"
assertDoesNotThrow (Registrar Médico)	hospital.registrarMedico(...)	No lanza excepción
assertDoesNotThrow (Eliminar Médico)	hospital.eliminarMedico("12433")	No lanza excepción
assertDoesNotThrow (Registrar Sala)	hospital.registrarSala("28")	No lanza excepción
assertDoesNotThrow (Registrar Cita Médica)	hospital.registrarCitaMedica(...)	No lanza excepción
assertDoesNotThrow (Eliminar Cita Médica)	hospital.eliminarCitaMedica(idCita)	No lanza excepción
assertEquals (Registrar Diagnóstico)	hospital.listDiagnosticos.size()	1
assertNotNull (Obtener Paciente por Cédula)	hospital.obtenerPacienteCedula("123")	Not null
assertNotNull (Obtener Médico por Cédula)	hospital.obtenerMedicoCedula("123")	Not null
assertTrue (Iniciar sesión)	hospital.iniciarSesion("paciente@test.com", "pass")	True
assertNotNull (Obtener Usuario por Correo)	hospital.obtenerUsuarioCorreo("paciente@test.com")	Not null

assertEquals (Cambiar Contraseña)	hospital.listPacientes.getFirst().getContraseña()	"newpass"
assertNotNull (Generar ID de Cita)	hospital.generarIdCita()	Not null
assertNotNull (Listar Médicos)	hospital.listarMedicosHospital()	Not null
assertNotNull (Listar Pacientes)	hospital.listarPacientesHospital()	Not null
assertNotNull (Listar Citas)	hospital.listarCitasHospital()	Not null
assertNotNull (Listar Salas)	hospital.listarSalasHospital()	Not null
assertNotNull (Listar Grupos Sanguíneos)	hospital.listarGruposSanguineos()	Not null
assertNotNull (Listar Especialidades Médicas)	hospital.listarEspecialidadesMedicas()	Not null
assertNotNull (Horarios disponibles para cita)	hospital.horariosDisponiblesCita()	Not null
assertNotNull (Horarios a elegir para médico)	hospital.horariosAElegirMedico()	Not null