



INFORME DE PROYECTO

I. PORTADA

Tema:	Planificación de proyecto Hospitalario
Unidad de Organización Curricular:	PROFESIONAL
Nivel y Paralelo:	5to “A”
Alumnos participantes:	Cholota Guaman Carlos Sebastian Mazabanda Pilamunga Diego Abraham Tixilema Puaquiza Kevin Alexander Tubon Chipantiza Danilo Alexander
Asignatura:	Sistemas de Bases De Datos Distribuidos
Docente:	Ing. Caiza Caizabano Jose Ruben

II. INFORME DE GUÍA PRÁCTICA

2.1 Objetivos

General:

Planificar el desarrollo del proyecto para la correcta implementación de una solución distribuida en el proyecto la gestión hospitalaria.

Específicos:

- Generar los requerimientos funcionales y no funcionales para tener un correcto diseño del proyecto
- Diseñar el esquema y modelo de la base de datos
- Diseñar la estructura o esquema a utilizar en el backend

2.2 Modalidad

Presencial

2.3 Tiempo de duración

Presenciales: 6

No presenciales: 0

2.4 Instrucciones

2.4.1 **Generar requerimientos funcionales y no funcionales**

- Identificar actores del sistema
- Levantar requerimientos funcionales
- Definir los requerimientos no funcionales

2.4.2 **Diseñar el esquema y modelo de la base de datos**

- Seleccionar el modelo de distribución
- Elegir el tipo de fragmentación
- Diseñar el modelo entidad-relacion
- Implementar el modelo en SQL Server

2.4.3 **Diseñar la estructura o esquema para el backend**

- Aplicar arquitectura MVC
- Definir estructura modular del proyecto

2.5 Listado de equipos, materiales y recursos

Listado de equipos y materiales generales empleados en la guía práctica:

- Docker DevOps
- Notion



- Editor de texto (Word)
- Internet

TAC (Tecnologías para el Aprendizaje y Conocimiento) empleados en la guía práctica:

- ☒ Plataformas educativas
- ☒ Simuladores y laboratorios virtuales
- ☐ Aplicaciones educativas
- ☐ Recursos audiovisuales
- ☐ Gamificación
- ☐ Inteligencia Artificial

Otros (Especifique): _____

2.6 Actividades desarrolladas

2.6.1 Generar requerimientos funcionales y no funcionales

- Identificar actores del sistema:
pacientes, médicos, administradores, personal técnico.
- Levantar requerimientos funcionales:
Registro y consulta de pacientes.
Gestión de citas médicas.
Administración de centros médicos y especialidades.
Acceso segmentado por roles (autenticación y permisos).
- Definir los requerimientos no funcionales:
Disponibilidad: replicación de datos entre nodos.
Escalabilidad: uso de contenedores Docker.
Seguridad: cifrado de datos, control de acceso.
Rendimiento: consultas optimizadas y fragmentación de datos.

2.6.2 Diseñar el esquema y modelo de la base de datos

- Seleccionar el modelo de distribución:
Distribuido (varios nodos con replicación).
- Elegir el tipo de fragmentación:
Híbrida: combinación según necesidades.
- Diseñar el modelo entidad-relacion:
Entidades clave: Paciente, Médico, Especialidad, Centro Médico, Consulta.
Relaciones: uno a muchos (Centro → Médicos), muchos a muchos (Paciente ↔ Consulta).
- Implementar el modelo en SQL Server

2.6.3 Diseñar la estructura o esquema para el backend

- Aplicar arquitectura MVC
Modelo: lógica de negocio y conexión con la base distribuida.
Controlador: endpoints RESTful para CRUD (Express.js).
Vista: respuestas JSON estructuradas para frontend.
- Definir estructura modular del proyecto



2.7 Resultados obtenidos

2.7.1 Generar requerimientos funcionales y no funcionales Requerimientos Funcionales

Número	Requerimiento	Prioridad
RF1	El sistema debe permitir que cada usuario inicie sesión mediante un nombre de usuario y contraseña.	Alta
RF2	Las credenciales deben ser verificadas frente a la tabla Usuarios en la base de datos.	Alta
RF3	El sistema debe implementar autenticación por tokens (JWT) o sesiones seguras para mantener el acceso del usuario.	Alta
RF4	El sistema debe permitir el cierre de sesión (logout) de manera segura.	Alta
RF5	Solo los usuarios autenticados podrán acceder a los módulos del sistema según su rol.	Alta
RF6	Los roles definen los permisos de acceso: Administrador, Médico, Empleado, Paciente.	Alta
RF7	El sistema debe permitir la recuperación o restablecimiento de contraseña.	Alta
RF8	Se debe registrar en un log de auditoría los intentos de inicio de sesión (exitosos o fallidos).	Alta
RF9	El sistema debe bloquear temporalmente un usuario tras varios intentos fallidos consecutivos.	Alta
RF10	El sistema debe permitir registrar nuevos centros médicos (nombre, código, dirección, ciudad, teléfono, sede).	Alta
RF11	Cada centro médico debe tener su propia base de datos local sincronizada con la base central.	Alta
RF12	Se debe poder consultar y actualizar la información de los centros médicos desde la sede central.	Alta
RF13	El sistema debe mostrar estadísticas generales (número de médicos, consultas, empleados) por centro médico.	Media/Alta
RF14	El sistema debe permitir registrar, consultar, modificar y eliminar médicos.	Alta
RF15	Cada médico debe estar asociado a una o varias especialidades médicas.	Alta
RF16	El sistema debe permitir registrar nuevas especialidades y asignarlas a los médicos.	Alta
RF17	Se debe poder consultar el listado de médicos por especialidad y por centro médico.	Media
RF18	Los médicos deben tener datos personales (nombres, apellidos, cédula, email, teléfono, dirección) y laborales (fecha de ingreso, centro, especialidad).	Alta
RF19	El sistema debe permitir registrar empleados administrativos (receptionistas, enfermeros, técnicos, etc.).	Alta
RF20	Cada empleado debe estar asociado a un centro médico y a un cargo específico.	Alta
RF21	Los empleados deben poder gestionar agendas, reportes y consultas médicas según su rol.	Media/Alta
RF22	El sistema debe permitir registrar pacientes (cédula, nombres, apellidos, dirección, fecha de nacimiento, teléfono, email).	Alta
RF23	El sistema debe guardar el historial de consultas de cada paciente.	Alta



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: AGOSTO – ENERO 2025



RF24	Se debe poder buscar pacientes por cédula o nombre desde cualquier sede.	Alta
RF25	El sistema debe permitir ver el historial médico completo de un paciente desde la base central.	Alta
RF26	El sistema debe registrar consultas médicas realizadas en cada centro.	Alta
RF27	Cada consulta debe incluir: paciente, médico, diagnóstico, fecha, motivo, tratamientos indicados y observaciones.	Alta
RF28	Las consultas deben almacenarse localmente y sincronizarse con la base central.	Alta
RF29	El sistema debe permitir consultar las citas por fecha, médico o paciente.	Alta
RF30	El sistema debe permitir cancelar, reagendar o actualizar consultas.	Alta
RF31	El sistema debe permitir emitir recetas médicas asociadas a las consultas.	Alta
RF32	Cada receta debe incluir los medicamentos, dosis y duración del tratamiento.	Alta
RF33	Los medicamentos deben estar registrados en un catálogo general con su descripción, fabricante y tipo.	Media/Alta
RF34	El sistema debe permitir consultar el historial de recetas emitidas a cada paciente.	Alta
RF35	El sistema debe generar facturas por las consultas, tratamientos o servicios prestados.	Alta
RF36	Cada factura debe estar asociada a un paciente y a un centro médico.	Alta
RF37	Debe registrar métodos de pago (efectivo, tarjeta, seguro médico).	Alta
RF38	El sistema debe generar reportes de ingresos por centro y periodo.	Media/Alta
RF39	El sistema debe permitir la autenticación de usuarios mediante nombre de usuario y contraseña.	Alta
RF40	Cada usuario debe tener un rol que determine sus permisos.	Alta
RF41	Los administradores deben poder crear, modificar y eliminar cuentas de usuario.	Alta
RF42	El sistema debe registrar las acciones importantes en un log de auditoría.	Alta
RF43	El sistema debe mantener sincronización de datos entre sedes mediante replicación distribuida.	Alta
RF44	Las operaciones CRUD realizadas en una sede deben reflejarse en la base central y en las demás sedes.	Alta
RF45	Debe permitir la reconstrucción de información global a partir de los fragmentos distribuidos.	Media/Alta
RF46	En caso de desconexión temporal, los nodos deben almacenar los cambios y sincronizarlos cuando se restablezca la conexión.	Alta
RF47	El sistema debe generar reportes sobre: consultas por especialidad, pacientes atendidos por médico, ingresos por centro y periodo, actividad y rendimiento de médicos.	Alta
RF48	Los reportes deben poder exportarse a PDF o Excel.	Alta
RF49	El sistema debe permitir visualizar gráficos estadísticos en la interfaz administrativa.	Media/Alta

Tabla 1. Requerimientos Funcionales



Requerimientos No Funcionales

Número	Requerimiento	Prioridad
RNF1	Soportar al menos 500 transacciones concurrentes sin pérdida de rendimiento.	Alta
RNF2	La replicación entre nodos debe tener latencia máxima de 3 segundos.	Alta
RNF3	Disponibilidad 24/7 con tiempo de inactividad menor al 1% mensual.	Alta
RNF4	Escalabilidad horizontal agregando nuevos nodos o sedes.	Alta
RNF5	Base de datos con consistencia e integridad referencial en todas las transacciones.	Alta
RNF6	Mecanismos de recuperación ante fallos y respaldo periódico.	Alta
RNF7	Datos replicados con integridad mediante replicación sincrónica o asincrónica.	Alta
RNF8	Autenticación y autorización seguras, contraseñas cifradas (bcrypt/SHA-256), manejo de sesiones y validación de tokens.	Alta
RNF9	Conexiones entre sedes cifradas (SSL/VPN).	Alta
RNF10	Acceso restringido por roles y autenticación segura.	Alta
RNF11	Contraseñas almacenadas con hash seguro (bcrypt o similar).	Alta
RNF12	Logs de acceso y eventos críticos.	Alta
RNF13	APIs con autenticación por tokens (JWT) o similar.	Alta
RNF14	Interfaces intuitivas y adaptadas a médicos, administrativos y pacientes.	Media/Alta
RNF15	Aplicación web responsive y compatible con navegadores modernos.	Alta
RNF16	Formularios con validación de cédula, correo y teléfono antes del envío.	Alta
RNF17	Despliegue en la nube (Azure) mediante contenedores Docker.	Alta
RNF18	Pipeline CI/CD en Azure DevOps para integración y despliegue continuo.	Alta
RNF19	Archivos .env seguros para configuración de entornos (puertos, credenciales, variables).	Alta
RNF20	Documentar procedimiento de despliegue y replicación.	Media/Alta
RNF21	Documentación técnica completa: diagrama de arquitectura, modelo ER, esquema de replicación y fragmentación, API (Swagger), manual de usuario y administrador.	Alta

Tabla 2. Requerimientos No Funcionales



2.7.2 Diseñar el esquema y modelo de la base de datos

2.7.2.1 Diseño Conceptual: Modelo Entidad–Relación (E/R) del Sistema Hospitalario Distribuido

Entidades Principales y Relaciones

A continuación se describen las entidades, sus atributos clave y las relaciones entre ellas.

El modelo se agrupa por áreas funcionales del sistema:

Módulo de Centros Médicos

Entidad: CentroMedico

Atributo	Tipo	Descripción
CentroID (PK)	INT	Identificador único del centro médico
Nombre	VARCHAR(100)	Nombre del centro (ej. “Centro Médico Quito”)
Ciudad	VARCHAR(100)	Ciudad donde se ubica
Direccion	VARCHAR(150)	Dirección exacta
Telefono	VARCHAR(20)	Teléfono de contacto
Email	VARCHAR(100)	Correo institucional
Sede	VARCHAR(50)	Nombre corto de la sede (HUACHI, INGAHURCO, QUEROCHACA)

Tabla 3. Centro medico

Relaciones:

- Un **CentroMedico** tiene muchos **Médicos**, **Empleados** y **Consultas**.

3. Módulo de Médicos y Especialidades

4. Entidad: Medico

Atributo	Tipo	Descripción
MedicoID (PK)	INT	Identificador único del médico
Cedula	VARCHAR(10)	Cédula o DNI
Nombres	VARCHAR(100)	Nombres del médico
Apellidos	VARCHAR(100)	Apellidos
Email	VARCHAR(100)	Correo institucional
Telefono	VARCHAR(20)	Teléfono
FechaIngreso	DATE	Fecha de ingreso al hospital
CentroID (FK)	INT	Relación con la sede en la que trabaja

Tabla 4. Medico

Relaciones:

- Un **Médico** pertenece a un **CentroMedico**.
- Un **Médico** tiene una o varias **Especialidades**.
- Un **Médico** atiende muchas **Consultas**.



Entidad: Especialidad

Atributo	Tipo	Descripción
EspecialidadID (PK)	INT	Identificador único
Nombre	VARCHAR(100)	Nombre de la especialidad
Descripcion	VARCHAR(255)	Descripción general

Tabla 5.Especialidad

Relaciones:

- Una **Especialidad** puede tener varios **Médicos**.

Entidad Intermedia: MedicoEspecialidad

Atributo	Tipo	Descripción
MedicoID (FK)	INT	Médico asociado
EspecialidadID (FK)	INT	Especialidad asociada

Tabla 6.MedicoEspecialidad

Claves:

- PK compuesta: (MedicoID, EspecialidadID)

3. . Módulo de Empleados

Entidad: Empleado

Atributo	Tipo	Descripción
EmpleadoID (PK)	INT	Identificador único
Cedula	VARCHAR(10)	Documento de identidad
Nombres	VARCHAR(100)	Nombres
Apellidos	VARCHAR(100)	Apellidos
Cargo	VARCHAR(100)	Puesto (enfermero, recepcionista, técnico, etc.)
Telefono	VARCHAR(20)	Teléfono
Email	VARCHAR(100)	Correo institucional
FechaIngreso	DATE	Fecha de ingreso
CentroID (FK)	INT	Sede donde trabaja

Tabla 7.Empleado

Relaciones:

- Un **CentroMedico** tiene muchos **Empleados**.



4. Módulo de Pacientes

Entidad: Paciente

Atributo	Tipo	Descripción
PacienteID (PK)	INT	Identificador único
Cedula	VARCHAR(10)	Cédula
Nombres	VARCHAR(100)	Nombres
Apellidos	VARCHAR(100)	Apellidos
FechaNacimiento	DATE	Fecha de nacimiento
Direccion	VARCHAR(150)	Dirección
Telefono	VARCHAR(20)	Teléfono
Email	VARCHAR(100)	Correo electrónico

Tabla 8. Paciente

Relaciones:

- Un **Paciente** puede tener muchas **Consultas** y **Facturas**.

5. Módulo de Consultas Médicas

Entidad: Consulta

Atributo	Tipo	Descripción
ConsultaID (PK)	INT	Identificador único
FechaConsulta	DATE	Fecha de la cita
Motivo	VARCHAR(255)	Motivo o síntoma
Diagnostico	TEXT	Diagnóstico médico
Observaciones	TEXT	Notas adicionales
MedicoID (FK)	INT	Médico que atiende
PacienteID (FK)	INT	Paciente atendido
CentroID (FK)	INT	Sede donde se realizó

Tabla 9. Consulta

Relaciones:

- Una **Consulta** pertenece a un **Médico**, un **Paciente** y un **CentroMedico**.
- Una **Consulta** puede tener varias **Recetas** y **Facturas** asociadas.

6. Módulo de Recetas y Medicamentos

Entidad: Receta

Atributo	Tipo	Descripción
RecetaID (PK)	INT	Identificador único
ConsultaID (FK)	INT	Consulta asociada
FechaEmision	DATE	Fecha de emisión
Observaciones	TEXT	Instrucciones médicas generales

Tabla 10. Receta



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: AGOSTO – ENERO 2025



Relaciones:

- Una **Receta** pertenece a una **Consulta**.
- Una **Receta** puede contener varios **Medicamentos** (relación N:M).

Entidad: Medicamento

Atributo	Tipo	Descripción
MedicamentoID (PK)	INT	Identificador único
Nombre	VARCHAR(100)	Nombre comercial
Descripcion	VARCHAR(255)	Detalles o composición
Fabricante	VARCHAR(100)	Laboratorio fabricante
Tipo	VARCHAR(50)	Categoría (analgésico, antibiótico, etc.)

Tabla 11. Medicamento

Entidad Intermedia: DetalleReceta

Atributo	Tipo	Descripción
RecetaID (FK)	INT	Receta asociada
MedicamentoID (FK)	INT	Medicamento asociado
Dosis	VARCHAR(50)	Cantidad o dosis indicada
Duracion	VARCHAR(50)	Días o tiempo de tratamiento

Tabla 12. Detalle Receta

Claves:

- PK compuesta: (RecetaID, MedicamentoID)

7. Módulo de Facturación

Entidad: Factura

Atributo	Tipo	Descripción
FacturaID (PK)	INT	Identificador único
ConsultaID (FK)	INT	Consulta asociada
PacienteID (FK)	INT	Paciente que paga
FechaEmision	DATE	Fecha de emisión
Total	DECIMAL(10,2)	Valor total
MetodoPago	VARCHAR(50)	Efectivo, tarjeta, seguro, etc.

Tabla 13. Factura

Relaciones:

- Una **Factura** pertenece a un **Paciente** y una **Consulta**.



8. Módulo de Seguridad y Usuarios

Entidad: Usuario

Atributo	Tipo	Descripción
UsuarioID (PK)	INT	Identificador único del usuario
NombreUsuario	VARCHAR(50)	Nombre de usuario único para login
Contrasenia	VARCHAR(255)	Contraseña cifrada (hash)
Rol	VARCHAR(50)	Rol asignado (Administrador, Médico, Empleado, Paciente)
Estado	VARCHAR(20)	Activo / Inactivo / Bloqueado
FechaCreacion	DATETIME	Fecha en que se creó la cuenta
UltimoAcceso	DATETIME	Última vez que inició sesión
ReferencialID	INT	FK que apunta a MedicoID, EmpleadoID o PacienteID según el rol

Tabla 14. Usuario

Relaciones:

- Un **Usuario** puede representar un **Médico**, **Empleado** o **Paciente**.
- Los accesos o sesiones del usuario se pueden registrar en una tabla de **auditoría**.

Entidad: AuditoriaAcceso

Atributo	Tipo	Descripción
AuditoriaID (PK)	INT	Identificador del evento
UsuarioID (FK)	INT	Usuario que intentó acceder
FechaHora	DATETIME	Momento del acceso
TipoAccion	VARCHAR(50)	LOGIN, LOGOUT, INTENTO_FALLIDO
IP	VARCHAR(45)	Dirección IP del usuario
Exitoso	BOOLEAN	Indica si el inicio de sesión fue correcto

Tabla 15. Auditoria acceso

Relaciones:

- Un **Usuario** puede tener muchos registros en **AuditoriaAcceso**.



9. Relaciones Globales entre Módulos

Relación	Tipo	Descripción
CentroMedico — Medico	1:N	Un centro tiene varios médicos
CentroMedico — Empleado	1:N	Un centro tiene varios empleados
CentroMedico — Consulta	1:N	Un centro registra muchas consultas
Medico — Especialidad	N:M	Un médico puede tener varias especialidades
Medico — Consulta	1:N	Un médico atiende muchas consultas
Paciente — Consulta	1:N	Un paciente puede tener muchas consultas
Consulta — Receta	1:N	Una consulta puede generar varias recetas
Receta — Medicamento	N:M	Una receta puede contener varios medicamentos
Consulta — Factura	1:1	Una consulta genera una factura
Usuario — Medico/Empleado/Paciente	1:1	Un usuario está vinculado a una persona en el sistema

Tabla 16. Relaciones de módulos

Diagramas UML - Modelado Conceptual por Módulos

1. Módulo de Centros Médicos

```
@startuml
entity "CentroMedico" as CentroMedico {
    *CentroID : INT <<PK>>
    --
    Nombre : VARCHAR(100)
    Ciudad : VARCHAR(100)
    Direccion : VARCHAR(150)
    Telefono : VARCHAR(20)
    Email : VARCHAR(100)
    Sede : VARCHAR(50)
}

CentroMedico ||--o{ Medico : "tiene"
CentroMedico ||--o{ Empleado : "tiene"
CentroMedico ||--o{ Consulta : "tiene"

@enduml
```

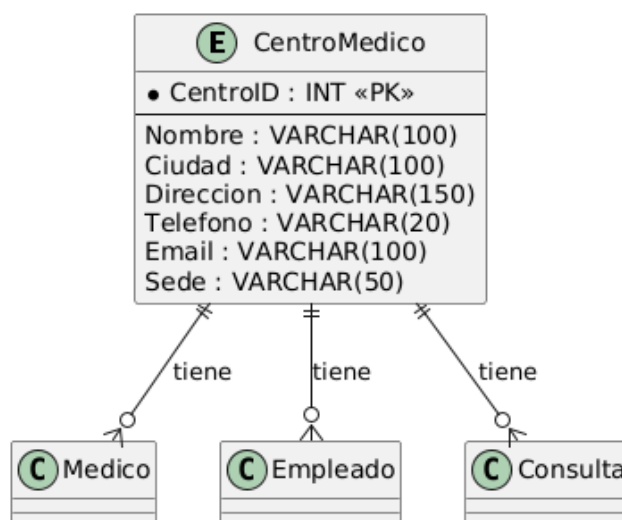


Ilustración 1. Diagrama centro medico



2. Módulo de Médicos y Especialidades

```
@startuml
entity "Medico" as Medico {
  *MedicoID : INT <<PK>>
  Cedula : VARCHAR(10)
  Nombres : VARCHAR(100)
  Apellidos : VARCHAR(100)
  Email : VARCHAR(100)
  Telefono : VARCHAR(20)
  FechaIngreso : DATE
  CentroID : INT <<FK>>
}

entity "Especialidad" as Especialidad {
  *EspecialidadID : INT <<PK>>
  Nombre : VARCHAR(100)
  Descripcion : VARCHAR(255)
}

entity "MedicoEspecialidad" as MedicoEspecialidad {
  *MedicoID : INT <<FK>>
  *EspecialidadID : INT <<FK>>
}

Medico ||--o{ MedicoEspecialidad : "posee"
Especialidad ||--o{ MedicoEspecialidad : "es parte de"
CentroMedico ||--o{ Medico : "trabaja en"

@enduml
```

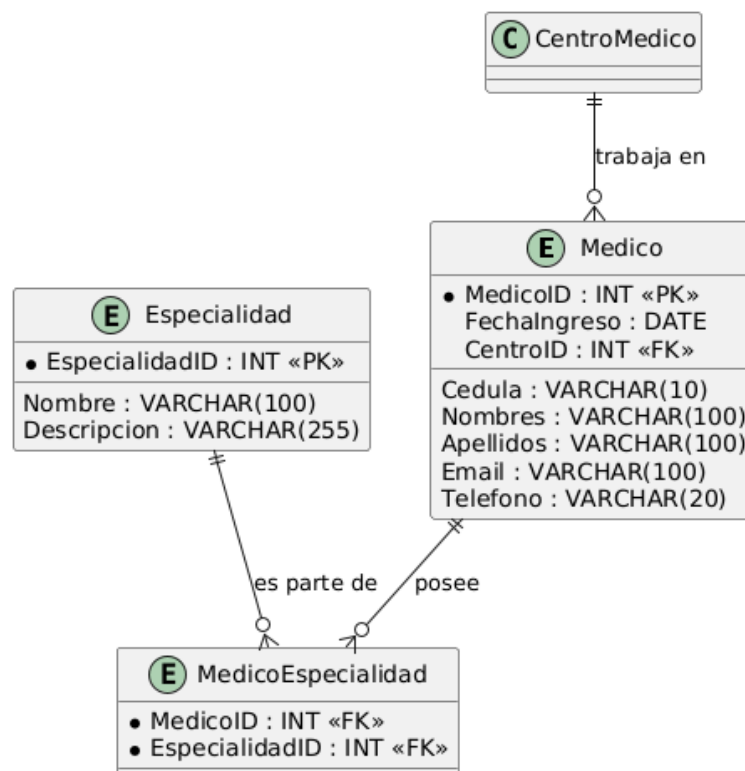


Ilustración 2. Diagrama médicos y especialidades



3. Módulo de Empleados

```
@startuml
entity "Empleado" as Empleado {
  *EmpleadoID : INT <<PK>>
  Cedula : VARCHAR(10)
  Nombres : VARCHAR(100)
  Apellidos : VARCHAR(100)
  Cargo : VARCHAR(100)
  Telefono : VARCHAR(20)
  Email : VARCHAR(100)
  FechaIngreso : DATE
  CentroID : INT <<FK>>
}

CentroMedico ||--o{ Empleado : "tiene"

@enduml
```

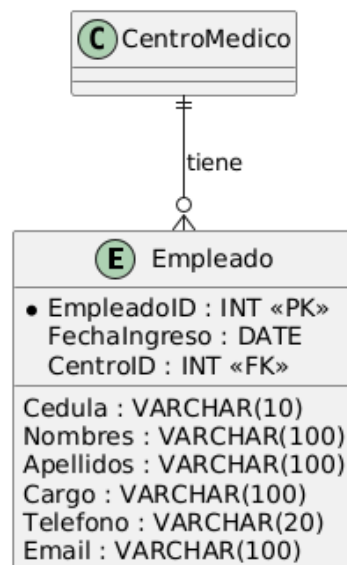


Ilustración 3. Diagrama empleado

4. Módulo de Pacientes

```
@startuml
entity "Paciente" as Paciente {
  *PacienteID : INT <<PK>>
  Cedula : VARCHAR(10)
  Nombres : VARCHAR(100)
  Apellidos : VARCHAR(100)
  FechaNacimiento : DATE
  Direccion : VARCHAR(150)
  Telefono : VARCHAR(20)
  Email : VARCHAR(100)
}

Paciente ||--o{ Consulta : "realiza"
Paciente ||--o{ Factura : "recibe"

@enduml
```

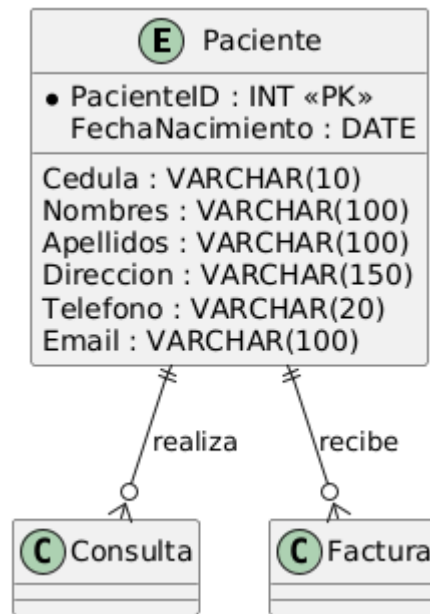


Ilustración 4. Diagrama Modulos de pacientes

5. Módulo de Consultas Médicas

@startuml

```
entity "Consulta" as Consulta {
    *ConsultaID : INT <<PK>>
    FechaConsulta : DATE
    Motivo : VARCHAR(255)
    Diagnostico : TEXT
    Observaciones : TEXT
    MedicoID : INT <<FK>>
    PacienteID : INT <<FK>>
    CentroID : INT <<FK>>
}
```

Medico ||--o{ Consulta : "atiende"

Paciente ||--o{ Consulta : "recibe"

CentroMedico ||--o{ Consulta : "realiza"

Consulta ||--o{ Receta : "genera"

Consulta ||--o{ Factura : "factura"

@enduml

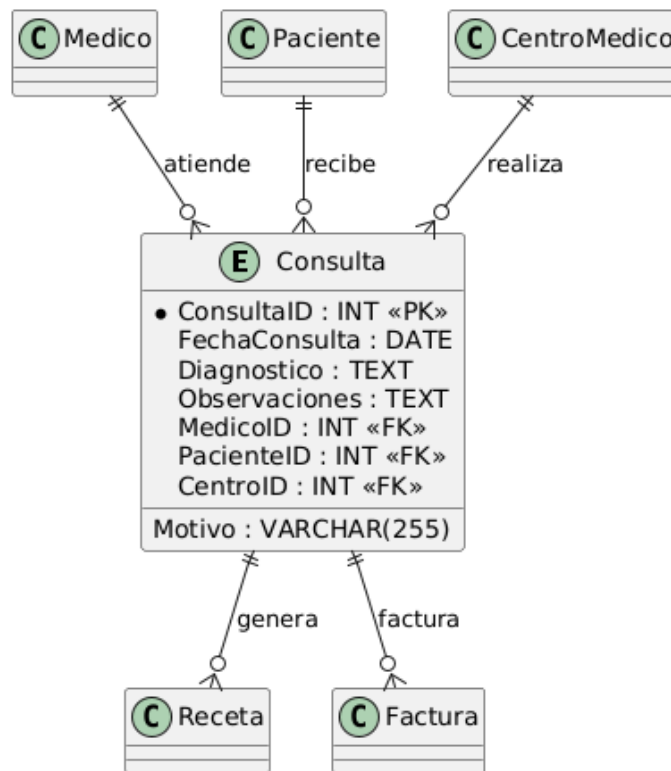


Ilustración 5. Diagrama Módulos consultas

6. Módulo de Recetas y Medicamentos

@startuml

```

entity "Receta" as Receta {
    *RecetaID : INT <<PK>>
    ConsultaID : INT <<FK>>
    FechaEmision : DATE
    Observaciones : TEXT
}
    
```

```

entity "Medicamento" as Medicamento {
    *MedicamentoID : INT <<PK>>
    Nombre : VARCHAR(100)
    Descripcion : VARCHAR(255)
    Fabricante : VARCHAR(100)
    Tipo : VARCHAR(50)
}
    
```

```

entity "DetalleReceta" as DetalleReceta {
    *RecetaID : INT <<FK>>
    *MedicamentoID : INT <<FK>>
    Dosis : VARCHAR(50)
    Duracion : VARCHAR(50)
}
    
```

```

Receta ||--o{ DetalleReceta : "contiene"
Medicamento ||--o{ DetalleReceta : "se prescribe en"
Consulta ||--o{ Receta : "genera"
    
```

@enduml

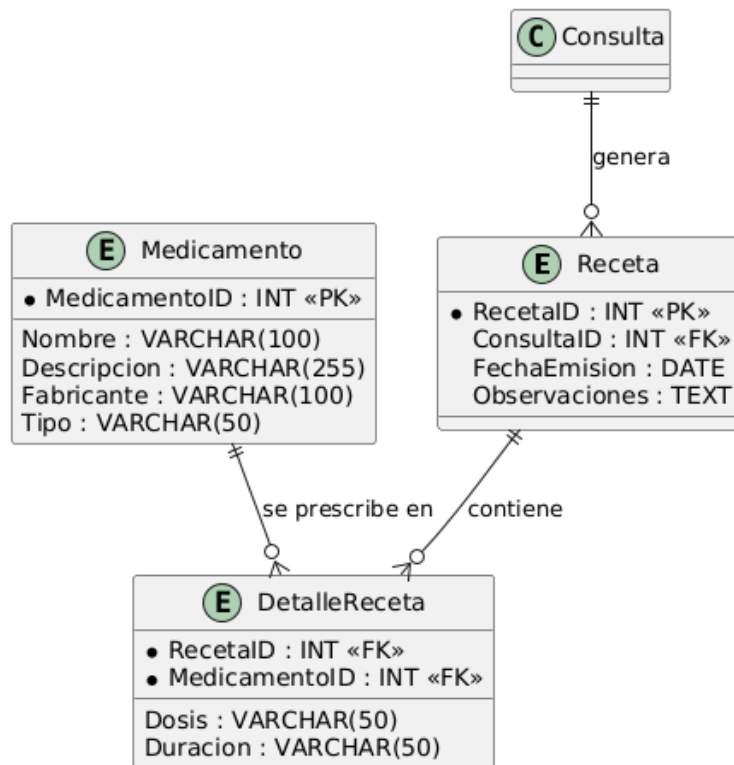


Ilustración 6. Diagrama Módulos recetas

7. Módulo de Facturación

@startuml

```

entity "Factura" as Factura {
    *FacturaID : INT <<PK>>
    ConsultaID : INT <<FK>>
    PacienteID : INT <<FK>>
    FechaEmision : DATE
    Total : DECIMAL(10,2)
    MetodoPago : VARCHAR(50)
}
    
```

Consulta ||--o{ Factura : "genera"

Paciente ||--o{ Factura : "recibe"

@enduml

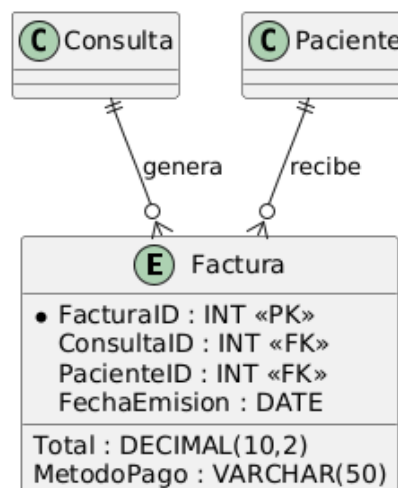


Ilustración 7. Diagrama Módulos factura



8. Módulo de Seguridad y Usuarios

```
@startuml
entity "Usuario" as Usuario {
  *UsuarioID : INT <<PK>>
  NombreUsuario : VARCHAR(50)
  Contraseña : VARCHAR(255)
  Rol : VARCHAR(50)
  Estado : VARCHAR(20)
  FechaCreacion : DATETIME
  UltimoAcceso : DATETIME
  ReferenciaID : INT <<FK>>
}

entity "AuditoriaAcceso" as AuditoriaAcceso {
  *AuditoriaID : INT <<PK>>
  UsuarioID : INT <<FK>>
  FechaHora : DATETIME
  TipoAccion : VARCHAR(50)
  IP : VARCHAR(45)
  Exitoso : BOOLEAN
}

Usuario ||--o{ AuditoriaAcceso : "registra"
Usuario ||--o{ Medico : "refiere a (rol Médico)"
Usuario ||--o{ Empleado : "refiere a (rol Empleado)"
Usuario ||--o{ Paciente : "refiere a (rol Paciente)"

@enduml
```

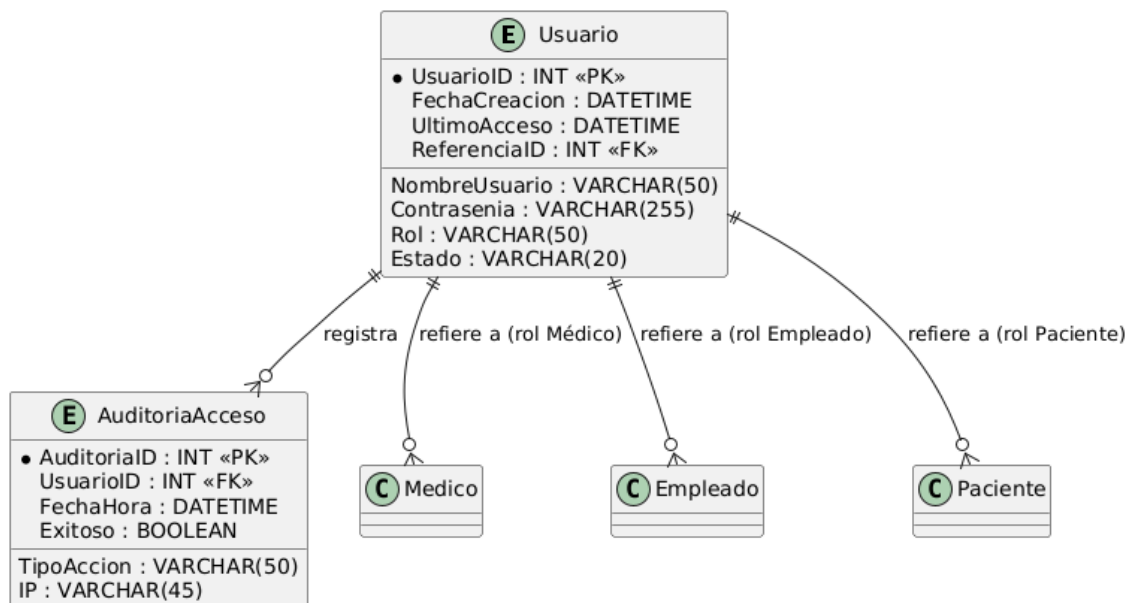


Ilustración 8. Diagrama Módulos seguridad

9. Relaciones Globales entre Módulos - General

```
@startuml
CentroMedico ||--o{ Medico
CentroMedico ||--o{ Empleado
CentroMedico ||--o{ Consulta
Medico ||--o{ Consulta
Paciente ||--o{ Consulta
Consulta ||--o{ Receta
Receta ||--o{ DetalleReceta
Medicamento ||--o{ DetalleReceta
Consulta ||--o{ Factura
Paciente ||--o{ Factura
Usuario ||--o{ AuditoriaAcceso
Usuario ||--o{ Medico
Usuario ||--o{ Empleado
Usuario ||--o{ Paciente
@enduml
```

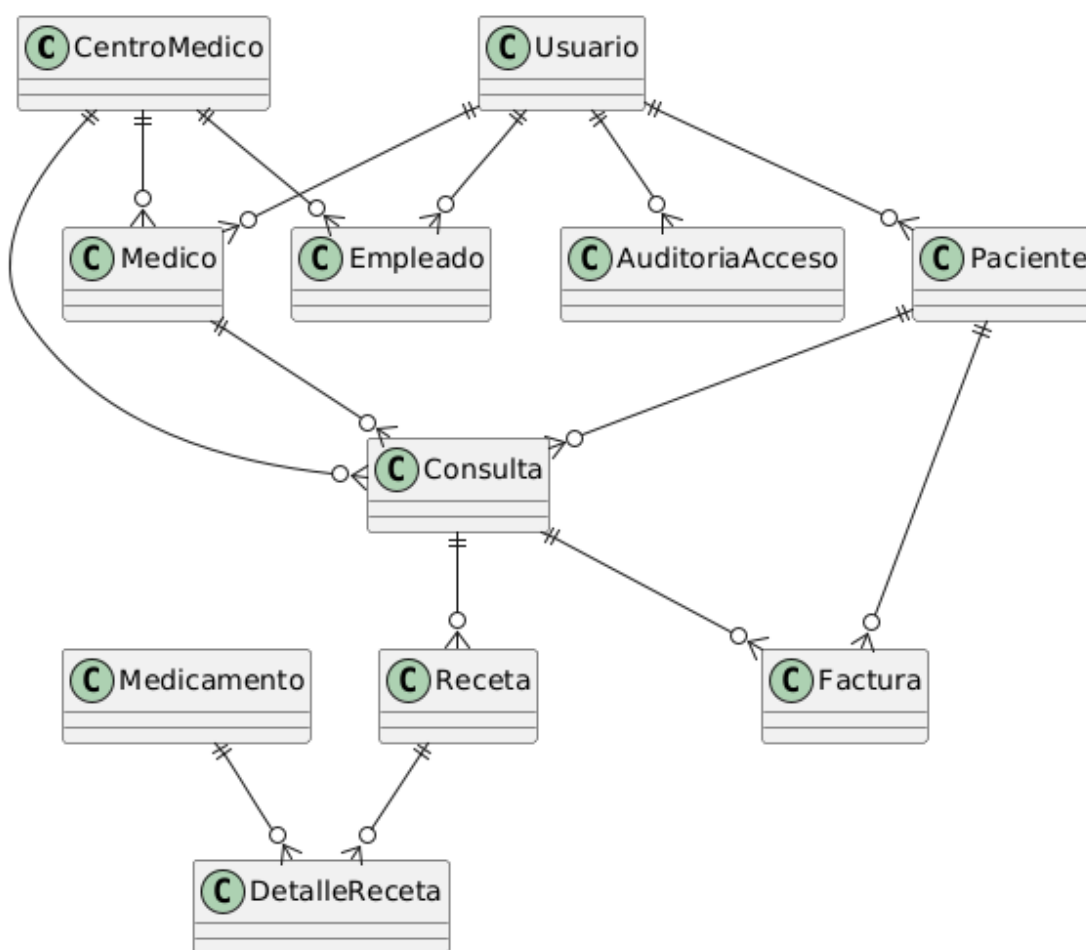


Ilustración 9. Diagrama Modulo de Relación Global



4.1.1 Diseñar la estructura o esquema para el backend

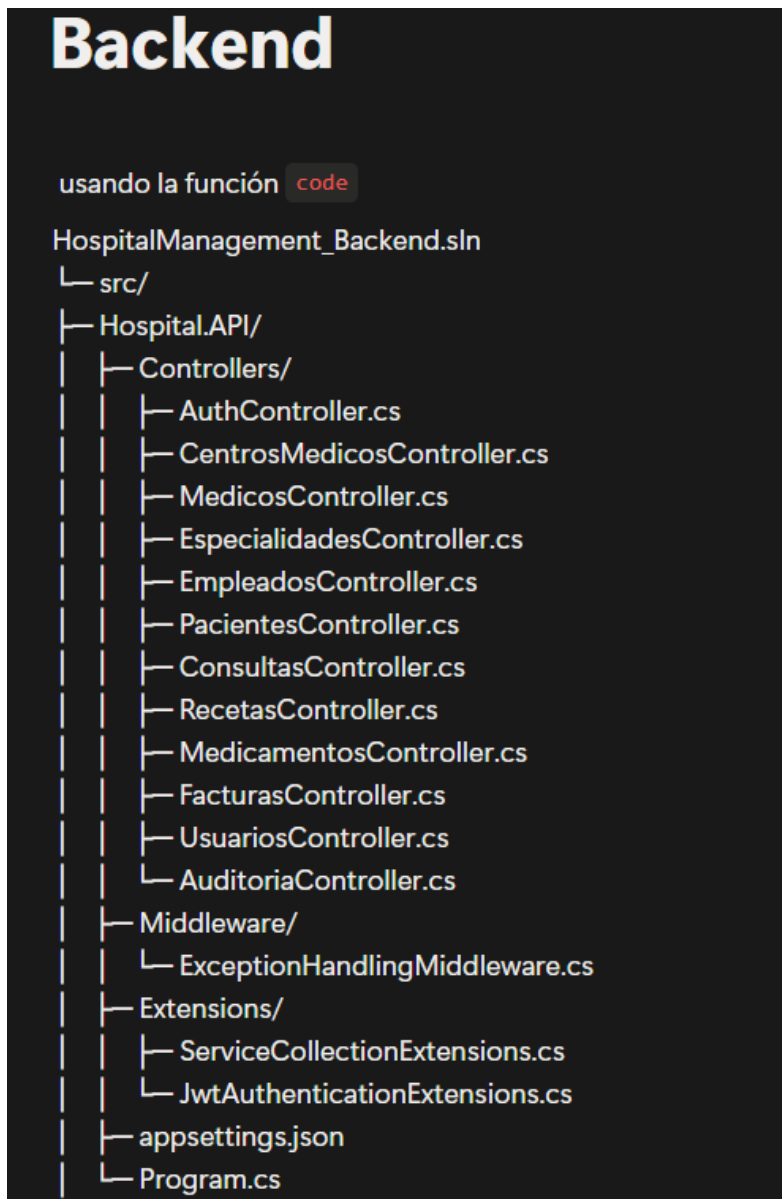


Ilustración 10: Estructura del Backend - Proyecto HospitalManagement

Descripción: Organización de la solución .NET (HospitalManagement_Backend.sln) mostrando la arquitectura modular con controladores por cada entidad del sistema, middleware personalizado y estructura de configuración basada en la arquitectura MVC.



Ilustración 11: Estructura de la Capa de Aplicación (Application Layer)

Descripción: Organización de la capa de servicios de aplicación, mostrando la separación entre interfaces e implementaciones de servicios, DTOs (Data Transfer Objects) organizados por entidades, validadores de datos y configuración de mapeo con AutoMapper

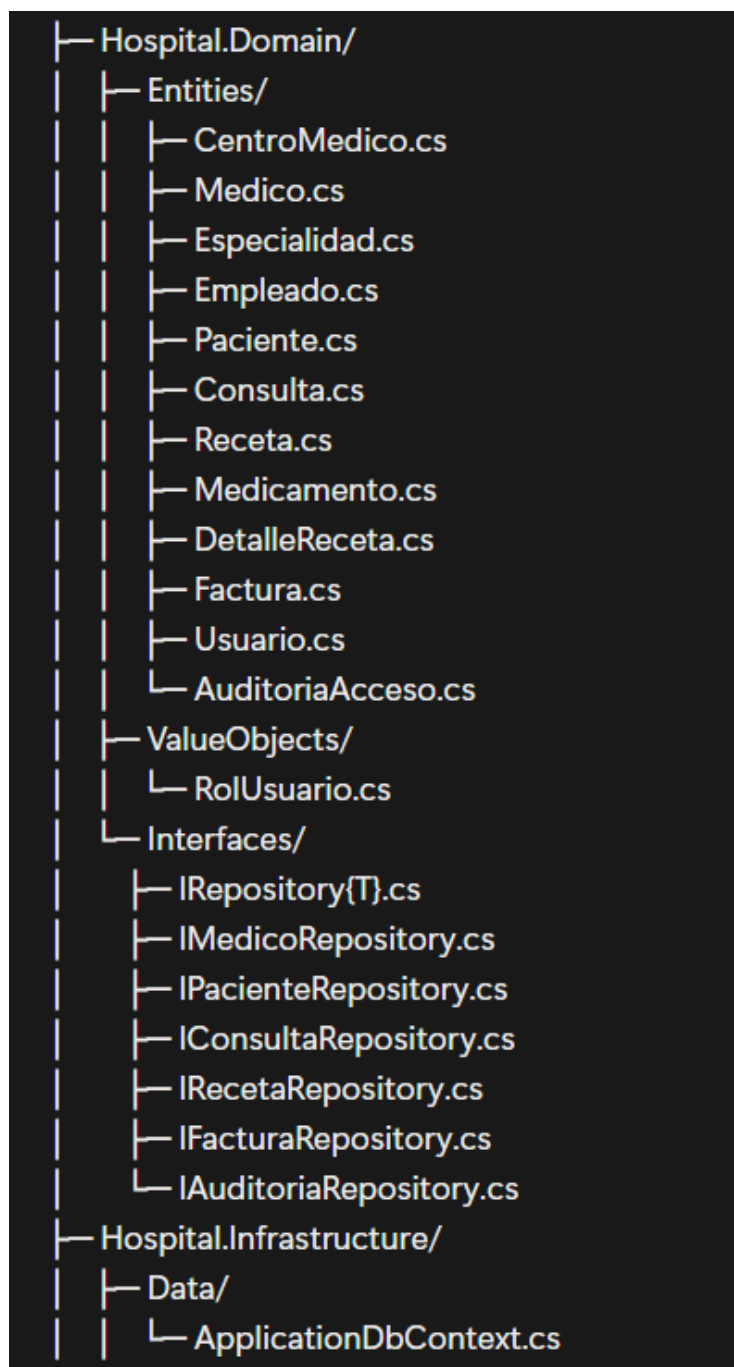


Ilustración 12: Estructura de la Capa de Dominio e Infraestructura (Domain & Infrastructure Layers)

Descripción: Organización de las entidades del dominio, interfaces de repositorio y configuración de infraestructura, mostrando la separación entre la capa de dominio (entidades y contratos) y la capa de infraestructura (persistencia de datos) siguiendo los principios de Domain-Driven Design.

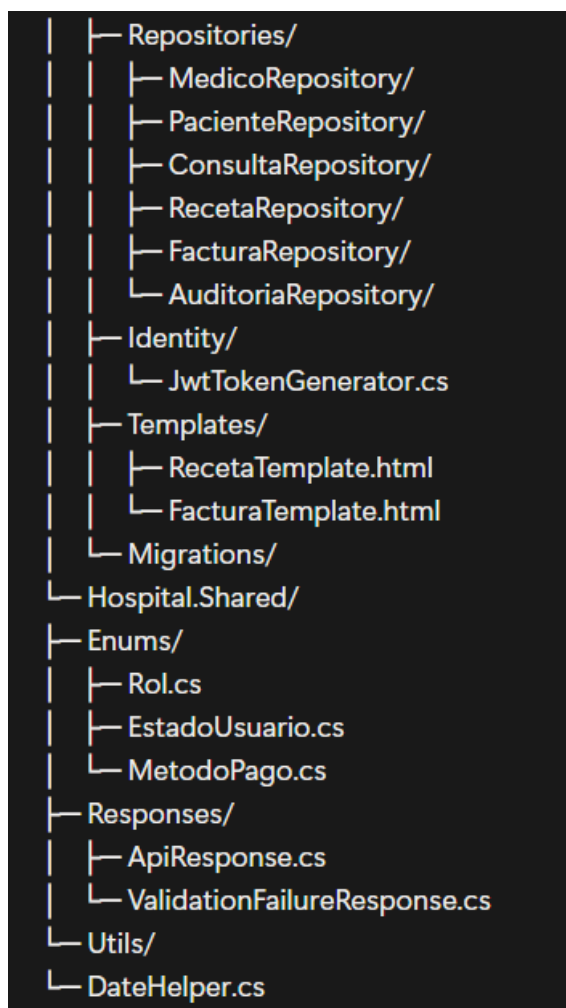


Ilustración 13: Estructura de Repositorios y Capa Compartida (Repositories & Shared Layer)

Descripción: Organización de los repositorios de datos por entidad, servicios de identidad para generación de tokens JWT, plantillas para documentos, migraciones de base de datos y biblioteca compartida con enumeraciones, respuestas API estandarizadas y utilidades comunes del sistema hospitalario.



4.1.2 Creación de la Bases de Datos y sus tablas

Bases de datos “CentroHospitalario”

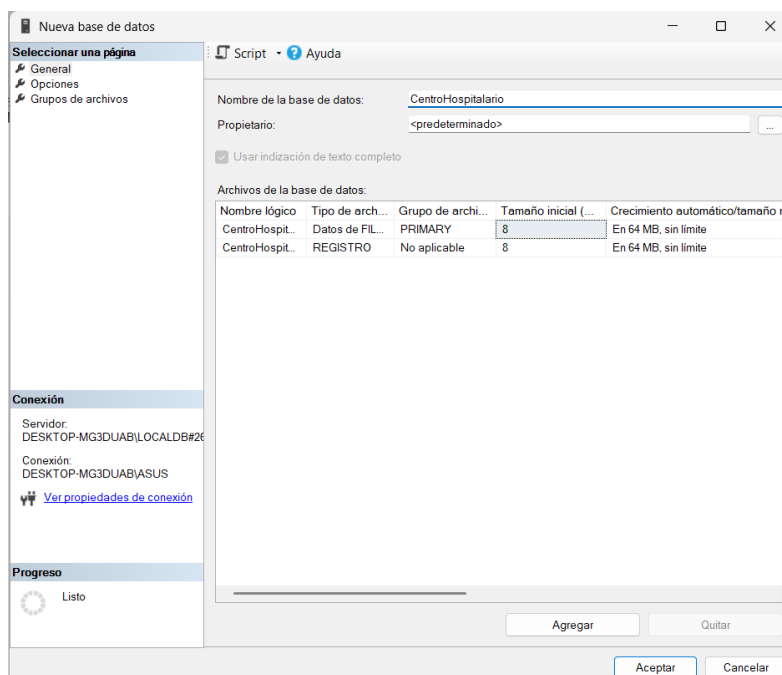


Ilustración 14. Creación de la Bases de Datos

1. Módulo de centros médicos

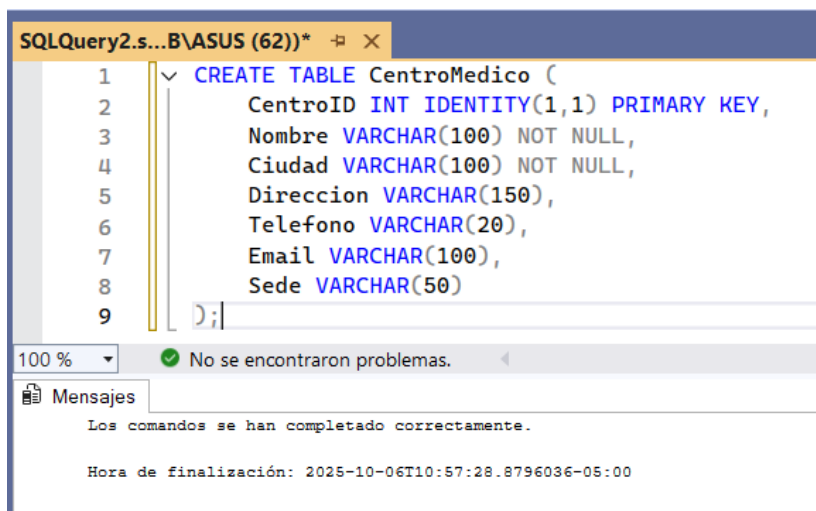


Tabla 17. Centro Medico

2. Módulo de Médicos y Especialidades

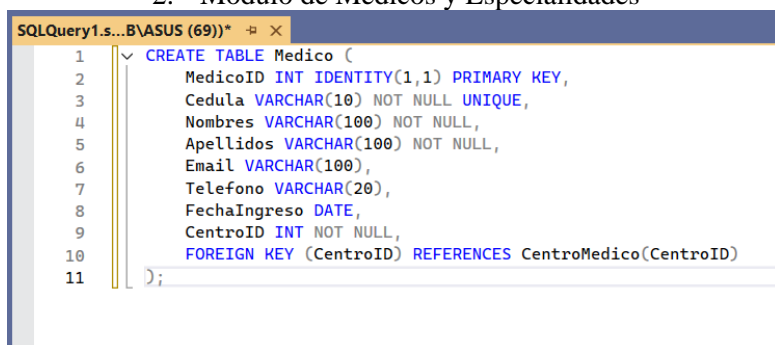


Tabla 18. Medico



```
SQLQuery2.s...B\ASUS (62))*  
1 CREATE TABLE Especialidad (  
2     EspecialidadID INT IDENTITY(1,1) PRIMARY KEY,  
3     Nombre VARCHAR(100) NOT NULL,  
4     Descripcion VARCHAR(255)  
5 );
```

Tabla 19. Especialidad

```
CREATE TABLE MedicoEspecialidad (  
    MedicoID INT NOT NULL,  
    EspecialidadID INT NOT NULL,  
    PRIMARY KEY (MedicoID, EspecialidadID),  
    FOREIGN KEY (MedicoID) REFERENCES Medico(MedicoID),  
    FOREIGN KEY (EspecialidadID) REFERENCES Especialidad(EspecialidadID)  
);
```

Tabla 20. Medico Especialidad

3. Módulo empleados

```
CREATE TABLE Empleado (  
    EmpleadoID INT IDENTITY(1,1) PRIMARY KEY,  
    Cedula VARCHAR(10) NOT NULL UNIQUE,  
    Nombres VARCHAR(100) NOT NULL,  
    Apellidos VARCHAR(100) NOT NULL,  
    Cargo VARCHAR(100),  
    Telefono VARCHAR(20),  
    Email VARCHAR(100),  
    FechaIn tipo de datos varchar(100),  
    CentroID INT NOT NULL,  
    FOREIGN KEY (CentroID) REFERENCES CentroMedico(CentroID)  
);
```

Tabla 21. Empleados

4. Módulo de pacientes

```
CREATE TABLE Paciente (  
    PacienteID INT IDENTITY(1,1) PRIMARY KEY,  
    Cedula VARCHAR(10) NOT NULL UNIQUE,  
    Nombres VARCHAR(100) NOT NULL,  
    Apellidos VARCHAR(100) NOT NULL,  
    FechaNacimiento DATE,  
    Direccion VARCHAR(150),  
    Telefono VARCHAR(20),  
    Email VARCHAR(100)  
);
```

Tabla 22. Paciente

5. Módulo de consultas medicas



```
CREATE TABLE Consulta (  
    ConsultaID INT IDENTITY(1,1) PRIMARY KEY,  
    FechaConsulta DATE NOT NULL,  
    Motivo VARCHAR(255),  
    Diagnostico TEXT,  
    Observaciones TEXT,  
    MedicoID INT NOT NULL,  
    PacienteID INT NOT NULL,  
    CentroID INT NOT NULL,  
    FOREIGN KEY (MedicoID) REFERENCES Medico(MedicoID),  
    FOREIGN KEY (PacienteID) REFERENCES Paciente(PacienteID),  
    FOREIGN KEY (CentroID) REFERENCES CentroMedico(CentroID)  
);
```

Tabla 23. Consulta

6. Modulo de Recetas y Medicamentos

```
CREATE TABLE Receta (  
    RecetaID INT IDENTITY(1,1) PRIMARY KEY,  
    ConsultaID INT NOT NULL,  
    FechaEmision DATE NOT NULL,  
    Observaciones TEXT,  
    FOREIGN KEY (ConsultaID) REFERENCES Consulta(ConsultaID)  
);
```

Tabla 24. Receta

```
CREATE TABLE Medicamento (  
    MedicamentoID INT IDENTITY(1,1) PRIMARY KEY,  
    Nombre VARCHAR(100) NOT NULL,  
    Descripcion VARCHAR(255),  
    Fabricante VARCHAR(100),  
    Tipo VARCHAR(50)  
);
```

Tabla 25. Medicamento

```
CREATE TABLE DetalleReceta (  
    RecetaID INT NOT NULL,  
    MedicamentoID INT NOT NULL,  
    Dosis VARCHAR(50),  
    Duracion VARCHAR(50),  
    PRIMARY KEY (RecetaID, MedicamentoID),  
    FOREIGN KEY (RecetaID) REFERENCES Receta(RecetaID),  
    FOREIGN KEY (MedicamentoID) REFERENCES Medicamento(MedicamentoID)  
);
```

Tabla 26. Detalle Receta

7. Modulo facturación



```
CREATE TABLE Factura (  
    FacturaID INT IDENTITY(1,1) PRIMARY KEY,  
    ConsultaID INT NOT NULL UNIQUE,  
    PacienteID INT NOT NULL,  
    FechaEmision DATE NOT NULL,  
    Total DECIMAL(10,2) NOT NULL,  
    MetodoPago VARCHAR(50),  
    FOREIGN KEY (ConsultaID) REFERENCES Consulta(ConsultaID),  
    FOREIGN KEY (PacienteID) REFERENCES Paciente(PacienteID)  
);
```

Tabla 27. Factura

8. Modulo seguridad y usuario

```
CREATE TABLE Usuario (  
    UsuarioID INT PRIMARY KEY IDENTITY(1,1),  
    NombreUsuario VARCHAR(50) NOT NULL UNIQUE,  
    Contraseña VARCHAR(50) NOT NULL, --> Hash de la contraseña  
    Rol VARCHAR(50) NOT NULL,  
    Estado VARCHAR(20) DEFAULT 'ACTIVO',  
    FechaCreacion DATETIME DEFAULT GETDATE(),  
    UltimoAcceso DATETIME,  
    ReferenciaID INT, --> FK dinámica según el rol  
);
```

Tabla 28. Usuario

```
CREATE TABLE AuditoriaAcceso (  
    AuditoriaID INT IDENTITY(1,1) PRIMARY KEY,  
    UsuarioID INT NOT NULL,  
    FechaHora DATETIME DEFAULT GETDATE(),  
    TipoAccion VARCHAR(50),  
    IP VARCHAR(45),  
    Exitoso BIT,  
    FOREIGN KEY (UsuarioID) REFERENCES Usuario(UsuarioID)  
);
```

Tabla 29. Auditoria Acceso

4.2 Habilidades blandas empleadas en la práctica

- ☐ Liderazgo
- ☒ Trabajo en equipo
- ☐ Comunicación asertiva
- ☐ La empatía
- ☐ Pensamiento crítico
- ☐ Flexibilidad
- ☐ La resolución de conflictos
- ☐ Adaptabilidad
- ☐ Responsabilidad

Trabajo en equipo

El Trabajo en equipo se desarrollará utilizando las herramientas como Azure DevOps y Notion

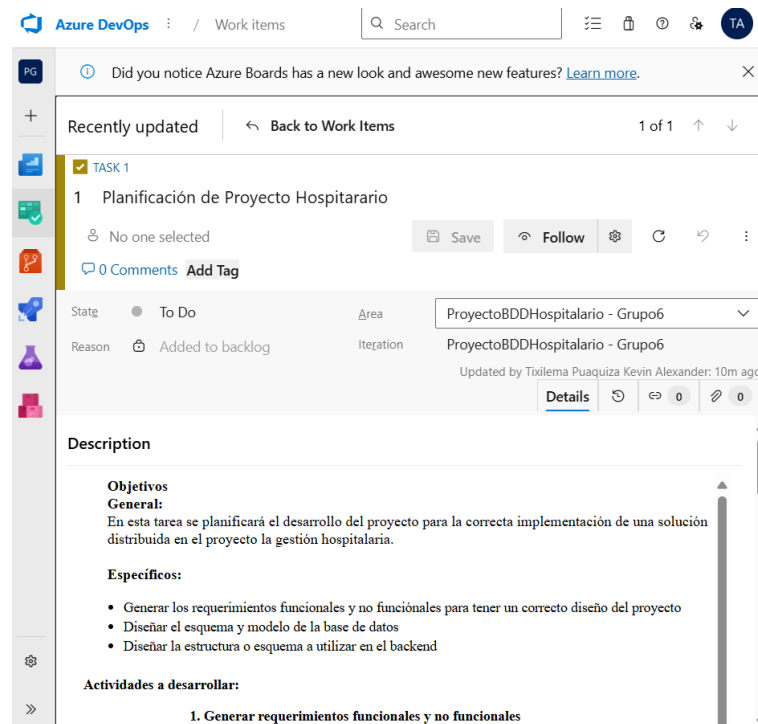
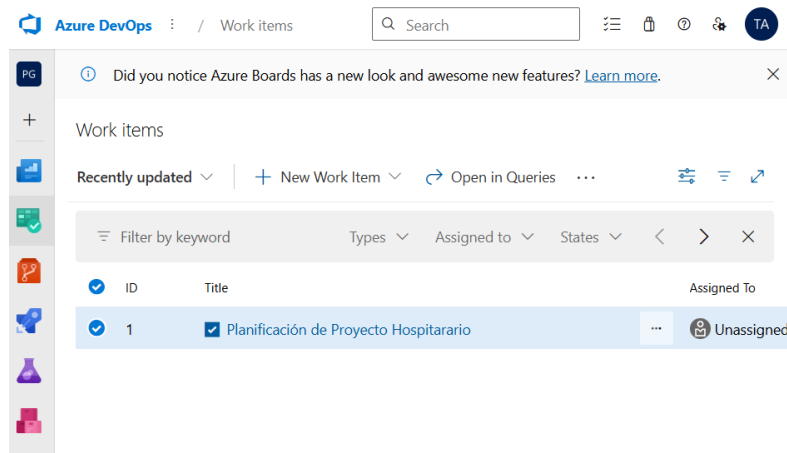
- Azure DevOps



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: AGOSTO – ENERO 2025



Se irá colocando las tareas a medida que se desarrolle el proyecto y así se trabajara en conjunto con los demás integrantes

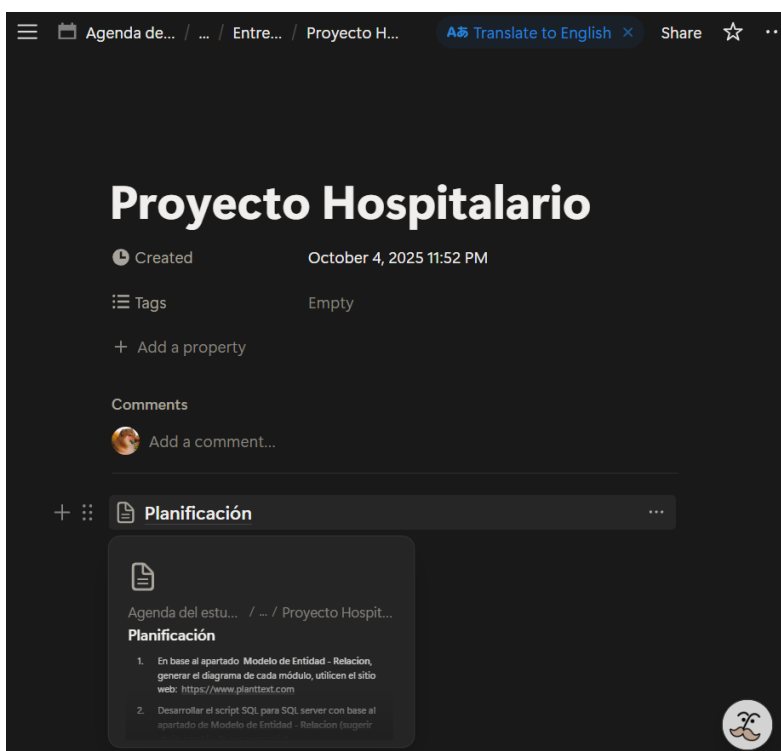


- **Notion**

En este apartado se realiza la planificación más específica, diálogos y sugerencias para realizar algún cambio o implementación adicional



https://www.notion.so/Proyecto-Hospitalario-283b9ff9620f80f3a20ff00569d61581?source=copy_link



4.3 Conclusiones

- Los requerimientos funcionales describen los servicios, operaciones y comportamientos que el sistema debe ofrecer. Se agrupan por módulos o áreas del sistema hospitalario. En cuanto a los requerimientos No funcionales, estos describen las propiedades de calidad, rendimiento, seguridad y mantenimiento del sistema.

4.4 Recomendaciones

Redactar las sugerencias o acciones concretas que se derivan de la guía práctica desarrollada.

4.5 Referencias bibliográficas

Insertar las referencias bibliográficas empleadas aplicando la norma IEEE.

4.6 Anexos

Script

-- 1. Módulo de Centros Médicos

```
CREATE TABLE CentroMedico (  
  CentroID INT IDENTITY(1,1) PRIMARY KEY,  
  Nombre VARCHAR(100) NOT NULL,  
  Ciudad VARCHAR(100) NOT NULL,  
  Direccion VARCHAR(150),  
  Telefono VARCHAR(20),  
  Email VARCHAR(100),  
  Sede VARCHAR(50)  
);
```



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: AGOSTO – ENERO 2025



-- 2. Módulo de Médicos y Especialidades

```
CREATE TABLE Medico (  
    MedicoID INT IDENTITY(1,1) PRIMARY KEY,  
    Cedula VARCHAR(10) NOT NULL UNIQUE,  
    Nombres VARCHAR(100) NOT NULL,  
    Apellidos VARCHAR(100) NOT NULL,  
    Email VARCHAR(100),  
    Telefono VARCHAR(20),  
    FechaIngreso DATE,  
    CentroID INT NOT NULL,  
    FOREIGN KEY (CentroID) REFERENCES CentroMedico(CentroID)  
);  
  
CREATE TABLE Especialidad (  
    EspecialidadID INT IDENTITY(1,1) PRIMARY KEY,  
    Nombre VARCHAR(100) NOT NULL,  
    Descripcion VARCHAR(255)  
);  
  
CREATE TABLE MedicoEspecialidad (  
    MedicoID INT NOT NULL,  
    EspecialidadID INT NOT NULL,  
    PRIMARY KEY (MedicoID, EspecialidadID),  
    FOREIGN KEY (MedicoID) REFERENCES Medico(MedicoID),  
    FOREIGN KEY (EspecialidadID) REFERENCES Especialidad(EspecialidadID)  
);
```

-- 3. Módulo de Empleados

```
CREATE TABLE Empleado (  
    EmpleadoID INT IDENTITY(1,1) PRIMARY KEY,  
    Cedula VARCHAR(10) NOT NULL UNIQUE,  
    Nombres VARCHAR(100) NOT NULL,  
    Apellidos VARCHAR(100) NOT NULL,  
    Cargo VARCHAR(100),  
    Telefono VARCHAR(20),  
    Email VARCHAR(100),  
    FechaIngreso DATE,  
    CentroID INT NOT NULL,  
    FOREIGN KEY (CentroID) REFERENCES CentroMedico(CentroID)  
);
```

-- 4. Módulo de Pacientes

```
CREATE TABLE Paciente (  
    PacienteID INT IDENTITY(1,1) PRIMARY KEY,  
    Cedula VARCHAR(10) NOT NULL UNIQUE,  
    Nombres VARCHAR(100) NOT NULL,  
    Apellidos VARCHAR(100) NOT NULL,  
    FechaNacimiento DATE,  
    Direccion VARCHAR(150),  
    Telefono VARCHAR(20),  
    Email VARCHAR(100)  
);
```

-- 5. Módulo de Consultas Médicas

```
CREATE TABLE Consulta (  
    ConsultaID INT IDENTITY(1,1) PRIMARY KEY,
```




UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: AGOSTO – ENERO 2025



```
FechaConsulta DATE NOT NULL,
Motivo VARCHAR(255),
Diagnostico TEXT,
Observaciones TEXT,
MedicoID INT NOT NULL,
PacienteID INT NOT NULL,
CentroID INT NOT NULL,
FOREIGN KEY (MedicoID) REFERENCES Medico(MedicoID),
FOREIGN KEY (PacienteID) REFERENCES Paciente(PacienteID),
FOREIGN KEY (CentroID) REFERENCES CentroMedico(CentroID)
);

-- 6. Módulo de Recetas y Medicamentos

CREATE TABLE Receta (
    RecetaID INT IDENTITY(1,1) PRIMARY KEY,
    ConsultaID INT NOT NULL,
    FechaEmision DATE NOT NULL,
    Observaciones TEXT,
    FOREIGN KEY (ConsultaID) REFERENCES Consulta(ConsultaID)
);

CREATE TABLE Medicamento (
    MedicamentoID INT IDENTITY(1,1) PRIMARY KEY,
    Nombre VARCHAR(100) NOT NULL,
    Descripcion VARCHAR(255),
    Fabricante VARCHAR(100),
    Tipo VARCHAR(50)
);

CREATE TABLE DetalleReceta (
    RecetaID INT NOT NULL,
    MedicamentoID INT NOT NULL,
    Dosis VARCHAR(50),
    Duracion VARCHAR(50),
    PRIMARY KEY (RecetaID, MedicamentoID),
    FOREIGN KEY (RecetaID) REFERENCES Receta(RecetaID),
    FOREIGN KEY (MedicamentoID) REFERENCES Medicamento(MedicamentoID)
);

-- 7. Módulo de Facturación

CREATE TABLE Factura (
    FacturaID INT IDENTITY(1,1) PRIMARY KEY,
    ConsultaID INT NOT NULL UNIQUE,
    PacienteID INT NOT NULL,
    FechaEmision DATE NOT NULL,
    Total DECIMAL(10,2) NOT NULL,
    MetodoPago VARCHAR(50),
    FOREIGN KEY (ConsultaID) REFERENCES Consulta(ConsultaID),
    FOREIGN KEY (PacienteID) REFERENCES Paciente(PacienteID)
);

-- 8. Módulo de Seguridad y Usuarios

CREATE TABLE Usuario (
    UsuarioID INT PRIMARY KEY IDENTITY(1,1),
    NombreUsuario VARCHAR(50) NOT NULL UNIQUE,
    Contraseña VARCHAR(50) NOT NULL, --> Hash de la contraseña
    Rol VARCHAR(50) NOT NULL,
```



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: AGOSTO – ENERO 2025



```
Estado VARCHAR(20) DEFAULT 'ACTIVO',  
FechaCreacion DATETIME DEFAULT GETDATE(),  
UltimoAcceso DATETIME,  
ReferenciaID INT, --> FK dinámica según el rol
```

```
);
```

```
CREATE TABLE AuditoriaAcceso (  
    AuditoriaID INT IDENTITY(1,1) PRIMARY KEY,  
    UsuarioID INT NOT NULL,  
    FechaHora DATETIME DEFAULT GETDATE(),  
    TipoAccion VARCHAR(50),  
    IP VARCHAR(45),  
    Exitoso BIT,  
    FOREIGN KEY (UsuarioID) REFERENCES Usuario(UsuarioID)  
);
```