



ESCUELA POLITECNICA NACIONAL



FACULTAD DE INGENIERÍA DE SISTEMAS

**Fundamentos de Bases de Datos –
GR1SW**

GRUPO: No. 2

**Manual de la Base de Datos del Proyecto del
Primer Bimestre**

Alumnos: Ayala Bryan, Chugá Juan

FECHA DE ENTREGA: 24-11-2025



Manual de la Base de Datos del Proyecto de Fundamentos de Bases de Datos del Primer Bimestre

1. Detalles generales de la Base de Datos

- Nombre del Sistema: “Sistema De Gestión De Citas Médicas En Consulta Externa Para el IEES”
- Ambiente de desarrollo: el sistema fue desarrollado en lenguaje Java, y con el IDE IntelliJIDEA.
- Autores: Bryan Ayala y Juan Chugá
- Fecha de Creación: 24/11/2025
- Nombre de la Base de Datos: sistemagestioncitas

2. Descripción de la Base de Datos

La base de datos descrita en este manual fue creada para soportar el ingreso y salida de datos de un sistema de gestión de citas creado para ser usado en un centro médico del IEES. Este centro médico soporta tres tipos de usuarios: médicos, pacientes y operadores de admisión del IEES.

Los pacientes pueden generar sus citas con algún médico de su preferencia en alguna fecha y horario libre. Las citas se toman desde las 8:00 hasta las 17:00. Los médicos pueden editar las citas que tienen agendadas con sus pacientes y añadir tratamientos que estos últimos deben seguir. Finalmente, los operadores de admisión pueden editar las citas desde el punto de vista de los pacientes, además de editar los datos de tanto los pacientes como de los médicos, aunque los operadores no pueden modificarse entre sí.

3. Arquitectura General de la Base de Datos

Se enlistan anotaciones con respecto a la implementación de la base de datos:

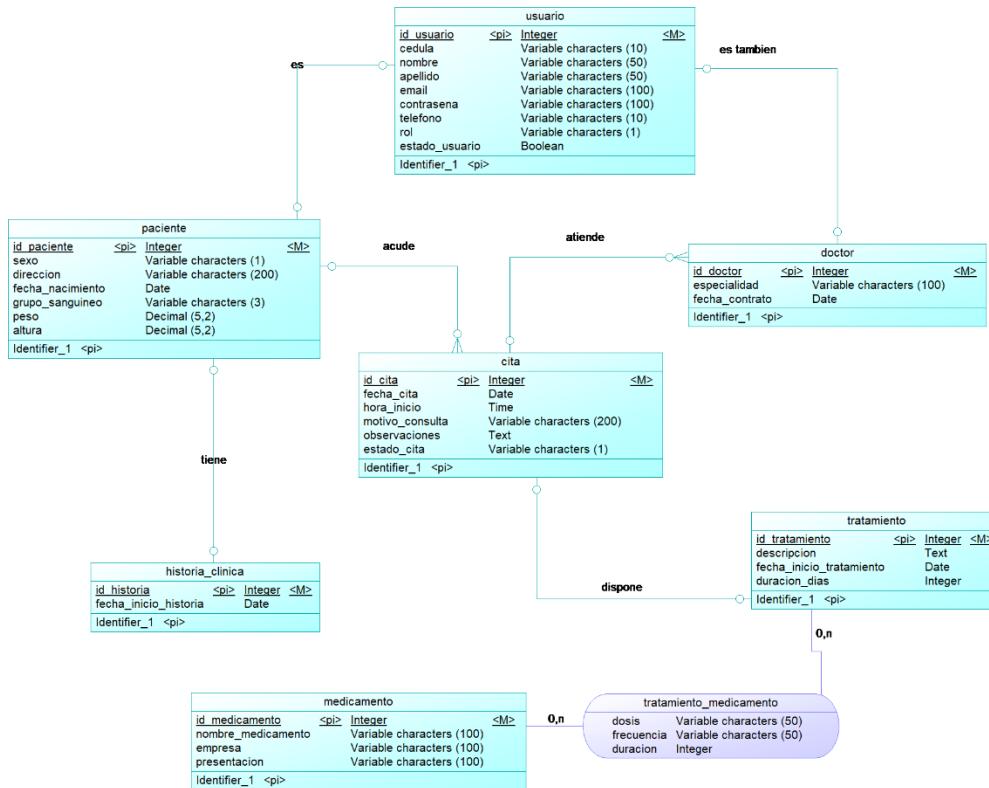
- Se usó *snake_case* como estándar de nombres, además de que se usaron los tipos de datos estándar de MySQL.
- La base de datos fue implementada usando ‘utf-8’, de tal forma que soporte caracteres especiales del alfabeto español.
- La base de datos se maneja enteramente con borrados o desactivaciones lógicas.
- Los dominios de la base de datos se aplicaron por medio de la comprobaciones al momento de la definición de la base de datos.

La base de datos se desarrolló en el motor de base de datos MySQL 8.0. A continuación se exponen los modelos entidad-relación y relacional de la base de datos.

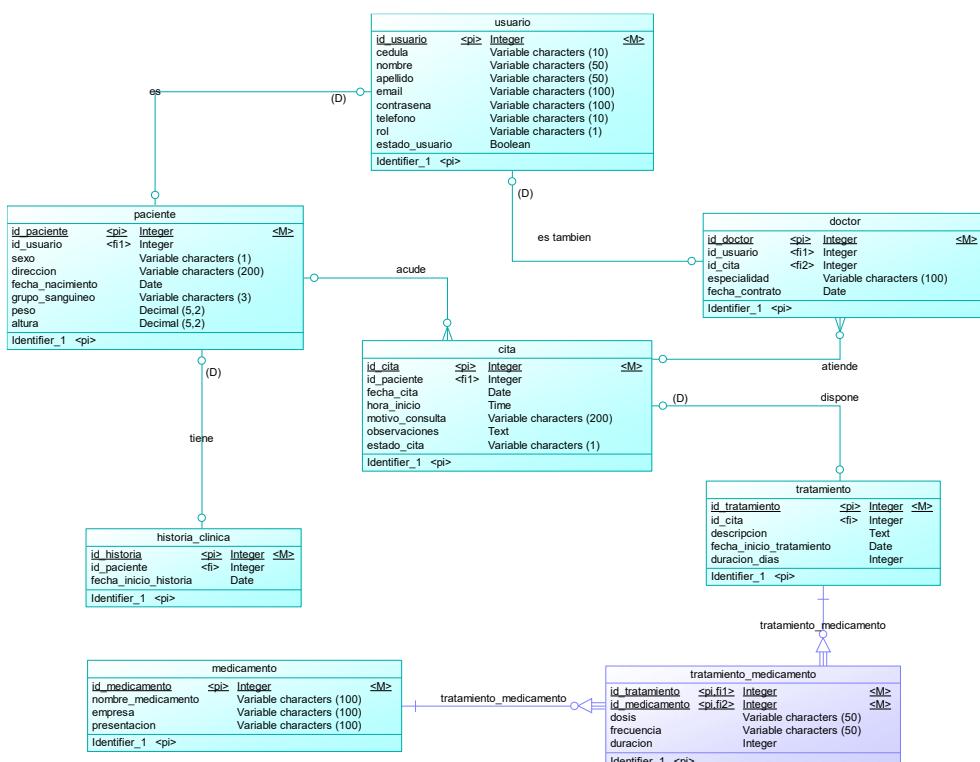


ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SOFTWARE

Modelo Entidad-Relación de la Base de Datos



Modelo Relacional de la Base de Datos



Tablas implementadas:

- Usuario: representa a cualquier ente que accede al sistema, es decir, tanto a los pacientes, como doctores y a los operadores de admisión, por eso tienen un atributo de rol que los identifica entre sí.
- Paciente: que representa a un paciente, cada registro de la tabla paciente está relacionado a un único registro de la tabla usuario.
- Doctor: que representa a un médico, cada registro de la tabla paciente está relacionado a un único registro de la tabla usuario.
- Cita: representa la interacción entre un paciente y un doctor, junto a todos los datos adjuntos como el horario, el motivo o las observaciones.
- Tratamiento: cada cita contiene un tratamiento, es decir, la retroalimentación que el doctor da al paciente. Este incluye las indicaciones que el paciente debe seguir, además de la fecha de inicio de tratamiento.
- Medicamento: incluye un listado de los medicamentos más comunes recetados por el médico en el Ecuador.
- Tratamiento_medicamento: indica los medicamentos que un paciente debe consumir como parte de un tratamiento, incluyendo su dosis, frecuencia y duración.
- Historia_Clinica: indica el inicio de la historia clínica del paciente, es decir, la fecha en que el cliente asistió a su primera cita médica.

4. Descripción de las Tablas

4.1 Tabla Usuario

Almacena la información básica y credenciales de todas las personas que interactúan con el sistema: pacientes, doctores u operadores/administradores.

Field	Type	Null	Key	Default	Extra
ID_USUARIO	int	NO	PRI	NULL	auto_increment
CEDULA	varchar(10)	YES	UNI	NULL	
NOMBRE	varchar(50)	YES		NULL	
APELLIDO	varchar(50)	YES		NULL	
EMAIL	varchar(100)	YES	UNI	NULL	
CONTRASENA	varchar(100)	YES		NULL	
TELEFONO	varchar(10)	YES		NULL	
ROL	varchar(1)	YES		NULL	
ESTADO_USUARIO	tinyint(1)	YES		NULL	

9 rows in set (0.00 sec)

Campos:

- ID_USUARIO — INT, tamaño estándar, no nulo, autoincremental. PK. Identificador único del usuario.
- CEDULA — VARCHAR(10), admite 10 caracteres, único, permite nulos. Número de cédula ecuatoriana; debe cumplir ‘^[0-9]{10}\$’.
- NOMBRE — VARCHAR(50), hasta 50 caracteres, permite nulos. Nombres del usuario.



- APELLIDO — VARCHAR(50), permite nulos. Apellidos del usuario.
- EMAIL — VARCHAR(100), hasta 100 caracteres, único, permite nulos. Correo electrónico de contacto.
- CONTRASENA — VARCHAR(100), permite nulos. Contraseña (debe almacenarse cifrada; no tiene valor por defecto).
- TELEFONO — VARCHAR(10), permite nulos. Teléfono de contacto; debe cumplir ‘^[0-9]{7,10}\$’.
- ROL — VARCHAR(1), permite nulos. Rol del usuario: ‘P’ (paciente), ‘D’ (doctor) u ‘O’ (operador).
- ESTADO_USUARIO — BOOLEAN, permite nulos. Indica si el usuario está activo (‘1’) o inactivo (‘0’).

Claves y restricciones:

- PK: ‘ID_USUARIO’.
- UNIQUE: ‘CEDULA’, ‘EMAIL’.
- Checks: validación de formato de cédula (10 dígitos), teléfono (7–10 dígitos), rol (‘P’, ‘D’, ‘O’), estado de usuario (‘0’, ‘1’).
- Campos nulos: excepto ‘ID_USUARIO’, el resto de campos no tienen restricciones ‘NOT NULL’, por lo que se permite ausencia de datos

4.2 Tabla Paciente

Guarda información médica y personal adicional de los usuarios que son pacientes. Amplía la tabla ‘USUARIO’ con datos clínicos relevantes.

mysql> describe paciente;						
Field	Type	Null	Key	Default	Extra	
ID_PACIENTE	int	NO	PRI	NULL	auto_increment	
ID_USUARIO	int	YES	UNI	NULL		
SEXO	varchar(1)	YES		NULL		
DIRECCION	varchar(200)	YES		NULL		
FECHA_NACIMIENTO	date	YES		NULL		
GRUPO_SANGUINEO	varchar(3)	YES		NULL		
PESO	decimal(5,2)	YES		NULL		
ALTURA	decimal(5,2)	YES		NULL		
8 rows in set (0.00 sec)						

Campos:

- ID_PACIENTE — INT, autoincremental, no nulo, PK. Identificador único del paciente.
- ID_USUARIO — INT, único, puede ser nulo. Clave foránea hacia ‘USUARIO.ID_USUARIO’; cada paciente está asociado a un usuario.
- SEXO — VARCHAR(1), permite nulos. Género (‘M’ masculino, ‘F’ femenino, ‘O’ otro).
- DIRECCION — VARCHAR(200), permite nulos. Dirección de domicilio.
- FECHA_NACIMIENTO — DATE, permite nulos. Fecha de nacimiento.
- GRUPO_SANGUINEO — VARCHAR(3), permite nulos. Grupo sanguíneo (‘A+’, ‘A-’, ‘B+’, ‘B-’, ‘AB+’, ‘AB-’, ‘O+’, ‘O-’).
- PESO — DECIMAL(5,2), permite nulos. Peso en kilogramos; debe ser positivo.
- ALTURA — DECIMAL(5,2), permite nulos. Altura en metros; debe ser positiva.

Claves y restricciones:



- PK: ‘ID_PACIENTE’.
- UNIQUE: ‘ID_USUARIO’ (un usuario no puede ser paciente y doctor simultáneamente).
- FK: ‘ID_USUARIO’ referencia a ‘USUARIO.ID_USUARIO’ (‘FK_PACIENTE_ES_USUARIO’).
- Checks: valores permitidos para ‘SEXO’ y ‘GRUPO_SANGUINEO’; ‘PESO’ y ‘ALTURA’ deben ser mayores que cero.

4.3 Tabla Doctor

Contiene información adicional de los usuarios que actúan como médicos en el sistema.

mysql> describe doctor;						
Field	Type	Null	Key	Default	Extra	
ID_DOCTOR	int	NO	PRI	NULL	auto_increment	
ID_USUARIO	int	YES	UNI	NULL		
ESPECIALIDAD	varchar(100)	YES		NULL		
FECHA_CONTRATO	date	YES		NULL		

4 rows in set (0.00 sec)

Campos:

- ID_DOCTOR — INT, autoincremental, no nulo, PK. Identificador del doctor.
- ID_USUARIO — INT, único, puede ser nulo. Clave foránea hacia ‘USUARIO.ID_USUARIO’ que identifica al usuario como doctor.
- ESPECIALIDAD — VARCHAR(100), permite nulos. Especialidad médica (p. ej., cardiología, pediatría).
- FECHA_CONTRATO — DATE, permite nulos. Fecha en la que el doctor se incorporó al centro.

Claves y Restricciones:

- PK: ‘ID_DOCTOR’.
- UNIQUE: ‘ID_USUARIO’ (evita duplicar médicos con el mismo usuario).
- FK: ‘ID_USUARIO’ referencia a ‘USUARIO.ID_USUARIO’ (‘FK_DOCTOR_ES_TAMBIE_USUARIO’).

4.4 Tabla Historia_Clinica

Registra la creación de la historia clínica asociada a cada paciente. Sirve como vínculo inicial para módulos que manejen el contenido clínico detallado.

mysql> describe historia_clinica;						
Field	Type	Null	Key	Default	Extra	
ID_HISTORIA	int	NO	PRI	NULL	auto_increment	
ID_PACIENTE	int	YES	UNI	NULL		
FECHA_INICIO_HISTORIA	date	YES		NULL		

3 rows in set (0.00 sec)



Campos:

- ID_HISTORIA — INT, autoincremental, no nulo, PK. Identificador único de la historia clínica.
- ID_PACIENTE — INT, único, puede ser nulo. Clave foránea hacia ‘PACIENTE.ID_PACIENTE’.
- FECHA_INICIO_HISTORIA — DATE, permite nulos. Fecha de apertura del historial clínico.

Claves y Restricciones:

- PK: ‘ID_HISTORIA’.
- UNIQUE: ‘ID_PACIENTE’. Un paciente puede tener a lo sumo una historia clínica.
- FK: ‘ID_PACIENTE’ referencia a ‘PACIENTE.ID_PACIENTE’ (‘FK_HISTORIA_TIENE2_PACIENTE’).

4.5 Tabla Cita

Almacena las citas médicas agendadas entre los pacientes y doctores, incluyendo el motivo de la consulta y su estado. Constituye una tabla transaccional clave del sistema.

mysql> describe cita;						
Field	Type	Null	Key	Default	Extra	
ID_CITA	int	NO	PRI	NULL	auto_increment	
ID_PACIENTE	int	YES	MUL	NULL		
ID_DOCTOR	int	YES	MUL	NULL		
FECHA_CITA	date	YES		NULL		
HORA_INICIO	time	YES		NULL		
MOTIVO_CONSULTA	varchar(200)	YES		NULL		
OBSERVACIONES	text	YES		NULL		
ESTADO_CITA	varchar(1)	YES		NULL		

8 rows in set (0.01 sec)

Campos:

- ID_CITA — INT, autoincremental, no nulo, PK. Identificador de la cita.
- ID_PACIENTE — INT, puede ser nulo. Clave foránea hacia ‘PACIENTE.ID_PACIENTE’ que indica quién solicita la cita.
- ID_DOCTOR — INT, puede ser nulo. Clave foránea hacia ‘DOCTOR.ID_DOCTOR’ que indica el profesional que atenderá la cita.
- FECHA_CITA — DATE, permite nulos. Fecha programada de la cita.
- HORA_INICIO — TIME, permite nulos. Hora de inicio de la cita.
- MOTIVO_CONSULTA — VARCHAR(200), permite nulos. Razón principal o síntomas que motivan la cita.
- OBSERVACIONES — TEXT, permite nulos. Comentarios adicionales registrados por el médico.
- ESTADO_CITA — VARCHAR(1), permite nulos. Estado de la cita: ‘P’ (pendiente), ‘C’ (cancelada) o ‘A’ (atendida).



Claves y Restricciones:

- PK: ‘ID_CITA’.
- FKs:
 - ‘FK_CITA_ACUDE_PACIENTE’: ‘ID_PACIENTE’ referencia a ‘PACIENTE.ID_PACIENTE’.
 - ‘FK_CITA_ATIENDE_DOCTOR’: ‘ID_DOCTOR’ referencia a ‘DOCTOR.ID_DOCTOR’.
- Check: validación del campo ‘ESTADO_CITA’ para aceptar solamente los valores ‘P’, ‘C’, ‘A’.

4.6 Tabla Tratamiento

Describe el tratamiento prescrito a raíz de una cita. Cada cita puede tener a lo sumo un tratamiento asociado.

```
mysql> describe tratamiento;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null | Key  | Default | Extra       |
+-----+-----+-----+-----+-----+-----+
| ID_TRATAMIENTO | int    | NO   | PRI   | NULL    | auto_increment |
| ID_CITA        | int    | YES  | UNI   | NULL    |              |
| DESCRIPCION    | text   | YES  |       | NULL    |              |
| FECHA_INICIO_TRATAMIENTO | date  | YES  |       | NULL    |              |
| DURACION_DIAS  | int    | YES  |       | NULL    |              |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Campos:

- ID_TRATAMIENTO — INT, autoincremental, no nulo, PK. Identificador del tratamiento.
- ID_CITA — INT, único, puede ser nulo. Clave foránea hacia ‘CITA.ID_CITA’. Relaciona el tratamiento con una cita específica.
- DESCRIPCION — TEXT, permite nulos. Descripción del tratamiento o diagnóstico.
- FECHA_INICIO_TRATAMIENTO — DATE, permite nulos. Fecha de inicio del tratamiento.
- DURACION_DIAS — INT, permite nulos. Duración del tratamiento en días; debe ser mayor o igual a cero.

Claves y Restricciones:

- PK: ‘ID_TRATAMIENTO’.
- UNIQUE: ‘ID_CITA’ (relación 1:1 con ‘CITA’).
- FK: ‘FK_TRATAMIE_DISPONE2_CITA’: ‘ID_CITA’ referencia a ‘CITA.ID_CITA’.
- Check: ‘DURACION_DIAS >= 0’.



4.7 Tabla Medicamento

Actúa como catálogo de medicamentos disponibles para ser prescritos en los tratamientos.

Field	Type	Null	Key	Default	Extra
ID_MEDICAMENTO	int	NO	PRI	NULL	auto_increment
NOMBRE_MEDICAMENTO	varchar(100)	YES		NULL	
EMPRESA	varchar(100)	YES		NULL	
PRESENTACION	varchar(100)	YES		NULL	

4 rows in set (0.00 sec)

Campos:

- ID_MEDICAMENTO — INT, autoincremental, no nulo, PK. Identificador del medicamento.
- NOMBRE_MEDICAMENTO — VARCHAR(100), permite nulos. Nombre comercial del medicamento.
- EMPRESA — VARCHAR(100), permite nulos. Laboratorio o empresa fabricante.
- PRESENTACION — VARCHAR(100), permite nulos. Forma de presentación (tableta, jarabe, inyectable, etc).

Claves y Restricciones:

- PK: ‘ID_MEDICAMENTO’.
- No existen restricciones de unicidad ni checks adicionales.

4.8 Tabla Tratamiento_Medicamento

Tabla intermedia que vincula cada tratamiento con uno o más medicamentos. Permite registrar la dosis y la frecuencia de administración.

Field	Type	Null	Key	Default	Extra
ID_TRATAMIENTO	int	NO	PRI	NULL	
ID_MEDICAMENTO	int	NO	PRI	NULL	
DOSIS	varchar(50)	YES		NULL	
FRECUENCIA	varchar(50)	YES		NULL	
DURACION	int	YES		NULL	

5 rows in set (0.00 sec)

Campos:

- ID_TRATAMIENTO — INT, no nulo. Parte de la clave primaria compuesta y FK hacia ‘TRATAMIENTO.ID_TRATAMIENTO’.
- ID_MEDICAMENTO — INT, no nulo. Parte de la clave primaria compuesta y FK hacia ‘MEDICAMENTO.ID_MEDICAMENTO’.
- DOSIS — VARCHAR(50), permite nulos. Cantidad de medicamento a administrar (p. ej., ‘500 mg’).



- FRECUENCIA — VARCHAR(50), permite nulos. Intervalo entre dosis (p. ej., ‘cada 8 horas’).
- DURACION — INT, permite nulos. Duración del medicamento en días; debe ser mayor o igual a cero.

Claves y Restricciones:

- PK: compuesta por ('ID_TRATAMIENTO', 'ID_MEDICAMENTO').
- FKs:
 - 'FK_TRATAMIE_TRAZAMIE_ID_TRATAMIE': 'ID_TRATAMIENTO' referencia a 'TRATAMIENTO.ID_TRATAMIENTO'.
 - 'FK_TRAZAMIE_TRAZAMIE_MEDICAME': 'ID_MEDICAMENTO' referencia a 'MEDICAMENTO.ID_MEDICAMENTO'.
- Check: 'DURACION >= 0'.

5. Descripción de las Relaciones entre Tablas

Se explica brevemente las r varias relaciones que unen las tablas ya descritas del modelo.

- USUARIO y PACIENTE / DOCTOR: un usuario puede ser paciente ('ROL = P') o doctor ('ROL = D'). Los campos 'ID_USUARIO' en 'PACIENTE' y 'DOCTOR' son únicos y referencian a 'USUARIO.ID_USUARIO', asegurando que cada usuario esté ligado a lo sumo a un paciente o a un doctor.
- PACIENTE y HISTORIA_CLINICA: relación uno a uno; 'HISTORIA_CLINICA.ID_PACIENTE' es único. Cada paciente dispone de una única historia clínica.
- PACIENTE y CITA: relación uno a muchos; 'CITA.ID_PACIENTE' referencia al paciente. Un paciente puede agendar múltiples citas.
- DOCTOR y CITA: relación uno a muchos; 'CITA.ID_DOCTOR' referencia al doctor. Un doctor puede atender muchas citas.
- CITA y TRATAMIENTO: relación uno a uno opcional; 'TRATAMIENTO.ID_CITA' es único. Una cita puede generar a lo sumo un tratamiento.
- TRATAMIENTO y TRATAMIENTO_MEDICAMENTO y MEDICAMENTO: relación muchos a muchos entre tratamientos y medicamentos, modelada por la tabla 'TRATAMIENTO_MEDICAMENTO'. Un tratamiento puede incluir varios medicamentos, y un medicamento puede estar presente en múltiples tratamientos.

6. Resumen de las verificaciones de los atributos (CHECKS)

Se enlista brevemente las consideraciones que algunos campos de ciertas entidades tienen para admitir datos.

- Formato de cédula: 'CEDULA' en 'USUARIO' debe tener exactamente 10 dígitos.
- Formato de teléfono: 'TELEFONO' en 'USUARIO' acepta de 7 a 10 dígitos.
- Valores de rol: 'ROL' en 'USUARIO' restringido a 'P', 'D', 'O'.
- Estado de usuario: 'ESTADO_USUARIO' acepta '0' o '1'.
- Sexo: 'SEXO' en 'PACIENTE' restringido a 'M', 'F', 'O'.
- Grupo sanguíneo: 'GRUPO_SANGUINEO' en 'PACIENTE' restringido a ocho combinaciones ('A+', 'A-', 'B+', 'B-', 'AB+', 'AB-', 'O+', 'O-').
- Peso y altura: en 'PACIENTE' deben ser positivos.



- Estado de cita: 'ESTADO_CITA' en 'CITA' acepta 'P', 'C', 'A'.
- Duración: 'DURACION_DIAS' en 'TRATAMIENTO' y 'DURACION' en 'TRATAMIENTO_MEDICAMENTO' deben ser mayores o iguales a cero.

7. Repositorio del Proyecto

El siguiente enlace lleva al repositorio del proyecto en GitHub, aquí se encuentran detalladas todas las partes de este proyecto, incluyendo este manual, los scripts de la base de datos, los documentos como este manual y la presentación usada para el proyecto, además del código fuente de la aplicación desarrollada.

<https://github.com/juanchuga06/Proyecto1BFBaseDatos>

La estructura de este proyecto es la siguiente, se detallan los ficheros más importantes:

- Database: se encuentran los scripts de SQL que detallan el DDL de la base de datos y una muestra de inserciones en las tablas si se desea comenzar a utilizar la base de datos,
- Design: se encuentran las imágenes en formato .png de los modelos entidad-relación, relacional y físico
- Documents: se encuentran los archivos .pdf de este manual y de la presentación creada para el proyecto.
- Lib: librerías y dependencias propias de un proyecto en Java
- Src: código fuente en Java implementado para la aplicación.