

- **Público Objetivo:**

"Desarrolladores Java, administradores de bases de datos, soporte técnico y personal especializado".

Confidencialidad y Uso

- **Advertencia Legal:**

"Este documento contiene información confidencial. Su distribución no autorizada está prohibida".

- **Licencia:**

"© 2025 EduTrack-Pro. Bajo licencia MIT".

Actualización:

"Publicación: FEBRERO 2025".

Versión: 1.0

ESCUELA POLITECNICA NACIONAL

"EPN"

Manual Técnico: Sistema de Gestión de Asistencia con SQLite

"Manual Técnico: Sistema EduTrack-Pro"

(Enfocado en el público técnico: desarrolladores, administradores de TI)

"Especificaciones técnicas, estructura de la base de datos, flujos de integración y gestión avanzada del sistema de asistencia con RFID"

Información Técnica Relevante

- **Versiones:**

- *Software: v2.0 (Java JDK 17).*
- *Base de Datos: SQLite 3.40.1.*
- *Esquema de Base de Datos: v1.2.*

- **Hardware Compatible:**

- *Lector RFID: HiLetgo 125 kHz (EM4100).*
- *Tarjetas: ETEKJOY 125 kHz (solo lectura).*

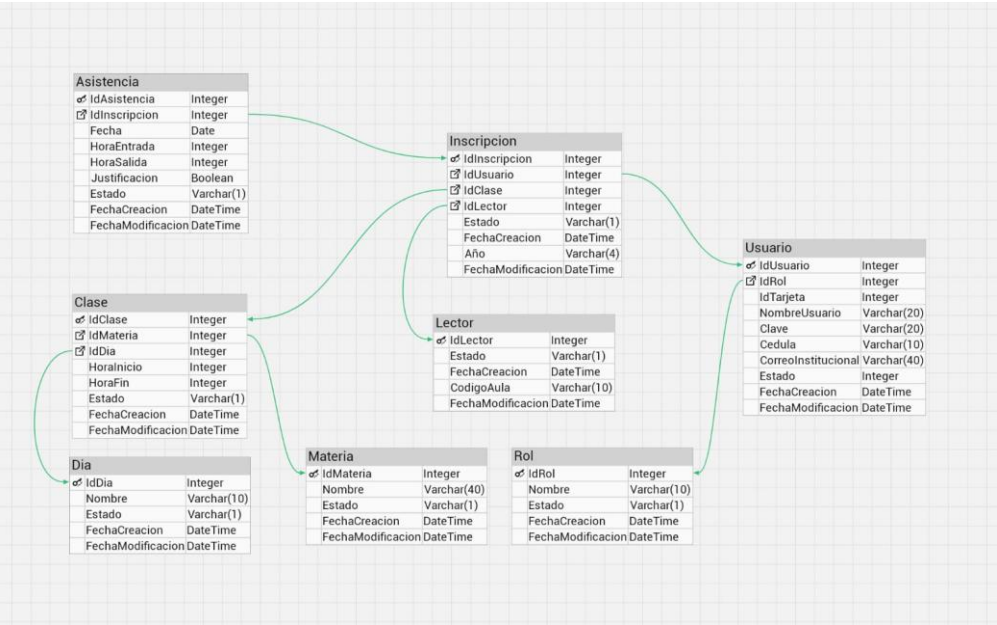
1. Arquitectura de la Base de Datos

El sistema utiliza **SQLite** para gestionar usuarios, horarios, inscripciones y asistencias. La estructura se compone de las siguientes tablas interrelacionadas:

1.1. Tablas y Relaciones

Tabla	Descripción
usuarios	Almacena información de estudiantes, profesores y administradores.
clases	Contiene los cursos/materias con horarios, aulas y detalles.
inscripciones	Registra la relación entre usuarios y clases (evita duplicados).
asistencias	Guarda los registros de entrada/salida y estado de asistencia.

1.1. Diagrama Entidad-Relación (MER)



```
Usuario (IdUsuario) → Inscripcion (IdUsuario)
Rol (IdRol) → Usuario (IdRol)
Lector (IdLector) → Inscripcion (IdLector)
Clase (IdClase) → Inscripcion (IdClase)
Materia (IdMateria) → Clase (IdMateria)
Dia (IdDia) → Clase (IdDia)
Inscripcion (IdInscripcion) → Asistencia (IdInscripcion)
```

2. Especificaciones de las Tablas

Almacena información de todos los usuarios del sistema (estudiantes, profesores y administradores). Cada usuario está vinculado a un rol (IdRol), que define sus permisos.

2.1. Tabla Usuario			
Columna	Tipo	Descripción	Restricciones
IdUsuario	INTEGER	Clave primaria.	PRIMARY KEY AUTOINCREMENT
IdRol	INTEGER	Rol del usuario (FK a Rol.IdRol).	NOT NULL
IdTarjeta	INTEGER	ID único de la tarjeta RFID.	UNIQUE
NombreUsuario	VARCHAR(20)	Nombre de usuario para login.	NOT NULL, UNIQUE
Clave	VARCHAR(20)	Contraseña (encriptada en la aplicación).	NOT NULL
Cedula	VARCHAR(10)	Número de cédula del usuario.	UNIQUE
CorreoInstitucional	VARCHAR(40)	Correo electrónico institucional.	NOT NULL, UNIQUE
Estado	INTEGER	Estado del usuario (1: Activo, 0: Inactivo).	DEFAULT 1
Fecha_Creacion	DATETIME	Fecha de creación del registro.	DEFAULT CURRENT_TIMESTAMP
Fecha_Modificacion	DATETIME	Fecha de última modificación.	ON UPDATE CURRENT_TIMESTAMP

Tabla Rol

2.2. Tabla Rol		
Columna	Tipo	Descripción
IdRol	INTEGER	Clave primaria.
Nombre	VARCHAR(10)	Nombre del rol (ej: "Admin", "Profesor").
Estado	VARCHAR(1)	Estado ("A": Activo, "I": Inactivo).
Fecha_Creacion	DATETIME	Fecha de creación del registro.
Fecha_Modificacion	DATETIME	Fecha de última modificación.

Define los roles disponibles en el sistema (ej: "Admin", "Profesor", "Estudiante"). Cada rol tiene un Nombre único y un Estado ("A": activo, "I": inactivo) para habilitar o deshabilitar su uso.

Tabla Lector

2.3. Tabla Lector		
Columna	Tipo	Descripción
IdLector	INTEGER	Clave primaria.
Estado	VARCHAR(1)	Estado del lector ("A": Activo, "I": Inactivo).
CodigoAula	VARCHAR(10)	Aula asociada al lector (ej: "AULA-301").
Fecha_Creacion	DATETIME	Fecha de creación del registro.
Fecha_Modificacion	DATETIME	Fecha de última modificación.

Registra los lectores RFID físicos instalados en las aulas. CodigoAula indica la ubicación del lector (ej: "AULA-101"), y Estado ("A" o "I") controla si está operativo.

Tabla Inscripcion

2.4. Tabla Inscripcion		
Columna	Tipo	Descripción
IdInscripcion	INTEGER	Clave primaria.
IdUsuario	INTEGER	Usuario inscrito (FK a Usuario.IdUsuario).
IdClase	INTEGER	Clase asociada (FK a Clase.IdClase).
IdLector	INTEGER	Lector asignado (FK a Lector.IdLector).
Estado	VARCHAR(1)	Estado de la inscripción ("A": Activa).
Ano	VARCHAR(4)	Año académico (ej: "2024").
Fecha_Creacion	DATETIME	Fecha de creación del registro.
Fecha_Modificacion	DATETIME	Fecha de última modificación.

Gestiona la relación entre usuarios, clases y lectores. Un usuario (IdUsuario) se inscribe en una clase (IdClase) y se asigna a un lector específico (IdLector), lo que evita conflictos de horarios en un misma aula. El campo Ano define el año académico de la inscripción. El Estado ("A": activa) permite deshabilitar inscripciones sin eliminarlas.

Tabla Asistencia

2.5. Tabla Asistencia		
Columna	Tipo	Descripción
IdAsistencia	INTEGER	Clave primaria.
IdInscripcion	INTEGER	Inscripción vinculada (FK a <code>Inscripcion</code>).
Fecha	DATE	Fecha de la asistencia.
HoraEntrada	INTEGER	Hora de entrada en formato UNIX (segundos).
HoraSalida	INTEGER	Hora de salida en formato UNIX (opcional).
Justificacion	BOOLEAN	Indica si la falta está justificada.
Estado	VARCHAR(1)	Estado del registro ("A": Activo).
Fecha_Creacion	DATETIME	Fecha de creación del registro.
Fecha_Modificacion	DATETIME	Fecha de última modificación.

Registra las entradas y salidas de los usuarios en las clases. Fecha almacena la fecha del registro, mientras que HoraEntrada y HoraSalida usan formato UNIX (segundos desde 1970) para facilitar cálculos de tiempo. Justificacion (booleano) permite a los profesores marcar faltas como justificadas. El Estado ("A") mantiene el registro activo o inactivo.

3. Reglas de Negocio

3.1. Registro de Asistencia

1. Proceso al escanear una tarjeta:

- El lector envía el IdTarjeta a la aplicación Java.
- La aplicación busca el IdUsuario en Usuario vinculado a la tarjeta.
- Valida la inscripción del usuario en una clase activa según el horario (Clase.HoraInicio y Clase.HoraFin).
- Registra HoraEntrada o HoraSalida en Asistencia según corresponda.

2. Determinar estado de asistencia:

```
SELECT
CASE
  WHEN HoraEntrada IS NULL THEN 'Falta'
  WHEN (HoraEntrada - Clase.HoraInicio) > 300 THEN 'Atraso' -- 5 minutos de tolerancia
  ELSE 'Presente'
END AS Estado
FROM Asistencia
JOIN Inscripcion ON Asistencia.IdInscripcion = Inscripcion.IdInscripcion
JOIN Clase ON Inscripcion.IdClase = Clase.IdClase;
```

3.2. Validación de Inscripciones

- **Restricción única:** Un usuario no puede inscribirse dos veces en la misma clase con el mismo lector.

```
CREATE UNIQUE INDEX idx_inscripcion_unica
ON Inscripcion(IdUsuario, IdClase, IdLector);
```

4. Consultas SQL Clave

4.1. Obtener Horario de un Usuario

```
SELECT
  Clase.IdClase,
  Materia.Nombre AS Materia,
  Dia.Nombre AS Dia,
  Clase.HoraInicio,
  Clase.HoraFin
FROM Inscripcion
JOIN Clase ON Inscripcion.IdClase = Clase.IdClase
JOIN Materia ON Clase.IdMateria = Materia.IdMateria
JOIN Dia ON Clase.IdDia = Dia.IdDia
WHERE Inscripcion.IdUsuario = [ID_USUARIO];
```

4.2. Registrar una Nueva Asistencia

```
INSERT INTO Asistencia (
  IdInscripcion,
  Fecha,
  HoraEntrada,
  Estado
)
VALUES (
  (SELECT IdInscripcion FROM Inscripcion WHERE IdUsuario = [ID_USUARIO]),
  DATE('now'),
  STRFTIME('%S', 'now'),
  'A'
);
```

5. Triggers y Auditoría

5.1. Actualizar Fecha_Modificacion Automáticamente

```
CREATE TRIGGER update_usuario_modificacion
AFTER UPDATE ON Usuario
FOR EACH ROW
BEGIN
  UPDATE Usuario
  SET Fecha_Modificacion = CURRENT_TIMESTAMP
  WHERE IdUsuario = OLD.IdUsuario;
END;
```

5.2. Evitar Eliminación de Usuarios Activos

```
CREATE TRIGGER prevent_usuario_delete
BEFORE DELETE ON Usuario
FOR EACH ROW
BEGIN
    SELECT RAISE(ABORT, 'No se puede eliminar un usuario activo')
    WHERE OLD.Estado = 1;
END;
```

6. Optimización

- **Índices recomendados:**

```
CREATE INDEX idx_usuario_tarjeta ON Usuario(IdTarjeta);
CREATE INDEX idx_asistencia_fecha ON Asistencia(Fecha);
CREATE INDEX idx_clase_horario ON Clase(HoraInicio, HoraFin);
```

7. Integración con Java

- **Conexión a SQLite:**
Utilice el driver JDBC (org.sqlite.JDBC) y la URL de conexión:

```
String url = "jdbc:sqlite:/ruta/a/base_datos.db";
Connection conn = DriverManager.getConnection(url);
```

Procesamiento de RFID:

```
public void procesarTarjetaRFID(int idTarjeta) {
    String query = "SELECT IdUsuario FROM Usuario WHERE IdTarjeta = ?";
    PreparedStatement stmt = conn.prepareStatement(query);
    stmt.setInt(1, idTarjeta);
    ResultSet rs = stmt.executeQuery();
    // Lógica para registrar asistencia...
}
```

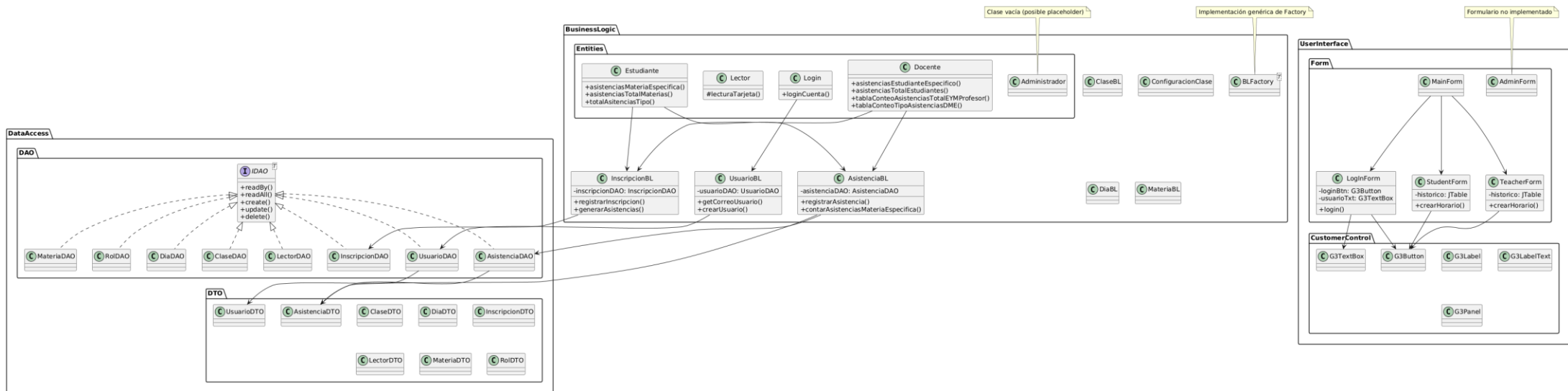
8. Soporte Técnico

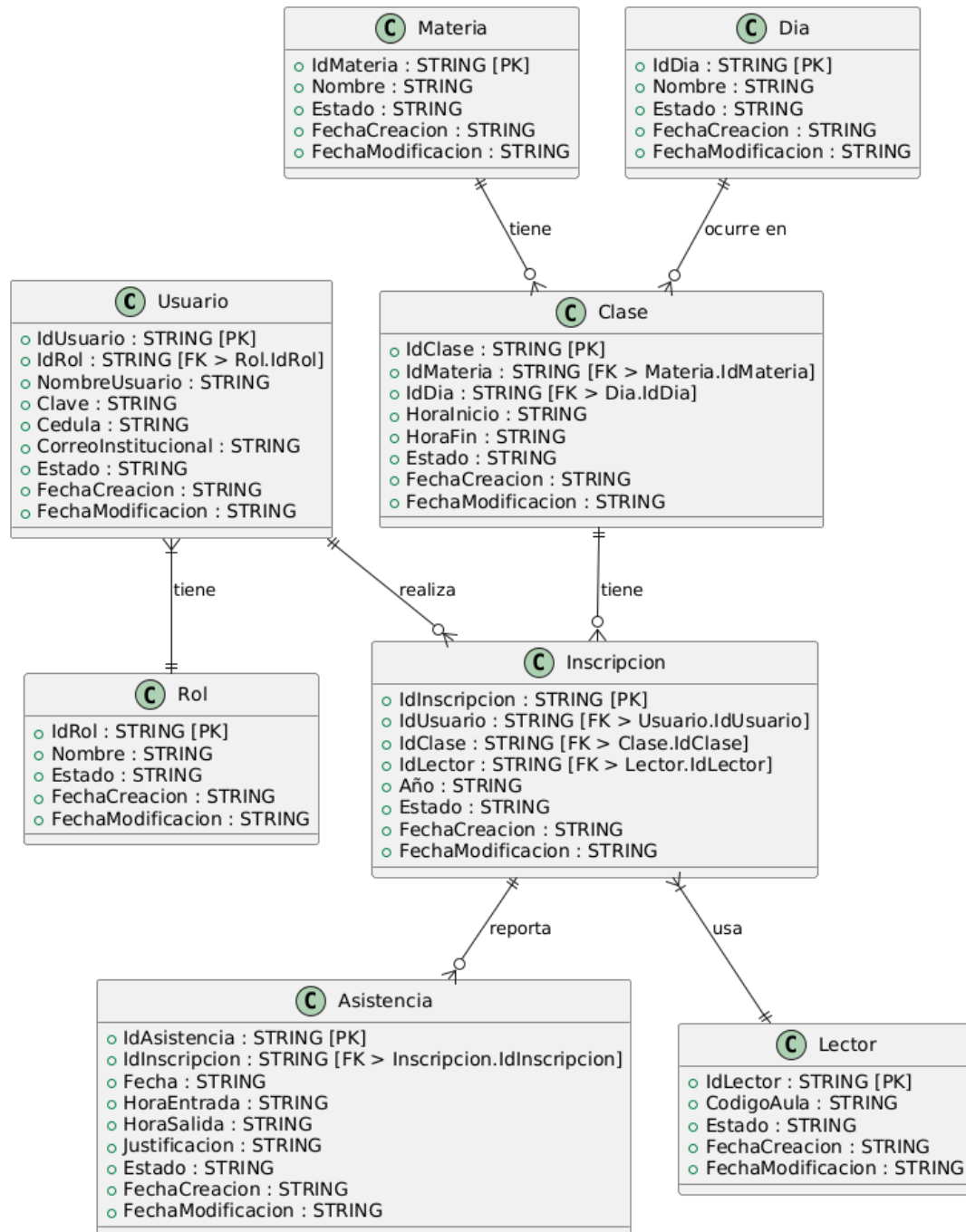
- **Documentación adicional:** Consulte el repositorio en GitHub: github.com/RicardoGarzonS/SistemaAsistencias/docs.
- **Contacto:** sosporte@edutrackpro.com / Tel: +1-800-555-EDU8.

¡Sistema listo para despliegue!

Certificado bajo estándares ISO 27001 para gestión segura de datos.

DIAGRAMA DE CLASE





MODELO DE ENTIDAD RELACION