BASE DE DATOS BIOGATE

Explicación

Este diseño de base de datos está estructurado para gestionar un sistema de control de acceso con autenticación biométrica. Se compone de 12 tablas: `***administradores***`, `***usuarios***`, `***roles***`, `***administran***`, `***dispositivos***`, `***usuarios_dispositivos***`, `***configuración_dispositivos***`, `***embeddings***`, `***historial***`, `***accesos***`, y `***reportes***`. Cada una de estas tablas cumple una función específica dentro del sistema y están relacionadas entre sí para garantizar integridad, seguridad y eficiencia en el manejo de la información.

ADMINISTRADORES

La tabla `***administradores***` almacena la información de los administradores encargados de gestionar el sistema. Cada administrador tiene un identificador único, nombre completo, un número de teléfono que facilita la comunicación y la autenticación, un nombre de usuario para ingresar al sistema y una contraseña protegida. La inclusión de esta tabla permite establecer un control jerárquico, donde ciertos administradores pueden estar asignados a la gestión de usuarios específicos.

USUARIOS

La tabla `***usuarios***` contiene la información personal de los usuarios registrados en el sistema. En lugar de almacenar un solo campo para el nombre completo, se han separado los datos en `nombre`, `*apellido_paterno*` y `*apellido_materno*`, lo que permite búsquedas más precisas y una mejor organización de la información. También se incluye un campo para el número de teléfono, un id de rol que referencia el puesto del usuario dentro de la organización y una imagen de identificación opcional, así como un apartado de autorización que representa el estado del usuario si está o no autorizado en el sistema.

La tabla `**roles**` define los diferentes niveles de permisos dentro del sistema. Cada rol tiene un identificador único y un nombre que describe su función (por ejemplo, "Administrador", "Usuario Estándar" o "Supervisor"). Esta tabla permite una gestión flexible de permisos, facilitando la asignación de responsabilidades dentro del sistema.

En lugar de almacenar el rol de cada usuario como un simple texto en la tabla usuarios, se crea una relación con esta tabla mediante una clave foránea. Esto evita redundancias y permite la escalabilidad del sistema, ya que en el futuro se pueden agregar nuevos roles sin modificar la estructura de la base de datos.

Para manejar la autenticación biométrica, la tabla `***embeddings***` almacena los datos biométricos de los usuarios en formato binario. Cada `*embedding*` representa un vector matemático del rostro del usuario, lo que facilita la verificación de identidad mediante reconocimiento facial. La relación entre esta tabla y `***usuarios***` se establece a través de `*id_usuario*`, asegurando que cada usuario tenga sus datos biométricos almacenados

correctamente. Se ha implementado la restricción `ON DELETE CASCADE`, lo que significa que, si un usuario es eliminado, sus datos biométricos también lo serán automáticamente. Esto garantiza la integridad de los datos y evita que queden registros huérfanos en la base de datos.

RELACION CON ADMINISTRADORES Y USUARIOS

La tabla `**administran**` gestiona la relación de muchos a muchos entre los administradores y los usuarios. Cada administrador puede estar a cargo de varios usuarios, y un usuario puede ser gestionado por múltiples administradores.

DISPOSITIVOS

La tabla `**dispositivos**` almacena la información de los distintos dispositivos utilizados para la autenticación biométrica dentro del sistema. Cada dispositivo tiene un identificador único, un nombre descriptivo, un tipo (como cámara, lector de huellas o escáner de iris) y una ubicación específica. Esta estructura permite gestionar múltiples tipos de dispositivos en distintos lugares, facilitando la escalabilidad del sistema a nuevas tecnologías de autenticación.

La tabla `**usuarios_dispositivos**` representa la relación entre los usuarios y los dispositivos que utilizan. Dado que un usuario puede autenticarse en múltiples dispositivos y un dispositivo puede ser usado por varios usuarios, esta tabla intermedia maneja la relación de muchos a muchos. Cada registro asocia un usuario con un dispositivo específico, permitiendo controlar en qué dispositivos ha sido registrado y autenticado un usuario.

La tabla `**configuraciones_dispositivos**` almacena los distintos parámetros de configuración de los dispositivos biométricos. Cada configuración tiene un identificador único, está asociada a un dispositivo específico y contiene un parámetro con su respectivo valor. Esto permite personalizar el comportamiento de cada dispositivo, como la sensibilidad del reconocimiento, el tiempo de espera entre intentos o el modo de operación. Al centralizar las configuraciones en esta tabla, se facilita la gestión y actualización de los dispositivos sin modificar su estructura base.

EVENTOS DE IDENTIFICACIÓN

El sistema también cuenta con la tabla `***historial***`, que se encarga de registrar los intentos de acceso y otros eventos relevantes dentro del sistema. Cada evento está asociado a un usuario mediante la clave foránea `*id_usuario*`, aunque esta relación puede aceptar valores nulos en caso de que el usuario no esté identificado. La tabla almacena la fecha y hora del evento, el id del dispositivo donde se registró y el lugar en el que ocurrió. Esta información es crucial para auditar accesos y detectar patrones sospechosos. La relación con `***usuarios***` permite analizar el comportamiento de los usuarios y hacer un seguimiento detallado de sus actividades. Se ha implementado `ON DELETE SET NULL`, de manera que, si un usuario

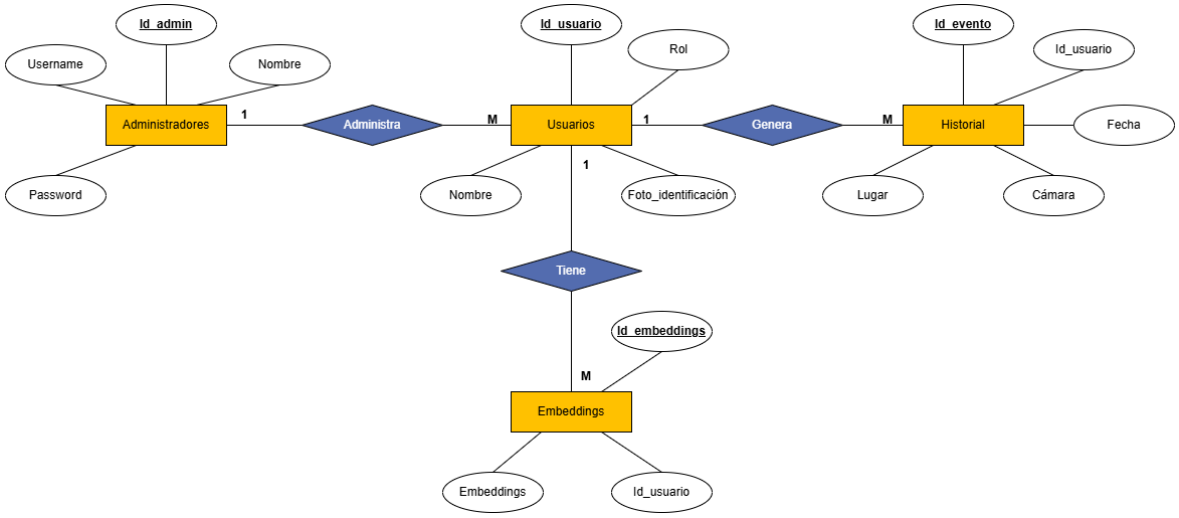
es eliminado, los eventos en los que participó permanecen en la base de datos, aunque sin referencia a la persona especifica.

Para documentar incidentes y facilitar su resolución, se incluye la tabla `***reportes***`. Esta tabla almacena información sobre eventos que requieran revisión, como accesos fallidos o intentos de ingreso no autorizados. Cada reporte está asociado a un evento en `***historial***` mediante la clave foránea `*id_evento*`. Se incluye una descripción detallada del incidente y un estado que indica si el problema sigue pendiente, ha sido revisado o ya se ha resuelto. Se ha aplicado la restricción `ON DELETE CASCADE`, lo que significa que, si un evento es eliminado, los reportes relacionados con ese evento también lo serán. Esto evita que existan reportes sin referencia válida dentro del sistema.

La tabla `**accesos**` almacena los registros de autenticaciones exitosas dentro del sistema. Cada entrada contiene un identificador único, el usuario autenticado, el dispositivo en el que se realizó el acceso, la fecha y hora del evento, y el estado de autorización (permitido o denegado). A diferencia del historial de eventos, que almacena todos los intentos de autenticación, esta tabla solo guarda los accesos que fueron correctamente autorizados, facilitando el análisis de entradas legítimas al sistema.

La tabla `**notificaciones**` almacenas alertas generadas automáticamente por el sistema o enviadas manualmente por los administradores. Cada notificación tiene un identificador único, un mensaje descriptivo, la fecha y hora en que fue generada y un estado que indica si ha sido vista o atendida. Estas notificaciones pueden ser activadas por eventos importantes, como intentos de acceso fallidos repetidos, detección de usuarios no autorizados o cambios en la configuración de dispositivos. La inclusión de esta tabla permite a los administradores recibir alertas en tiempo real y tomar decisiones oportunas para mejorar la seguridad del sistema.

Primera versión de la base de datos



Proceso de normalización

El proceso de normalización de esta base de datos se llevó a cabo en varias etapas, siguiendo las formas normales establecidas en la teoría de bases de datos relacionales. La normalización tiene como objetivo reducir la redundancia, mejorar la integridad de los datos y optimizar la eficiencia en las consultas.

Primera Forma Normal (1FN)

Definición:

En la Primera Forma Normal, se deben cumplir las siguientes condiciones:

- Cada columna debe contener un **único valor atómico** (no listas ni datos repetidos en una celda).
- Cada fila debe ser identificable de manera única mediante una clave primaria.

Problema detectado

- En la entidad Administradores y Usuarios, el campo nombre almacenaba el nombre completo en una sola columna. Esto genera dificultades para realizar búsquedas por nombre o apellido y no cumple con la atomicidad.

- En la entidad Historial, el campo cámara almacenaba dispositivos de detección de manera poco flexible, lo que limita la escalabilidad del sistema.
- La columna rol en Usuarios contenía valores repetitivos que podrían gestionarse mediante una tabla separada.
- No existía una tabla dedicada a Notificaciones, lo que significaba que las alertas y eventos estaban dispersos en el sistema sin una estructura adecuada.

Solución aplicada

- Se dividió el campo nombre en nombre, apellido_paterno y apellido_materno para garantizar la atomicidad.
- Se creó una nueva entidad Dispositivos para permitir la detección mediante distintos tipos de dispositivos (cámaras, lectores de huellas, escáneres de iris, etc.).
- Se eliminó la columna rol en Usuarios y en su lugar se creó la tabla Roles, con una clave foránea id_rol en Usuarios.
- Se creó la tabla Notificaciones para almacenar eventos generados por el sistema sin sobrecargar otras tablas.

Segunda Forma Normal (2FN)

Definición:

Para cumplir con la Segunda Forma Normal, se debe **eliminar cualquier dependencia parcial de la clave primaria** en aquellas tablas con claves compuestas. Cada atributo debe depender completamente de la clave primaria.

Problema detectado

- En la tabla Usuarios, los atributos rol y foto_identificacion podían no depender completamente de la clave primaria id_usuario.
- En la tabla Historial, el campo id_usuario no siempre era obligatorio, ya que algunos eventos pueden registrarse sin una identificación biométrica previa.

- En la tabla Dispositivos, algunos parámetros de configuración (umbral_sensibilidad, modo_operacion) no dependían directamente de la clave primaria id_dispositivo, sino de su tipo.
- El campo lugar en Historial no dependía directamente de un evento, sino del dispositivo que registró la acción.

Solución aplicada

- Se creó la tabla Roles, eliminando la dependencia parcial de Usuarios.
- Se permitió que id_usuario en Historial pudiera ser nulo, permitiendo registrar eventos sin identificación biométrica.
- Se creó la tabla Configuracion_Dispositivos, separando los parámetros específicos de cada dispositivo, reduciendo redundancia en Dispositivos.
- Se eliminó lugar de Historial y se vinculó a Configuracion_Dispositivos mediante id_dispositivo para obtener la ubicación.

Tercera Forma Normal (3FN)

Definición:

La Tercera Forma Normal busca eliminar **dependencias transitivas**, es decir, que un atributo no clave dependa de otro atributo que no sea la clave primaria.

Problema detectado

- En la tabla Historial, la relación con los dispositivos era directa a través del campo cámara. Esto significaba que si en el futuro se agregaban nuevos tipos de dispositivos (huella, iris, etc.), sería necesario modificar la estructura de la tabla.
- En la tabla Administradores, la relación con Usuarios era uno a muchos, lo que limitaba la posibilidad de que varios administradores gestionaran un mismo usuario.
- El campo estado en Notificaciones y Reportes contenía valores repetitivos que podían normalizarse.

Solución aplicada

- Se eliminó la dependencia del campo cámara en Historial y se reemplazó por id_dispositivo, haciendo referencia a la nueva tabla Dispositivos.

- Se creó la tabla Administran, que actúa como una tabla intermedia entre Administradores y Usuarios, permitiendo que varios administradores gestionen los mismos usuarios y viceversa.
- Se normalizó el estado de Notificaciones y Reportes, asegurando valores consistentes (Pendiente, Revisado, Resuelto).
- Se estableció que Reportes dependiera de Historial y no directamente de Usuarios, asegurando la integridad de los eventos reportados.

Forma Normal de Boyce-Codd (BCNF)

Definición:

BCNF es una versión más estricta de la 3FN y asegura que **no existan dependencias funcionales donde un atributo no clave determine una parte de la clave primaria.**

Problema detectado

- La relación entre Usuarios y Administradores debía permitir múltiples asignaciones sin depender de la tabla Usuarios.
- La tabla Usuarios_Dispositivos debía garantizar que un usuario no estuviera registrado más de una vez en el mismo dispositivo.
- En Accesos, la combinación id_usuario e id_dispositivo debía ser única por cada intento de acceso.

Solución aplicada

- Se definió que Usuarios_Dispositivos tuviera una clave primaria compuesta (id_usuario, id_dispositivo), evitando registros duplicados.
- Se estableció la clave primaria compuesta en Accesos (id_usuario, id_dispositivo, fecha_hora), asegurando que cada intento de acceso fuera único.
- Se aplicaron restricciones ON DELETE CASCADE para eliminar datos relacionados de manera automática y evitar registros huérfanos.

Diccionario de datos

Administrador

Columna	Tipo de datos	Restricciones	Descripción

Id_admin	SERIAL	PRIMARY KEY	Identificador único del administrador.
Nombre	VARCHAR (100)	NOT NULL	Primer nombre del administrador.
Apellido_paterno	VARCHAR (100)	NOT NULL	Apellido paterno del administrador.
Apellido_materno	VARCHAR (100)	Opcional	Apellido materno del administrador.
Teléfono	VARCHAR (15)	UNIQUE NOT NULL	Número de teléfono del administrador.
Username	VARCHAR (50)	UNIQUE NOT NULL	Nombre de usuario del administrador.
Password	TEXT	NOT NULL	Contraseña del administrador.

Usuarios (Usuario registrado)

Columna	**Tipo de datos**	**Restricciones**	**Descripción**
Id_usuario	SERIAL	PRIMARY KEY	Identificador único del usuario.
Nombre	VARCHAR (100)	NOT NULL	Primer nombre del usuario.
Apellido_paterno	VARCHAR(100)	NOT NULL	Apellido paterno del usuario.
Apellido_materno	VARCHAR(100)	Opcional	Apellido materno del usuario.
Telefono	VARCHAR(15)	UNIQUE NOT NULL	Número de teléfono del usuario.
Id_rol	INT	FOREIGN KEY	Relación con la tabla de roles.
Foto_identificacion	TEXT	Opcional	Ruta o base64 de la foto de identificación.
Autorizacion	VARCHAR(10)	NOT NULL, DEFAULT ‘DENEGADO’, CHECK (utorización	Indica si el usuario tiene permiso para acceder al sistema. Solo puede tomar

		IN ('PERMITIDO', 'DENEGADO'))	los valores "PERMITIDO" o "DENEGADO".

Administran (Relación entre administradores y usuarios)

Columna	Tipo de datos	Restricciones	Descripción
Id_admin	INT	REFERENCES admins(id_admin) ON DELETE CASCADE	Administrador responsable del usuario.
Id_usuario	INT	REFERENCES users(id_usuario) ON DELETE CASCADE	Usuario administrado.
Clave primaria compuesta	(id_admin, id_usuario)		Permite que un usuario tenga varios administradores y viceversa.

Embeddings (Datos biométricos)

Columna	Tipo de datos	Restricciones	Descripción
id_embedding	SERIAL	PRIMARY KEY	Identificador único del embedding.
id_usuario	INT	REFERENCES users(id_usuario) ON DELETE CASCADE	Relación con el usuario dueño del embedding.
embedding	BYTEA	NOT NULL	Datos biométricos del rostro (vector binario).

Roles

Columna	Tipo de datos	Restricciones	Descripción
id_rol	SERIAL	PRIMARY KEY	Identificador único del rol.
Nombre	VARCHAR(50)	UNIQUE, NOT NULL	Nombre del rol (ej. Administrador,

			Empleado, Invitado).

Dispositivos (Dispositivos de detección)

Columna	Tipo de datos	Restricciones	Descripción
Id_dispositivo	SERIAL	PRIMARY KEY	Identificador único del dispositivo.
Nombre	VARCHAR(100)	NOT NULL	Nombre descriptivo del dispositivo.
Tipo	VARCHAR(50)	NOT NULL	Tipo de dispositivo (Cámara, Huella, Iris, etc.).
Ubicación	VARCHAR(255)	Opcional	Ubicación física del dispositivo.

Usuarios_dispositivos (Relación entre usuarios y dispositivos)

Columna	Tipo de datos	Restricciones	Descripción
Id_usuario	INT	REFERENCES users(id_usuario) ON DELETE CASCADE	Usuario que tiene acceso al dispositivo.
Id_dispositivo	INT	REFERENCES dispositivos(id_dispositivo) ON DELETE CASCADE	Dispositivo autorizado para el usuario.
Clave primaria compuesta	(id_usuario, id_dispositivo)		Permite gestionar accesos a múltiples dispositivos.

Configuracion_dispositivos (Relación entre usuarios y dispositivos)

Columna	Tipo de datos	Restricciones	Descripción
id_configuracion	INT	REFERENCES users(id_usuario) ON DELETE CASCADE	Usuario que tiene acceso al dispositivo.
Id_dispositivo	INT	REFERENCES dispositivos(id_dispositivo) ON DELETE CASCADE	Dispositivo autorizado para el usuario.

Parametro	VARCHAR(100)	NOT NULL	Nombre del parámetro de configuración (ej. 'Umbral_Sensibilidad', 'Modo_Operación').
Valor	VARCHAR(255)	NOT NULL	Valor asignado al parámetro (ej. '0.8' para umbral o 'Automático' para modo de operación).

Historial (Eventos de autenticación)

Columna	Tipo de datos	Restricciones	Descripción
Id_evento	SERIAL	PRIMARY KEY	Identificador único del evento.
Id_usuario	INT	REFERENCES users(id_usuario) ON DELETE SET NULL (Opcional)	Relación con el usuario.
Fecha	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Fecha y hora del evento (por defecto es la fecha actual).
Id_dispositivo	INT	REFERENCES dispositivos(id_dispositivo) ON DELETE SET NULL (Opcional)	Dispositivo en el que se registró el evento.
lugar	VARCHAR(255)	Opcional	Ubicación o lugar donde ocurrió el evento.

Accesos

Columna	Tipo de datos	Restricciones	Descripción
Id_acceso	SERIAL	PRIMARY KEY	Identificador único del intento de acceso.

Id_usuario	INT	FOREIGN KEY → usuarios(id_usuario), NULLABLE	Usuario que intentó acceder.
Id_dispositivo	INT	FOREIGN KEY → dispositivos(id_dispositivo)	Dispositivo utilizado.
Fecha	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Fecha y hora del acceso.
Resultado	VARCHAR(10)	CHECK ('Exitoso', 'Fallido')	Indica si el acceso fue exitoso o no.

Reportes (Reportes de incidentes)

Columna	Tipo de datos	Restricciones	Descripción
Id_reporte	SERIAL	PRIMARY KEY	Identificador único del reporte.
Id_evento	INT	REFERENCES historial_eventos(id_evento) ON DELETE CASCADE	Relación con el evento correspondiente.
Descripción	TEXT	NOT NULL	Descripción detallada del incidente o reporte.
Estado	VARCHAR(50)	CHECK (estado IN ('Pendiente', 'Revisado', 'Resuelto'))	Estado del reporte.

Notificaciones

Columna	Tipo de datos	Restricciones	Descripción
Id_notificacion	SERIAL	PRIMARY KEY	Identificador único de la notificación.
Id_usuario	INT	FOREIGN KEY → usuarios(id_usuario), NULLABLE	Usuario al que está dirigida la notificación.

Mensaje	TEXT	NOT NULL	Contenido de la notificación.
Fecha	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Fecha y hora en que se envió la notificación.
Estado	VARCHAR(50)	CHECK (‘Pendiente’, ‘Visto’, ‘Archivado’)	Estado de la notificación.

Diagrama ENTIDAD – RELACIÓN de la base de datos

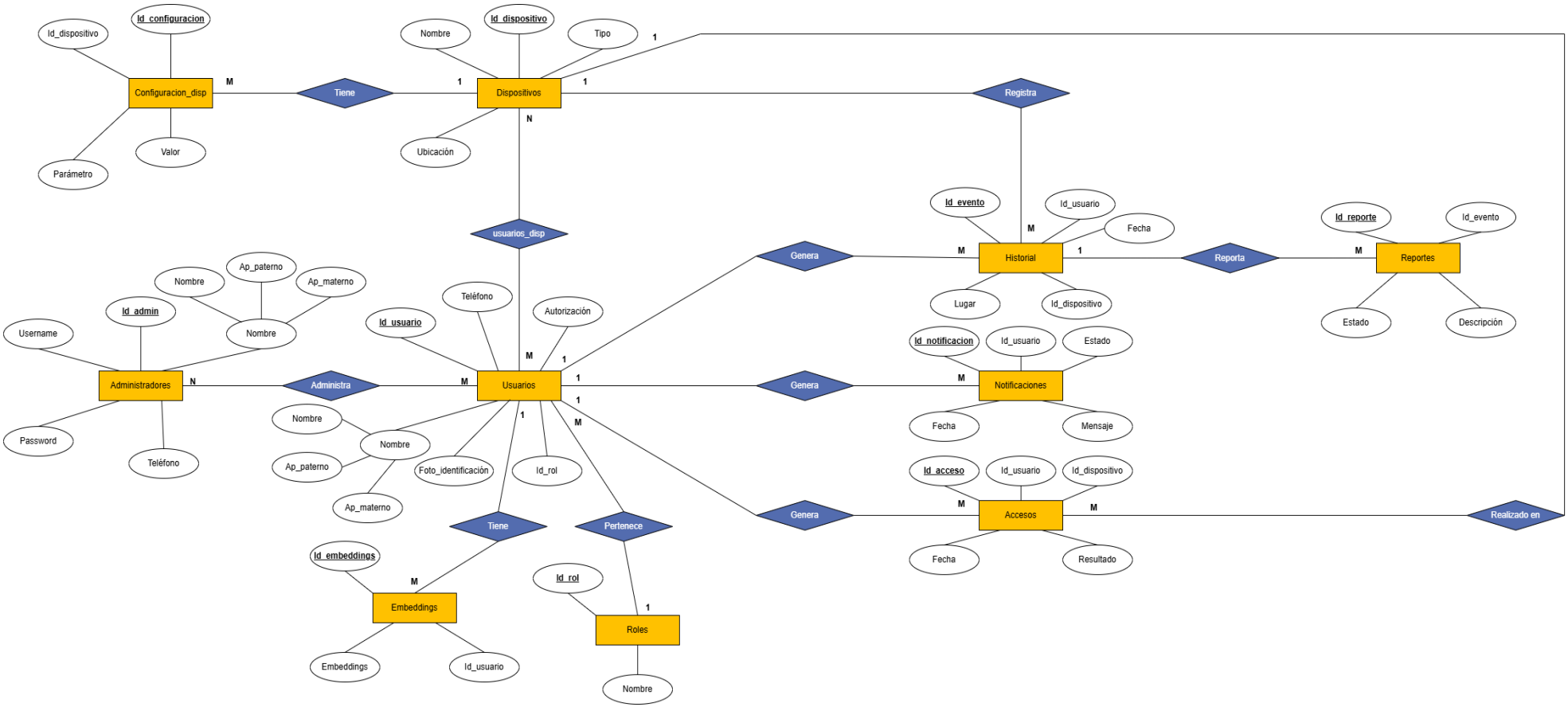


Diagrama RELACIONAL