

**Resolución de Problemas Aplicando el Modelo Relacional,  
Cardinalidad y Normalización**

**Proyecto: Censo Rural – Modelo Relacional, Cardinalidad y  
Normalización**

**GA6-220501096-AA1-EV01**



**Isidro J Gallardo Navarro**

**Ficha:3070299**

**2025**

**Tecnología en Análisis y Desarrollo de  
Software.**

**ADSO**

1. Introducción
2. Contexto del Proyecto "Censo Rural"
3. Identificación de Entidades y Requerimientos de Datos
4. Aplicación de Normalización hasta Tercera Forma Normal
5. Establecimiento de Cardinalidad entre Entidades
6. Definición de Esquemas de Seguridad e Integridad
7. Modelo Entidad-Relación y Herramientas
8. Validación del Diseño
9. Conclusiones y Recomendaciones
10. Referencias Bibliográficas
11. Anexos

## **1. Introducción**

### **1.1 Presentación**

El presente documento desarrolla la evidencia de aprendizaje GA6-220501096-AA1-EV01, correspondiente a la resolución de problemas mediante la aplicación del modelo relacional, establecimiento de cardinalidad y normalización de bases de datos. El proyecto seleccionado es "Censo Rural", un sistema de información diseñado para optimizar la recolección, gestión y análisis de datos personales de habitantes en zonas rurales apartadas de Colombia.

### **1.2 Objetivos**

#### **Objetivo**

#### **General:**

Diseñar una base de datos robusta, íntegra y segura para el proyecto "Censo Rural", aplicando los principios del modelo relacional, técnicas de normalización hasta la Tercera Forma Normal (3NF), establecimiento correcto de cardinalidades y definición de esquemas de seguridad e integridad referencial.

#### **Objetivos Específicos:**

1. Identificar y justificar las entidades principales del sistema basándose en los requerimientos del software
2. Aplicar técnicas de normalización para eliminar redundancias
3. Establecer relaciones entre entidades con cardinalidades apropiadas
4. Definir esquemas de seguridad para datos personales sensibles
5. Documentar el Modelo Entidad-Relación (MER) del sistema

### **1.3 Alcance**

Esta evidencia abarca el diseño lógico-relacional de la base de datos, incluyendo análisis conceptual de entidades, aplicación de formas normales (1NF, 2NF, 3NF), definición de relaciones y especificación de restricciones de integridad. Se utiliza SQLite3 como sistema gestor de base de datos por sus características offline-first requeridas para operar en zonas rurales con conectividad limitada.

## 2. Contexto del Proyecto "Censo Rural"

### 2.1 Problemática

Las zonas rurales apartadas de Colombia presentan desafíos únicos para la recolección de información demográfica: dispersión geográfica, conectividad limitada o nula a internet, infraestructura precaria y población vulnerable con necesidades específicas de atención. Estos factores dificultan la implementación de sistemas tradicionales que requieren conexión permanente.

### 2.2 Solución Propuesta

El proyecto "Censo Rural" implementa un sistema con arquitectura **offline-first**, permitiendo captura de datos en dispositivos móviles sin conectividad, almacenamiento local en SQLite3 y sincronización diferida cuando se disponga de conexión. Los componentes principales incluyen:

- Aplicación móvil para encuestadores (Android/iOS)
- Base de datos local SQLite3 en cada dispositivo
- Servidor central con PostgreSQL para consolidación
- API REST para sincronización de datos
- Dashboard web para supervisores y administradores

### 2.3 Requerimientos Funcionales Clave

**RF01 - Captura de datos personales:** Registro de información personal completa según campos especificados los requerimientos del software (primerNombre, segundoNombre, primerApellido, segundoApellido, tipoDocumento, numeroDocumento, fechaNacimiento, sexo, email, numeroCelular, Domicilio, direccion, barrio).

**RF02 - Geolocalización:** Captura de coordenadas GPS y ubicación detallada (vereda, municipio, departamento).

**RF03 - Gestión de usuarios:** Diferenciación de roles (Encuestador, Supervisor, Administrador) con permisos específicos.

**RF04 - Operación offline:** Funcionamiento completo sin conectividad, con sincronización posterior.

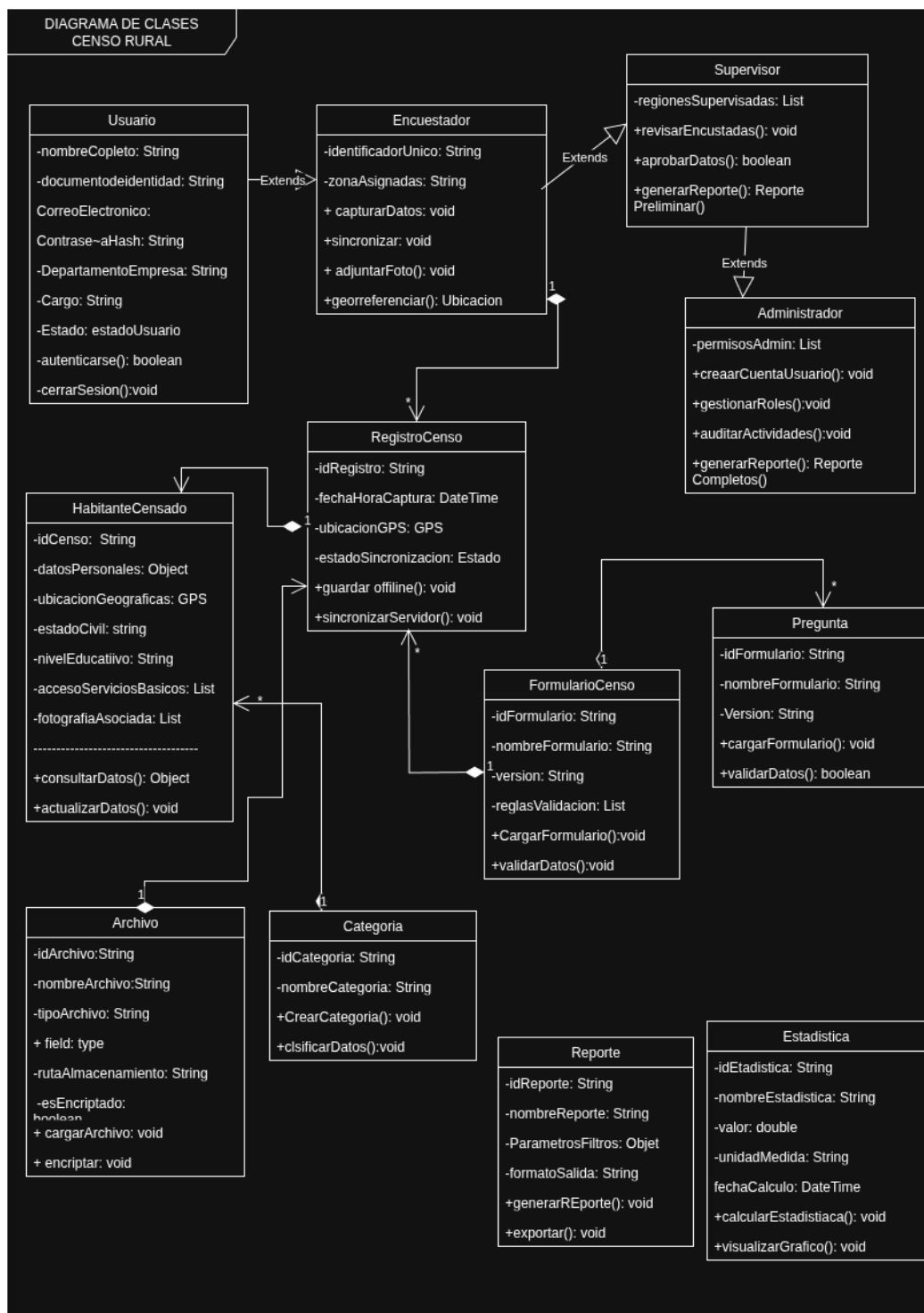
**RF05 - Trazabilidad:** Registro de quién capturó cada dato, cuándo y dónde.

**RF06 - Seguridad:** Protección de datos personales sensibles mediante cifrado y control de acceso, cumpliendo la Ley 1581 de 2012 (Colombia).

### **3. Identificación de Entidades y Requerimientos de Datos**

#### **3.1 Criterio 1 EV01: Entidades Identificadas**

La identificación de entidades se realizó mediante análisis estructurado de requisitos, revisión y aplicación de criterios de modelado conceptual. Cada entidad debe tener identidad propia, atributos descriptivos y relaciones significativas.



[LINK DEL DIAGRAMA DRAWIO](#) (DEBE CONTAR CON CUENTA DE GOOGLE PARA PODER ACCEDER, Tabla de Contenidos)

### 3.1.1 HabitanteCensado

**Definición:** Representa a cada persona censada, almacenando información personal, demográfica y de contacto.

**Justificación:** Es la entidad central del sistema, conteniendo los datos sensibles que son objetivo principal del censo.

**Atributos mapeados desde:**

1. Campo	Atributo BD	Tipo de Dato	Restricción
primerNombre	primer_nombre	VARCHAR(50)	NOT NULL
segundoNombre	segundo_nombre	VARCHAR(50)	NULL
primerApellido	primer_apellido	VARCHAR(50)	NOT NULL
segundoApellido	segundo_apellido	VARCHAR(50)	NOT NULL
tipoDocumento	tipo_documento	VARCHAR(10)	NOT NULL, CHECK
numeroDocumento	numero_documento	VARCHAR(20)	NOT NULL, UNIQUE
fechaNacimiento	fecha_nacimiento	DATE	NOT NULL
sexo	sexo	CHAR(1)	NOT NULL, CHECK
email	email	VARCHAR(100)	NULL
numeroCelular	numero_celular	VARCHAR(15)	NULL
Domicilio	domicilio	VARCHAR(200)	NOT NULL
direccion	direccion_detalle	VARCHAR(200)	NULL
barrio	barrio	VARCHAR(100)	NULL

**Atributos adicionales de control:**

- idHabitante: Clave primaria autoincremental
- idUbicacion: Clave foránea hacia Ubicacion (normalización)
- fecha\_registro: Timestamp de creación
- verificado: Booleano indicando validación de datos
- estado\_consentimiento: Control de consentimiento ('activo', 'revocado', 'anonimizado')

**Reglas de negocio:**

- El número de documento debe ser único en todo el sistema
- Tipo de documento: CC, TI, CE, PA, RC, NUI
- Sexo: M (Masculino), F (Femenino), O (Otro)

### 3.1.2 Ubicacion

**Definición:** Almacena información geográfica detallada del lugar de residencia.

**Justificación:** Separada de HabitanteCensado por normalización (3NF). Múltiples habitantes pueden compartir una ubicación (familia), evitando redundancia.

**Atributos principales:**

- idUbicacion: Clave primaria
- vereda, municipio, departamento: Unidades administrativas
- direccion\_rural: Descripción textual de ubicación
- latitud, longitud: Coordenadas GPS (DECIMAL, NOT NULL)
- altitud\_msnm: Altitud sobre el nivel del mar
- referencia\_geografica: Referencias adicionales

#### **Reglas de negocio:**

- Coordenadas GPS obligatorias para geolocalización precisa
- Latitud: -90 a 90 grados
- Longitud: -180 a 180 grados

### **3.1.3 Usuario**

**Definición:** Entidad genérica que representa actores del sistema con credenciales de acceso.

**Justificación:** Centraliza autenticación, permitiendo especialización posterior (Encuestador, Supervisor, Administrador).

#### **Atributos:**

- idUsuario: Clave primaria
- nombre\_completo: Nombre del usuario
- email: Correo electrónico único (UNIQUE)
- contrasena\_hash: Hash criptográfico (bcrypt/Argon2)
- fecha\_creacion, ultimo\_acceso: Control temporal
- estado\_activo: Cuenta habilitada/deshabilitada
- intentos\_fallidos: Control de intentos de login

#### **Reglas de negocio:**

- Email único en el sistema
- Contraseñas nunca en texto plano (solo hash)
- Bloqueo temporal tras 5 intentos fallidos

### **3.1.4 Encuestador**

**Definición:** Especialización de Usuario para trabajadores de campo.

**Justificación:** Hereda autenticación de Usuario y añade información del rol operativo.



**Atributos adicionales:**

- idUsuario: PK y FK hacia Usuario (herencia 1:1)
- idZona: FK hacia ZonaGeografica
- documento\_identidad: Cédula del encuestador (UNIQUE)
- meta\_diaria: Objetivo de registros por día
- fecha\_contratacion, fecha\_asignacion: Control temporal
- estado\_disponibilidad: 'disponible', 'en\_campo', 'descanso', 'inactivo'

**3.1.5 RegistroCenso**

**Definición:** Entidad transaccional que representa cada instancia de captura.

**Justificación:** Proporciona trazabilidad completa del proceso de censo.

**Atributos:**

- idRegistro: Clave primaria
- idHabitante, idEncuestador, idFormulario: Claves foráneas
- fecha\_hora\_captura: Timestamp de captura
- latitud\_captura, longitud\_captura: GPS del encuestador
- estado\_sincronizacion: 'pendiente', 'sincronizado', 'error', 'conflicto'
- duracion\_captura\_min: Tiempo de entrevista
- calidad\_datos: 'excelente', 'buena', 'regular', 'pobre'
- consentimiento\_informado: Booleano (NOT NULL)

**3.1.6 FormularioCenso**

**Definición:** Define estructura y validaciones del cuestionario.

**Justificación:** Permite versionamiento sin afectar registros históricos.

**Atributos:**

- idFormulario: Clave primaria
- nombre\_formulario, version: Identificación (UNIQUE combinado)
- estructura\_json: Definición de campos en JSON
- fecha\_creacion, fecha\_vigencia\_inicio, fecha\_vigencia\_fin
- estado\_activo: Disponibilidad para uso

### 3.1.7 ZonaGeografica

**Definición:** Áreas territoriales de cobertura para organización operativa.

**Atributos:**

- idZona: Clave primaria
- codigo\_zona: Código oficial (UNIQUE)
- nombre\_zona, municipio, departamento
- area\_km2, poblacion\_estimada
- nivel\_prioridad: 1 (urgente) a 5 (baja)
- poligono\_geojson: Delimitación geográfica

### 3.1.8 LogAuditoria

**Definición:** Registro inmutable de operaciones significativas.

**Justificación:** Cumplimiento normativo (Ley 1581/2012) y trazabilidad.

**Atributos:**

- idLog: Clave primaria
- idUsuario: FK hacia Usuario
- tipo\_accion: 'INSERT', 'UPDATE', 'DELETE', 'LOGIN', 'LOGOUT'
- tabla\_afectada, registro\_afectado
- fecha\_hora: Timestamp preciso
- valores\_anteriores, valores\_nuevos: JSON con datos

## 4. Aplicación de Normalización hasta Tercera Forma Normal

### 4.1 Fundamentos Teóricos

La normalización es un proceso sistemático propuesto por Edgar F. Codd en 1970 para organizar datos reduciendo redundancia y mejorando integridad. Las formas normales progresivas son:

- **1NF:** Eliminar grupos repetitivos, garantizar atomicidad
- **2NF:** Eliminar dependencias parciales de la clave primaria
- **3NF:** Eliminar dependencias transitivas entre atributos no clave

## 4.2 Criterio 3 EV01: Análisis de Normalización

### 4.2.1 HabitanteCensado: Alcanzando 3NF

**Estado inicial (antes de normalización):**

Tabla monolítica con todos los campos incluyendo ubicación:

HABITANTE(idHabitante, primer\_nombre, ..., vereda, municipio, departamento, latitud, longitud)

**Verificación** **de** **1NF:**

**Cumple:** Valores atómicos, sin grupos repetitivos, filas únicas.

**Verificación** **de** **2NF:**

**Cumple:** Clave primaria simple (idHabitante), no hay dependencias parciales.

**Verificación** **de** **3NF:**

**NO cumple inicialmente** por dependencia transitiva:

idHabitante → (latitud, longitud) → vereda, municipio, departamento

**Solución aplicada:** Separación en dos entidades:



```
HABITANTE_CENSADO(  
    idHabitante [PK],  
    idUbicacion [FK],  
    primer_nombre, segundo_nombre, primer_apellido, segundo_apellido,  
    tipo_documento, numero_documento, fecha_nacimiento, sexo,  
    email, numero_celular  
)
```

```
UBICACION(  
    idUbicacion [PK],  
    vereda, municipio, departamento,  
    direccion_rural, latitud, longitud, altitud_msnm  
)
```

**Ahora cumple 3NF:** No hay dependencias transitivas, cada atributo depende directamente de su clave primaria.

**Beneficios:**

1. Eliminación de redundancia (ubicación compartida por familia)
2. Consistencia en actualizaciones de coordenadas GPS
3. Integridad garantizada
4. Escalabilidad facilitada

#### 4.2.2 Usuario y Encuestador: Implementación de Herencia

**Patrón de herencia mediante relación 1:1:**

```
USUARIO(  
    idUsuario [PK],  
    nombre_completo, email, contrasena_hash,  
    fecha_creacion, ultimo_acceso, estado_activo  
)  
  
ENCUESTADOR(  
    idUsuario [PK, FK → Usuario],  
    idZona [FK → ZonaGeografica],  
    documento_identidad, meta_diaria, estado_disponibilidad  
)
```

**Ambas cumplen 3NF:**

- Usuario: Atributos de autenticación sin dependencias transitivas
- Encuestador: Atributos operativos dependen directamente de idUsuario

### Beneficios:

- Eliminación de redundancia en datos de autenticación
- Extensibilidad para nuevos roles
- Separación de responsabilidades

#### 4.2.3 RegistroCenso: Entidad Asociativa en 3NF

```
REGISTRO_CENSO(  
    idRegistro [PK],  
    idHabitante [FK], idEncuestador [FK], idFormulario [FK],  
    fecha_hora_captura, latitud_captura, longitud_captura,  
    estado_sincronizacion, duracion_captura_min, calidad_datos  
)
```

### Análisis:

```
REGISTRO_CENSO(  
    idRegistro [PK],  
    idHabitante [FK], idEncuestador [FK], idFormulario [FK],  
    fecha_hora_captura, latitud_captura, longitud_captura,  
    estado_sincronizacion, duracion_captura_min, calidad_datos  
)
```

### Cumple 3NF:

- Atributos contextuales (fecha, ubicación GPS del encuestador, duración) pertenecen al evento de registro
- No hay dependencias transitivas
- Cada atributo depende de idRegistro

**Justificación de cardinalidad 1:N con HabitanteCensado:**  
Permite censos de seguimiento temporal manteniendo 3NF, ya que cada registro es una entidad independiente con contexto temporal propio.

#### 4.2.4 FormularioCenso: Versionamiento y 3NF

```
FORMULARIO_CENSO(  
  idFormulario [PK],  
  nombre_formulario, version,  
  estructura_json, fecha_creacion,  
  UNIQUE(nombre_formulario, version)  
)
```

**Cumple 3NF:** Todos los atributos dependen de idFormulario, sin dependencias transitivas.

**Beneficio del diseño:** Permite evolución del cuestionario sin migración de datos históricos.

### 5. Establecimiento de Cardinalidad entre Entidades

#### 5.1 Fundamentos de Cardinalidad

La cardinalidad define restricciones cuantitativas en relaciones entre entidades, especificando cuántas instancias pueden asociarse.

**Notación:** (min, max) en cada extremo de la relación.

#### 5.2 Criterio 2 EV01: Catálogo de Relaciones

##### 5.2.1 Encuestador CAPTURA RegistroCenso

**Cardinalidad:** Uno a Muchos (1:N)

Encuestador (1,1) — CAPTURA — (0,N) RegistroCenso

**Interpretación:**

- Cada RegistroCenso es capturado por exactamente UN Encuestador

- Un Encuestador puede haber capturado CERO o MUCHOS RegistroCenso

#### Justificación:

1. **Responsabilidad individual:** Cada registro tiene responsable identificable
2. **Productividad:** Un encuestador realiza 15-25 registros/día
3. **Trazabilidad:** Permite análisis de desempeño por encuestador

#### Implementación:

```
CREATE TABLE ENCUESTADOR (
  idUsuario INTEGER PRIMARY KEY, -- PK y FK simultáneamente
  FOREIGN KEY (idUsuario) REFERENCES Usuario(idUsuario)
  ON DELETE CASCADE -- Encuestador no existe sin Usuario
);
```

### 5.2.2 HabitanteCensado TIENE RegistroCenso

**Cardinalidad:** Uno a Muchos (1:N)

HabitanteCensado (1,1) — TIENE — (1,N) RegistroCenso

#### Interpretación:

- Cada RegistroCenso corresponde a exactamente UN HabitanteCensado
- Un HabitanteCensado debe tener AL MENOS UN RegistroCenso

#### Justificación:

1. **Censos periódicos:** Seguimiento temporal del habitante
2. **Corrección de errores:** Nuevo registro sin eliminar histórico
3. **Historial temporal:** Análisis de cambios en situación del habitante

#### Atributo sugerido para control:

registro\_vigente BOOLEAN DEFAULT TRUE -- Identifica registro más reciente

### 5.2.3 HabitanteCensado RESIDE\_EN Ubicacion

**Cardinalidad:** Muchos a Uno (N:1)

HabitanteCensado (0,N) ——— RESIDE\_EN ——— (1,1) Ubicacion

**Interpretación:**

- Cada HabitanteCensado reside en exactamente UNA Ubicacion
- Una Ubicacion puede tener CERO o MUCHOS HabitanteCensado

**Justificación:**

1. **Realidad rural:** Familias comparten vivienda
2. **Eficiencia:** Coordenadas GPS almacenadas una sola vez
3. **Análisis espacial:** Densidad poblacional por ubicación

**Ejemplo real:**

Ubicacion: Finca Los Robles (4.6010, -74.0001)

|—— Pedro Gómez (padre, 45 años)

|—— María López (madre, 42 años)

|—— Juan Gómez (hijo, 18 años)

|—— Ana Gómez (hija, 15 años)

### 5.2.4 ZonaGeografica ASIGNADA\_A Encuestador

**Cardinalidad:** Uno a Muchos (1:N)

ZonaGeografica (1,1) ——— ASIGNADA\_A ——— (0,N) Encuestador

**Interpretación:**

- Cada Encuestador está asignado a exactamente UNA ZonaGeografica
- Una ZonaGeografica puede tener CERO o MUCHOS Encuestador

**Justificación:**

1. **Organización territorial:** División en zonas para planificación
2. **Especialización geográfica:** Conocimiento profundo de su zona
3. **Distribución de carga:** Múltiples encuestadores por área extensa



### 5.2.5 FormularioCenso DEFINE RegistroCenso

**Cardinalidad:** Uno a Muchos (1:N)

FormularioCenso (1,1) ——— DEFINE ——— (0,N) RegistroCenso

**Interpretación:**

- Cada RegistroCenso utiliza exactamente UN FormularioCenso
- Un FormularioCenso puede ser usado en CERO o MUCHOS RegistroCenso

**Justificación:**

1. **Trazabilidad de versiones:** Saber qué cuestionario se aplicó
2. **Evolución sin migración:** Actualizar preguntas sin afectar histórico
3. **Análisis comparativo:** Identificar impacto de cambios en formulario

**Restricción:**

FOREIGN KEY (idFormulario) REFERENCES FormularioCenso(idFormulario)  
ON DELETE RESTRICT -- No eliminar formularios con registros

### 5.2.6 Usuario HEREDA Encuestador

**Cardinalidad:** Uno a Uno (1:1)

Usuario (1,1) ——— HEREDA ——— (0,1) Encuestador

**Interpretación:**

- Cada Encuestador está asociado a exactamente UN Usuario
- Un Usuario puede ser opcionalmente un Encuestador

**Justificación:**

1. **Patrón de herencia:** Especialización en modelo relacional
2. **Separación de responsabilidades:** Autenticación vs operaciones
3. **Extensibilidad:** Facilita añadir roles (Supervisor, Admin)

**Implementación:**

```

CREATE TABLE ENCUESTADOR (
    idUsuario INTEGER PRIMARY KEY, -- PK y FK simultáneamente
    FOREIGN KEY (idUsuario) REFERENCES Usuario(idUsuario)
        ON DELETE CASCADE -- Encuestador no existe sin Usuario
);

```

## 6. Definición de Esquemas de Seguridad e Integridad

### 6.1 Importancia en Datos Personales Sensibles

El proyecto maneja información clasificada como **datos personales sensibles** según Ley 1581/2012 (Colombia), incluyendo identificación, ubicación GPS, datos demográficos y contacto.

### 6.2 Criterio 1 EV01: Esquemas Implementados

#### 6.2.1 Integridad Referencial mediante Claves Foráneas

Implementación:

```

ALTER TABLE REGISTRO_CENSO
ADD CONSTRAINT fk_registro_habitante
    FOREIGN KEY (idHabitante) REFERENCES HABITANTE_CENSADO(idHabitante)
        ON DELETE RESTRICT ON UPDATE CASCADE;

```

Políticas:

- **ON DELETE RESTRICT:** Protege integridad histórica y auditoría
- **ON DELETE CASCADE:** Solo para herencia (Usuario → Encuestador)
- **ON UPDATE CASCADE:** Propaga actualizaciones automáticamente

**Beneficios:** Previene registros huérfanos, garantiza trazabilidad completa.

#### 6.2.2 Restricciones de Unicidad

```
-- HabitanteCensado: documento único
numero_documento VARCHAR(20) NOT NULL UNIQUE

-- Usuario: email único
email VARCHAR(100) NOT NULL UNIQUE

-- FormularioCenso: combinación nombre-versión única
UNIQUE(nombre_formulario, version)
```

#### Importancia:

- Evita censar la misma persona dos veces
- Previene cuentas duplicadas
- Garantiza unicidad de versiones de formularios

#### 6.2.3 Validación de Dominios mediante CHECK

##### Validaciones implementadas

```
CHECK (sexo IN ('M', 'F', 'Otro'))
CHECK (tipo_documento IN ('CC', 'TI', 'CE', 'PA', 'RC', 'NUI'))
CHECK (latitud BETWEEN -90 AND 90)
CHECK (longitud BETWEEN -180 AND 180)
CHECK (estado_sincronizacion IN ('pendiente', 'sincronizado', 'error', 'conflicto'))
CHECK (nivel_prioridad BETWEEN 1 AND 5)
```

**Beneficios:** Previene datos inconsistentes, facilita reportes con valores predecibles.

#### 6.2.4 Almacenamiento Seguro de Contraseñas

**Principio:** NUNCA almacenar contraseñas en texto plano.

```
CREATE TABLE USUARIO (
    contrasena_hash VARCHAR(255) NOT NULL, -- Almacena hash, no contraseña
);
```

**Algoritmo recomendado: bcrypt**

```
import bcrypt

Al crear usuario
hashed = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt(rounds=12))

Al verificar login
if bcrypt.checkpw(password.encode('utf-8'), stored_hash):
    Autenticación exitosa
```

**Política de contraseñas:**

- Longitud mínima: 8 caracteres
- Complejidad: 1 mayúscula, 1 minúscula, 1 número
- Rotación: Cambio cada 90 días
- Bloqueo temporal: 5 intentos fallidos

**6.2.5 Control de Acceso Basado en Roles (RBAC)**

**Matriz de permisos:**

Operación	Encuestador	Supervisor	Admin
Crear HabitanteCensado	✓	✓	✓
Actualizar propio registro	✓	✓	✓
Actualizar registro de otros	✗	✓	✓
Eliminar registros	✗	✗	✓
Ver estadísticas globales	✗	✗	✓
Gestionar usuarios	✗	Parcial	✓
Ver logs de auditoría	✗	Zona	✓

## Implementación en aplicación:

```
function requireRole(allowedRoles) {  
  return (req, res, next) => {  
    if (allowedRoles.includes(req.user.rol)) {  
      next();  
    } else {  
      res.status(403).json({ error: 'Acceso denegado' });  
    }  
  };  
}
```

### 6.2.6 Auditoría Completa mediante LogAuditoria

#### Trigger de auditoría automática:

```
CREATE TRIGGER trg_audit_habitante_insert  
AFTER INSERT ON HABITANTE_CENSADO  
FOR EACH ROW  
BEGIN  
  INSERT INTO LOG_AUDITORIA (  
    idUsuario, tipo_accion, tabla_afectada,  
    registro_afectado, descripcion  
  ) VALUES (  
    (SELECT idUsuarioActual FROM SessionContext LIMIT 1),  
    'INSERT', 'HABITANTE_CENSADO', NEW.idHabitante,  
    'Nuevo habitante: ' || NEW.primer_nombre || ' ' || NEW.primer_apellido  
  );  
END;
```

#### Eventos auditados:

1. Todas las operaciones sobre HabitanteCensado
2. Intentos de autenticación (exitosos y fallidos)

3. Cambios en usuarios
4. Exportación de datos personales
5. Modificaciones en FormularioCenso

#### Características de LogAuditoria:

- **Insert-only:** No se modifica ni elimina (integridad forense)
- **Cumplimiento normativo:** Requerido por Ley 1581/2012
- **Investigación de incidentes:** Capacidad forense completa

#### 6.2.7 Cifrado de Datos Sensibles

##### Estrategias recomendadas:

1. **Cifrado de base de datos completa:** SQLCipher para dispositivos móviles
2. **Cifrado a nivel de aplicación:** AES-256 para campos específicos
3. **Cifrado en tránsito:** TLS 1.3 para sincronización

##### Campos que requieren cifrado:

- Números de documento
- Email y teléfono
- Coordenadas GPS precisas

##### Implementación SQLCipher:

A code editor window with a dark blue background and a yellow border. It contains Python code for connecting to a SQLite database using SQLCipher. The code includes imports for sqlite3 and dbapi2, a comment in Spanish, and the connection and execution of a PRAGMA statement to set a key.

```
import sqlite3
from pysqlcipher3 import dbapi2 as sqlite

Conexión con cifrado
conn = sqlite.connect('censo_rural.db')
conn.execute("PRAGMA key='clave_segura_compleja'")
```

#### 6.2.8 Cumplimiento de Derecho de Habeas Data

##### Derecho de acceso:

```
SELECT * FROM HABITANTE_CENSADO
WHERE numero_documento = ? AND tipo_documento = ?;
Derecho de rectificación:
UPDATE HABITANTE_CENSADO
SET email = ?, numero_celular = ?
WHERE idHabitante = ?;
-- Trigger registra la modificación en LogAuditoria
```

**Derecho de supresión (anonimización preferida):**

```
UPDATE HABITANTE_CENSADO
SET primer_nombre = 'ANONIMIZADO',
    numero_documento = 'ANONIMIZADO-' || idHabitante,
    email = NULL, numero_celular = NULL,
    estado_consentimiento = 'anonimizado'
WHERE idHabitante = ?;
```