**MODELADO DE BASE DE DATOS - FUNDACIÓN ATMA NAMASTÉ**

**📊 RESUMEN GENERAL**

**Total de Entidades:** 14  
**Total de ENUMs:** 20  
**Enfoque:** ENUMs nativos para catálogos estables  
**Base de Datos:** PostgreSQL  
**ORM:** Prisma

**🏗️ MÓDULOS Y SUS ENTIDADES**

**MÓDULO 1: CORE (3 entidades)**

1. Persona
2. TutorMenor
3. Usuario

**MÓDULO 2: PROGRAMAS (4 entidades)**

1. Programa
2. ProgramaResponsable
3. Inscripcion
4. Asistencia

**MÓDULO 3: APADRINAMIENTO (3 entidades)**

1. Padrino
2. Apadrinamiento
3. Donacion

**MÓDULO 4: RECURSOS (3 entidades)**

1. Entrega
2. EntregaBeneficiario
3. Gasto

**MÓDULO 5: AUDITORÍA (1 entidad)**

1. Auditoria

**📐 ENTIDADES DETALLADAS**

**1. PERSONA**

**Descripción:** Entidad central que representa a cualquier individuo vinculado a la fundación.

**Campos:**

* id - UUID, PK
* tipoDocumento - ENUM: TipoDocumento
* numeroDocumento - String(20), UNIQUE, NOT NULL
* primerNombre - String(50), NOT NULL
* segundoNombre - String(50), NULL
* primerApellido - String(50), NOT NULL
* segundoApellido - String(50), NULL
* fechaNacimiento - Date, NOT NULL
* edad - Integer, computed (calculado automáticamente)
* genero - ENUM: Genero, NOT NULL
* direccion - String(200), NOT NULL
* barrio - String(100), NULL
* telefono - String(20), NOT NULL
* email - String(100), UNIQUE, NULL
* tipoPersona - ENUM: TipoPersona, NOT NULL
* estado - ENUM: EstadoGeneral, NOT NULL, DEFAULT: ACTIVO
* observaciones - Text, NULL
* fotografiaUrl - String(500), NULL
* consentimientoInformado - Boolean, NOT NULL, DEFAULT: false
* documentoConsentimientoUrl - String(500), NULL
* creadoPor - UUID, FK → Usuario, NOT NULL
* modificadoPor - UUID, FK → Usuario, NULL
* createdAt - DateTime, NOT NULL, AUTO
* updatedAt - DateTime, NOT NULL, AUTO

**Índices:**

* numeroDocumento (UNIQUE)
* email (UNIQUE, where not null)
* (tipoPersona, estado) (compuesto)
* (primerNombre, primerApellido) (búsqueda)

**Reglas de Negocio:**

* Si tipoPersona = MENOR → consentimientoInformado debe ser true
* numeroDocumento único en el sistema
* email único si existe
* edad = YEAR(NOW()) - YEAR(fechaNacimiento)

**Relaciones:**

* 1:1 con **Usuario** (opcional)
* 1:N con **TutorMenor** (como tutor)
* 1:N con **TutorMenor** (como menor)
* 1:N con **Inscripcion**
* 1:N con **ProgramaResponsable** (si es docente)
* 1:N con **Apadrinamiento** (si es menor)
* 1:N con **Entrega**
* 1:N con **EntregaBeneficiario**
* 1:N con **Gasto** (si recibe pagos)

**2. TUTOR\_MENOR**

**Descripción:** Relación legal entre tutores y menores de edad.

**Campos:**

* id - UUID, PK
* tutorId - UUID, FK → Persona, NOT NULL
* menorId - UUID, FK → Persona, NOT NULL
* parentesco - ENUM: Parentesco, NOT NULL
* parentescoOtro - String(50), NULL
* esTutorPrincipal - Boolean, NOT NULL, DEFAULT: false
* fechaInicio - Date, NOT NULL, DEFAULT: today()
* fechaFin - Date, NULL
* estado - ENUM: EstadoGeneral, NOT NULL, DEFAULT: ACTIVO
* createdAt - DateTime, NOT NULL, AUTO
* updatedAt - DateTime, NOT NULL, AUTO

**Índices:**

* (tutorId, estado) (compuesto)
* (menorId, estado) (compuesto)
* (menorId, esTutorPrincipal, estado) (UNIQUE where esTutorPrincipal=true AND estado=ACTIVO)

**Reglas de Negocio:**

* Un menor DEBE tener al menos un tutor con estado ACTIVO
* Solo UN tutor puede tener esTutorPrincipal = true por menor activo
* Si parentesco = OTRO → parentescoOtro es obligatorio
* tutorId debe ser Persona con tipoPersona que incluya TUTOR
* menorId debe ser Persona con tipoPersona = MENOR

**Relaciones:**

* N:1 con **Persona** (tutor)
* N:1 con **Persona** (menor)

**3. USUARIO**

**Descripción:** Credenciales de acceso al sistema.

**Campos:**

* id - UUID, PK
* personaId - UUID, FK → Persona, UNIQUE, NOT NULL
* username - String(50), UNIQUE, NOT NULL
* email - String(100), UNIQUE, NOT NULL
* password - String(255), NOT NULL (hash bcrypt)
* rol - ENUM: RolUsuario, NOT NULL
* estado - ENUM: EstadoUsuario, NOT NULL, DEFAULT: ACTIVO
* ultimoAcceso - DateTime, NULL
* createdAt - DateTime, NOT NULL, AUTO
* updatedAt - DateTime, NOT NULL, AUTO

**Índices:**

* username (UNIQUE)
* email (UNIQUE)
* personaId (UNIQUE)
* (estado, rol) (compuesto)

**Reglas de Negocio:**

* Solo Persona con tipoPersona = DOCENTE o ADMINISTRATIVO puede tener usuario
* password debe hashearse con bcrypt (10 rounds)
* username debe ser único en todo el sistema
* email debe ser único y válido

**Relaciones:**

* 1:1 con **Persona**
* 1:N con **Auditoria**
* 1:N con todas las tablas que tienen campos creadoPor, modificadoPor, registradoPor

**4. PROGRAMA**

**Descripción:** Programas académicos, lúdicos y de apoyo.

**Campos:**

* id - UUID, PK
* codigo - String(20), UNIQUE, NOT NULL
* nombre - String(100), NOT NULL
* descripcion - Text, NULL
* tipoPrograma - ENUM: TipoPrograma, NOT NULL
* publicoObjetivo - ENUM: PublicoObjetivo, NOT NULL
* edadMinima - Integer, NULL
* edadMaxima - Integer, NULL
* diasActividad - ENUM[]: DiaSemana, NULL (array)
* horaInicio - Time, NULL
* horaFin - Time, NULL
* requiereDocente - Boolean, NOT NULL, DEFAULT: true
* cupoMaximo - Integer, NULL
* estado - ENUM: EstadoPrograma, NOT NULL, DEFAULT: ACTIVO
* creadoPor - UUID, FK → Usuario, NOT NULL
* createdAt - DateTime, NOT NULL, AUTO
* updatedAt - DateTime, NOT NULL, AUTO

**Índices:**

* codigo (UNIQUE)
* (estado, tipoPrograma) (compuesto)
* publicoObjetivo

**Reglas de Negocio:**

* Si requiereDocente = true → debe tener al menos un ProgramaResponsable activo
* edadMaxima > edadMinima si ambos existen
* horaFin > horaInicio si ambos existen
* codigo formato sugerido: DANZA-01, MUSICA-02

**Relaciones:**

* 1:N con **ProgramaResponsable**
* 1:N con **Inscripcion**
* 1:N con **Donacion**
* 1:N con **Entrega**
* 1:N con **Gasto**

**5. PROGRAMA\_RESPONSABLE**

**Descripción:** Asignación de docentes a programas.

**Campos:**

* id - UUID, PK
* programaId - UUID, FK → Programa, NOT NULL
* personaId - UUID, FK → Persona, NOT NULL
* esResponsablePrincipal - Boolean, NOT NULL, DEFAULT: false
* fechaAsignacion - Date, NOT NULL, DEFAULT: today()
* fechaFinalizacion - Date, NULL
* estado - ENUM: EstadoGeneral, NOT NULL, DEFAULT: ACTIVO
* createdAt - DateTime, NOT NULL, AUTO
* updatedAt - DateTime, NOT NULL, AUTO

**Índices:**

* (programaId, estado) (compuesto)
* (personaId, estado) (compuesto)
* (programaId, esResponsablePrincipal, estado) (UNIQUE where esResponsablePrincipal=true AND estado=ACTIVO)

**Reglas de Negocio:**

* personaId debe ser Persona con tipoPersona = DOCENTE
* Un programa debe tener al menos un responsable con esResponsablePrincipal = true
* Un docente puede ser responsable de múltiples programas

**Relaciones:**

* N:1 con **Programa**
* N:1 con **Persona** (docente)

**6. INSCRIPCION**

**Descripción:** Beneficiarios inscritos en programas.

**Campos:**

* id - UUID, PK
* programaId - UUID, FK → Programa, NOT NULL
* personaId - UUID, FK → Persona, NOT NULL
* fechaInscripcion - Date, NOT NULL, DEFAULT: today()
* fechaRetiro - Date, NULL
* estado - ENUM: EstadoInscripcion, NOT NULL, DEFAULT: ACTIVO
* motivoRetiro - Text, NULL
* observaciones - Text, NULL
* inscritoPor - UUID, FK → Usuario, NOT NULL
* createdAt - DateTime, NOT NULL, AUTO
* updatedAt - DateTime, NOT NULL, AUTO

**Índices:**

* (programaId, estado) (compuesto)
* (personaId, estado) (compuesto)
* (programaId, personaId, estado) (UNIQUE where estado=ACTIVO)
* fechaInscripcion

**Reglas de Negocio:**

* UNIQUE(programaId, personaId) WHERE estado = ACTIVO
* Una persona no puede tener dos inscripciones ACTIVAS al mismo programa
* Si estado = RETIRADO → fechaRetiro y motivoRetiro obligatorios
* Validar edad de persona vs edadMinima/edadMaxima del programa
* Validar cupos: COUNT(inscripciones activas) < programa.cupoMaximo

**Relaciones:**

* N:1 con **Programa**
* N:1 con **Persona**
* 1:N con **Asistencia**

**7. ASISTENCIA**

**Descripción:** Registro de asistencia a clases.

**Campos:**

* id - UUID, PK
* inscripcionId - UUID, FK → Inscripcion, NOT NULL
* fecha - Date, NOT NULL
* estado - ENUM: EstadoAsistencia, NOT NULL
* observaciones - Text, NULL
* registradoPor - UUID, FK → Usuario, NOT NULL
* createdAt - DateTime, NOT NULL, AUTO
* updatedAt - DateTime, NOT NULL, AUTO

**Índices:**

* (inscripcionId, fecha) (UNIQUE)
* fecha
* (fecha, estado) (compuesto)

**Reglas de Negocio:**

* UNIQUE(inscripcionId, fecha)
* Solo UN registro de asistencia por inscripción por fecha
* La inscripción debe estar en estado ACTIVO
* fecha no puede ser futura

**Relaciones:**

* N:1 con **Inscripcion**
* N:1 con **Usuario** (registradoPor)

**8. PADRINO**

**Descripción:** Personas u organizaciones que apadrinan niños.

**Campos:**

* id - UUID, PK
* personaId - UUID, FK → Persona, NULL (si es empresa/organización)
* tipo - ENUM: TipoPadrino, NOT NULL
* nombreCompleto - String(200), NOT NULL
* nombreEmpresa - String(200), NULL
* pais - String(100), NOT NULL
* ciudad - String(100), NULL
* email - String(100), NOT NULL
* telefono - String(20), NULL
* tipoDonacion - ENUM: TipoDonacionPadrino, NOT NULL
* frecuenciaDonacion - ENUM: FrecuenciaDonacion, NOT NULL
* preferenciaPrivacidad - ENUM: Privacidad, NOT NULL, DEFAULT: PRIVADA
* fechaInicio - Date, NOT NULL, DEFAULT: today()
* estado - ENUM: EstadoGeneral, NOT NULL, DEFAULT: ACTIVO
* observaciones - Text, NULL
* creadoPor - UUID, FK → Usuario, NOT NULL
* createdAt - DateTime, NOT NULL, AUTO
* updatedAt - DateTime, NOT NULL, AUTO

**Índices:**

* estado
* pais
* email

**Reglas de Negocio:**

* Si tipo != PERSONA\_NATURAL → nombreEmpresa es obligatorio
* email debe tener formato válido
* Información completamente confidencial (solo acceso admin)

**Relaciones:**

* N:1 con **Persona** (opcional)
* 1:N con **Apadrinamiento**
* 1:N con **Donacion**

**9. APADRINAMIENTO**

**Descripción:** Relación entre padrino y niño apadrinado.

**Campos:**

* id - UUID, PK
* padrinoId - UUID, FK → Padrino, NOT NULL
* menorId - UUID, FK → Persona, NOT NULL
* fechaInicio - Date, NOT NULL, DEFAULT: today()
* fechaFin - Date, NULL
* estado - ENUM: EstadoApadrinamiento, NOT NULL, DEFAULT: ACTIVO
* motivoFinalizacion - Text, NULL
* observaciones - Text, NULL
* creadoPor - UUID, FK → Usuario, NOT NULL
* createdAt - DateTime, NOT NULL, AUTO
* updatedAt - DateTime, NOT NULL, AUTO

**Índices:**

* (padrinoId, estado) (compuesto)
* (menorId, estado) (compuesto)
* menorId (UNIQUE where estado=ACTIVO)

**Reglas de Negocio:**

* UNIQUE(menorId) WHERE estado = ACTIVO
* Un menor solo puede tener UN apadrinamiento activo a la vez
* Un padrino puede tener múltiples apadrinamientos activos
* menorId debe ser Persona con tipoPersona = MENOR
* Si estado = FINALIZADO → fechaFin y motivoFinalizacion obligatorios

**Relaciones:**

* N:1 con **Padrino**
* N:1 con **Persona** (menor)
* 1:N con **Donacion**
* 1:N con **Entrega**

**10. DONACION**

**Descripción:** Registro de donaciones recibidas.

**Campos:**

* id - UUID, PK
* tipo - ENUM: TipoDonacion, NOT NULL
* padrinoId - UUID, FK → Padrino, NULL
* apadrinamientoId - UUID, FK → Apadrinamiento, NULL
* programaId - UUID, FK → Programa, NULL
* donadorNombre - String(200), NULL
* esAnonima - Boolean, NOT NULL, DEFAULT: false
* tipoRecurso - ENUM: TipoRecurso, NOT NULL
* monto - Decimal(12,2), NULL
* moneda - String(3), NOT NULL, DEFAULT: 'COP'
* descripcionEspecie - Text, NULL
* fechaRecepcion - Date, NOT NULL, DEFAULT: today()
* metodoPago - ENUM: MetodoPago, NULL
* numeroReferencia - String(100), NULL
* descripcion - Text, NULL
* observaciones - Text, NULL
* comprobanteUrl - String(500), NULL
* registradoPor - UUID, FK → Usuario, NOT NULL
* createdAt - DateTime, NOT NULL, AUTO
* updatedAt - DateTime, NOT NULL, AUTO

**Índices:**

* tipo
* fechaRecepcion (DESC)
* (tipo, fechaRecepcion) (compuesto)
* padrinoId
* programaId

**Reglas de Negocio:**

* Si tipo = PADRINO → padrinoId y apadrinamientoId obligatorios
* Si tipo = PARA\_PROGRAMA → programaId obligatorio
* Si tipo = GENERAL → donadorNombre obligatorio (a menos que esAnonima = true)
* Si tipoRecurso = MONETARIA → monto obligatorio y > 0
* Si tipoRecurso = EN\_ESPECIE → descripcionEspecie obligatoria
* fechaRecepcion no puede ser futura
* Donaciones de padrinos son confidenciales

**Relaciones:**

* N:1 con **Padrino**
* N:1 con **Apadrinamiento**
* N:1 con **Programa**
* N:1 con **Usuario** (registradoPor)

**11. ENTREGA**

**Descripción:** Ayudas y recursos entregados a beneficiarios.

**Campos:**

* id - UUID, PK
* tipo - ENUM: TipoEntrega, NOT NULL
* programaId - UUID, FK → Programa, NULL
* apadrinamientoId - UUID, FK → Apadrinamiento, NULL
* personaId - UUID, FK → Persona, NULL (NULL si es masiva)
* tipoRecurso - ENUM: TipoRecurso, NOT NULL
* monto - Decimal(12,2), NULL
* descripcionItems - Text, NOT NULL
* fechaProgramada - Date, NOT NULL
* fechaEntrega - Date, NULL
* estado - ENUM: EstadoEntrega, NOT NULL, DEFAULT: PROGRAMADA
* motivoCancelacion - Text, NULL
* confirmacionRecibido - Boolean, NOT NULL, DEFAULT: false
* firmaUrl - String(500), NULL
* comprobanteUrl - String(500), NULL
* observaciones - Text, NULL
* registradoPor - UUID, FK → Usuario, NOT NULL
* entregadoPor - UUID, FK → Usuario, NULL
* createdAt - DateTime, NOT NULL, AUTO
* updatedAt - DateTime, NOT NULL, AUTO

**Índices:**

* tipo
* estado
* fechaProgramada
* (estado, fechaProgramada) (compuesto)
* personaId
* programaId

**Reglas de Negocio:**

* Si es entrega individual → usar personaId
* Si es masiva → personaId = NULL y usar EntregaBeneficiario
* Si estado = ENTREGADA → fechaEntrega, confirmacionRecibido y entregadoPor obligatorios
* Si estado = CANCELADA → motivoCancelacion obligatorio
* Si tipoRecurso = MONETARIA o MIXTA → monto obligatorio y > 0
* fechaEntrega >= fechaProgramada

**Relaciones:**

* N:1 con **Programa**
* N:1 con **Apadrinamiento**
* N:1 con **Persona** (beneficiario individual)
* 1:N con **EntregaBeneficiario** (si es masiva)
* N:1 con **Usuario** (registradoPor)
* N:1 con **Usuario** (entregadoPor)

**12. ENTREGA\_BENEFICIARIO**

**Descripción:** Tabla intermedia para entregas masivas.

**Campos:**

* id - UUID, PK
* entregaId - UUID, FK → Entrega, NOT NULL
* personaId - UUID, FK → Persona, NOT NULL
* confirmado - Boolean, NOT NULL, DEFAULT: false
* fechaConfirmacion - DateTime, NULL
* firmaUrl - String(500), NULL
* observaciones - Text, NULL
* createdAt - DateTime, NOT NULL, AUTO

**Índices:**

* entregaId
* personaId
* (entregaId, personaId) (UNIQUE)

**Reglas de Negocio:**

* UNIQUE(entregaId, personaId)
* Un beneficiario no puede estar dos veces en la misma entrega
* Si confirmado = true → fechaConfirmacion obligatoria
* Solo se usa cuando Entrega.personaId = NULL

**Relaciones:**

* N:1 con **Entrega**
* N:1 con **Persona**

**13. GASTO**

**Descripción:** Registro de gastos de la fundación.

**Campos:**

* id - UUID, PK
* categoria - ENUM: CategoriaGasto, NOT NULL
* subcategoria - String(100), NOT NULL
* concepto - String(200), NOT NULL
* personaId - UUID, FK → Persona, NULL
* programaId - UUID, FK → Programa, NULL
* monto - Decimal(12,2), NOT NULL
* fechaGasto - Date, NOT NULL, DEFAULT: today()
* metodoPago - ENUM: MetodoPago, NOT NULL
* numeroReferencia - String(100), NULL
* descripcion - Text, NULL
* comprobanteUrl - String(500), NULL
* observaciones - Text, NULL
* registradoPor - UUID, FK → Usuario, NOT NULL
* createdAt - DateTime, NOT NULL, AUTO
* updatedAt - DateTime, NOT NULL, AUTO

**Índices:**

* categoria
* fechaGasto (DESC)
* (categoria, fechaGasto) (compuesto)
* personaId
* programaId

**Reglas de Negocio:**

* monto > 0
* fechaGasto no puede ser futura
* Si categoria = PAGO\_PERSONAL → personaId obligatorio
* Si categoria = GASTO\_PROGRAMA → programaId obligatorio

**Relaciones:**

* N:1 con **Persona** (si es pago a personal)
* N:1 con **Programa** (si es gasto de programa)
* N:1 con **Usuario** (registradoPor)

**14. AUDITORIA**

**Descripción:** Registro de trazabilidad de acciones críticas.

**Campos:**

* id - UUID, PK
* usuarioId - UUID, FK → Usuario, NOT NULL
* accion - ENUM: AccionAuditoria, NOT NULL
* entidad - String(50), NOT NULL (nombre de la tabla)
* entidadId - UUID, NULL (ID del registro afectado)
* datosAnteriores - JSON, NULL (estado antes del cambio)
* datosNuevos - JSON, NULL (estado después del cambio)
* createdAt - DateTime, NOT NULL, AUTO

**Índices:**

* usuarioId
* createdAt (DESC)
* (usuarioId, createdAt) (compuesto)
* (entidad, entidadId) (compuesto)

**Reglas de Negocio:**

* Registra automáticamente: CREATE, UPDATE, DELETE
* Registra LOGIN y LOGOUT
* Registra EXPORT de reportes
* Los registros nunca se eliminan (solo lectura histórica)

**Relaciones:**

* N:1 con **Usuario**

**🎨 ENUMS COMPLETOS**

// Documentos e Identidad

TipoDocumento = 'CC' | 'TI' | 'CE' | 'PASAPORTE' | 'RC'

Genero = 'MASCULINO' | 'FEMENINO' | 'OTRO'

// Personas

TipoPersona = 'MENOR' | 'TUTOR' | 'BENEFICIARIO\_ADULTO' | 'DOCENTE' | 'ADMINISTRATIVO' | 'PERSONAL\_APOYO'

Parentesco = 'PADRE' | 'MADRE' | 'ABUELO' | 'ABUELA' | 'TIO' | 'TIA' | 'HERMANO' | 'HERMANA' | 'OTRO'

// Estados

EstadoGeneral = 'ACTIVO' | 'INACTIVO'

EstadoUsuario = 'ACTIVO' | 'INACTIVO' | 'BLOQUEADO'

EstadoPrograma = 'ACTIVO' | 'INACTIVO' | 'EN\_PAUSA'

EstadoInscripcion = 'ACTIVO' | 'RETIRADO' | 'SUSPENDIDO'

EstadoAsistencia = 'PRESENTE' | 'AUSENTE' | 'JUSTIFICADO' | 'TARDANZA'

EstadoApadrinamiento = 'ACTIVO' | 'FINALIZADO'

EstadoEntrega = 'PROGRAMADA' | 'ENTREGADA' | 'CANCELADA'

// Usuarios

RolUsuario = 'ADMIN' | 'DOCENTE'

// Programas

TipoPrograma = 'ACADEMICO' | 'LUDICO' | 'APOYO\_SOCIAL' | 'ESPIRITUAL'

PublicoObjetivo = 'NINOS' | 'JOVENES' | 'ADULTOS' | 'TODOS'

DiaSemana = 'LUNES' | 'MARTES' | 'MIERCOLES' | 'JUEVES' | 'VIERNES' | 'SABADO' | 'DOMINGO'

// Padrinos

TipoPadrino = 'PERSONA\_NATURAL' | 'EMPRESA' | 'ORGANIZACION'

TipoDonacionPadrino = 'MONETARIA' | 'EN\_ESPECIE' | 'AMBAS'

FrecuenciaDonacion = 'MENSUAL' | 'TRIMESTRAL' | 'SEMESTRAL' | 'ANUAL' | 'OCASIONAL'

Privacidad = 'PUBLICA' | 'PRIVADA' | 'ANONIMA'

// Donaciones y Recursos

TipoDonacion = 'PADRINO' | 'GENERAL' | 'ANONIMA' | 'PARA\_PROGRAMA'

TipoRecurso = 'MONETARIA' | 'EN\_ESPECIE' | 'MIXTA'

MetodoPago = 'TRANSFERENCIA' | 'GIRO\_INTERNACIONAL' | 'EFECTIVO' | 'CHEQUE' | 'OTRO'

TipoEntrega = 'MERCADO' | 'BECA\_BESPRO' | 'UTILES\_ESCOLARES' | 'ROPA\_ZAPATOS' | 'AYUDA\_PUNTUAL' | 'OTRO'

CategoriaGasto = 'PAGO\_PERSONAL' | 'GASTO\_OPERATIVO' | 'GASTO\_PROGRAMA' | 'EVENTO' | 'OTRO'

// Auditoría

AccionAuditoria = 'CREATE' | 'UPDATE' | 'DELETE' | 'LOGIN' | 'LOGOUT' | 'EXPORT'