DIRECCIÓN NACIONAL DE DERECHO DE AUTOR

Subdirección de Tecnologías de la Información

DOCUMENTO TÉCNICO ARQUITECTÓNICO

Módulo de Expediente

Gestor de Correspondencia

Versión	1.0
Fecha	26 de mayo de 2025
Estado	Borrador para revisión
Clasificación	Interno – Uso Restringido
Elaborado por	Ing. Carlos Alberto Hernández Rincón

1. Introducción y Contexto

El presente documento establece las recomendaciones arquitectónicas para el Módulo de Expediente dentro del Gestor de Correspondencia de la Dirección Nacional de Derecho de Autor (DNDA). El análisis abarca tres dimensiones: (i) el modelo de datos propuesto y sus mejoras, (ii) la decisión de incorporar el módulo al microservicio existente CorrespondenceManager o crearlo como un servicio independiente, y (iii) los lineamientos de arquitectura frontend.

Un expediente (dossier) es la agrupación lógica y cronológica de una o varias correspondencias relacionadas con un asunto, trámite o proceso institucional. Dado que su ciclo de vida, sus reglas de negocio y sus actores son distintos a los de la correspondencia individual, requiere tratamiento arquitectónico propio.

2. Análisis del Modelo de Datos Propuesto

A continuación se revisa cada tabla propuesta, identificando observaciones, campos faltantes y correcciones de nomenclatura. El modelo mejorado se presenta en la sección 2.4.

2.1 Tabla Expedient

Campos propuestos y observaciones

Campo	Tipo sugerido	Observación
ExpedientId	UNIQUEIDENTIFIER (PK)	Correcto. Usar GUID para portabilidad entre microservicios.
CorrespondencePrincipalId	UNIQUEIDENTIFIER (FK)	Correcto. Referencia a la correspondencia que origina el expediente.
Name	NVARCHAR(200)	Correcto. Nombre descriptivo del expediente.
Code	VARCHAR(50) UNIQUE	Correcto. Código único, preferiblemente con formato institucional (p.ej. EXP-2025-00001).
CreatedDate	DATETIME2	Correcto. Usar DATETIME2 por precisión. Valor default: GETUTCDATE().
UserCreated	NVARCHAR(100)	Observación: se recomienda usar un UserId (FK a tabla de usuarios) en lugar del nombre en texto plano.
Descriptions	NVARCHAR(MAX)	⚠ CORRECCIÓN: debe ser Description (singular). Permite detallar el propósito del expediente.
ExpirationDate	DATE (nullable)	Correcto. Fecha de vencimiento o archivo. Puede ser nulo si no aplica.
State	TINYINT / NVARCHAR(30)	Recomendación: usar un catálogo ExpedientState (Activo, Cerrado, Archivado, Anulado).

📌 **Campos faltantes detectados:** UpdatedDate (DATETIME2), UserModified (FK a usuarios), ExpedientTypeId (FK catálogo de tipos), AreaId / DependencyId (unidad organizacional responsable), Priority (Baja/Media/Alta/Urgente), ClosedDate (DATE nullable), IsDeleted (BIT, soft delete).

2.2 Tabla ExpedientCorrespondence

Campos propuestos y observaciones

Campo	Tipo sugerido	Observación
ExpedientCorrespondenceId	UNIQUEIDENTIFIER (PK)	Correcto.
--- ExpedientId ---	⚠ FALTANTE	⛔ CRÍTICO: falta la FK a Expedient. Sin este campo la tabla no puede relacionarse con su expediente padre.
CorrespondenceId	UNIQUEIDENTIFIER (FK)	Correcto. FK a la tabla Correspondence del CorrespondenceManager.
CreatedDate	DATETIME2	Correcto. Default: GETUTCDATE().
UserCreated	NVARCHAR(100) / FK	Ver observación UserCreated en sección 2.1.

📌 **Campos faltantes detectados:** ExpedientId (FK – CRÍTICO), OrderNumber (INT, orden de la correspondencia dentro del expediente), RelationshipType (catálogo: Origen, Respuesta, Adjunta, Relacionada), Notes (NVARCHAR(500) nullable).

2.3 Tabla ExpedientAttachement

Campos propuestos y observaciones

Campo	Tipo sugerido	Observación
ExpedientAttachementId	UNIQUEIDENTIFIER (PK)	Correcto. Nota: "Attachement" tiene un typo; la forma correcta es Attachment.
ExpedientId	UNIQUEIDENTIFIER (FK)	Correcto.
CorrespondenceId	UNIQUEIDENTIFIER (FK nullable)	Opcional. Permite saber a qué correspondencia pertenece el adjunto dentro del expediente.
AttachmentId	UNIQUEIDENTIFIER (FK)	Correcto. FK al repositorio central de archivos.

AttachmentTypeId	INT (FK catálogo)	Correcto. Ejemplo: Resolución, Oficio, Soporte, Otro.
FileName	NVARCHAR(260)	Correcto.
MimeType	VARCHAR(100)	Correcto.
SizeBytes	BIGINT	Correcto. Usar BIGINT para soportar archivos >2 GB.
StoragePath	NVARCHAR(500)	Correcto. Ruta en almacenamiento (S3, Azure Blob, NAS).
UploadedAt	DATETIME2	Correcto.
Descriptions	NVARCHAR(500)	⚠ CORRECCIÓN: debe ser Description (singular).
CreatedDate	DATETIME2	Redundante con UploadedAt. Mantener solo uno o diferenciar su semántica claramente.
UserCreated	NVARCHAR(100) / FK	Ver observación UserCreated en sección 2.1.

> 📌 **Campos faltantes detectados:** IsDeleted (BIT, soft delete), DeletedDate (DATETIME2 nullable), UserDeleted, ChecksumHash (VARCHAR(64), para integridad SHA-256), IsConfidential (BIT).

2.4 Modelo de Datos Mejorado (Recomendado)

A continuación se presenta el esquema de tablas corregido y enriquecido con los campos identificados en el análisis previo. Los campos en negrita son añadidos respecto al diseño original.

Tabla: Expedient (mejorada)

Campo	Tipo	Restricción
ExpedientId	UNIQUEIDENTIFIER	PK, DEFAULT NEWSEQUENTIALID()
Code	VARCHAR(50)	UNIQUE NOT NULL
Name	NVARCHAR(200)	NOT NULL
Description	NVARCHAR(MAX)	NULL
CorrespondencePrincipalId	UNIQUEIDENTIFIER	FK → Correspondence, NOT NULL
ExpedientTypeId	INT	FK → ExpedientType (catálogo), NOT NULL
AreaId	INT	FK → Area/Dependency, NOT NULL
Priority	TINYINT	NOT NULL, DEFAULT 2 (1=Baja…4=Urgente)
State	TINYINT	FK → ExpedientState (catálogo), NOT NULL
ExpirationDate	DATE	NULL
ClosedDate	DATE	NULL

IsDeleted	BIT	NOT NULL, DEFAULT 0
CreatedDate	DATETIME2	NOT NULL, DEFAULT GETUTCDATE()
UserCreatedId	UNIQUEIDENTIFIER	FK → Users, NOT NULL
UpdatedDate	DATETIME2	NULL
UserUpdatedId	UNIQUEIDENTIFIER	FK → Users, NULL

Tabla: ExpedientCorrespondence (mejorada)

Campo	Tipo	Restricción
ExpedientCorrespondenceId	UNIQUEIDENTIFIER	PK, DEFAULT NEWSEQUENTIALID()
ExpedientId	UNIQUEIDENTIFIER	FK → Expedient, NOT NULL
CorrespondenceId	UNIQUEIDENTIFIER	FK → Correspondence, NOT NULL
RelationshipType	TINYINT	FK → RelationshipType (catálogo), NOT NULL
OrderNumber	INT	NOT NULL, DEFAULT 1 (orden dentro del expediente)
Notes	NVARCHAR(500)	NULL
CreatedDate	DATETIME2	NOT NULL, DEFAULT GETUTCDATE()
UserCreatedId	UNIQUEIDENTIFIER	FK → Users, NOT NULL

Tabla: ExpedientAttachment (mejorada – typo corregido)

Campo	Tipo	Restricción
ExpedientAttachmentId	UNIQUEIDENTIFIER	PK, DEFAULT NEWSEQUENTIALID()
ExpedientId	UNIQUEIDENTIFIER	FK → Expedient, NOT NULL
CorrespondenceId	UNIQUEIDENTIFIER	FK → Correspondence, NULL (opcional)
AttachmentId	UNIQUEIDENTIFIER	FK → repositorio central de archivos, NOT NULL
AttachmentTypeId	INT	FK → AttachmentType (catálogo), NOT NULL
FileName	NVARCHAR(260)	NOT NULL
MimeType	VARCHAR(100)	NOT NULL
SizeBytes	BIGINT	NOT NULL
StoragePath	NVARCHAR(500)	NOT NULL
ChecksumHash	VARCHAR(64)	NULL (SHA-256 integridad)
IsConfidential	BIT	NOT NULL, DEFAULT 0
Description	NVARCHAR(500)	NULL

UploadedAt	DATETIME2	NOT NULL, DEFAULT GETUTCDATE()
UserCreatedId	UNIQUEIDENTIFIER	FK → Users, NOT NULL
IsDeleted	BIT	NOT NULL, DEFAULT 0
DeletedDate	DATETIME2	NULL
UserDeletedId	UNIQUEIDENTIFIER	FK → Users, NULL

3. Decisión Arquitectónica: Microservicio Independiente vs. CorrespondenceManager

Esta sección presenta un análisis comparativo entre las dos opciones principales de despliegue del módulo de Expediente, seguido de la recomendación justificada.

3.1 Contexto de la Decisión

El Gestor de Correspondencia de la DNDA ya cuenta con el microservicio CorrespondenceManager, responsable de la gestión del ciclo de vida de las comunicaciones institucionales. El módulo de Expediente introduce un nuevo contexto delimitado (bounded context) que agrupa, clasifica y da seguimiento a conjuntos de correspondencias bajo un mismo asunto o trámite.

Los factores clave para la decisión son: acoplamiento de dominio, escalabilidad independiente, complejidad operacional, gobernanza de datos y madurez del equipo para operar múltiples servicios.

3.2 Opción A – Incluir en CorrespondenceManager

Ventajas y desventajas de la Opción A

Criterio	Evaluación	Detalle
Simplicidad operacional	✅ Ventaja	Un solo servicio para desplegar, monitorear y escalar.
Transacciones locales	✅ Ventaja	Operaciones sobre Correspondence y Expediente en la misma BD, sin sagas distribuidas.
Tiempo de implementación	✅ Ventaja	Menor tiempo de entrega al reutilizar infraestructura existente.
Acoplamiento de dominio	⚠ Riesgo	Mezcla dos bounded contexts. Aumenta la complejidad del servicio con el tiempo.
Escalabilidad diferenciada	❌ Limitación	No es posible escalar Expediente de forma independiente a Correspondence.
Responsabilidad única	❌ Limitación	Viola SRP: un solo servicio conoce demasiados dominios.

3.3 Opción B – Microservicio ExpedienteManager Independiente (Recomendada)

Ventajas y desventajas de la Opción B

Criterio	Evaluación	Detalle
Separación de bounded contexts	✅ Ventaja	Cada microservicio tiene un dominio claro y responsabilidad única.
Escalabilidad independiente	✅ Ventaja	Expediente puede escalar según demanda de consultas masivas o reportes.
Despliegue autónomo	✅ Ventaja	El equipo puede actualizar ExpedienteManager sin afectar Correspondence.

Evolución tecnológica	✅ Ventaja	Libertad de elegir motor de búsqueda o BD documental para el futuro.
Complejidad operacional	⚠ Riesgo	Requiere gestión de contratos de API, versionamiento y mayor infraestructura CI/CD.
Comunicación entre servicios	⚠ Riesgo	Llamadas HTTP/gRPC entre servicios; se mitiga con eventos de dominio (mensajería).
Consistencia eventual	⚠ Riesgo	Las operaciones que afectan ambos dominios requieren patrones como Saga u Outbox.

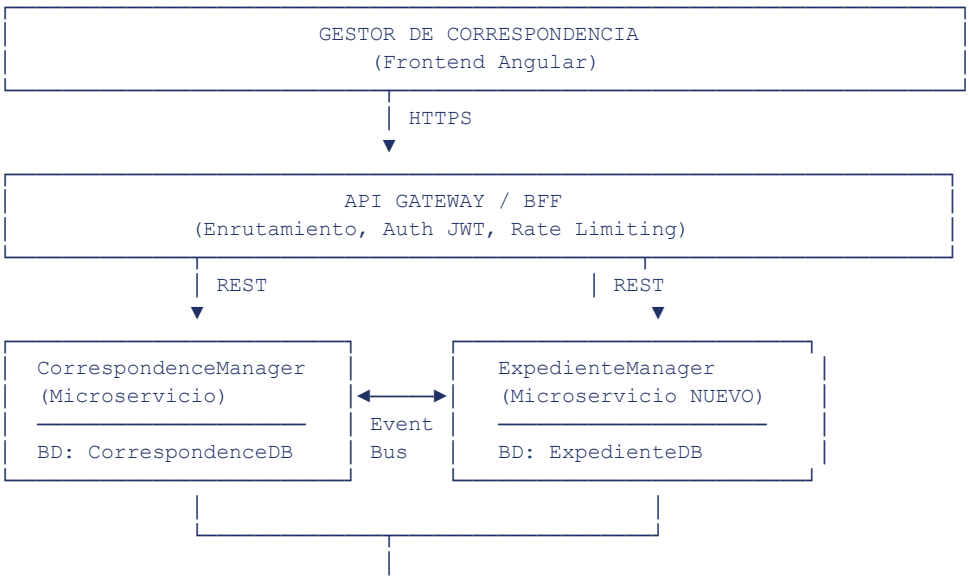
3.4 Recomendación y Justificación (ADR)

> 📌 **Decisión Arquitectónica (ADR-001):** Se recomienda crear el microservicio ExpedienteManager como un servicio independiente, con su propia base de datos, expuesto a través del API Gateway institucional. La comunicación con CorrespondenceManager se realizará mediante eventos de dominio (patrón Outbox + Message Broker) y consultas REST síncronas solo donde sea estrictamente necesario.

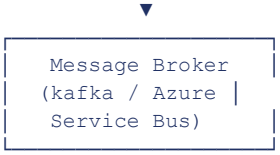
La justificación principal es que el Expediente tiene su propio ciclo de vida (creación, asignación, vencimiento, cierre, archivo), sus propios actores (funcionarios que gestionan trámites vs. operadores de correspondencia) y sus propias reglas de negocio (numeración automática, plazos legales, control de acceso por dependencia). Unir ambos dominios en un solo servicio genera deuda técnica que se vuelve costosa a medida que el sistema crece.

3.5 Diagrama de Componentes (Vista de Alto Nivel)

El siguiente diagrama describe la interacción entre los componentes principales:




DNDA
Dirección Nacional
de Derecho de Autor
Ministerio del Interior


Message Broker
(kafka / Azure
Service Bus)

4. Propuesta de API REST – ExpedienteManager

El microservicio ExpedienteManager expone una API RESTful versionada bajo la base /api/v1/expedientes. Los contratos siguen el estándar JSON:API. La autenticación es mediante JWT Bearer Token validado por el API Gateway.

4.1 Gestión de Expedientes

Método	Ruta	Descripción	Roles
GET	/api/v1/expedientes	Listar expedientes (paginado, filtros)	Consultor, Gestor, Admin
GET	/api/v1/expedientes/{id}	Obtener detalle de un expediente	Consultor, Gestor, Admin
POST	/api/v1/expedientes	Crear nuevo expediente	Gestor, Admin
PUT	/api/v1/expedientes/{id}	Actualizar expediente completo	Gestor, Admin
PATCH	/api/v1/expedientes/{id}/state	Cambiar estado del expediente	Gestor, Admin
DELETE	/api/v1/expedientes/{id}	Eliminación lógica (soft delete)	Admin

4.2 Correspondencias del Expediente

Método	Ruta	Descripción	Roles
GET	/api/v1/expedientes/{id}/correspondencias	Listar correspondencias vinculadas	Consultor+
POST	/api/v1/expedientes/{id}/correspondencias	Vincular correspondencia existente	Gestor, Admin
DELETE	/api/v1/expedientes/{id}/correspondencias/{cId}	Desvincular correspondencia	Admin

4.3 Adjuntos del Expediente

Método	Ruta	Descripción	Roles
GET	/api/v1/expedientes/{id}/adjuntos	Listar adjuntos del expediente	Consultor+
POST	/api/v1/expedientes/{id}/adjuntos	Subir y vincular adjunto (multipart)	Gestor, Admin
GET	/api/v1/expedientes/{id}/adjuntos/{aId}	Descargar adjunto	Consultor+

DELETE	/api/v1/expedientes/{id}/adjuntos/{aId}	Eliminar adjunto (soft delete)	Admin

4.4 Búsqueda y Reportes

Método	Ruta	Descripción
GET	/api/v1/expedientes/search	Búsqueda full-text por código, nombre o descripción
GET	/api/v1/expedientes/reportes/vencimiento	Reporte de expedientes próximos a vencer
GET	/api/v1/expedientes/reportes/estado	Dashboard de expedientes por estado y área

5. Propuesta de Arquitectura Frontend

El módulo de Expediente se integra como un feature module lazy-loaded dentro del proyecto Angular existente del Gestor de Correspondencia, siguiendo la misma design language y componentes del sistema actual.

5.1 Estructura de Módulo Angular

```
src/
└── app/
    └── features/
        └── expediente/                ← Feature Module (lazy-loaded)
            ├── expediente.module.ts
            ├── expediente.routes.ts
            ├── models/
            │   ├── expediente.model.ts
            │   ├── expediente-correspondence.model.ts
            │   └── expediente-attachment.model.ts
            ├── services/
            │   └── expediente.service.ts   ← HTTP client → ExpedienteManager API
            ├── store/                      ← NgRx (si se usa state management)
            │   ├── expediente.actions.ts
            │   ├── expediente.reducer.ts
            │   └── expediente.effects.ts
            └── pages/
                ├── expediente-list/        ← Vista de listado
                ├── expediente-detail/      ← Vista de detalle / línea de tiempo
                ├── expediente-form/        ← Crear / Editar
                └── expediente-search/      ← Búsqueda avanzada
```

5.2 Vistas y Componentes Principales

Inventario de componentes frontend

Vista / Componente	Descripción	Funcionalidad clave

ExpedienteListComponent	Listado principal de expedientes	Tabla paginada con filtros por estado, área, tipo y rango de fechas. Acciones rápidas: ver, editar, cambiar estado.
ExpedienteDetailComponent	Vista de detalle del expediente	Header con datos principales, línea de tiempo de correspondencias vinculadas, sección de adjuntos con previsualización, historial de cambios.
ExpedienteFormComponent	Crear / editar expediente	Formulario reactivo (Reactive Forms) con validaciones. Selector de correspondencia principal con búsqueda type-ahead.
LinkCorrespondenceModalComponent	Modal para vincular correspondencias	Búsqueda en tiempo real dentro de CorrespondenceManager. Permite seleccionar tipo de relación y orden.
ExpedienteAttachmentsComponent	Gestión de adjuntos	Drag & drop para subir archivos, preview en modal, descarga directa, eliminación con confirmación.
ExpedienteSearchComponent	Búsqueda avanzada	Filtros avanzados: full-text, estado, área, tipo, rango de vencimiento. Exportación a Excel / PDF.
ExpedienteStatsBannerComponent	Dashboard / KPIs (opcional)	Tarjetas con métricas: total activos, por vencer, cerrados, por área.

5.3 Flujo de Usuario Principal

Crear Expediente: Usuario accede a "Expedientes" → hace clic en "Nuevo Expediente" → completa el formulario (código, nombre, tipo, área, correspondencia principal) → vincula correspondencias adicionales → guarda → el sistema asigna código automático y muestra confirmación.

Consultar y Filtrar: Usuario accede al listado → aplica filtros por estado/área → selecciona un expediente → accede al detalle con línea de tiempo de correspondencias y adjuntos.

Vincular Correspondencia: Desde el detalle del expediente → "Agregar Correspondencia" → modal de búsqueda → selección con tipo de relación → confirmación → actualización en tiempo real del listado.

Gestionar Adjuntos: Desde el detalle → sección "Adjuntos" → drag & drop de archivo → selección de tipo de adjunto → subida al StoragePath → registro en ExpedientAttachment.

Cerrar / Archivar: Desde el detalle → "Cambiar Estado" → selección de estado destino con observación → confirmación → el sistema registra ClosedDate y notifica a interesados.

6. Consideraciones de Seguridad y Control de Acceso

Matriz de roles y permisos

Acción	Consultor	Gestor	Jefe de Área	Admin TI
Ver listado de expedientes de su área	✅	✅	✅	✅
Ver listado de todos los expedientes	❌	❌	✅	✅
Crear expediente	❌	✅	✅	✅
Editar expediente	❌	✅	✅	✅
Vincular / desvincular correspondencia	❌	✅	✅	✅
Subir adjuntos	❌	✅	✅	✅
Eliminar adjuntos	❌	❌	✅	✅
Cambiar estado del expediente	❌	✅	✅	✅
Cerrar / archivar expediente	❌	❌	✅	✅
Eliminar expediente (soft delete)	❌	❌	❌	✅
Ver reportes y métricas	✅	✅	✅	✅
Exportar reportes	❌	✅	✅	✅

Adicionalmente se recomiendan las siguientes medidas de seguridad:

- Cifrado en reposo y en tránsito (TLS 1.3) para todos los adjuntos almacenados.
- Registro de auditoría (Audit Log) inmutable para cada operación sobre un expediente.
- Validación de tipo MIME en el servidor para la carga de adjuntos (no confiar en la extensión).
- Límite de tamaño máximo de adjunto configurable por variable de entorno.
- Expiración automática de tokens de descarga de adjuntos (URL firmada con TTL ≤ 15 min).
- Separación de la BD de Expediente de la BD de Correspondencia (aislamiento de datos).

7. Conclusiones y Próximos Pasos

El módulo de Expediente representa un bounded context diferenciado dentro del Gestor de Correspondencia de la DNDA. La recomendación arquitectónica es crear el microservicio ExpedienteManager de forma independiente, con su propia base de datos, expuesto a través del API Gateway y comunicado con CorrespondenceManager mediante eventos de dominio.

El modelo de datos propuesto es sólido en su estructura base, y las mejoras identificadas — especialmente la corrección del campo faltante ExpedientId en ExpedientCorrespondence y la normalización de nomenclatura— son críticas antes de iniciar el desarrollo.

Fase	Actividad

Sprint 10	Validar el modelo de datos mejorado con el equipo de negocio y DBA.
Sprint 10	Definir el catálogo de tablas auxiliares: ExpedientType, ExpedientState, RelationshipType.
Sprint 10	Configurar el proyecto ExpedienteManager (estructura, CI/CD, Docker, BD).
Sprint 11	Implementar CRUD básico de Expediente con sus endpoints REST.
Sprint 11	Implementar vinculación de correspondencias y adjuntos.
Sprint 11	Desarrollar el Feature Module Angular: list, detail, form.
Sprint 11	Integración con Message Broker para eventos de dominio.
Sprint 11	Implementar reportes y búsqueda avanzada.
Sprint 11	Pruebas de integración, seguridad y performance. Go-live en ambiente QA.