

# ?? Servicios (app/Services)

1. **ExpedienteService**: Gestiona el flujo del movimiento de los **Expedientes** entre las distintas **Dependencias**. Resuelve lógicas para pases en paralelo y comprueba que dependencias tienen permisos para adjuntar archivos a un expediente específico.

## Funciones Activas (Core del Servicio)

- **obtenerDependenciasEnPaseParaleloActivo(int \$expedienteId)** **Funcionalidad:** Es una consulta de bajo nivel (`DB::table`) a la tabla pivote `dependencia_expediente` que relaciona un expediente con varias oficinas simultáneamente. **Qué hace:** Retorna en forma de un simple *array* los IDs numéricos de todas las dependencias con las cuales un expediente comparte el estado de "pase paralelo" en estado "activo".
- **sePuedeTransferir(int \$expedienteId, int \$dependenciaActualId)** **Funcionalidad:** Evalúa lógicamente si el usuario puede "Transferir/Enviar" un expediente. **Regla de negocio actual:** Busca en la BD si el expediente pertenece a esa dependencia actual (`holder_id === $dependenciaActualId`) y además comprueba que NO tenga pases paralelos distribuidos (`!$relacionesActivas`). Por lo tanto, un expediente en medio de un "pase en paralelo" activo hacia varias oficinas **no se puede** transferir de raíz al próximo paso hasta que se cierre el paralelo.
- **dependenciaPuedeAgregarArchivos(int \$expedienteId, int \$dependenciaId)** **Funcionalidad:** Decide si una dependencia en concreto tiene permitido el alta de nuevos documentos. **Regla de negocio actual:** Si la dependencia está involucrada dentro del listado de "Pases en paralelo activo", entonces le permite añadir archivos (`return true;`). Si no está en paralelo, evalúa si es el simple poseedor primario del expediente (`holder_id`). *Dato interesante del código:* Antes había un bloque de validación más rígido que impedía subir archivos, pero fue comentado asumiendo el nuevo permiso de que "AHORA PERMITIMOS QUE SE AGREGUE DOCUMENTACION AUNQUE ESTE EN PARALELO".
- **yoTengoExpediente(int \$expedienteId, int \$dependenciaId)** **Funcionalidad:** Simple comprobación binaria para uso rápido en vistas (if/else). **Qué hace:** Solo evalúa si la dependencia X pedida coincide con el poseedor primario (`holder_id`) cargado en el Objeto Expediente.
- **yoTengoExpedienteEnParalelo(int \$expedienteId, int \$dependenciaId)** **Funcionalidad:** Complementa la función de arriba. **Qué hace:** Llama al método `obtenerDependenciasEnPaseParaleloActivo()` y revisa con un simple `in_array` si la dependencia X está contenida en ese listado de oficinas

involucradas en un pase conjunto.

2. **FirmaDigitalService**: Responsable de integrar funciones a nivel de sistema operativo para validar firmas digitales en los PDF (usando comandos como `pdfsig` y bases `NSS`) y aplica una "firma por sistema" a los documentos.
  3. **ManifiestoExpedienteService**: Maneja la integridad de la documentación aplicando un modelo estilo "blockchain". Sella expedientes (`sellar()`) adjuntando manifiestos con *hash canonicos* estado por estado para garantizar que los documentos no hayan sido adulterados leyendo el historial (`verificarCadena()`).
  4. **OpenAIService**: Centraliza la conexión y peticiones a la API de ChatGPT para sugerir o generar de forma automática respuestas (`generateReply()`) a correos y comunicaciones simulando asistentes virtuales.
- 

Revision #3

Created 2026-03-03 13:41:56 UTC by Admin

Updated 2026-03-03 13:51:37 UTC by Admin