HU-JZ-001: Control de Acceso Basado en Propiedad para Jueces

📋 Información General

Campo	Valor
ID	HU-JZ-001
Nombre	Control de Acceso a Expedientes por Propiedad
Módulo	Seguridad y Control de Acceso
Sprint	Sprint 2
Estado	☑ Implementado
Prioridad	CRÍTICA
Fecha Implementación	2026-01-05

🎯 Objetivo

Garantizar que **un juez solo pueda acceder a los expedientes, documentos y audiencias de las causas que tiene asignadas**, implementando un control de acceso basado en atributos (FIA_ATD.1) para prevenir accesos no autorizados (ataques IDOR).

📖 Historia de Usuario

Como Juez del Sistema Judicial
Quiero que el sistema me permita acceder únicamente a las causas que tengo asignadas
Para garantizar la confidencialidad de la información y cumplir con el principio de compartimentación de datos sensibles

🔐 Requisitos de Seguridad Common Criteria

FIA_ATD.1 (User Attribute Definition)

- ☑ **Control de acceso basado en atributos**: Validación del atributo `juez_asignado_id`
- ☑ **Verificación en tiempo real**: Consulta a la base de datos en cada solicitud
- ☑ **No dependencia exclusiva del token**: JWT como identificador, BD como fuente de verdad

FDP_ACC.1 (Subset Access Control)

- ☑ **Acceso restringido por propiedad**: Solo recursos propios
- ☑ **Separación de privilegios**: ADMIN_CJ y SECRETARIO tienen acceso completo
- ☑ **Aplicación consistente**: Middleware en todas las rutas protegidas

FAU_GEN.1 (Audit Data Generation)

- ☑ **Registro de accesos denegados**: Severidad ALTA para posibles ataques

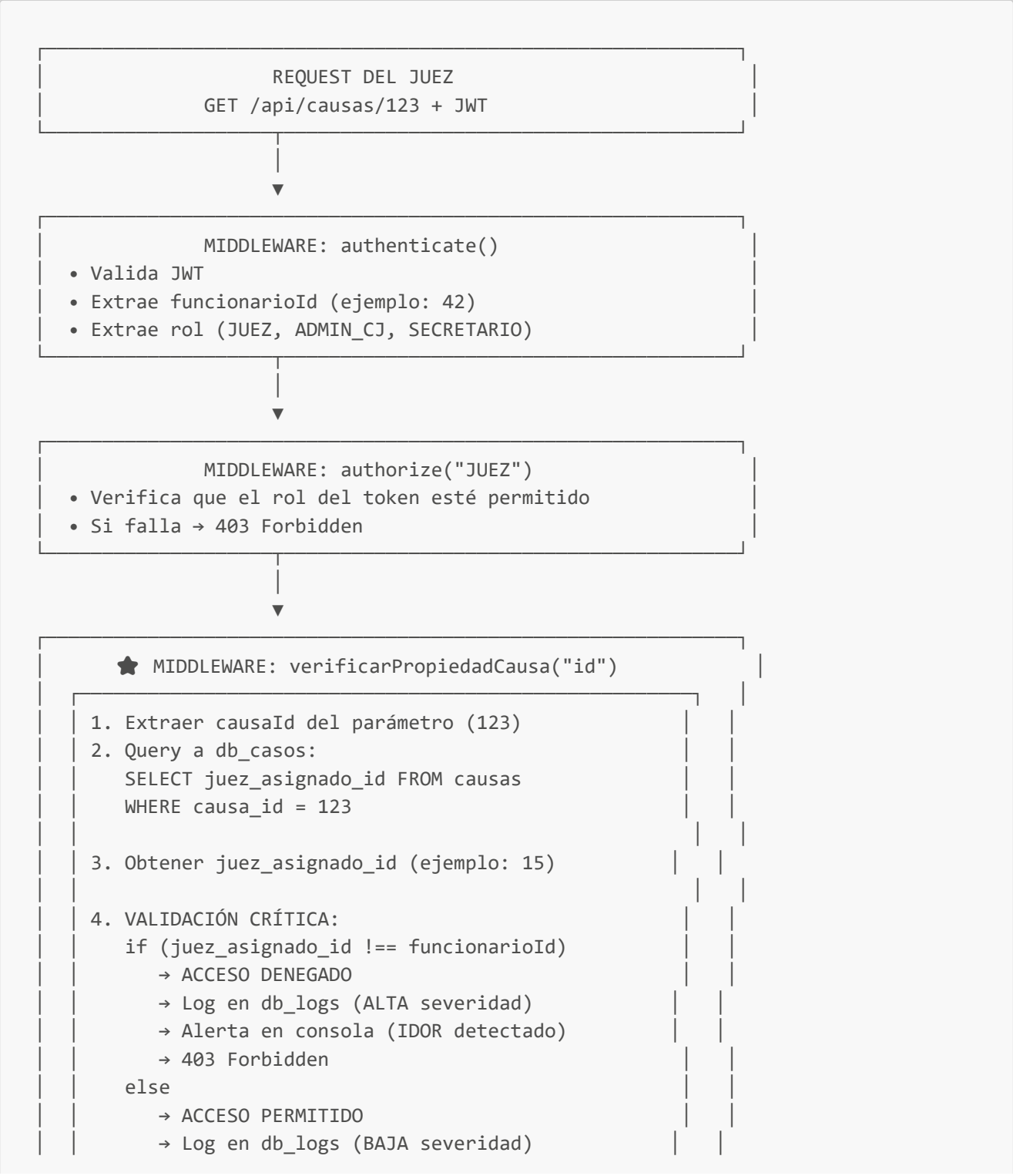
- ☑ **Registro de accesos permitidos**: Severidad BAJA para trazabilidad
- ☑ **Datos completos**: IP, User-Agent, causa, jueces involucrados

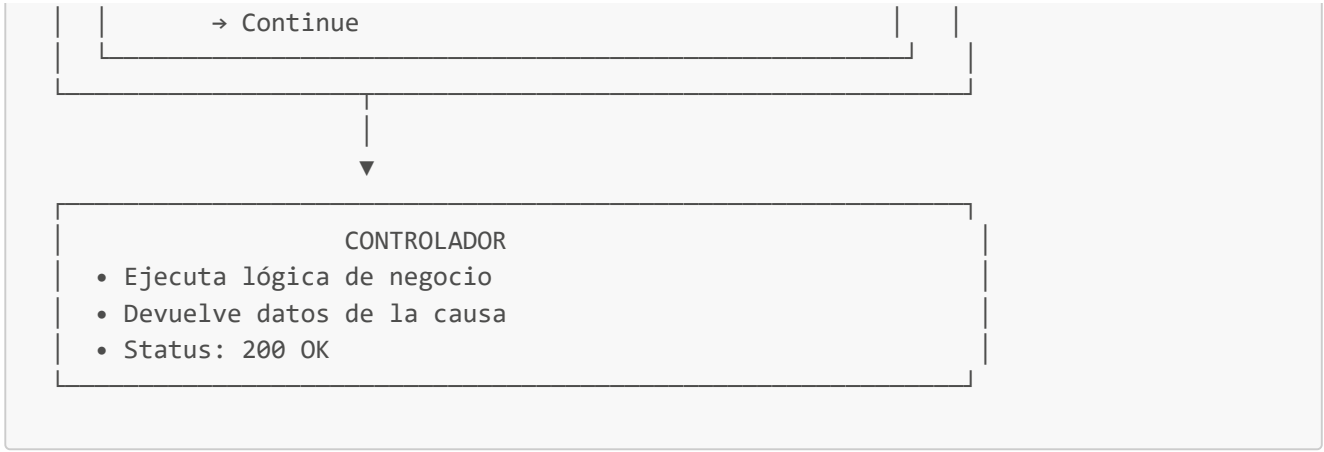
FAU_SAR.1 (Audit Review)

- ☑ **Alertas en tiempo real**: Logs en consola para monitoreo
- ☑ **Datos estructurados**: JSON para análisis automatizado
- ☑ **Integridad del log**: Hash SHA-256 en cada evento

🏗 Arquitectura de la Solución

Componentes Implementados





💻 Implementación Técnica

1. Middleware de Control de Acceso

Archivo: `backend/src/middleware/access-control.middleware.ts`

Función Principal: `verificarPropiedadCausa(paramName)`

```
export function verificarPropiedadCausa(paramName: string = "id") {
  return async (req: Request, res: Response, next: NextFunction): Promise<void> =>
{
    // 1. BYPASS para roles privilegiados
    if (req.user.rol !== "JUEZ") {
      next();
      return;
    }

    // 2. Extraer ID de la causa
    const causaId = parseInt(req.params[paramName]);

    // 3. CONSULTA A BASE DE DATOS (Fuente de verdad)
    const result = await client.query(
      `SELECT
        causa_id, juez_asignado_id, numero_proceso,
        estado_procesal, materia, unidad_judicial
       FROM causas WHERE causa_id = $1`,
      [causaId]
    );

    const causa = result.rows[0];
    const juezAsignadoDB = causa.juez_asignado_id;
    const juezTokenID = req.user.funcionarioId;

    // 4. VALIDACIÓN CRÍTICA DE PROPIEDAD
    if (juezAsignadoDB !== juezTokenID) {
      // ACCESO DENEGADO
      await auditService.log({
        tipoEvento: "ACCESO_DENEGADO",
        usuarioId: juezTokenID,
        moduloAfectado: "CASOS",
```

```
        descripcion: `[ALTA] Intento de acceso a causa no asignada`,
        datosAfectados: {
          causaId,
          juezAsignadoReal: juezAsignadoDB,
          juezIntentandoAcceder: juezTokenID,
          // ... más contexto
        }
      });

      res.status(403).json({
        success: false,
        error: "No tiene autorización para acceder a esta causa",
        code: "FORBIDDEN_RESOURCE"
      });
      return;
    }

    // ACCESO PERMITIDO
    next();
  };
}
```

2. Middlewares Relacionados

`verificarPropiedadDocumento(documentoParamName)`

- Valida que el documento pertenezca a una causa del juez
- JOIN entre `documentos` y `causas` para obtener `juez_asignado_id`
- Mismo flujo de validación

`verificarPropiedadAudiencia(audienciaParamName)`

- Valida que la audiencia pertenezca a una causa del juez
- JOIN entre `audiencias` y `causas` para obtener `juez_asignado_id`
- Mismo flujo de validación

3. Aplicación en Rutas

Archivo: `backend/src/routes/causas.routes.ts`

```
import { verificarPropiedadCausa } from "../middleware/access-
control.middleware.js";

// Ruta protegida con control de propiedad
router.get(
  "/:id",
  authenticate,                    // 1. Valida JWT
  authorize("ADMIN_CJ", "JUEZ", "SECRETARIO"),  // 2. Valida rol
  verificarPropiedadCausa("id"),  // 3. ⭐ Valida propiedad
  async (req, res, next) => {
    // Si llegamos aquí, el acceso está autorizado
    const causa = await causasService.getCausaById(id);
```

```
    res.json({ success: true, data: causa });
  }
);
```

Rutas Protegidas

Causas

Ruta	Método	Middleware	Descripción
`/api/causas/:id`	GET	`verificarPropiedadCausa("id")`	Detalle de causa
`/api/causas/:id/expediente`	GET	`verificarPropiedadCausa("id")`	Expediente completo

Documentos

Ruta	Método	Middleware	Descripción
`/api/documentos/:id`	GET	`verificarPropiedadDocumento("id")`	Documento individual
`/api/documentos/causa/:causaId`	GET	`verificarPropiedadCausa("causaId")`	Documentos de causa

Audiencias

Ruta	Método	Middleware	Descripción
`/api/audiencias/:id/estado`	PATCH	`verificarPropiedadAudiencia("id")`	Cambiar estado
`/api/audiencias/:id/reprogramar`	PATCH	`verificarPropiedadAudiencia("id")`	Reprogramar

Auditoría y Monitoreo

Evento: ACCESO_DENEGADO

Estructura del log (ALTA severidad):

```
{
  "log_id": 1234,
  "fecha_evento": "2026-01-05T10:30:45.123Z",
  "tipo_evento": "ACCESO_DENEGADO",
  "usuario_id": 42,
  "usuario_correo": "juez.perez@judicial.gob.ec",
  "modulo_afectado": "CASOS",
  "descripcion": "[ALTA] Intento de acceso a causa no asignada. Juez 42 intentó
acceder a causa 123 asignada a juez 15",
  "datos_afectados": {
    "causaId": 123,
    "numeroProceso": "17281-2026-00123",
```

```
    "juezAsignadoReal": 15,
    "juezIntentandoAcceder": 42,
    "ruta": "/api/causas/123",
    "metodo": "GET",
    "estado": "EN_TRAMITE",
    "materia": "CIVIL",
    "unidadJudicial": "Unidad Judicial Civil 1"
  },
  "ip_origen": "192.168.1.100",
  "user_agent": "Mozilla/5.0...",
  "hash_evento": "a3f5b8c9..."
}
```

Alerta en Consola:

```
[SEGURIDAD] ACCESO_DENEGADO - Posible IDOR:
Juez 42 (juez.perez@judicial.gob.ec) intentó acceder a causa 123
desde IP 192.168.1.100
```

Evento: ACCESO_CAUSA

Estructura del log (BAJA severidad):

```
{
  "log_id": 1235,
  "fecha_evento": "2026-01-05T10:31:00.456Z",
  "tipo_evento": "ACCESO_CAUSA",
  "usuario_id": 42,
  "usuario_correo": "juez.perez@judicial.gob.ec",
  "modulo_afectado": "CASOS",
  "descripcion": "[BAJA] Acceso autorizado a causa 456",
  "datos_afectados": {
    "causaId": 456,
    "numeroProceso": "17281-2026-00456",
    "ruta": "/api/causas/456",
    "metodo": "GET"
  },
  "ip_origen": "192.168.1.100",
  "user_agent": "Mozilla/5.0...",
  "hash_evento": "d7e2a4f1..."
}
```

🩹 Casos de Prueba

Caso 1: Acceso Autorizado ☑

Precondiciones:

- Usuario: Juez ID 42

- Token JWT válido con `funcionarioId: 42`
- Causa 456 tiene `juez_asignado_id = 42`

Pasos:

1. Cliente envía: `GET /api/causas/456` con Bearer token
2. `authenticate()` → Extrae `funcionarioId: 42`
3. `authorize("JUEZ")` → Rol permitido
4. `verificarPropiedadCausa()` → Query a DB
5. Validación: `42 === 42` → ☑ MATCH

Resultado Esperado:

- Status: `200 OK`
- Body: Datos de la causa 456
- Log: `ACCESO_CAUSA` con severidad BAJA

Caso 2: Acceso Denegado (IDOR Detectado) ❌

Precondiciones:

- Usuario: Juez ID 42
- Token JWT válido con `funcionarioId: 42`
- Causa 123 tiene `juez_asignado_id = 15`

Pasos:

1. Cliente envía: `GET /api/causas/123` con Bearer token
2. `authenticate()` → Extrae `funcionarioId: 42`
3. `authorize("JUEZ")` → Rol permitido
4. `verificarPropiedadCausa()` → Query a DB
5. Validación: `15 !== 42` → ❌ NO MATCH

Resultado Esperado:

- Status: `403 Forbidden`
- Body:

```
{
  "success": false,
  "error": "No tiene autorización para acceder a esta causa",
  "code": "FORBIDDEN_RESOURCE"
}
```

- Log: `ACCESO_DENEGADO` con severidad ALTA
- Alerta en consola con detalles del intento

Caso 3: Bypass para Administradores ☑

Precondiciones:

- Usuario: Admin ID 1
- Token JWT válido con `funcionarioId: 1`, `rol: "ADMIN_CJ"`
- Causa 123 tiene `juez_asignado_id = 15`

Pasos:

1. Cliente envía: `GET /api/causas/123` con Bearer token
2. `authenticate()` → Extrae `funcionarioId: 1`
3. `authorize("ADMIN_CJ")` → Rol permitido
4. `verificarPropiedadCausa()` → Detecta rol ADMIN_CJ → **BYPASS**

Resultado Esperado:

- Status: `200 OK`
- Body: Datos de la causa 123
- No hay validación de propiedad (acceso total)

Caso 4: Documento de Causa No Asignada ❌

Precondiciones:

- Usuario: Juez ID 42
- Documento DOC-789 pertenece a causa 123
- Causa 123 tiene `juez_asignado_id = 15`

Pasos:

1. Cliente envía: `GET /api/documentos/DOC-789`
2. `verificarPropiedadDocumento()` → JOIN con causas
3. Validación: `15 !== 42` → ❌ NO MATCH

Resultado Esperado:

- Status: `403 Forbidden`
- Log: `ACCESO_DENEGADO` en módulo `DOCUMENTOS`

🔍 Consultas SQL Utilizadas

Validación de Propiedad de Causa

```
SELECT
  causa_id,
  numero_proceso,
  juez_asignado_id,
  juez_pseudonimo,
  estado_procesal,
  materia,
  unidad_judicial
FROM causas
WHERE causa_id = $1;
```

Validación de Propiedad de Documento

```
SELECT
  d.documento_id,
  d.causa_id,
  c.numero_proceso,
  c.juez_asignado_id,  -- ← Campo crítico
  d.tipo,
  d.nombre
FROM documentos d
JOIN causas c ON d.causa_id = c.causa_id
WHERE d.documento_id = $1;
```

Validación de Propiedad de Audiencia

```
SELECT
  a.audiencia_id,
  a.causa_id,
  c.numero_proceso,
  c.juez_asignado_id,  -- ← Campo crítico
  a.tipo,
  a.fecha_hora_programada
FROM audiencias a
JOIN causas c ON a.causa_id = c.causa_id
WHERE a.audiencia_id = $1;
```

⚠ Vectores de Ataque Mitigados

1. IDOR (Insecure Direct Object Reference)

Ataque:

- Juez modifica ID en URL: `/api/causas/123` → `/api/causas/124`
- Intenta acceder a causas de otros jueces

Mitigación:

- ☑ Validación en cada request contra la base de datos
- ☑ No se confía en el token JWT como única fuente
- ☑ Registro de intentos en auditoría
- ☑ Respuesta genérica 403 (sin revelar si el recurso existe)

2. Token Manipulation

Ataque:

- Modificar claim `funcionarioId` en token JWT

Mitigación:

- ☑ Token firmado con secret (HMAC-SHA256)
- ☑ Validación de firma en `authenticate()`
- ☑ Base de datos como fuente de verdad de asignación

3. Privilege Escalation

Ataque:

- Secretario intenta acceder con rol JUEZ

Mitigación:

- ☑ Middleware `authorize()` valida roles permitidos
- ☑ Separación clara de privilegios por rol

4. Session Replay

Ataque:

- Reutilizar token antiguo de sesión anterior

Mitigación:

- ☑ Token con expiración (JWT_EXPIRES_IN)
- ☑ Validación de expiración en cada request
- ☑ IP tracking en logs para detectar anomalías

📊 Métricas de Seguridad

Indicadores Clave

Métrica	Descripción	Umbral de Alerta
Tasa de ACCESO_DENEGADO	% de requests con 403	> 5% del total
Intentos IDOR por usuario	Cantidad de accesos denegados por juez	> 3 en 1 hora
Causas accedidas por juez	Promedio de causas accedidas por sesión	Outliers estadísticos
Tiempo de respuesta middleware	Latencia agregada por validación	> 100ms

Consulta de Análisis (SQL)

```
-- Jueces con múltiples intentos de acceso no autorizado
SELECT
  usuario_id,
  usuario_correo,
  COUNT(*) as intentos_idor,
  array_agg(DISTINCT (datos_afectados->>'causaId')) as causas_intentadas
FROM logs_auditoria
WHERE tipo_evento = 'ACCESO_DENEGADO'
  AND modulo_afectado = 'CASOS'
```

```
  AND fecha_evento >= NOW() - INTERVAL '24 hours'
GROUP BY usuario_id, usuario_correo
HAVING COUNT(*) >= 3
ORDER BY intentos_idor DESC;
```

🚀 Extensiones Futuras

Fase 1: Rate Limiting Dinámico

- [ ] Bloqueo temporal tras N intentos IDOR
- [ ] CAPTCHA después de 3 intentos fallidos
- [ ] Notificación automática a ADMIN_CJ

Fase 2: Análisis de Comportamiento

- [ ] Machine Learning para detectar patrones anómalos
- [ ] Scoring de riesgo por usuario
- [ ] Dashboard de seguridad en tiempo real

Fase 3: Auditoría Avanzada

- [ ] Integración con SIEM externo
- [ ] Exportación de logs a formato CEF
- [ ] Alertas a canales externos (Slack, email)

📚 Referencias

- **Common Criteria**: ISO/IEC 15408
  - FIA_ATD.1: User attribute definition
  - FDP_ACC.1: Subset access control
  - FAU_GEN.1: Audit data generation
- **OWASP Top 10 2021**
  - A01:2021 – Broken Access Control
  - A04:2021 – Insecure Design
- **NIST SP 800-53**
  - AC-3: Access Enforcement
  - AU-2: Audit Events

☑ Criterios de Aceptación

Criterio	Estado	Evidencia
Juez solo ve sus causas	☑	Middleware `verificarPropiedadCausa()`
Admin tiene acceso total	☑	Bypass para rol ADMIN_CJ
Intentos IDOR son loggeados	☑	Log con severidad ALTA en db_logs

Criterio	Estado	Evidencia
Respuesta 403 genérica	☑	No revela existencia del recurso
Validación en DB por request	☑	Query en cada llamada al middleware
Auditoría con hash integridad	☑	SHA-256 en cada evento

👥 Roles y Permisos

Rol	Acceso a Causas	Bypass Validación
JUEZ	Solo causas asignadas	❌ No
SECRETARIO	Causas de su unidad/materia	☑ Sí
ADMIN_CJ	Todas las causas	☑ Sí

Documento generado: 2026-01-05
Sprint: 2 - Operativa del Expediente y Audiencias
Estado: ☑ Implementado y Validado
Próxima Revisión: Sprint 3