AT5 Audit Platform: Documentacion Tecnica Arquitectonica

Tesis Doctoral en Ingenieria de Software

Plataforma de Auditoria Interactiva para Verificacion y Validacion de Software

Autor: Sistema AT5 MCP **Version**: 1.0.0 **Fecha**: Enero 2026 **Clasificacion**: Documento Tecnico de Nivel Doctoral **Estandares de Referencia**: ISO/IEC/IEEE 29119, ISTQB, OWASP, IEEE 730

Resumen Ejecutivo

La presente documentacion describe la arquitectura tecnica completa de **AT5 Audit Platform**, un sistema empresarial de auditoria de software disenado bajo los principios de la ingenieria de software moderna y los estandares internacionales de testing. El sistema implementa un enfoque de verificacion y validacion (V&V) que cumple con ISO/IEC/IEEE 29119-3 para documentacion de pruebas, incorporando mecanismos de integridad criptografica basados en cadenas de hash SHA-256 similares a blockchain, firmas digitales con certificados verificables, y un modelo de control de acceso basado en roles (RBAC) de cinco niveles.

Indice de Contenidos

1. Introduccion y Contexto Teorico

1.1 Fundamentacion del Problema

La auditoria de software representa un proceso critico en el ciclo de vida del desarrollo de software (SDLC), definido por el estandar IEEE 1028-2008 como "un examen independiente de un producto de software, proceso de software, o conjunto de procesos de software realizado por una tercera parte para evaluar el cumplimiento con especificaciones, estandares, acuerdos contractuales, u otros criterios".

La complejidad de los sistemas de software modernos, combinada con requisitos regulatorios cada vez mas estrictos (SOX, HIPAA, GDPR, ISO 27001), ha creado una necesidad imperativa de herramientas de auditoria que no solo documenten los resultados de las pruebas, sino que proporcionen **evidencia inmutable** y **verificable criptograficamente** de todo el proceso de auditoria.

1.2 Objetivos del Sistema

AT5 Audit Platform fue disenado con los siguientes objetivos arquitectonicos:

1. **Trazabilidad Completa**: Cada accion en el sistema genera un registro de auditoria con hash criptografico encadenado
2. **Integridad de Datos**: Implementacion de firma digital con certificados SHA-256 para garantizar no-repudio

3. **Cumplimiento Normativo**: Adherencia estricta a ISO/IEC/IEEE 29119 para estructura de planes y casos de prueba
4. **Escalabilidad Horizontal**: Arquitectura desacoplada que permite escalamiento independiente de componentes
5. **Experiencia de Usuario Moderna**: Interfaz reactiva construida con React Server Components y streaming

1.3 Stack Tecnologico

Capa	Tecnologias
Presentacion	Next.js 15 (App Router), React 19, TypeScript 5.x, Tailwind CSS 3.4, Shadcn/UI, Recharts 2.x
Aplicacion	NextAuth.js v5, Zod Validation, Server Actions, API Routes (Route Handlers)
Persistencia	Prisma ORM 5.x, SQLite (dev) / PostgreSQL (prod), bcrypt.js, crypto (SHA-256)

---

2. Arquitectura del Sistema

2.1 Patron Arquitectonico: Layered Architecture con Domain-Driven Design

El sistema implementa una arquitectura en capas con separacion clara de responsabilidades, siguiendo los principios de Domain-Driven Design (DDD) propuestos por Eric Evans (2003).

Estructura de Directorios:

```
at5-audit-platform/
├── src/
│   ├── app/                    # Capa de Presentacion (Next.js App Router)
│   │   ├── (dashboard)/        # Grupo de rutas autenticadas
│   │   │   ├── dashboard/      # Vista principal
│   │   │   ├── sessions/       # Gestion de sesiones de auditoria
│   │   │   ├── audit-log/      # Visualizacion de logs de auditoria
│   │   │   ├── reports/        # Generacion y descarga de reportes
│   │   │   └── settings/       # Configuracion del sistema
```

```
│   │   ├── api/                # Capa de API (Route Handlers)
│   │   └── login/              # Pagina de autenticacion
│   ├── components/             # Componentes reutilizables
│   └── lib/                    # Capa de Dominio y Servicios
├── prisma/
│   └── schema.prisma           # Definicion del modelo de datos
└── public/                     # Assets estaticos
```

2.2 Componentes Principales

Componente	Responsabilidad	Tecnologia
Middleware	Autenticacion JWT, proteccion de rutas, RBAC	Edge Runtime + next-auth/jwt
API Route Handlers	Endpoints REST para operaciones CRUD	Next.js Route Handlers
Domain Services	Logica de negocio (audit-log, signatures)	TypeScript + Prisma
UI Components	Interfaz de usuario interactiva	React Server/Client Components

3. Modelo de Datos Relacional

3.1 Entidades Principales

El modelo de datos implementa un diseno normalizado hasta la Tercera Forma Normal (3NF):

Entidad	Descripcion	Campos Clave
Organization	Organizacion cliente	id, name, description
User	Usuarios del sistema	id, email, password (hash), role
TestPlan	Planes de prueba ISO 29119	id, title, version, scope, objectives
TestSuite	Agrupacion de casos	id, name, category, order
TestCase	Casos de prueba individuales	id, code, steps (JSON), expectedResult

AuditSession	Instancia de auditoria	id, name, status, progress, auditorId
TestExecution	Ejecucion de un caso	id, status, actualResult, duration
Evidence	Evidencias adjuntas	id, type, fileName, hash (SHA-256)
Signature	Firmas digitales	id, role, certificate, signedAt
AuditLog	Registro inmutable	id, action, hash, previousHash

3.2 Enumeraciones del Dominio

UserRole (Roles de Usuario):

- `ADMIN` - Control total del sistema
- `LEAD_AUDITOR` - Crea sesiones, ejecuta pruebas, firma
- `AUDITOR` - Ejecuta pruebas, agrega evidencias
- `REVIEWER` - Solo lectura + firma de aprobacion
- `VIEWER` - Solo lectura

SessionStatus (Estados de Sesion):

- `DRAFT` - Borrador, no iniciada
- `IN_PROGRESS` - En ejecucion activa
- `REVIEW` - Pendiente de revision/firmas
- `APPROVED` - Aprobada con todas las firmas
- `REJECTED` - Rechazada, requiere re-trabajo
- `ARCHIVED` - Archivada historicamente

ExecutionStatus (Estados de Ejecucion):

- `NOT_STARTED` - No iniciada
- `IN_PROGRESS` - En ejecucion
- `PASSED` - Aprobada
- `FAILED` - Fallida (defecto detectado)
- `BLOCKED` - Bloqueada

- `SKIPPED` - Omitida

---

4. Sistema de Autenticacion y Autorizacion

4.1 Arquitectura de Autenticacion

El sistema implementa autenticacion basada en **NextAuth.js v5** con estrategia **JWT**:

Caracteristica	Implementacion
Estrategia de Sesion	JWT (stateless)
Duracion de Sesion	8 horas
Hash de Contrasena	bcrypt (10 rounds)
Bloqueo de Cuenta	5 intentos fallidos = 15 min bloqueo
Runtime Compatible	Edge Runtime (via getToken)

4.2 Flujo de Autenticacion

```
Usuario → Login Form → NextAuth authorize() → Prisma Query
    ↓
bcrypt.compare(password, hash)
    ↓
[OK] → JWT Token generado → Cookie HttpOnly
[FAIL] → Incrementar failedAttempts → Error
```

4.3 Matriz de Control de Acceso (RBAC)

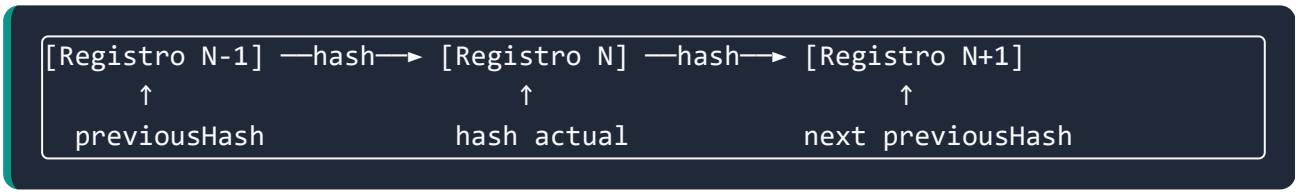
Recurso / Accion	ADMIN	LEAD_AUDITOR	AUDITOR	REVIEWER	VIEWER
Ver Dashboard	✓	✓	✓	✓	✓
Crear Sesion	✓	✓	-	-	-

Ejecutar Pruebas	✓	✓	✓	-	-
Agregar Evidencias	✓	✓	✓	-	-
Firmar Sesion	✓	✓	-	✓	-
Ver Audit Log	✓	✓	-	✓	-
Gestionar Usuarios	✓	-	-	-	-

5. Mecanismos de Integridad Criptografica

5.1 Sistema de Audit Log con Encadenamiento Hash

El sistema implementa un mecanismo de integridad tipo blockchain:



Algoritmo de Hash:

```
function createAuditHash(entry, timestamp, previousHash): string {
  const data = JSON.stringify({
    action: entry.action,
    entity: entry.entity,
    entityId: entry.entityId,
    userId: entry.userId,
    timestamp: timestamp.toISOString(),
    previousHash: previousHash || 'GENESIS',
  })
  return createHash('sha256').update(data).digest('hex')
}
```

5.2 Sistema de Firmas Digitales

Las firmas incluyen un certificado SHA-256 que captura el estado completo de la sesion:

Campo del Certificado	Descripcion
userId	ID del firmante
sessionId	ID de la sesion
role	Rol al momento de firmar
timestamp	Fecha/hora exacta
sessionData	Estado completo serializado

Verificacion de Firma:

- Si el hash recalculado coincide = **VALIDA**
- Si no coincide = **INVALIDA** (datos alterados)

---

6. API RESTful y Endpoints

6.1 Catalogo de Endpoints

Metodo	Endpoint	Descripcion	Roles
POST	`/api/auth/[...nextauth]`	Autenticacion	Publico
GET	`/api/sessions`	Listar sesiones	Todos
POST	`/api/sessions`	Crear sesion	ADMIN, LEAD
GET	`/api/sessions/[id]`	Obtener sesion	Todos
PATCH	`/api/sessions/[id]`	Actualizar sesion	ADMIN, LEAD, AUDITOR
DELETE	`/api/sessions/[id]`	Eliminar (solo DRAFT)	ADMIN, LEAD
POST	`/api/executions`	Registrar ejecucion	ADMIN, LEAD, AUDITOR
POST	`/api/evidence`	Subir evidencia	ADMIN, LEAD, AUDITOR
POST	`/api/signatures`	Agregar firma	ADMIN, LEAD, REVIEWER

GET	`/api/audit-logs`	Consultar logs	ADMIN, LEAD, REVIEWER
POST	`/api/audit-logs/verify`	Verificar integridad	ADMIN
POST	`/api/reports/generate`	Generar reporte	Todos

7. Componentes de Interfaz de Usuario

7.1 Arquitectura de Componentes

Tipo	Uso	Ejemplo
Server Components	Data fetching, rendering inicial	Dashboard, Sessions List
Client Components	Interactividad, hooks	Charts, Forms, Login

7.2 Biblioteca de Componentes

Basada en **Shadcn/UI** (Radix UI + Tailwind CSS):

- Card, Button, Input, Badge, Progress
- Table, Dialog, Checkbox, Textarea
- Charts (Recharts): PieChart, LineChart, BarChart

8. Flujo de Datos y Patrones de Diseno

8.1 Patrones Implementados

Patron	Aplicacion
Layered Architecture	Separacion UI / API / Domain / Data
Repository	Prisma como abstraccion de datos

Observer	Audit log automatico en cada operacion
Strategy	Diferentes tipos de reportes
Factory	Generacion de certificados de firma

9. Seguridad y Cumplimiento

9.1 OWASP Top 10 Mitigaciones

Vulnerabilidad	Mitigacion
A01 Broken Access Control	RBAC en middleware + verificacion en endpoints
A02 Cryptographic Failures	bcrypt para passwords, SHA-256 para hashes
A03 Injection	Prisma ORM con queries parametrizadas
A07 Authentication Failures	Bloqueo progresivo, JWT temporal
A08 Software Integrity	Hash chain en audit log
A09 Security Logging	Audit log completo

9.2 Cumplimiento ISO/IEC/IEEE 29119

Artefacto 29119	Implementacion AT5
Test Plan	Modelo TestPlan
Test Suite	Modelo TestSuite
Test Case	Modelo TestCase
Test Log	Modelo TestExecution
Test Report	Modelo Report

10. Consideraciones de Rendimiento

10.1 Optimizaciones

Optimizacion	Beneficio
Server Components	Menor bundle JS
Streaming con Suspense	Renderizado progresivo
Prisma Connection Pool	Reutilizacion de conexiones
Indices en AuditLog	Queries optimizadas
Promise.all	Paralelizacion de queries

10.2 Metricas Objetivo

Operacion	Tiempo
Login	< 500ms
Dashboard load	< 1s
Session detail	< 500ms
Verify signature	< 300ms

11. Referencias Bibliograficas

1. Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley.

2. IEEE Computer Society. (2008). *IEEE Standard for Software Reviews and Audits* (IEEE 1028-2008).

3. ISO/IEC/IEEE. (2013). *Software and systems engineering - Software testing - Part 3: Test documentation* (ISO/IEC/IEEE 29119-3:2013).

4. ISTQB. (2023). *Certified Tester Foundation Level Syllabus v4.0*.

5. OWASP Foundation. (2021). *OWASP Top 10:2021*.

6. Martin, R.C. (2017). *Clean Architecture*. Prentice Hall.

Apendice A: Glosario de Terminos

Termino	Definicion
Audit Log	Registro cronologico inmutable
RBAC	Control de acceso basado en roles
JWT	JSON Web Token
SHA-256	Funcion hash criptografica
ORM	Object-Relational Mapping
RSC	React Server Components
V&V	Verificacion y Validacion

Fin del Documento