

**Maestría en Ingeniería de Software**

Producto Académico Colaborativo:

Elaboración de un Informe sobre el Caso:

**Gestión de la Etapa de Juicio dentro del Proceso Judicial**

**Integrantes:**

Cangalaya Carrasco Henry Miguel

Guzmán Vargas Jhessel

Sebastian Castillo Jerson

Yanqui Rivera Maria de los Angeles

Docente: FABIO RODRIGUEZ MELENDEZ

Lima – Perú

2025

**INDICE**

[RESUMEN 1](#_Toc199276268)

[ABSTRACT 2](#_Toc199276269)

[INTRODUCCIÓN 3](#_Toc199276270)

[1. PLANIFICACIÓN Y DISEÑO 4](#_Toc199276271)

[1.1. PROBLEMA CENTRAL 4](#_Toc199276272)

[1.2. ANÁLISIS DE CAUSAS 4](#_Toc199276273)

[1.2.1. Causas Directas 4](#_Toc199276274)

[1.2.2. Causas Indirectas 5](#_Toc199276275)

[1.3. ANÁLISIS DE LAS CONSECUENCIAS 5](#_Toc199276276)

[1.4. DISEÑO 6](#_Toc199276277)

[1.5. MODELO DE PROCESOS DE NEGOCIO 6](#_Toc199276278)

[1.5.1. MODELO TO BE 6](#_Toc199276279)

[1.5.1.1. Nivel 1 – Etapa de Juicio 6](#_Toc199276280)

[1.5.1.2. Nivel 2 – Presentación de Pruebas 6](#_Toc199276281)

[1.5.1.3. Nivel 2 – Interrogatorio 7](#_Toc199276282)

[1.5.1.4. Nivel 2 – Deliberación del Juez 8](#_Toc199276283)

[1.5.1.5. Nivel 2 – Emisión de la Sentencia 9](#_Toc199276284)

[1.5.2. MODELOS UML 10](#_Toc199276285)

[1.5.2.1. Diagrama de Clases 10](#_Toc199276286)

[1.5.2.2. Diagrama de Componentes 39](#_Toc199276287)

[1.5.2.3. Diagrama de Actividades 40](#_Toc199276288)

[1.5.2.4. Diagrama de Secuencias 41](#_Toc199276289)

[1.5.2.5. Diagrama de Casos de Uso 42](#_Toc199276290)

[1.5.3. MODELO DE DATOS 42](#_Toc199276291)

[1.5.3.1. MODELO CONCEPTUAL 42](#_Toc199276292)

[1.5.3.2. MODELO LÓGICO 43](#_Toc199276293)

[1.5.3.3. MODELO FÍSICO 45](#_Toc199276294)

[1.6. DETERMINACIÓN DE ÉPICAS, USER STORIES Y CRITERIOS DE ACEPTACIÓN 1](#_Toc199276295)

[1.7. PROPUESTA TECNOLÓGICA EMERGENTE 10](#_Toc199276296)

[1.7.1. Introducción: Respuesta a la Crisis Judicial 10](#_Toc199276297)

[1.7.2. Arquitectura General 12](#_Toc199276298)

[1.7.2.1. Tecnologías Específicas 13](#_Toc199276299)

[1.7.3. Beneficios de la Implementación 14](#_Toc199276300)

[1.7.4. Consideraciones específicas para el contexto local 15](#_Toc199276301)

[1.8. DEFINICIÓN DE ESTRATEGIAS PARA GARANTIZAR LA CALIDAD DEL PRODUCTO SW 15](#_Toc199276302)

[1.8.1. Estrategias de Aseguramiento de Calidad 15](#_Toc199276303)

[1.8.2. Enfoque DevSecOps para el Expediente Fiscal Electrónico 16](#_Toc199276304)

[1.8.3. Capacitación y Concientización 17](#_Toc199276305)

[1.8.4. Herramientas y Tecnologías 17](#_Toc199276306)

[1.9. ESTRATEGIAS DE PRUEBAS ESPECIALIZADAS 17](#_Toc199276307)

[1.9.1. Matriz de pruebas por dominio judicial 17](#_Toc199276308)

[1.9.2. Estrategias de pruebas no funcionales 18](#_Toc199276309)

[1.10. PLANIFICACIÓN DE PRUEBAS PARA EL MÓDULO DE ETAPA DE JUICIO 18](#_Toc199276310)

[1.10.1. Principales Áreas Funcionales a Probar 18](#_Toc199276311)

[1.10.2. Estrategia de Análisis de Calidad con SonarQube 19](#_Toc199276312)

[1.10.3. Plan de Pruebas por Sprint 20](#_Toc199276313)

[1.10.4. Métricas de Calidad Específicas para la Etapa de Juicio 20](#_Toc199276314)

[1.10.5. Control de Calidad de Datos Judiciales 20](#_Toc199276315)

[1.10.5.1. Validación y Verificación de Integridad 20](#_Toc199276316)

[1.10.5.2. Calidad de Base de Datos 21](#_Toc199276317)

[1.10.6. Aseguramiento de Calidad en Producción 21](#_Toc199276318)

[1.10.6.1. Monitoreo Proactivo 21](#_Toc199276319)

[1.10.6.2. Mejora Continua 21](#_Toc199276320)

[1.10.6.3. Estrategias de Validación Legal y Cumplimiento 22](#_Toc199276321)

[1.10.7. Gestión de Riesgos de Calidad 22](#_Toc199276322)

[1.11. DEFINICIÓN DE ESTÁNDARES DE CODIFICACIÓN 23](#_Toc199276323)

[1.11.1. Convenciones Generales 23](#_Toc199276324)

[1.11.1.1. Estructura de Archivos 23](#_Toc199276325)

[1.11.1.2. Nomenclatura 23](#_Toc199276326)

[1.11.1.3. Documentación 24](#_Toc199276327)

[1.11.2. Estándares Específicos por Lenguaje 24](#_Toc199276328)

[1.11.2.1. C# / .NET 24](#_Toc199276329)

[1.11.2.2. SQL Server 24](#_Toc199276330)

[1.11.3. Seguridad y Calidad 25](#_Toc199276331)

[1.11.3.1. Prácticas de Seguridad 25](#_Toc199276332)

[1.11.3.2. Control de Calidad 25](#_Toc199276333)

[1.11.3.3. Estándares para APIs 25](#_Toc199276334)

[1.11.4. Control de Versiones 26](#_Toc199276335)

[1.12. HERRAMIENTAS A UTILIZAR 26](#_Toc199276336)

[1.12.1. Herramientas de Desarrollo 26](#_Toc199276337)

[1.12.1.1. Entornos de Desarrollo Integrados (IDEs) 26](#_Toc199276338)

[1.12.1.2. Modelado y Diseño 26](#_Toc199276339)

[1.12.1.3. Frameworks de Desarrollo 26](#_Toc199276340)

[1.12.1.4. Bases de Datos 26](#_Toc199276341)

[1.12.2. Frameworks de Prueba 27](#_Toc199276342)

[1.12.2.1. Pruebas Unitarias 27](#_Toc199276343)

[1.12.2.2. Pruebas de Integración 27](#_Toc199276344)

[1.12.2.3. Pruebas Específicas 27](#_Toc199276345)

[1.12.3. Herramientas de Gestión de Código 27](#_Toc199276346)

[1.12.3.1. Control de Versiones 27](#_Toc199276347)

[1.12.3.2. Integración y Despliegue Continuo 27](#_Toc199276348)

[1.12.3.3. Gestión de Dependencias 28](#_Toc199276349)

[1.12.4. Calidad y Documentación 28](#_Toc199276350)

[2. CICLO TDD PARA EL DESARROLLO DE LA ETAPA DE JUICIO 28](#_Toc199276351)

[2.1. Enfoque General del TDD 28](#_Toc199276352)

[2.2. Plan de Implementación por Funcionalidades 28](#_Toc199276353)

[2.2.1. Gestión de Pruebas Judiciales 28](#_Toc199276354)

[2.2.2. Gestión de Interrogatorios 29](#_Toc199276355)

[2.2.3. Deliberación Judicial 29](#_Toc199276356)

[2.2.4. Emisión de Sentencia 30](#_Toc199276357)

[2.3. Ejemplos de Pruebas Concretas 30](#_Toc199276358)

[2.3.1. Para Presentación de Pruebas 30](#_Toc199276359)

[2.3.2. Para Interrogatorio 30](#_Toc199276360)

[2.3.3. Para Deliberación 30](#_Toc199276361)

[2.3.4. Para Sentencia 31](#_Toc199276362)

[2.4. Herramientas a Utilizar 31](#_Toc199276363)

[CONCLUSIONES 31](#_Toc199276364)

[RECOMENDACIONES 32](#_Toc199276365)

[REFERENCIA BIBLIOGRÁFICA 33](#_Toc199276366)

**INDICE DE TABLAS**

[Tabla 1: *EPIC1: Gestión de Denuncias* 1](#_Toc199626669)

[Tabla 2: *EPIC2 - Evaluación Preliminar* 2](#_Toc199626670)

[Tabla 3: *Investigación Criminal* 4](#_Toc199626671)

[Tabla 4: *Acusación Formal* 6](#_Toc199626672)

[Tabla 5: *EPIC5 - Gestión de Expedientes* 7](#_Toc199626673)

[Tabla 6: *Matriz de Pruebas por Dominio Judicial* 16](#_Toc199626674)

[Tabla 7: 18](#_Toc199626675)

[Tabla 8: 19](#_Toc199626676)

**INDICE DE FIGURAS**

[Figura 1: *Nivel 1 - Etapa de Juicio* 6](#_Toc199626736)

[Figura 2: *Nivel 2 - Presentación de Pruebas* 7](#_Toc199626737)

[Figura 3: *Nivel 2 - Interrogatorio* 7](#_Toc199626738)

[Figura 4: *Nivel 2 - Deliberación del Juez* 8](#_Toc199626739)

[Figura 5: *Nivel 2 - Emisión de la Sentencia* 9](#_Toc199626740)

[Figura 6: *Diagrama de Clases* 10](#_Toc199626741)

[Figura *7*: *Diagrama de Componentes* 39](#_Toc199626742)

[Figura 8: *Diagrama de Actividades* 40](#_Toc199626743)

[Figura 9: *Diagrama de Secuencias* 41](#_Toc199626744)

[Figura 10: *Diagrama de Casos de Uso* 42](#_Toc199626745)

[Figura 11: *Modelo Conceptual* 42](#_Toc199626746)

[Figura 12: *Modelo Lógico* 44](#_Toc199626747)

[Figura 13: *Modelo físico* 45](#_Toc199626748)

[Figura 14: *Arquitectura General* 11](#_Toc199626749)

[Figura 15: *Estrategias de Aseguramiento de Calidad* 14](#_Toc199626750)

[Figura 16: 22](#_Toc199626751)

# RESUMEN

El documento presenta un informe sobre la gestión de la etapa de juicio dentro del proceso judicial en un contexto latinoamericano, caracterizado por altos niveles de criminalidad organizada, corrupción e impunidad. Se identifican problemas centrales como la dependencia de medios físicos, tecnología obsoleta y falta de interoperabilidad entre sistemas, lo que genera retrasos, pérdida de documentos y sobrecarga laboral. La propuesta incluye un modelo de procesos de negocio, diagramas UML, una arquitectura tecnológica emergente basada en microservicios y blockchain, y estrategias de calidad y pruebas especializadas. El enfoque metodológico sigue el ciclo TDD (Test-Driven Development) para garantizar la robustez del sistema.

# ABSTRACT

This collaborative academic report addresses the management of the trial stage within the judicial process in a Latin American context marked by organized crime, corruption, and impunity. Central issues include reliance on physical media, outdated technology, and lack of system interoperability, leading to delays, document loss, and operational inefficiencies. The proposal features a business process model, UML diagrams, an emerging technological architecture based on microservices and blockchain, and specialized quality and testing strategies. The methodological approach follows TDD (Test-Driven Development) to ensure system robustness.

# INTRODUCCIÓN

La administración de justicia en América Latina enfrenta serios desafíos estructurales derivados de la corrupción, la criminalidad organizada y la fragmentación tecnológica. En este contexto, la etapa de juicio, una de las más críticas del proceso penal, sufre de ineficiencias sistémicas que afectan directamente el acceso a una justicia oportuna y transparente. Este informe propone una transformación digital del proceso judicial centrado en la implementación de un Expediente Fiscal Electrónico (EFE), apoyado en tecnologías emergentes, metodologías ágiles de desarrollo como TDD y prácticas DevSecOps para asegurar la calidad, trazabilidad y transparencia de los procesos.

1. PLANIFICACIÓN Y DISEÑO

Se describe una problemática compleja del sistema judicial de un país latinoamericano caracterizado por altos niveles de criminalidad organizada, corrupción e impunidad. La falta de un sistema ágil y transparente ha provocado una percepción de injusticia y desconfianza en la ciudadanía.

* 1. PROBLEMA CENTRAL

La gestión de expedientes judiciales y fiscales se realiza principalmente a través de medios físicos y sistemas informáticos desactualizados. Esto genera:

* Retrasos significativos en los tiempos de resolución de casos.
* Pérdida o manipulación de documentos físicos.
* Sobrecarga de trabajo para los operadores de justicia.
* Dificultad en el acceso a la información, favoreciendo escenarios de corrupción.
  1. ANÁLISIS DE CAUSAS
     1. Causas Directas
* **Dependencia de medios físicos:** Los expedientes judiciales se almacenan en documentos físicos con cientos de hojas, lo que incrementa el riesgo de pérdida, manipulación o destrucción.
* **Desplazamiento frecuente:** Los usuarios deben presentarse al menos cinco veces al año en las sedes judiciales, lo que implica altos costos y barreras para la población vulnerable.
* **Larga duración de los procesos:** Los casos se extienden más de dos años y no cumplen con los tiempos normativos, afectando el acceso a la justicia.
* **Tecnología obsoleta:** La mitad del parque informático tiene más de seis años de antigüedad y los sistemas actuales son incompatibles con tecnologías modernas.
* **Sistemas informáticos no integrados:** El SIJ (Sistema de Información Judicial) se limita a ser un sistema informativo sin capacidad para gestionar expedientes digitales.
  + 1. Causas Indirectas
* **Infraestructura judicial inadecuada:** Edificios estrechos y sedes alquiladas sin condiciones tecnológicas adecuadas.
* **Falta de interoperabilidad:** Cada institución usa plataformas aisladas sin integración, lo que impide la trazabilidad de los casos.
* **Desactualización normativa:** No existen marcos regulatorios claros para la digitalización y gestión electrónica de los expedientes.
  1. ANÁLISIS DE LAS CONSECUENCIAS
* **Retrasos en la resolución de casos:** Personas permanecen más de 15 años sin sentencia, lo cual vulnera sus derechos humanos.
* **Aumento de la criminalidad organizada:** La falta de control y trazabilidad permite que extorsionadores y sicarios operen desde las cárceles.
* **Corrupción y manipulación de expedientes:** La gestión manual de los expedientes permite la desaparición de documentos y la manipulación de pruebas.
* **Percepción de impunidad:** La ciudadanía percibe un sistema incapaz de impartir justicia de manera eficiente y transparente.
* **Impacto en zonas alejadas:** Las comunidades rurales, especialmente indígenas, enfrentan mayores barreras para acceder a la justicia, lo que perpetúa la discriminación y la marginalización.
  1. MODELO DE PROCESOS DE NEGOCIO
     1. MODELO TO BE
        1. Nivel 1 – Etapa de Juicio

Figura 1:   
*Nivel 1 - Etapa de Juicio*

Diagrama

El contenido generado por IA puede ser incorrecto.

Fuente: Elaboración Propia

* + - 1. Nivel 2 – Presentación de Pruebas

Figura 2:   
*Nivel 2 - Presentación de Pruebas*

Diagrama

El contenido generado por IA puede ser incorrecto.  
Fuente: Elaboración Propia

* + - 1. Nivel 2 – Interrogatorio

Figura 3:   
*Nivel 2 - Interrogatorio*

Diagrama

El contenido generado por IA puede ser incorrecto.

Fuente: Elaboración Propia

* + - 1. Nivel 2 – Deliberación del Juez

Figura 4:   
*Nivel 2 - Deliberación del Juez*

Diagrama

El contenido generado por IA puede ser incorrecto.  
Fuente: Elaboración Propia

* + - 1. Nivel 2 – Emisión de la Sentencia

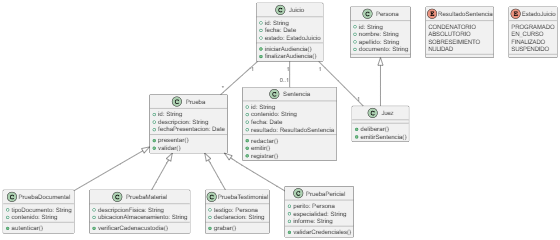
Figura 5:   
*Nivel 2 - Emisión de la Sentencia*

Diagrama, Esquemático

El contenido generado por IA puede ser incorrecto.  
Fuente: Elaboración Propia

* + 1. MODELOS UML
       1. Diagrama de Clases

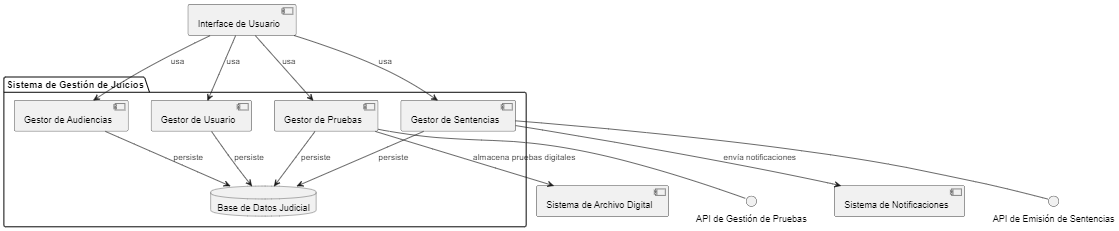
Figura 6:   
*Diagrama de Clases*



Fuente : Elaboración Propia

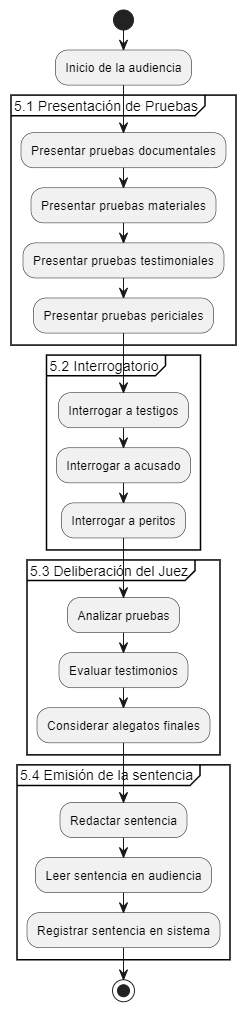
* + - 1. Diagrama de Componentes

Figura *7*:   
*Diagrama de Componentes*

  
Fuente: Elaboración Propia

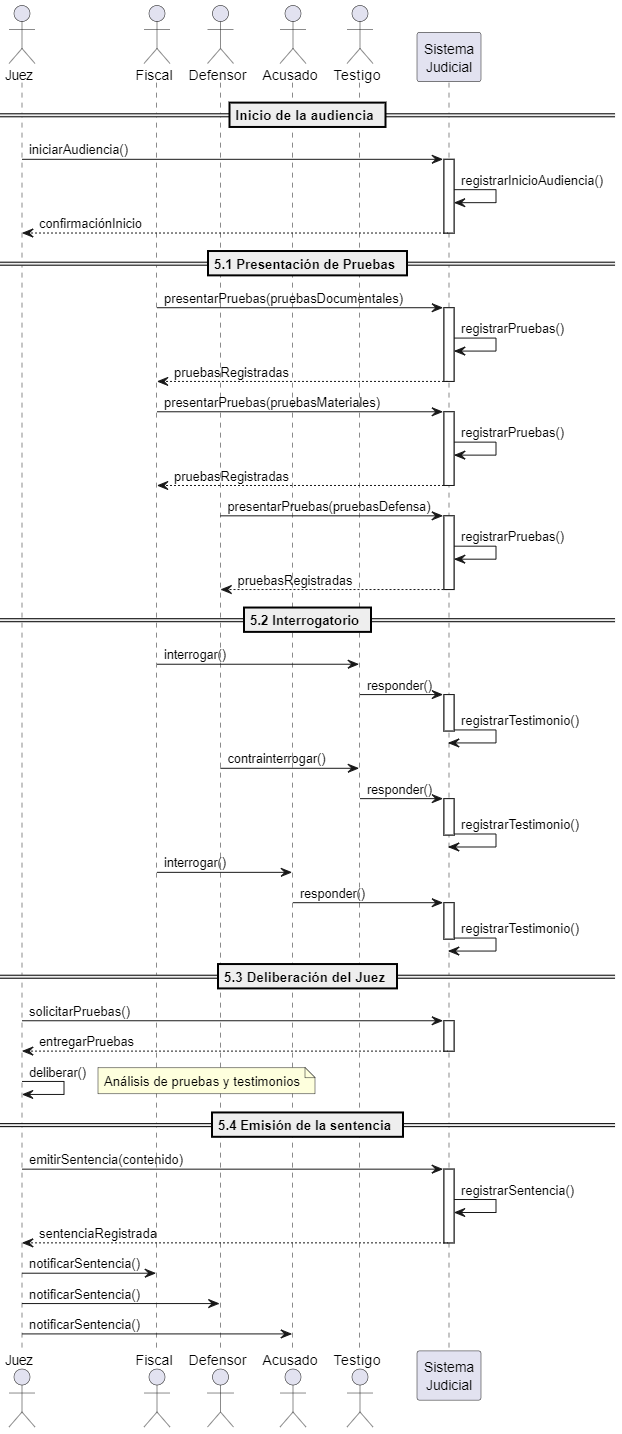
* + - 1. Diagrama de Actividades

Figura 8:   
*Diagrama de Actividades*

  
**Fuente: Elaboración Propia**

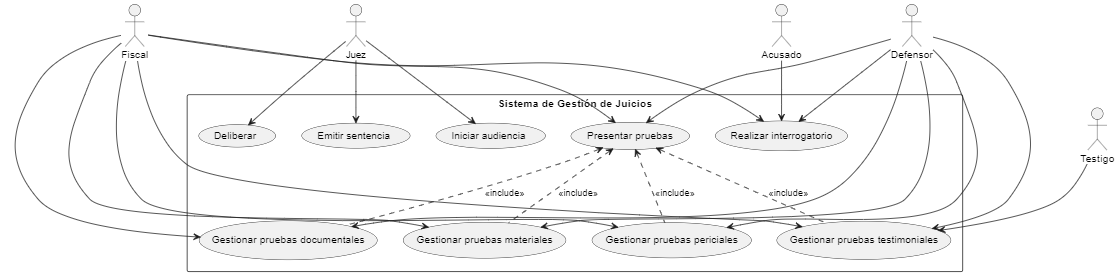
* + - 1. Diagrama de Secuencias

Figura 9:   
*Diagrama de Secuencias*

  
Fuente: Elaboración Propia

* + - 1. Diagrama de Casos de Uso

Figura 10:   
*Diagrama de Casos de Uso*

  
Fuente: Elaboración Propia

* + 1. MODELO DE DATOS
       1. MODELO CONCEPTUAL

Figura 11:   
*Modelo Conceptual*

Diagrama

El contenido generado por IA puede ser incorrecto.  
Fuente: Elaboración Propia

* + - 1. MODELO LÓGICO

Figura 12:   
*Modelo Lógico*

Diagrama

El contenido generado por IA puede ser incorrecto.  
Fuente: Elaboración Propia

* + - 1. MODELO FÍSICO

Figura 13:   
*Modelo físico*

Diagrama, Dibujo de ingeniería

El contenido generado por IA puede ser incorrecto.Fuente: Elaboración Propia

* 1. DETERMINACIÓN DE ÉPICAS, USER STORIES Y CRITERIOS DE ACEPTACIÓN

Tabla 1:   
*EPIC1: Gestión de Denuncias*

|  |
| --- |
| **EPIC1: GESTIÓN DE DENUNCIAS** |
| **US1: Registro de denuncia ciudadana** |
| **Descripción:**  **Como** ciudadano  **Quiero** registrar una denuncia en el sistema  **Para** iniciar un proceso judicial sobre un hecho delictivo |
| **Aceptación Criteria** |
| **E1: Registro exitoso de denuncia**  **Dado** que soy un ciudadano con documento válido  **Cuando** completo el formulario de denuncia con todos los datos obligatorios    Y adjunto evidencias digitales    Y confirmo el envío de la denuncia  **Entonces** el sistema asigna un número único de denuncia    Y cambia el estado a "recibida"    Y notifica la recepción exitosa  **E2: Registro de denuncia con datos incompletos**  **Dado** que soy un ciudadano con documento válido  **Cuando** intento enviar un formulario de denuncia incompleto  **Entonces** el sistema muestra mensajes de error    Y no permite finalizar el registro hasta completar los campos obligatorios |
| **US2: Asignación de fiscal a denuncia** |
| **Descripción:**  **Como** administrador del sistema  **Quiero** asignar un fiscal a cada denuncia recibida  **Para** iniciar el proceso de evaluación |
| **Aceptación Criteria** |
| **E1: Asignación automática por carga de trabajo**  **Dado** que hay una nueva denuncia en estado "recibida"  **Cuando** el sistema ejecuta la asignación automática  **Entonces** se asigna un fiscal con menor carga de trabajo    Y la denuncia cambia a estado "asignada"    Y se notifica al fiscal asignado  **E2: Asignación manual por especialidad**  **Dado** que hay una denuncia compleja en estado "recibida"  **Cuando** el administrador selecciona la opción de asignación manual    Y elige un fiscal especializado en el tipo de delito  **Entonces** la denuncia se asigna al fiscal seleccionado    Y cambia a estado "asignada"    Y se registra el motivo de la asignación manual |

Fuente: Elaboración Propia

Tabla 2:   
*EPIC2 - Evaluación Preliminar*

|  |
| --- |
| **EPIC2: EVALUACIÓN PRELIMINAR** |
| **US1: Evaluación inicial de denuncia** |
| **Descripción:**  **Como** fiscal  **Quiero** realizar una evaluación preliminar de la denuncia  **Para** determinar si procede la investigación |
| **Aceptación Criteria** |
| **E1: Evaluación con procedencia**  **Dado** que soy un fiscal con una denuncia asignada  **Cuando** completo el formulario de evaluación    Y marco la denuncia como "procedente"    Y fundamento mi decisión  **Entonces** el sistema cambia el estado a "en\_investigacion"    Y genera un expediente de evaluación    Y notifica al denunciante  **E2: Evaluación sin procedencia**  **Dado** que soy un fiscal con una denuncia asignada  **Cuando** completo el formulario de evaluación    Y marco la denuncia como "no procedente"    Y documento la justificación detallada  **Entonces** el sistema cambia el estado a "archivada"    Y notifica al denunciante incluyendo los motivos |
| **US2: Programación de diligencias de evaluación** |
| **Descripción:**  **Como** fiscal  **Quiero** programar diligencias preliminares  **Para** obtener información adicional sobre la denuncia |
| **Aceptación Criteria** |
| **E1: Programación exitosa de diligencia**  **Dado** que estoy evaluando una denuncia  **Cuando** creo una nueva diligencia en el sistema    Y asigno tipo, prioridad y fecha límite  **Entonces** la diligencia queda registrada en el sistema    Y se asocia al expediente de evaluación    Y se notifica a los involucrados  **E2: Reprogramación de diligencia**  **Dado** que existe una diligencia programada  **Cuando** modifico la fecha límite    Y registro el motivo de la reprogramación  **Entonces** el sistema actualiza la diligencia    Y notifica el cambio a los involucrados |

Fuente: Elaboración Propia

Tabla 3:   
*Investigación Criminal*

|  |
| --- |
| **EPIC3: INVESTIGACIÓN CRIMINAL** |
| **US1: Apertura de investigación** |
| **Descripción:**  **Como** fiscal  **Quiero** abrir una investigación formal  **Para** recopilar pruebas sobre un presunto delito |
| **Aceptación Criteria** |
| **E1: Apertura desde evaluación positiva**  **Dado** que una denuncia tiene evaluación "procedente"  **Cuando** inicio la fase de investigación   Y completo la información inicial requerida  **Entonces** se crea un expediente de investigación    Y se vincula con la evaluación preliminar    Y el sistema cambia estado a "en\_investigacion"  **E2: Definición de plan de investigación**  **Dado** que he iniciado una investigación  **Cuando** ingreso el plan de investigación    Y defino plazos y recursos necesarios  **Entonces** el sistema registra el plan    Y permite seguimiento del avance por etapas |
| **US2: Gestión de pruebas** |
| **Descripción:**  **Como** fiscal  **Quiero** registrar y organizar pruebas obtenidas  **Para** fundamentar la investigación |
| **Aceptación Criteria** |
| **E1: Registro de nueva prueba**  **Dado** que tengo una investigación activa  **Cuando** agrego una nueva prueba    Y clasifico su tipo y relevancia    Y adjunto los archivos digitales correspondientes  **Entonces** la prueba queda registrada en el sistema    Y se asocia al expediente de investigación  **E2: Valoración de pruebas**  **Dado** que existen pruebas registradas  **Cuando** asigno valoración a cada prueba    Y documento el análisis realizado  **Entonces** el sistema actualiza la valoración    Y calcula el índice de solidez probatoria |

Fuente: Elaboración Propia

Tabla 4:   
*Acusación Formal*

|  |
| --- |
| **EPIC4: ACUSACIÓN FORMAL** |
| **US1: Elaboración de acusación** |
| **Descripción:**  **Como** fiscal  **Quiero** redactar una acusación formal  **Para** presentarla ante el juzgado correspondiente |
| **Aceptación Criteria** |
| **E1: Creación de borrador de acusación**  **Dado** que tengo una investigación con pruebas suficientes  **Cuando** inicio la redacción de acusación    Y completo todos los elementos requeridos    Y selecciono las pruebas que sustentarán la acusación  **Entonces** el sistema guarda el borrador    Y permite revisión antes de formalizar  **E2: Formalización de acusación**  **Dado** que tengo un borrador de acusación completo  **Cuando** selecciono "formalizar acusación"    Y confirmo la acción  **Entonces** el sistema cambia el estado a "formalizada"    Y genera un número de expediente judicial    Y prepara el documento para presentación al juzgado |
| **US2: Selección de juzgado competente** |
| **Descripción:**  **Como** fiscal  **Quiero** seleccionar el juzgado adecuado  **Para** remitir la acusación según jurisdicción y competencia |
| **Aceptación Criteria** |
| **E1: Asignación automática de juzgado**  **Dado** que tengo una acusación formalizada  **Cuando** solicito asignación de juzgado  **Entonces** el sistema determina el juzgado competente    Y asocia la acusación al juzgado seleccionado    Y muestra la información de contacto  **E2: Asignación manual con justificación**  **Dado** que tengo una acusación formalizada  **Cuando** selecciono manualmente un juzgado diferente al sugerido    Y documento el motivo de la selección  **Entonces** el sistema registra la asignación manual    Y asocia la acusación al juzgado seleccionado |

Fuente: Elaboración Propia

Tabla 5:   
*EPIC5 - Gestión de Expedientes*

|  |
| --- |
| **EPIC5: GESTIÓN DE EXPEDIENTES** |
| **US1: Consulta de estado de expediente** |
| **Descripción:**  **Como** ciudadano denunciante  **Quiero** consultar el estado de mi denuncia  **Para** conocer el avance del proceso |
| **Aceptación Criteria** |
| **E1:Consulta con número de denuncia**  **Dado** que soy un ciudadano con una denuncia registrada  **Cuando** ingreso el número de denuncia en el sistema    Y me autentico con mi documento  **Entonces** el sistema muestra el estado actual    Y las últimas actualizaciones del expediente    Y los próximos pasos del proceso  **E2: Recepción de notificaciones de cambios**  **Dado** que soy un denunciante registrado  **Cuando** ocurre un cambio significativo en mi expediente  **Entonces** recibo una notificación por correo electrónico    Y puedo ver el detalle del cambio en la plataforma |
| **US2: Integración entre fiscalía y juzgado** |
| **Descripción:**  **Como** administrador del sistema  **Quiero** sincronizar información entre fiscalía y juzgado  **Para** mantener la integridad del expediente judicial |
| **Aceptación Criteria** |
| **E1: Transferencia de expediente fiscal a judicial**  **Dado** que existe una acusación formalizada  **Cuando** se completa la remisión al juzgado  **Entonces** el sistema transfiere digitalmente el expediente    Y mantiene la trazabilidad de documentos    Y registra la recepción por parte del juzgado  **E2: Actualización bidireccional de información**  **Dado** que un expediente está en fase judicial  **Cuando** el juzgado registra una actuación  **Entonces** el sistema notifica a la fiscalía    Y actualiza el estado visible para todas las partes    Y mantiene sincronizados ambos sistemas |

Fuente: Elaboración Propia

* 1. PROPUESTA TECNOLÓGICA EMERGENTE
     1. Introducción: Respuesta a la Crisis Judicial

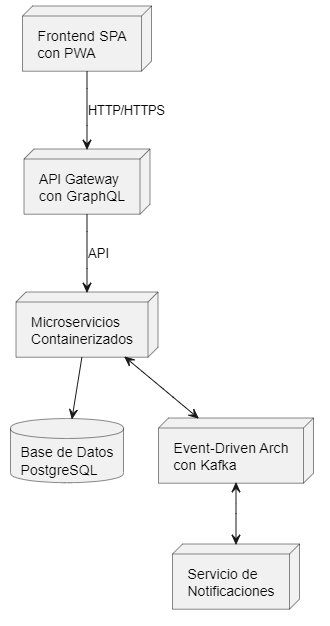
La presente propuesta tecnológica emergente responde directamente a la crisis sin precedentes que enfrenta el sistema judicial del país latinoamericano, caracterizada por el aumento de la delincuencia organizada, la impunidad sistémica y la falta de transparencia en la administración de justicia. El Expediente Fiscal Electrónico (EFE) que proponemos implementar aborda específicamente:

* La **fragmentación y obsolescencia** de los sistemas actuales que operan de forma aislada
* La **pérdida y manipulación** de documentos físicos que facilitan la corrupción
* La **falta de acceso a la justicia** en zonas remotas donde habitan comunidades nativas
* Los **tiempos excesivos de procesamiento** que mantienen a inocentes en prisión sin sentencia
* La **ausencia de trazabilidad** que permite la manipulación de casos por influencias o sobornos

Nuestra propuesta aprovecha el financiamiento del Banco Internacional para construir una plataforma digital integral que no solo digitaliza documentos, sino que transforma fundamentalmente los procesos judiciales, garantizando transparencia, accesibilidad y reducción drástica de los tiempos procesales.

* + 1. Arquitectura General

Figura 14:   
*Arquitectura General*



Fuente: Elaboración Propia

* + - 1. Tecnologías Específicas

**Backend y Procesamiento**

* **Microservicios Containerizados**: Implementación con Docker y Kubernetes para cada módulo funcional (Denuncias, Evaluación, Investigación, Acusación)
* **Event-Driven Architecture**: Apache Kafka para coordinación asíncrona entre etapas del proceso judicial
* **GraphQL API**: Para consultas optimizadas de expedientes y reducción de overhead en comunicaciones

**Almacenamiento y Recuperación**

* **Base de Datos**: PostgreSQL con extensiones PostGIS (para geolocalización de denuncias) y TimescaleDB (para series temporales de actividades)
* **Blockchain Permisionada**: Hyperledger Fabric para registro inmutable de evidencias y pruebas críticas
* **Vector Database**: Para búsqueda semántica en documentos judiciales (Weaviate/Pinecone)

**Frontend y Experiencia de Usuario**

* **Progressive Web App (PWA)**: Acceso móvil y desktop para ciudadanos y funcionarios judiciales
* **Diseño Adaptativo**: Interfaces específicas para diferentes roles (Fiscalía, Policía, Juzgado)
* **Autenticación Biométrica**: Para acceso seguro a información confidencial

**Inteligencia y Automatización**

* **Procesamiento de Lenguaje Natural (NLP)**: Para análisis preliminar de denuncias y categorización automática
* **Machine Learning**: Para detección de patrones en evidencias y priorización de diligencias

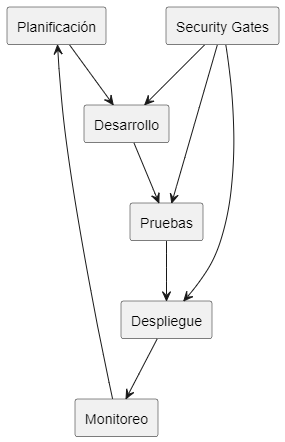
**Seguridad y Cumplimiento**

* **Zero Trust Architecture**: Microperímetros de seguridad para cada componente del sistema
* **Firma Digital Avanzada**: Con infraestructura PKI para documentos legales
* **Auditoría Automatizada**: Registro inmutable de todas las interacciones con datos sensibles
  + 1. Beneficios de la Implementación
* **Agilidad Procesal**: Reducción de tiempos en cada fase del proceso judicial
* **Transparencia**: Trazabilidad completa del proceso para ciudadanos y fiscalías
* **Interoperabilidad**: Integración fluida con sistemas externos (policiales, penitenciarios)
* **Escalabilidad**: Capacidad de absorber incrementos de carga sin degradación del servicio
* **Resiliencia**: Alta disponibilidad mediante replicación y recuperación automática
  + 1. Consideraciones específicas para el contexto local
* **Acceso Remoto**: Implementación de puntos de acceso satelital en zonas rurales para comunidades nativas
* **Interfaz Multilingüe**: Soporte para idiomas indígenas locales además del español
* **Módulo Anti-Corrupción**: Sistema de alertas tempranas ante patrones sospechosos de manipulación de expedientes
* **Operación Offline**: Capacidad de funcionamiento sin conectividad con sincronización posterior
* **Participación Comunitaria**: Plataforma para veedores ciudadanos que monitorean transparencia procesal

Esta propuesta tecnológica emergente permitiría transformar radicalmente el sistema judicial del país, reduciendo la impunidad, combatiendo la corrupción y restaurando la confianza ciudadana en las instituciones de justicia mediante un Expediente Fiscal Electrónico accesible, seguro y eficiente.

* 1. DEFINICIÓN DE ESTRATEGIAS PARA GARANTIZAR LA CALIDAD DEL PRODUCTO SW
     1. Estrategias de Aseguramiento de Calidad

Figura 15:   
*Estrategias de Aseguramiento de Calidad*



Fuente: Elaboración Propia

* + 1. Enfoque DevSecOps para el Expediente Fiscal Electrónico
* **Integración Continua/Despliegue Continuo (CI/CD):** Automatización de pruebas y despliegues en entornos seguros.
* **Control de Calidad en Tiempo Real:** Monitoreo constante de métricas de calidad y seguridad durante el desarrollo.
* **Colaboración Interdisciplinaria:** Equipos de desarrollo, seguridad y operaciones trabajando en conjunto desde el inicio.
* **Feedback Rápido:** Ciclos cortos de retroalimentación para corrección de errores y ajustes de calidad.
  + 1. Capacitación y Concientización
* **Programas de Capacitación Continua:** Actualización constante en mejores prácticas de desarrollo seguro y de calidad.
* **Talleres Prácticos:** Ejercicios hands-on sobre análisis de código, pruebas de seguridad y uso de herramientas de calidad.
* **Simulacros de Respuesta a Incidentes:** Entrenamiento en la identificación y respuesta a brechas de seguridad o fallos de calidad.
  + 1. Herramientas y Tecnologías
* **SonarQube:** Para análisis de calidad de código estático y gestión de deuda técnica.
* **Jenkins/GitLab CI:** Para automatización de pipelines de integración y despliegue continuo.
* **Snyk/Veracode:** Para análisis y corrección de vulnerabilidades en dependencias y código.
  1. ESTRATEGIAS DE PRUEBAS ESPECIALIZADAS
     1. Matriz de pruebas por dominio judicial

Tabla 6:   
*Matriz de Pruebas por Dominio Judicial*

|  |  |  |  |
| --- | --- | --- | --- |
| Dominio Judicial | Tipo de Prueba | Herramienta Sugerida | Frecuencia |
| Penal | Funcional, Seguridad | Selenium, JUnit | Por iteración |
| Civil | Funcional, Rendimiento | LoadRunner, JUnit | Mensual |
| Familiar | Usabilidad, Accesibilidad | Selenium, Axe | Por iteración |
| Laboral | Funcional, Seguridad | Selenium, JUnit | Por iteración |
| Contencioso Administrativo | Rendimiento, Seguridad | LoadRunner, Veracode | Trimestral |

Fuente: Elaboración Propia

* + 1. Estrategias de pruebas no funcionales
* **Pruebas de Rendimiento**: Simulación de carga y estrés para evaluar comportamiento ante picos de actividad.
* **Pruebas de Seguridad**: Escaneo y penetración para identificar y remediar vulnerabilidades.
* **Pruebas de Usabilidad**: Evaluación de la interfaz y experiencia de usuario con enfoque en la accesibilidad.
* **Pruebas de Compatibilidad**: Asegurar funcionamiento adecuado en diferentes navegadores y dispositivos.
  1. PLANIFICACIÓN DE PRUEBAS PARA EL MÓDULO DE ETAPA DE JUICIO
     1. Principales Áreas Funcionales a Probar

Tabla 7:

*Áreas Funcionales a Probar*

|  |  |  |
| --- | --- | --- |
| Área Funcional | Descripción | Criticidad |
| Presentación de Pruebas | Carga, validación y gestión de material probatorio | Alta |
| Registro de Testimonios | Captura y transcripción de declaraciones de testigos y peritos | Media |
| Deliberación y Veredicto | Documentación de conclusiones y decisiones judiciales | Alta |
| Notificación de Sentencias | Generación y distribución de resoluciones a las partes | Alta |
| Gestión de Recursos | Registro y seguimiento de apelaciones y recursos | Media |

Fuente: Elaboración Propia

* + 1. Estrategia de Análisis de Calidad con SonarQube
* **Implementación de Quality Gates**: Configuración de umbrales específicos para el módulo de Etapa de Juicio con métricas de:
  + Cobertura de código: Mínimo 85% para funcionalidades críticas del juicio
  + Deuda técnica: Máximo 3 días permitidos
  + Duplicación: Menor al 3% del código
  + Vulnerabilidades: 0 críticas o altas permitidas
  + Code smells: Máximo 10 por componente
* **Análisis Continuo**: Integración de SonarQube en el pipeline CI/CD específico para el módulo de juicio
  + Escaneo automático tras cada commit al repositorio
  + Análisis de calidad previo a cada despliegue a entornos de prueba
  + Dashboards de seguimiento específicos para el equipo de desarrollo
* **Reglas Personalizadas**: Creación de perfiles de calidad específicos para lógica judicial
  + Reglas de validación para nomenclatura de expedientes judiciales
  + Detección de potenciales fugas de información confidencial
  + Verificación de patrones de seguridad en acceso a documentos judiciales
    1. Plan de Pruebas por Sprint

Tabla 8:

*Plan de Pruebas por Sprint*

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Enfoque | Tipos de Pruebas | Criterios de Aceptación |
| 1 | Gestión de Audiencias | Unitarias, Integración | Quality Gate SonarQube, 0 bugs críticos |
| 2 | Presentación de Pruebas | Unitarias, Seguridad | Quality Gate SonarQube, Test de penetración |
| 3 | Registro de Testimonios | Unitarias, UI/UX | Quality Gate SonarQube, Pruebas de usabilidad |
| 4 | Deliberación y Veredicto | Unitarias, Rendimiento | Quality Gate SonarQube, Tiempo de respuesta < 2s |
| 5 | Notificación y Recursos | Integración, E2E | Quality Gate SonarQube, Flujo completo verificado |

Fuente: Elaboración Propia

* + 1. Métricas de Calidad Específicas para la Etapa de Juicio
* **Precisión Jurídica**: Validación con expertos legales de la correcta implementación de reglas procesales
* **Integridad Documental**: Verificación de no repudio en documentos generados durante el juicio
* **Trazabilidad de Acciones**: Auditoría completa de cada acción realizada sobre expedientes durante la fase de juicio
* **Tiempos de Respuesta**: Monitoreo de performance en momentos críticos como la carga de pruebas multimedia
* **Concurrencia**: Capacidad para manejar múltiples sesiones de juicio simultáneas sin degradación
  + 1. Control de Calidad de Datos Judiciales
       1. Validación y Verificación de Integridad
* **Validación Jurídica**: Implementación de validadores especializados para documentos legales
* **Consistencia de Datos**: Verificación de integridad entre etapas procesales
* **Trazabilidad Completa**: Auditoría de cada acción realizada sobre expedientes
* **Firmas Digitales**: Validación criptográfica de documentos oficiales
  + - 1. Calidad de Base de Datos
* **Normalización Optimizada**: Esquemas optimizados para búsquedas judiciales frecuentes
* **Gestión de Datos Maestros**: Para entidades como juzgados, fiscalías y códigos penales
* **Indexación Especializada**: Para búsqueda eficiente de precedentes y jurisprudencia
* **Archivado Legal**: Políticas de retención conforme a normativa procesal vigente
  + 1. Aseguramiento de Calidad en Producción
       1. Monitoreo Proactivo
* **Observabilidad End-to-End**: Seguimiento de KPIs procesales (tiempos de respuesta, resolución de casos)
* **Alertas Tempranas**: Detección de anomalías en flujos de trabajo judiciales
* **APM Especializado**: Monitoreo de rendimiento en puntos críticos del proceso judicial
  + - 1. Mejora Continua
* **Análisis de Incidentes**: Metodología de causa raíz para fallos en procesos judiciales
* **Retroalimentación Estructurada**: Sesiones periódicas con fiscalías y juzgados
* **Métricas de Calidad**: Dashboard de indicadores de efectividad del sistema judicial
  + - 1. Estrategias de Validación Legal y Cumplimiento
* **Revisión por Expertos Legales**: Validación de conformidad con códigos procesales
* **Auditorías de Compliance**: Verificación periódica de cumplimiento normativo
* **Simulacros de Auditoría**: Preparación para inspecciones de órganos de control judicial
* **Ethical Hacking**: Evaluación de posibilidades de manipulación de expedientes
  + 1. Gestión de Riesgos de Calidad

Figura 16:

*Gestion de Riesgos de Calidad*

Fuente: Elaboración Propia

* **Análisis FMEA**: Para cada etapa crítica del proceso judicial
* **Plan de Contingencia**: Para escenarios de fallos en etapas críticas (ej. formalización de acusación)
* **Gestión de Cambios**: Procedimientos reforzados para modificaciones en módulos sensibles

La implementación de estas estrategias permitirá garantizar un Expediente Fiscal Electrónico con la confiabilidad, disponibilidad e integridad necesarias para transformar el sistema judicial, reduciendo la impunidad y aumentando la transparencia del proceso judicial.

* 1. DEFINICIÓN DE ESTÁNDARES DE CODIFICACIÓN

Se establece los estándares de codificación y guías de estilo que el equipo seguirá durante la construcción del **Sistema de Gestión Procesal Penal**. Estos estándares son obligatorios para todos los miembros del equipo y tienen como objetivo garantizar la calidad, mantenibilidad, seguridad y consistencia del código fuente.

* + 1. Convenciones Generales
       1. Estructura de Archivos
* Organizar los módulos según la estructura del proceso judicial (recepción, evaluación, investigación, formalización, juicio, sentencia, ejecución).
* Utilizar nombres descriptivos para los archivos que reflejen su funcionalidad.
* Limitar cada archivo a una sola responsabilidad o componente.
  + - 1. Nomenclatura
* **PascalCase** para clases y componentes: EmisionSentencia, GestionExpediente.
* **camelCase** para variables locales: expedienteActual.
* **PascalCase** para métodos y propiedades: ObtenerSentencia(), ExpedienteActual.
* **\_camelCase** con guión bajo para campos privados: \_repositorio, \_servicioUsuarios.
* **MAYÚSCULAS\_CON\_GUIONES** para constantes: ESTADO\_ARCHIVADO.
* Prefijos claros para interfaces: ISentencia, IRepositorioDocumentos.
  + - 1. Documentación
* Documentar todas las clases, métodos y funciones con comentarios XML (///).
* Incluir información sobre parámetros, valores de retorno y excepciones.
* Mantener la documentación actualizada cuando se modifique el código.
  + 1. Estándares Específicos por Lenguaje
       1. C# / .NET
* Indentación de 4 espacios (no tabuladores).
* Apertura de llaves en una nueva línea.
* Una instrucción por línea.
* Líneas no superiores a 120 caracteres.
* Uso obligatorio de bloques try-catch para manejo de excepciones.
* Preferir tipos de propiedades automáticas cuando sea posible.
* Utilizar async/await para operaciones asíncronas.
* Aplicar principios SOLID en el diseño de clases y componentes.
* Uso de inyección de dependencias mediante el contenedor IoC de .NET.
  + - 1. SQL Server
* Palabras clave en MAYÚSCULAS.
* Nombres de tablas en PascalCase.
* Cada cláusula en una línea distinta.
* Comentarios explicativos para consultas complejas.
* Utilizar procedimientos almacenados para operaciones complejas.
* Índices adecuados para optimizar consultas frecuentes.
* Seguir convenciones de T-SQL para comandos específicos de SQL Server.
  + 1. Seguridad y Calidad
       1. Prácticas de Seguridad
* No incluir credenciales o tokens en el código fuente.
* Implementar validación de entrada para todos los datos externos.
* Utilizar consultas parametrizadas para prevenir inyección SQL.
* Implementar encriptación para datos sensibles.
* Utilizar Identity Server o Azure AD para la autenticación cuando sea posible.
* Seguir las directrices de seguridad de OWASP para aplicaciones .NET.
  + - 1. Control de Calidad
* Cobertura mínima de pruebas unitarias: **80%**.
* Revisión de código obligatoria antes de integración.
* Análisis estático de código con **SonarQube** y **StyleCop**.
* Prohibido hacer commit de código con warnings o errores.
* Implementar pruebas de integración para APIs y servicios.
  + - 1. Estándares para APIs
* Seguir principios REST para APIs web.
* Implementar APIs con ASP.NET Core.
* Versionado explícito de endpoints: /api/v1/sentencias.
* Respuestas consistentes con códigos HTTP apropiados.
* Documentación automática con **Swagger/OpenAPI**.
* Utilizar DTOs para transferencia de datos entre capas.
  + 1. Control de Versiones
* Utilizar Git con modelo de ramificación **GitFlow**.
* Formato de mensajes de commit:
  + Descriptivos y claros sobre el cambio realizado.
* Prohibido hacer commit directamente a las ramas principal y desarrollo.
  1. HERRAMIENTAS A UTILIZAR
     1. Herramientas de Desarrollo
        1. Entornos de Desarrollo Integrados (IDEs)
* **Visual Studio Code**: Para desarrollo general y edición de código
* **IntelliJ IDEA**: Para desarrollo Java/Backend
* **Eclipse**: Alternativa para desarrollo Java
  + - 1. Modelado y Diseño
* **Visual Paradigm**: Para creación de diagramas UML (clases, secuencias, casos de uso)
* **Bizagi Modeler**: Para modelado de procesos de negocio BPMN
* **PlantUML**: Para generación de diagramas UML mediante código
  + - 1. Frameworks de Desarrollo
* **Spring Boot**: Para el desarrollo del backend
* **Angular/React**: Para interfaces de usuario
* **Hibernate**: Para mapeo objeto-relacional
* **REST/SOAP**: Para servicios web y APIs
  + - 1. Bases de Datos
* **PostgreSQL**: Base de datos principal con soporte para auditoría
* **MongoDB**: Para almacenamiento de documentos no estructurados
* **Redis**: Para caché y sesiones
  + 1. Frameworks de Prueba
       1. Pruebas Unitarias
* **JUnit**: Para pruebas unitarias en Java
* **Jest**: Para pruebas de componentes frontend
* **Mockito**: Para simulación de objetos en pruebas
  + - 1. Pruebas de Integración
* **Postman**: Para pruebas de API REST
* **SoapUI**: Para pruebas de servicios web
* **Selenium/Cypress**: Para pruebas end-to-end de interfaces
  + - 1. Pruebas Específicas
* **JMeter**: Para pruebas de rendimiento y carga
* **OWASP ZAP**: Para pruebas de seguridad automatizadas
* **TestRail**: Para gestión de casos de prueba
  + 1. Herramientas de Gestión de Código
       1. Control de Versiones
* **Git**: Sistema de control de versiones
* **GitHub/GitLab**: Plataforma para repositorios y colaboración
  + - 1. Integración y Despliegue Continuo
* **Jenkins**: Para automatización de builds y despliegues
* **SonarQube**: Para análisis estático de código
* **Docker**: Para contenerización de servicios
* **Kubernetes**: Para orquestación de contenedores
  + - 1. Gestión de Dependencias
* **Maven/Gradle**: Para gestión de dependencias Java
* **npm/yarn**: Para gestión de dependencias JavaScript
  + 1. Calidad y Documentación
* **Swagger/OpenAPI**: Para documentación de APIs
* **Javadoc**: Para documentación de código Java
* **Markdown**: Para documentación de proyecto

1. CICLO TDD PARA EL DESARROLLO DE LA ETAPA DE JUICIO
   1. Enfoque General del TDD

El desarrollo del módulo "5. Etapa de Juicio" seguirá el ciclo Red-Green-Refactor aplicado a una arquitectura DDD hexagonal orientada a eventos:

**Red:** Escribir pruebas que fallen inicialmente para validar requisitos de dominio

**Green:** Implementar el mínimo código necesario para pasar las pruebas

**Refactor**: Mejorar la calidad del código sin romper las pruebas

* 1. Organización de Pruebas por Capas de Arquitectura
     1. Pruebas de Dominio (EtapaDeJuicio.Domain.Tests)

**Validan las reglas del negocio en su forma más pura:**

* Entidades y Agregados
* Value Objects
* Reglas de dominio
* Eventos de dominio
* Políticas y servicios de dominio
  + 1. Pruebas de Aplicación (EtapaDeJuicio.Application.Tests)

**Verifican la orquestación de casos de uso:**

* Comandos y sus manejadores
* Consultas y sus manejadores
* Sagas para procesos de negocio transaccionales
* Eventos y sus manejadores
* Servicios de aplicación
  + 1. Pruebas de Infraestructura (EtapaDeJuicio.Infrastructure.Tests)

**Comprueban los adaptadores técnicos:**

* Repositorios y persistencia
* Implementaciones de UnitOfWork
* Event Bus
* Servicios externos
* Adaptadores y gateways
  + 1. Pruebas de API (EtapaDeJuicio.API.Tests)

**Validan la interfaz con el exterior:**

* Controllers
* DTOs
* Filtros y middleware
* Seguridad y autenticación
  + 1. Pruebas de Integración (EtapaDeJuicio.Integration.Tests)

**Aseguran la correcta interacción entre componentes:**

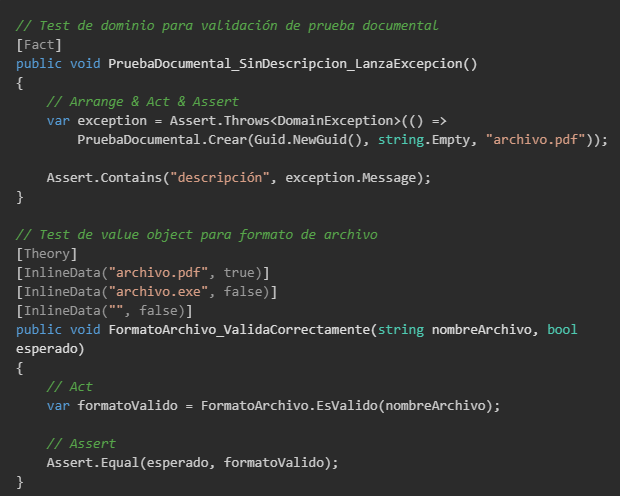
* Flujos de trabajo completos
* Comunicación entre microservicios
* Consistencia eventual
  1. Plan de Implementación por Funcionalidades
     1. Gestión de Pruebas Judiciales

**Dominio**

**Ciclo TDD:**

* **Red:** Crear pruebas para el agregado PruebaJudicial y sus entidades relacionadas
* **Green:** Implementar entidades, value objects y reglas de dominio
* **Refactor:** Mejorar con validaciones específicas por tipo de prueba

**Ejemplos específicos:**

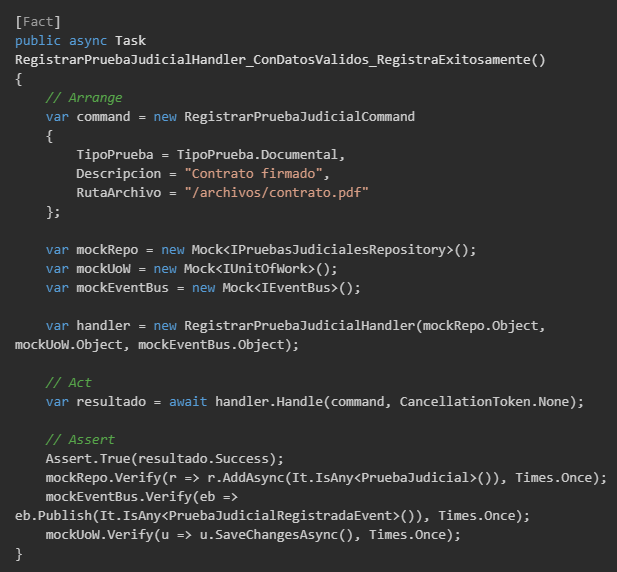


**Aplicación**

**Ciclo TDD:**

* **Red**: Probar comandos como RegistrarPruebaJudicialCommand y sus handlers
* **Green**: Implementar manejadores de comandos y consultas
* **Refactor**: Mejorar con validaciones específicas y manejo de errores

Ejemplos específicos:



**Infraestructura**

**Ciclo TDD:**

* **Red**: Probar implementaciones de repositorios y persistencia
* **Green**: Implementar repositorios y UnitOfWork
* **Refactor**: Optimizar consultas y manejo de transacciones

Ejemplos específicos:

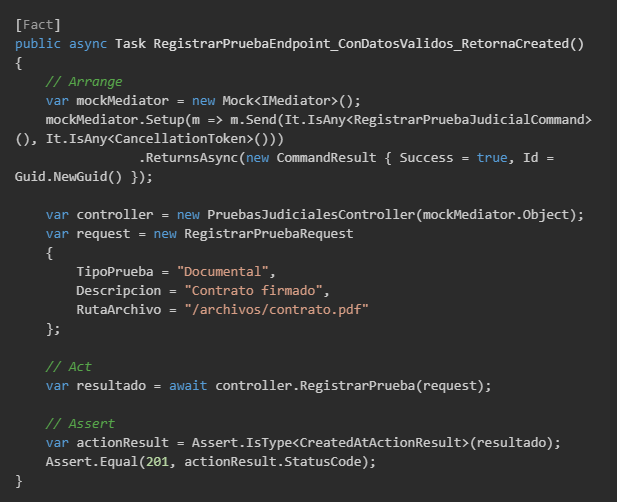


**API**

**Ciclo TDD:**

* **Red**: Probar endpoints de API para gestión de pruebas
* **Green**: Implementar controladores y acciones
* **Refactor**: Mejorar validación de entrada y manejo de errores

Ejemplos específicos:



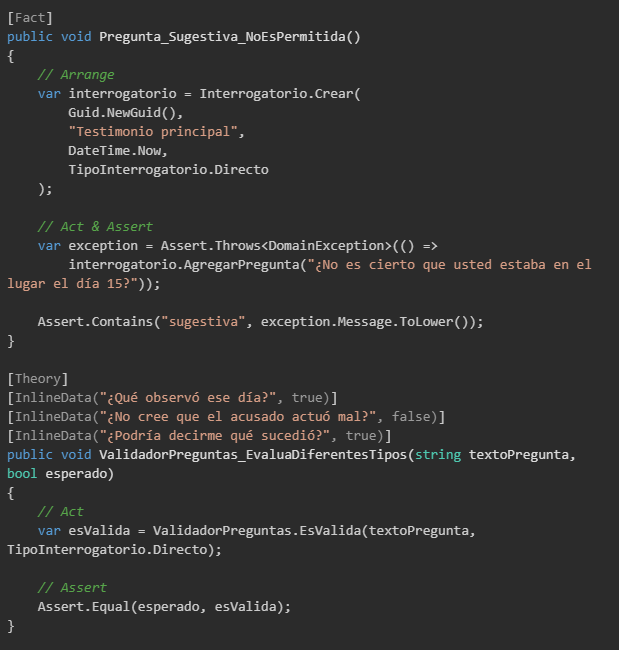
* + 1. Gestión de Interrogatorios

**Dominio**

**Ciclo TDD:**

* **Red**: Probar entidades como Interrogatorio, Pregunta y Testimonio
* **Green**: Implementar entidades y reglas de dominio
* **Refactor**: Mejorar con reglas de validación de preguntas improcedentes

Ejemplos específicos:

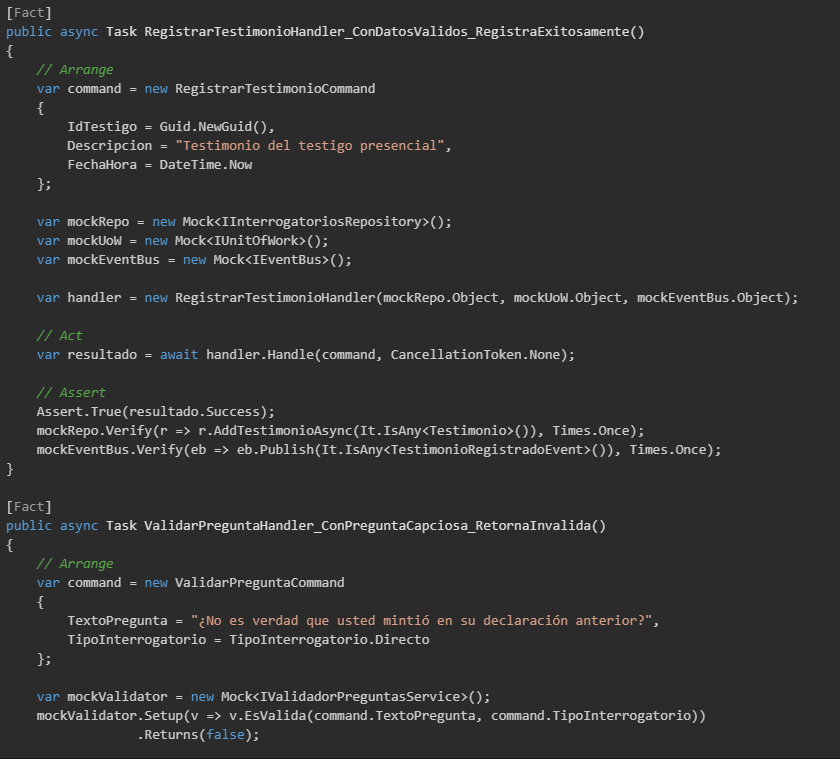


**Aplicación**

**Ciclo TDD:**

* **Red**: Probar comandos y handlers para registro y validación de interrogatorios
* **Green**: Implementar handlers y validaciones
* **Refactor**: Mejorar con validaciones más sofisticadas

Ejemplos específicos:



* + 1. Deliberación Judicial

**Dominio**

**Ciclo TDD:**

* **Red**: Probar entidades de Deliberacion, ConsiderandoLegal y ValoracionPrueba
* **Green**: Implementar entidades y servicios de dominio
* **Refactor**: Mejorar con algoritmos de valoración probatoria

Ejemplos específicos:



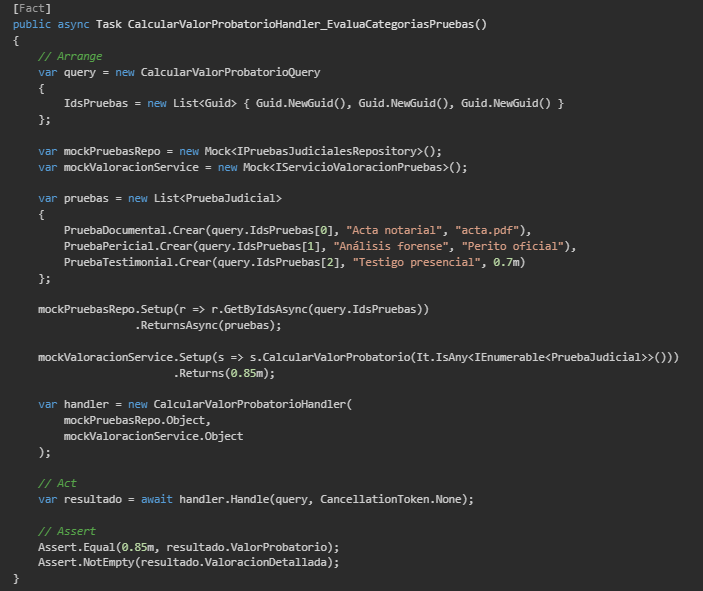
**Aplicación**

**Ciclo TDD:**

* **Red**: Probar generación de considerandos y análisis de pruebas
* **Green**: Implementar manejadores para deliberación
* **Refactor**: Mejorar algoritmos y manejo de casos especiales

Ejemplos específicos:





* + 1. Emisión de Sentencia

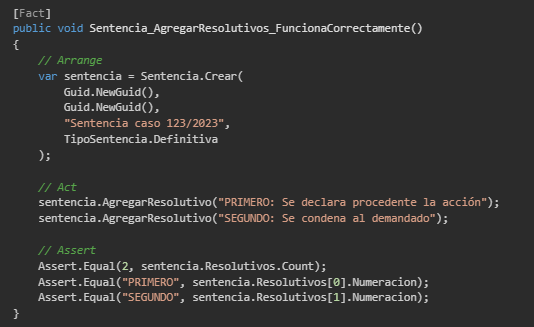
**Dominio**

**Ciclo TDD:**

* **Red**: Probar entidades para Sentencia, Resolutivo y Notificacion
* **Green**: Implementar entidades, value objects y reglas de dominio
* **Refactor**: Mejorar validaciones y metadatos adicionales

Ejemplos específicos:





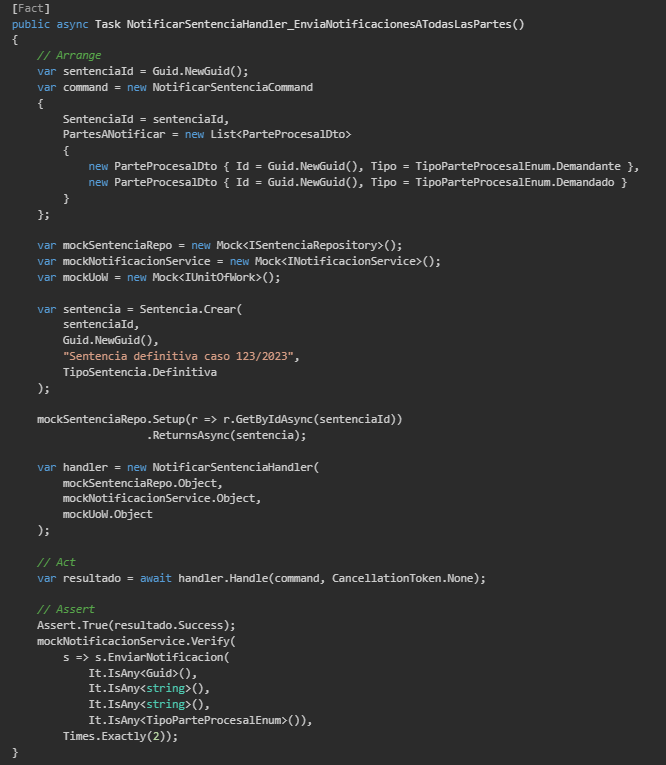
**Aplicación**

**Ciclo TDD:**

* **Red**: Probar comandos para emitir sentencias y generar notificaciones
* **Green**: Implementar handlers de comandos y eventos
* **Refactor**: Mejorar validaciones y manejo de errores

Ejemplos específicos:





* 1. Ejemplos de Pruebas por Casos de Uso
     1. Para Presentación de Pruebas

**Pruebas de Dominio**

* Verificar que rechace pruebas documentales sin descripción
* Verificar validación de formatos de archivo permitidos
* Verificar creación y validación de pruebas periciales
* Verificar clasificación correcta por tipo de prueba

**Pruebas de Aplicación**

* Verificar registro exitoso de prueba documental
* Verificar que no permita registrar pruebas duplicadas
* Verificar registro de metadatos adicionales de una prueba

**Pruebas de Infraestructura**

* Verificar persistencia correcta en base de datos
* Verificar recuperación por ID y por tipo de prueba
* Verificar publicación de eventos tras registro de pruebas
  + 1. Para Interrogatorio

**Pruebas de Dominio**

* Verificar limitación de preguntas por interrogatorio
* Verificar reglas de validación de preguntas según tipo de interrogatorio
* Verificar registro correcto de testimonios y respuestas

**Pruebas de Aplicación**

* Verificar flujo del interrogatorio mediante una saga
* Verificar validadores de preguntas improcedentes
* Verificar cambios de estado durante el interrogatorio

**Pruebas de Integración**

* Verificar límites de tiempo para interrogatorios
* Verificar flujo completo de interrogatorio con preguntas y respuestas
* Verificar interacción con el repositorio de testigos
  + 1. Para Deliberación

**Pruebas de Dominio**

* Verificar cálculo de valor probatorio por categoría de prueba
* Verificar generación de considerandos basados en pruebas
* Verificar reglas de fundamentación legal en considerandos

**Pruebas de Aplicación**

* Verificar ponderación de pruebas combinadas
* Verificar generación de considerandos legales
* Verificar vinculación de pruebas con hechos probados

**Pruebas de API**

* Verificar consultas de avance de deliberación
* Verificar registro de fundamentos legales
* Verificar historial de modificaciones en deliberación
  + 1. Para Sentencia

**Pruebas de Dominio**

* Verificar estructura correcta de la sentencia
* Verificar validación de datos obligatorios (juez, descripción)
* Verificar adición y numeración correcta de resolutivos

**Pruebas de Aplicación**

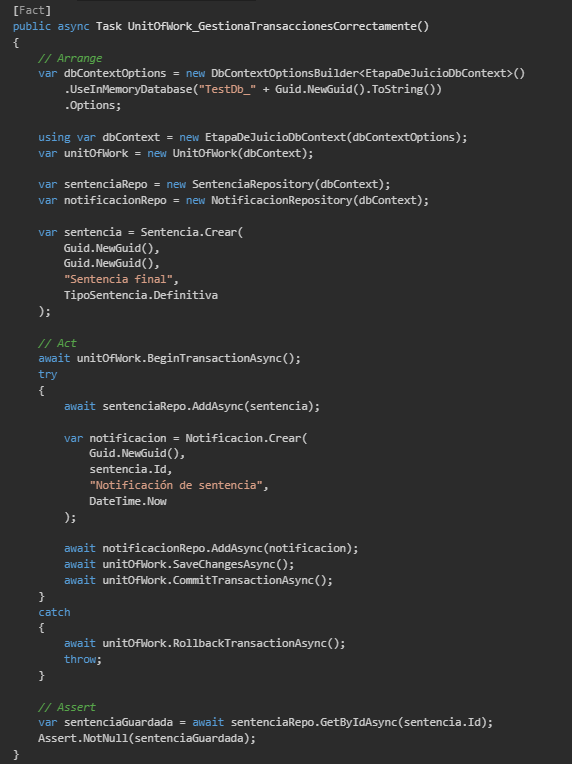
* Verificar emisión de sentencia completa
* Verificar notificación a todas las partes procesales
* Verificar generación de versión oficial firmada

**Pruebas de Integración**

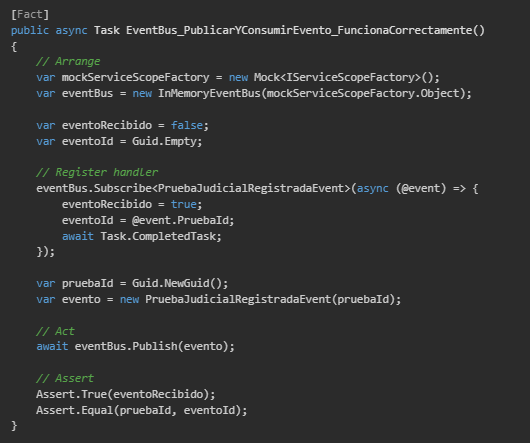
* Verificar flujo completo desde deliberación hasta sentencia
* Verificar firma digital del juez
* Verificar disponibilidad de sentencia para consulta
  1. Estrategias de Pruebas para Componentes de Arquitectura
     1. Pruebas para Repository Pattern



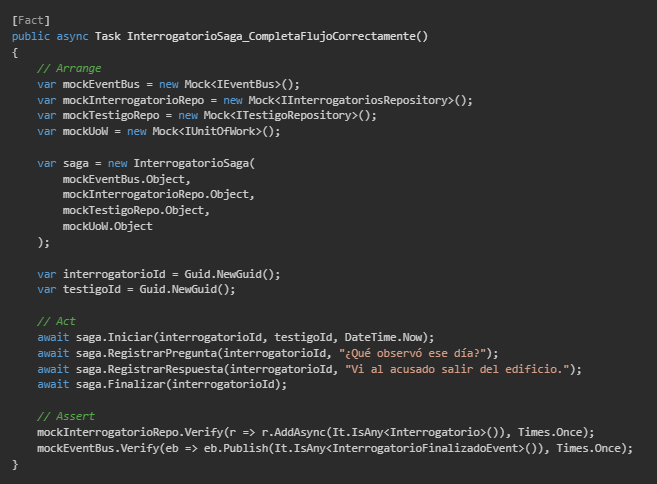
* + 1. Pruebas para Unit of Work



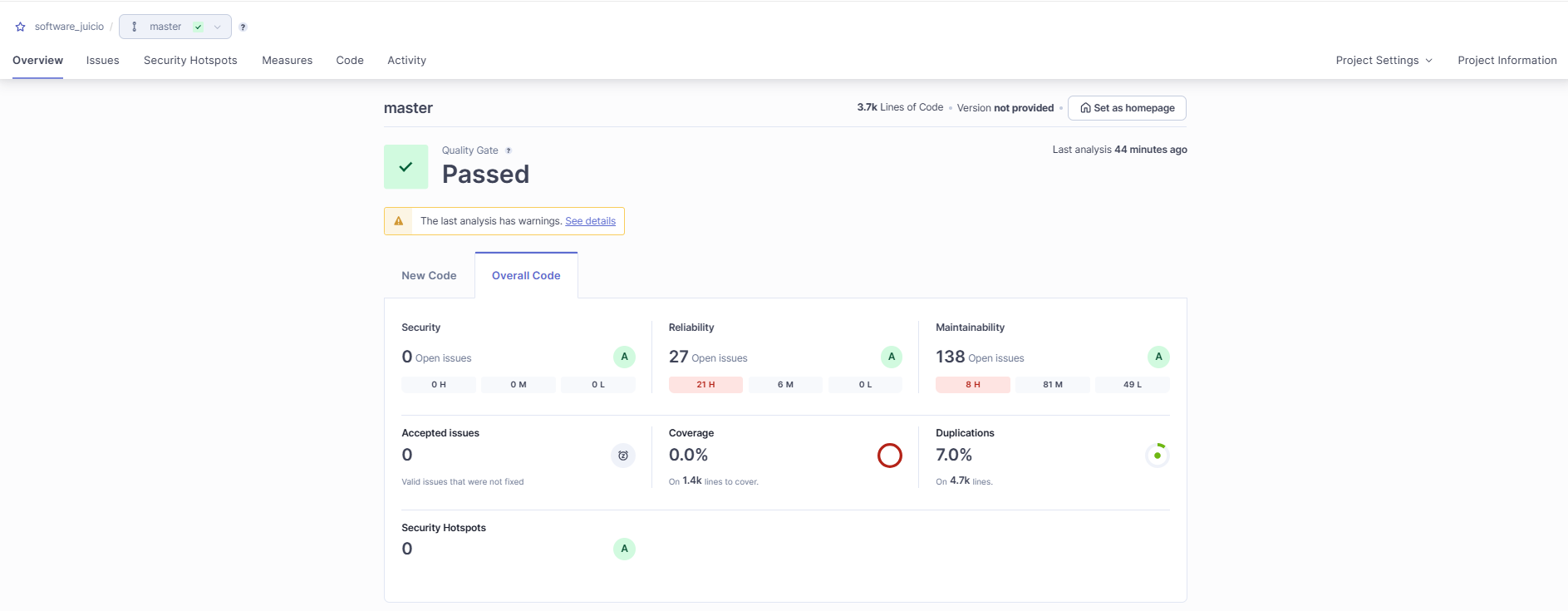
* + 1. Pruebas para Event Bus

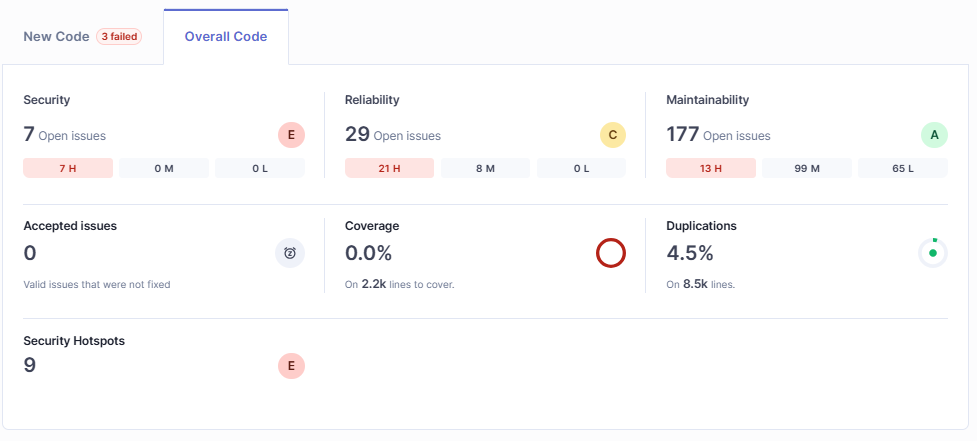


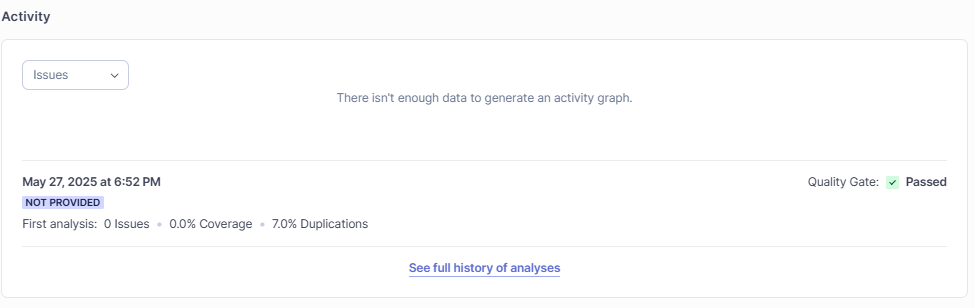
* + 1. Pruebas para Saga Pattern

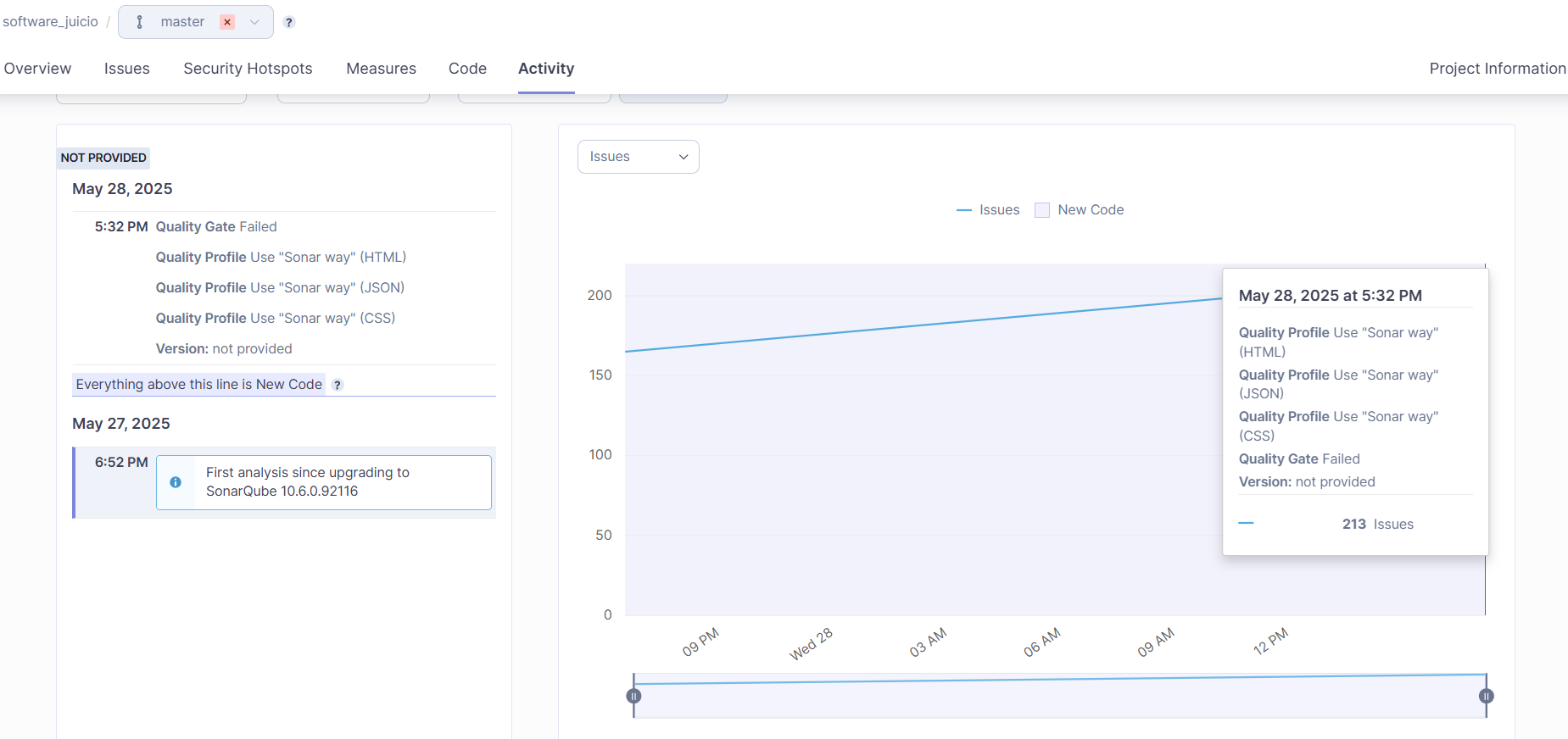


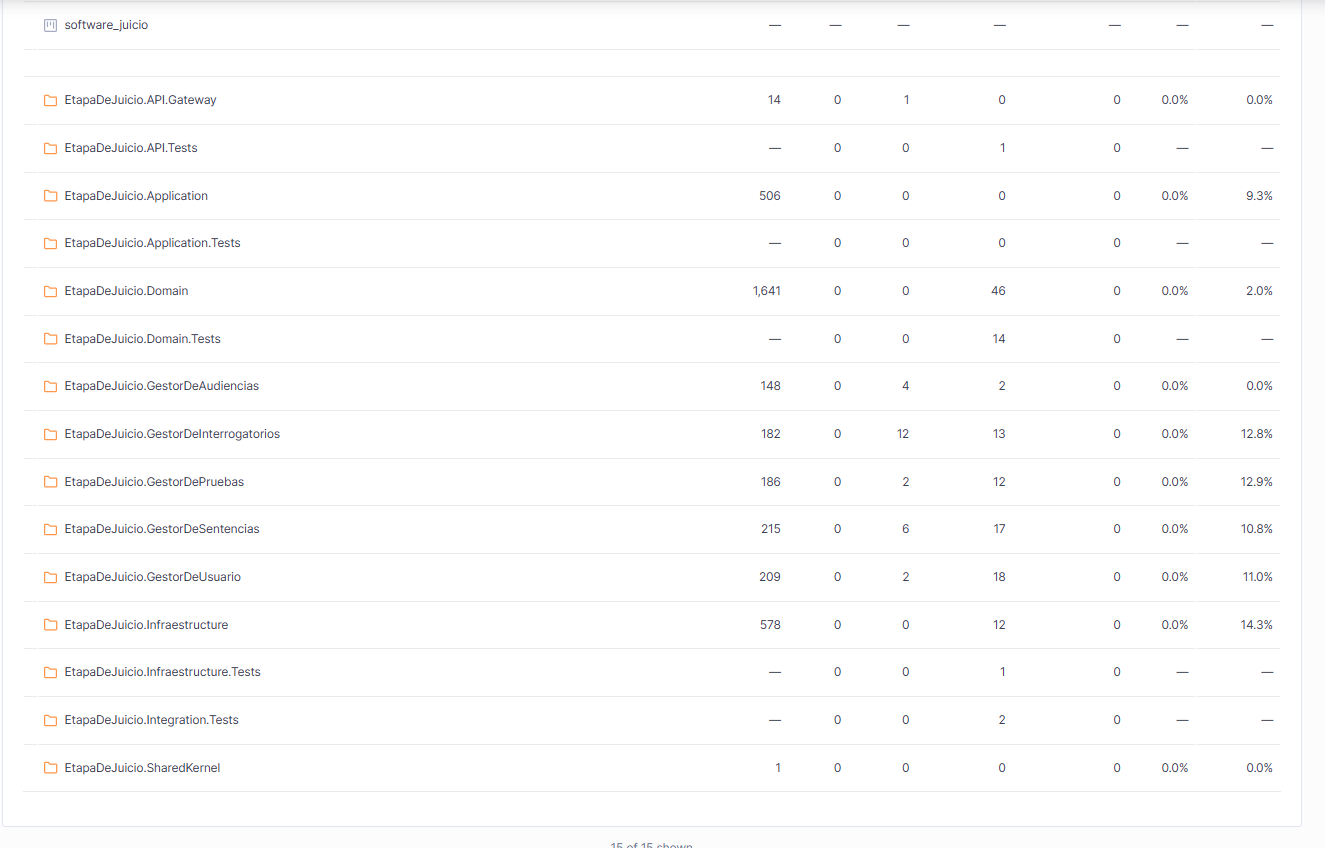
* 1. Herramientas a Utilizar
* **xUnit: Framework de pruebas principal para .NET**
* **Moq: Para la creación de mocks en pruebas unitarias**
* **FluentAssertions: Para aserciones más expresivas**
* **EF Core InMemory: Para pruebas de repositorios sin base de datos real**
* **Testcontainers: Para pruebas con dependencias containerizadas**
* **Coverlet: Para medición de cobertura de código**
  1. ****Configuración del Flujo CI/CD para TDD****
* **Pre-commit: Ejecución de pruebas unitarias y análisis estático de código**
* **CI Pipeline: Ejecución de pruebas unitarias, integración y análisis de cobertura**
* **Quality Gate: Verificación de cobertura mínima del 80% antes de aceptar PR**
* **Reporting: Generación de informes de pruebas y cobertura**
  1. Pruebas realizadas con SonarQube

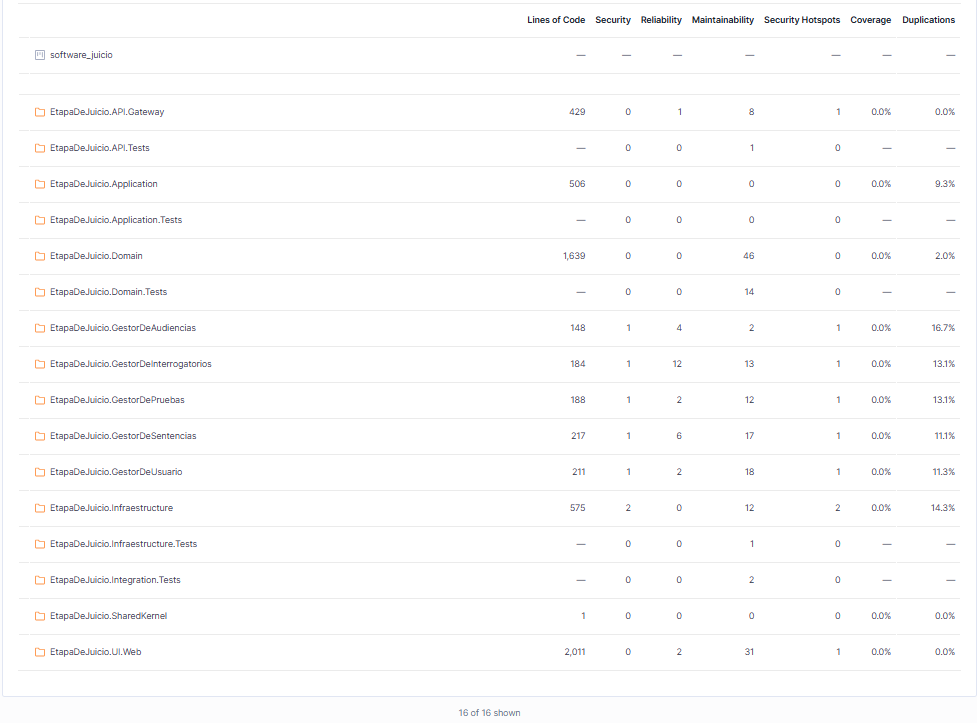


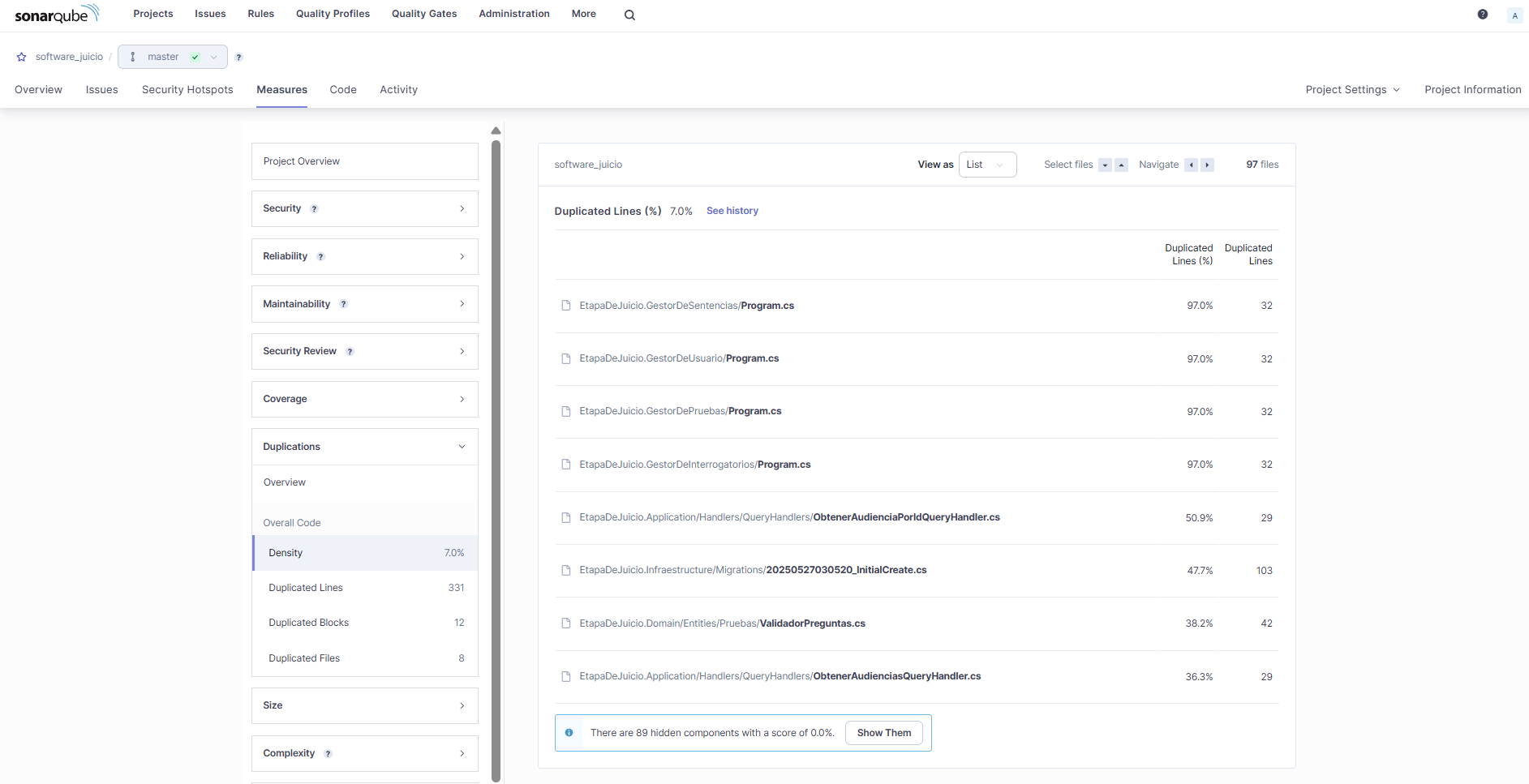


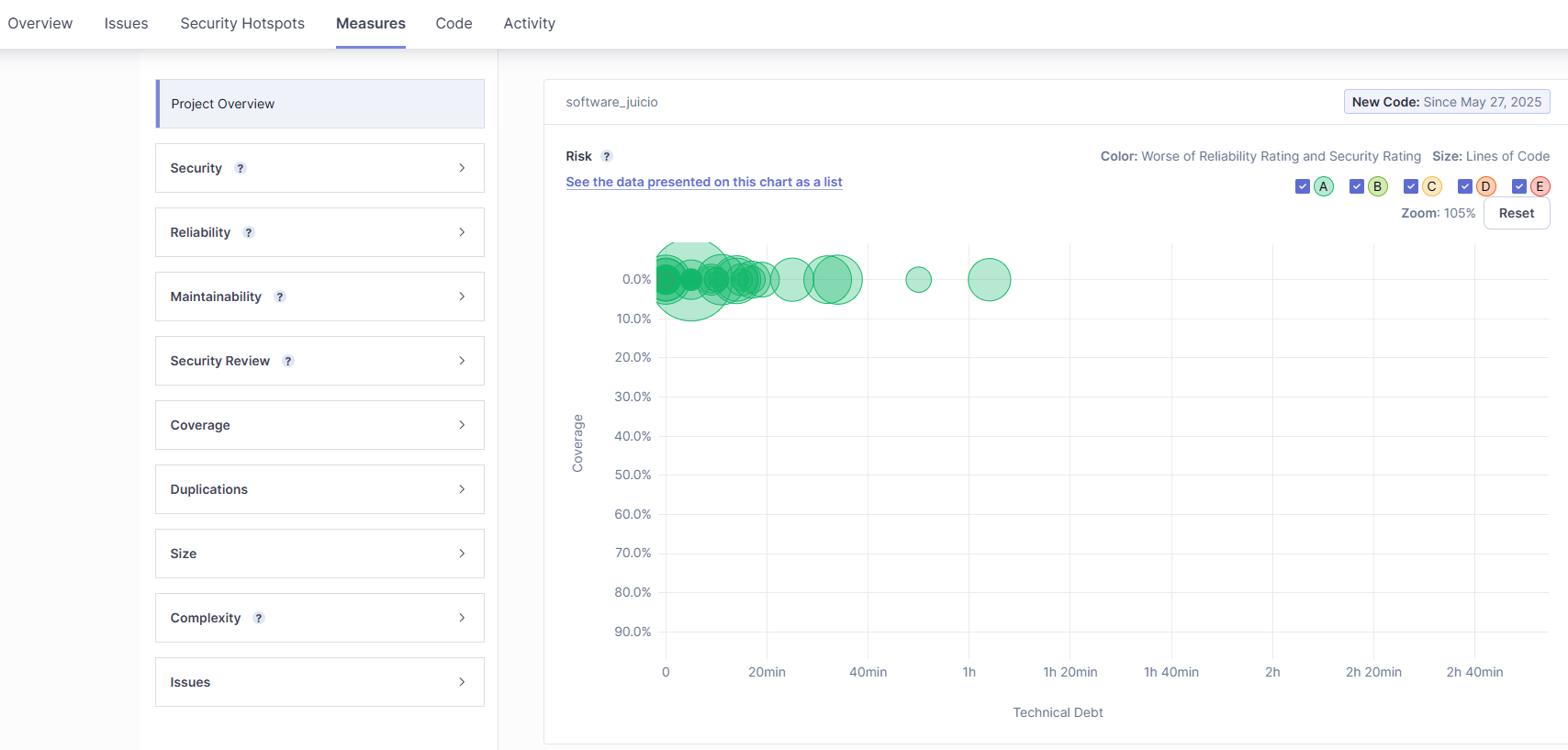


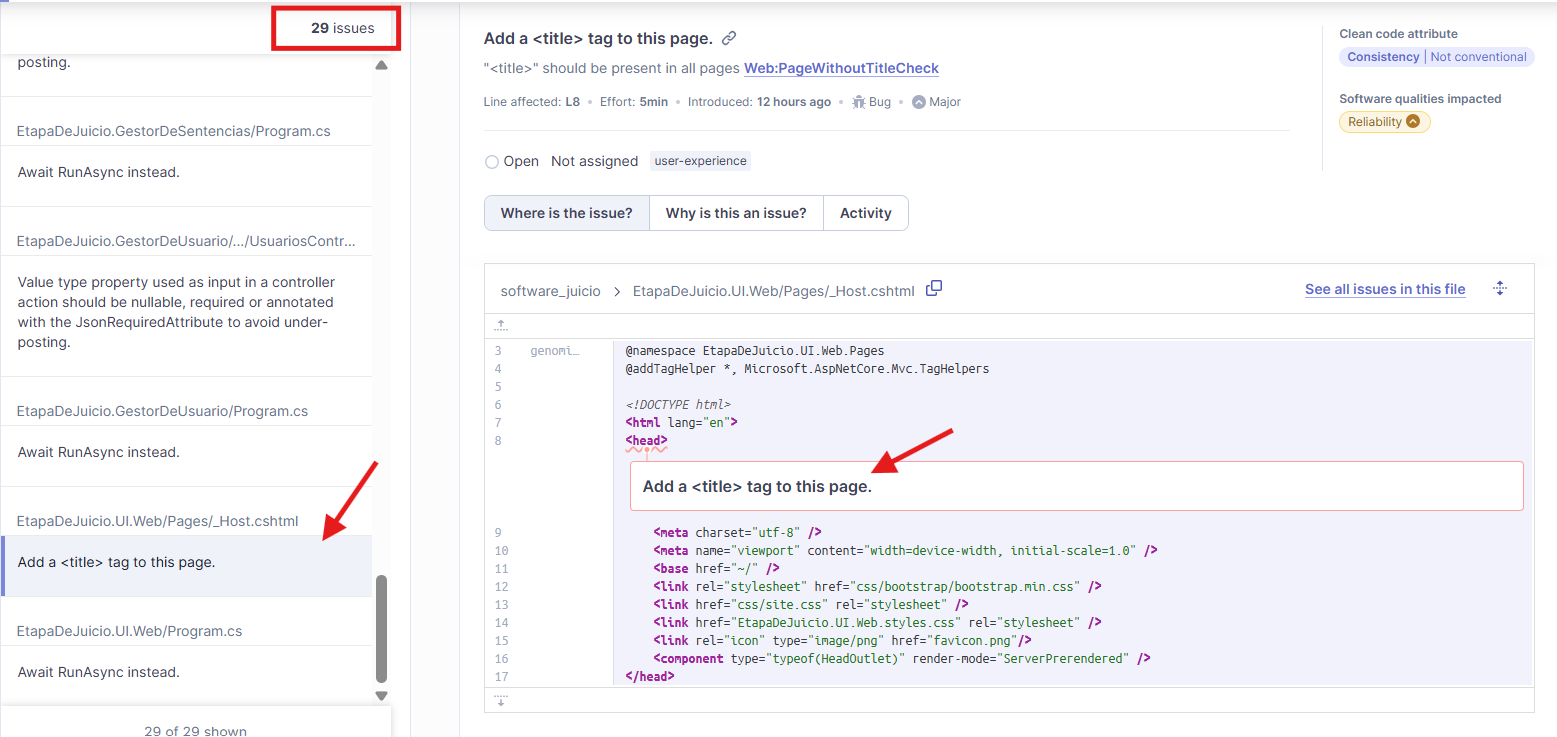


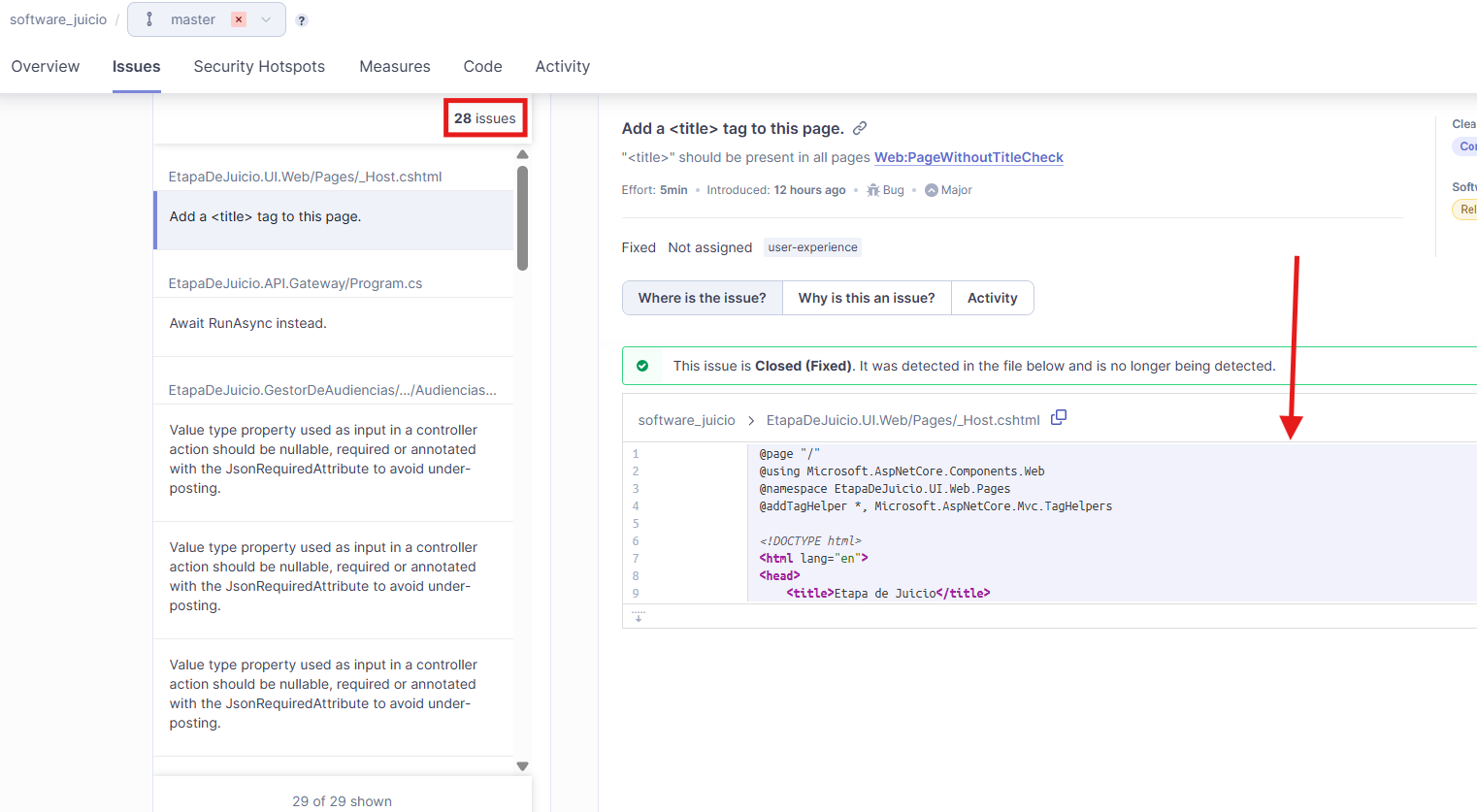












CONCLUSIONES

* Los estándares de codificación fueron acordados por todo el equipo y su cumplimiento será supervisado durante las revisiones de código. Se realizarán actualizaciones a este documento según sea necesario para adaptarse a los requisitos cambiantes del proyecto.
* Problemática y causas: El sistema judicial analizado enfrenta desafíos críticos como la obsolescencia tecnológica y la fragmentación de procesos, que perpetúan la impunidad y la desconfianza ciudadana (Ministerio de Justicia, 2023). La dependencia de expedientes físicos y la falta de integración entre instituciones son causas directas de estos problemas.
* Propuesta tecnológica: La implementación de un Expediente Fiscal Electrónico (EFE) con arquitectura basada en microservicios, blockchain para trazabilidad y herramientas como SonarQube para calidad de código, representa una solución viable (Rodríguez, 2022). Esta propuesta alinea con estándares internacionales como los principios de justicia abierta de la OEA (2021).
* Metodología y calidad: El enfoque TDD y las estrategias DevSecOps garantizan un desarrollo robusto y seguro. Las métricas de calidad específicas (ej., cobertura de código del 85%) y las pruebas no funcionales (rendimiento, seguridad) son esenciales para un sistema crítico como este (IEEE, 2020).
* Impacto social: La digitalización y transparencia en la etapa de juicio pueden reducir tiempos procesales y mejorar el acceso a la justicia, especialmente en zonas rurales (PNUD, 2023). Sin embargo, su éxito depende de la capacitación continua y la adaptación al contexto local.

RECOMENDACIONES

* Capacitación continua: Formar permanentemente a fiscales, jueces y operadores sobre el uso del nuevo sistema digital, con especial énfasis en seguridad y ética judicial.
* Marco normativo: Promover reformas legales que respalden el uso de tecnologías como blockchain y firmas digitales en procedimientos judiciales.
* Gobernanza del dato: Establecer una estrategia nacional de gobernanza de datos judiciales, con énfasis en interoperabilidad, protección de datos y auditoría ciudadana.
* Pilotaje y escalabilidad: Iniciar la implementación del EFE en distritos piloto, evaluando indicadores clave de desempeño antes de escalar a nivel nacional.
* Monitoreo de impacto: Crear una unidad técnica que evalúe el impacto del EFE en términos de reducción de tiempos procesales, percepción ciudadana y trazabilidad.

REFERENCIA BIBLIOGRÁFICA

* Beck, K. (2003). Test Driven Development: By Example. Addison-Wesley.
* Hall, B. (2020). xUnit Test Patterns in .NET. O'Reilly Media.
* IEEE. (2020). DevSecOps: Integrating security into DevOps. IEEE Software, 37(3), 20–25.
* Lewis, J., & Fowler, M. (2014). Microservices: a definition of this new architectural term. https://martinfowler.com/articles/microservices.html
* Organización de Estados Americanos (OEA). (2021). Justicia abierta en América Latina y el Caribe: Principios y mejores prácticas. https://www.oas.org
* Programa de las Naciones Unidas para el Desarrollo (PNUD). (2023). Acceso a la justicia en América Latina. https://www.undp.org
* Rodríguez, F. (2022). Arquitecturas resilientes para sistemas judiciales latinoamericanos. Revista Latinoamericana de Ingeniería de Software, 14(2), 77–92.
* Tapscott, D., & Tapscott, A. (2016). Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world. Penguin.