**Documentación Unificada del Sistema OCR Asíncrono Empresarial**

**1. Introducción**

**1.1. Presentación**

El presente documento consolida la arquitectura, el diseño y las especificaciones del "Sistema OCR Asíncrono Empresarial", un sistema automatizado de reconocimiento óptico de caracteres especializado en la extracción de datos de recibos de pagos móviles venezolanos. Este sistema se ha diseñado con un enfoque riguroso en la Integridad Total, la Perfección Continua y la Zero-Fault Detection, sirviendo como la verdad absoluta de lo que el sistema debe hacer y cómo debe comportarse.

**1.2. Objetivo del Sistema**

El objetivo primordial de este sistema es estructurar y entregar información de pagos móviles a sistemas downstream (n8n) de manera asíncrona y con capacidades espaciales avanzadas, garantizando la precisión operativa y la satisfacción del usuario final.

**1.3. Justificación**

La necesidad de este sistema surge de la complejidad inherente al procesamiento de recibos de pago en un entorno empresarial, donde la ambigüedad y los fallos pueden degradar la integridad de los datos y comprometer el rendimiento. Este enfoque sistemático y blindado es indispensable para asegurar un funcionamiento predecible y robusto, persiguiendo la excelencia en cada etapa del diseño y la ejecución.

**2. Filosofía y Principios Fundamentales**

La base inmutable de nuestro proyecto se asienta en una filosofía central que guía cada decisión de diseño y ejecución. Esta filosofía, detallada a continuación, establece los pilares de la Integridad Total, la Zero-Fault Detection, la Persistencia de Correcciones, la Interface Excellence, la Coherencia de Referencias Críticas y la Comprensión Profunda del Contexto de Dominio.

**LA FILOSOFÍA CENTRAL: EL ADN INMUTABLE DEL PROYECTO**

**"Integridad Total y Perfección Continua"**

**PRELUDIO Y CUESTIONAMIENTO FUNDAMENTAL:**

**¿Es realmente posible alcanzar la "Integridad Total" y la "Perfección Continua" en un sistema de software? ¿No es acaso un ideal inalcanzable, una utopía en un campo tan propenso al error humano y a los cambios constantes? La respuesta es que, si bien la perfección absoluta puede ser una meta móvil, la persecución implacable de la misma y la aceptación de que la "Integridad Total" no es un objetivo, sino un estado operacional no negociable, es lo que define nuestro enfoque. Esta filosofía no es una lista de deseos; es un mandato operativo, un contrato de calidad intrínseco con cada bit de información y cada ciclo de procesamiento. Representa la ética inquebrantable que debe guiar cada acción, cada decisión de diseño y cada línea de código. Es el ADN inmutable que asegura la supervivencia y la excelencia de nuestro sistema a través del tiempo y frente a cualquier desafío.**

**1. INTEGRIDAD TOTAL: El Santuario de los Datos y la Ejecución Atómica**

**Definición Profunda y Cuestionamiento:**

**La Integridad Total se erige como el valor supremo, una verdad absoluta dentro de nuestro sistema. No es una característica; es la condición de existencia. Significa que cada dato, desde su nacimiento hasta su archivado, es inviolable en su forma, contenido y relación. Pero, ¿cómo garantizamos esa inviolabilidad en un entorno donde los datos viajan, se transforman y se almacenan? A través de la ejecución atómica y transaccional. Cada operación que altera el estado del sistema —ya sea la creación de un registro, la actualización de un campo, la eliminación de un recurso o la ejecución de una parte de la lógica del OCR— debe ser un evento todo o nada. No puede haber estados intermedios inconsistentes, datos parciales o huellas de una operación fallida. Si un paso falla, la operación completa debe ser revertida (rollback) como si nunca hubiera ocurrido. Esto se extiende más allá de las bases de datos a cada función y servicio.**

**Cuestionamiento Interno: ¿Estamos forzando la atomicidad incluso en las operaciones más triviales? ¿Hemos identificado todos los puntos de posible inconsistencia (ej. después de una llamada a una API externa que falla a mitad de camino)? ¿Nuestra estrategia de compensación es robusta y automatizada?**

**Implicación para el Desarrollo y Ejecución (Más Allá de la Base de Datos):**

**Contratos de Datos Rigurosos: Validación implacable de esquemas de datos (JSON Schema, Pydantic, TypeScript) en cada límite (APIs, colas de mensajes, módulos internos). Rechazo inmediato y reportado de cualquier dato que no cumpla.**

**Transacciones Multicapa: Aplicación del concepto ACID no solo a la base de datos, sino a secuencias de operaciones en el código. Uso de patrones de Sagas para transacciones distribuidas o complejas que abarcan múltiples servicios, con lógicas de compensación claramente definidas.**

**Idempotencia Fundamental: Cada operación que pueda ser reintentada (subidas de archivos, llamadas a APIs externas) debe ser diseñada para producir el mismo resultado si se ejecuta múltiples veces. Esto es vital para la resiliencia en sistemas distribuidos.**

**Estado Inmutable por Defecto: Siempre que sea posible, preferir estructuras de datos inmutables. Si un estado debe cambiar, hacerlo a través de transformaciones controladas y explícitas que produzcan un nuevo estado válido.**

**2. ZERO-FAULT DETECTION: Inmunidad Activa al Error y Resiliencia Inquebrantable**

**Definición Profunda y Cuestionamiento:**

**Esta filosofía trasciende la mera corrección de errores; es una postura proactiva de inmunidad y contención. El sistema no espera que los errores ocurran para reaccionar; los previene activamente en cada punto de entrada y transformación. Esto es una "vacuna" constante. Cualquier dato, entrada o condición que no cumpla con las expectativas debe ser interceptado en el punto más temprano posible, sin ser forzado o propagado. No se trata de "manejar errores", sino de "evitar que los errores existan en primer lugar" o, si se materializan, de encapsularlos y aislarlos instantáneamente. La resiliencia implica que el sistema puede absorber fallos (internos o externos) sin comprometer su funcionalidad principal o su Integridad Total.**

**Cuestionamiento Interno: ¿Estamos realmente deteniendo los errores en su origen o los estamos dejando que se propaguen y esperando que una validación más adelante los capture? ¿Hemos identificado todos los puntos ciegos donde un dato anómalo podría colarse? ¿Nuestras caídas de sistema son manejadas con gracia o de forma abrupta?**

**Implicación para el Desarrollo y Ejecución:**

**Validación Omnipresente (Input/Output/Domain Validation): No solo validar las entradas externas, sino validar en cada límite de módulo, en cada transformación de datos, y en la salida de cada función. La validación debe ser granular y contextual.**

**Diseño Defensivo Extremo: Asumir que toda entrada es potencialmente maliciosa o incorrecta. Utilizar tipado estricto, assertions, y cláusulas de guarda (guard clauses) para verificar precondiciones y postcondiciones en cada función crítica.**

**Manejo Graceful de Excepciones: Las excepciones son eventos, no fallos catastróficos. Deben ser capturadas, registradas con detalle contextual, y el sistema debe continuar su operación de forma segura (ej., devolviendo un error claro al cliente, o utilizando valores por defecto seguros).**

**Circuit Breakers y Fallbacks: Para dependencias externas (otras APIs, bases de datos), implementar patrones de Circuit Breaker para evitar cascadas de fallos, y mecanismos de fallback para degradar la funcionalidad de forma controlada cuando un servicio externo no está disponible.**

**Monitoreo de Anomalías: No solo registrar errores, sino también monitorear desviaciones del comportamiento esperado (datos inesperados, latencias anormales, etc.) para una alerta proactiva.**

**3. PRUEBAS INTEGRALES Y PERSISTENCIA DE CORRECCIONES: El Ciclo de Retroalimentación Incesante**

**Definición Profunda y Cuestionamiento:**

**Nuestra aproximación a las pruebas no es una fase final, sino el pulso continuo del desarrollo. Cada corrección, cada mejora, cada nueva línea de código debe nacer con la expectativa de ser validada automáticamente y exhaustivamente. Las "correcciones" no son meros arreglos temporales; son soluciones arquitectónicas permanentes que se demuestran resistentes a la regresión. Esto implica una cultura de Test-Driven Development (TDD), donde las pruebas no solo validan el código, sino que también guían su diseño, y donde cada error pasado se convierte en un nuevo caso de prueba que vive para siempre. La Persistencia de Correcciones significa que un error resuelto no puede reaparecer.**

**Cuestionamiento Interno: ¿Estamos escribiendo pruebas solo para que pasen, o para que realmente validen el comportamiento esperado y los límites? ¿Cada error que encontramos se convierte en una nueva prueba para evitar su recurrencia? ¿Nuestras pruebas son rápidas, fiables y se ejecutan automáticamente en cada cambio? ¿Confiamos ciegamente en la IA para generar pruebas, o las revisamos críticamente?**

**Implicación para el Desarrollo y Ejecución:**

**Pirámide de Pruebas Robusta:**

**Unitarias (TDD): Cubren la lógica de negocio más pequeña y aislada de cada función/clase. Altamente automatizadas y rápidas. Guían el diseño.**

**Integración: Verifican la interacción entre módulos, servicios y bases de datos.**

**Componentes (para Frontend): Pruebas que validan un componente UI en aislamiento, simulando sus inputs y eventos.**

**Contract Testing: Aseguran que las APIs (backend y frontend) cumplen sus contratos.**

**Regresión: Un conjunto de pruebas críticas que se ejecutan constantemente para asegurar que los cambios no rompen funcionalidades existentes. Cada bug corregido se convierte en una prueba de regresión.**

**End-to-End (E2E): Flujos de usuario completos, probando la integración de todo el sistema.**

**Rendimiento/Carga: Crucial para tu prioridad de "ultra veloz". Simulan la demanda real para identificar cuellos de botella.**

**Seguridad: Pruebas de vulnerabilidad y penetración.**

**CI/CD Inquebrantable: Cualquier pull request que no pase las pruebas automatizadas (unidad, integración, regresión, linters) debe bloquear el despliegue. La automatización es el guardián de la Persistencia de Correcciones.**

**Cobertura de Código Inteligente: No solo cantidad, sino calidad. Asegurar que las líneas de código cubiertas son las críticas y las más propensas a errores.**

**4. INTERFACE EXCELLENCE: La Transparencia y Usabilidad Sistémica**

**Definición Profunda y Cuestionamiento:**

**La excelencia en la interfaz no se limita a una interfaz de usuario atractiva; se extiende a cada punto de interacción dentro y fuera del sistema: APIs, contratos de módulos, formatos de datos, mensajes de error y logs. Una interfaz excelente es intuitiva, predecible y a prueba de errores del consumidor, ya sea otro sistema, un desarrollador o un usuario final. Su transparencia permite entender rápidamente qué hace, cómo se usa y qué esperar. La ambigüedad es el enemigo de la excelencia de la interfaz.**

**Cuestionamiento Interno: ¿Es nuestra API tan fácil de usar como una interfaz de usuario bien diseñada? ¿Nuestros mensajes de error son crípticos o guían al desarrollador o usuario hacia una solución? ¿Estamos documentando activamente cada interfaz de forma que sea usable por un tercero sin contacto directo?**

**Implicación para el Desarrollo y Ejecución:**

**Diseño de API Contract-First: Definir el contrato de la API (ej., con OpenAPI/Swagger) antes de la implementación, para que sirva de guía inmutable.**

**Consistencia Absoluta: Nomenclatura uniforme, uso consistente de verbos HTTP, formatos de respuesta uniformes, y patrones de paginación/filtrado predecibles en todas las APIs.**

**Documentación Viva y Automática: Generar la documentación (para APIs, componentes UI, esquemas de datos) directamente del código o del contrato, asegurando que esté siempre actualizada y sea la fuente de verdad. Incluir ejemplos de uso y de errores.**

**Mensajes de Error Claros y Accionables: Los errores deben ser legibles por humanos y máquinas. Deben incluir un código de error, una descripción clara del problema y, si es posible, una sugerencia sobre cómo resolverlo o a qué se debe.**

**Versionado Explícito: Implementar estrategias de versionado (ej., v1, v2) para APIs o módulos para permitir la evolución sin romper la compatibilidad con clientes existentes.**

**5. COHERENCIA DE REFERENCIAS CRÍTICAS: El Tejido Inquebrantable de la Información**

**Definición Profunda y Cuestionamiento:**

**El sistema es una red de referencias: IDs únicos, relaciones entre entidades, punteros a recursos. La Coherencia de Referencias Críticas significa que cada referencia dentro del sistema debe ser válida, única y consistente en todo momento y en todas las capas. Una referencia a un ID inexistente, un dato huérfano o una relación inconsistente es una corrupción de la Integridad Total y debe ser imposible o, al menos, detectada y corregida de inmediato. Este es el tejido conectivo de nuestro software.**

**Cuestionamiento Interno: ¿Podría existir un registro en la base de datos que apunta a un usuario que ya no existe? ¿Nuestros identificadores son realmente únicos en el tiempo y en el espacio (distribución)? ¿Estamos manejando correctamente las eliminaciones en cascada o la actualización de referencias cuando un elemento cambia?**

**Implicación para el Desarrollo y Ejecución:**

**Identificadores Universales (UUIDs/GUIDs): Utilizar identificadores únicos globalmente para recursos, eventos, transacciones. Esto simplifica la trazabilidad en sistemas distribuidos y reduce el riesgo de colisiones.**

**Integridad Referencial de Base de Datos: Imponer restricciones de claves foráneas a nivel de base de datos (FOREIGN KEY) para asegurar que las relaciones sean siempre válidas.**

**Validación de Referencias a Nivel de Aplicación: Antes de ejecutar operaciones que dependan de referencias (ej., "buscar el documento con ID X"), verificar la existencia y validez de dicho ID a nivel de código.**

**Estrategias de Concurrencia y Consistencia: Implementar bloqueos optimistas/pesimistas para manejar accesos concurrentes a recursos críticos y asegurar que las actualizaciones de referencias se hagan de forma atómica.**

**Trazabilidad Completa (Correlation IDs): Asegurar que cada solicitud o transacción se pueda rastrear de principio a fin a través de todos los microservicios y componentes, usando IDs de correlación. Esto ayuda a depurar y verificar la coherencia.**

**6. COMPRENSIÓN PROFUNDA DEL CONTEXTO DE DOMINIO: La Inteligencia Localizada y de Negocio**

**Definición Profunda y Cuestionamiento:**

**El software no es solo lógica, es una representación viva del negocio. Esta filosofía exige que la aplicación no solo manipule datos, sino que comprenda la semántica profunda, las reglas de negocio, las normativas y las heurísticas implícitas del dominio en el que opera. Implica que el software puede inferir intenciones, resolver ambigüedades y tomar decisiones alineadas con los objetivos de negocio, incluso cuando los datos brutos no son explícitos. Es la capacidad de trascender el "cómo" y entender el "por qué" y el "qué significa".**

**Cuestionamiento Interno: ¿Nuestro software simplemente procesa datos, o realmente "entiende" lo que está procesando (ej., un número en un recibo es un monto total, no solo un número)? ¿Estamos capturando la "sabiduría" del experto de negocio en el código? ¿Qué sucede cuando una regla de negocio cambia; el software puede adaptarse fácilmente o requiere una reescritura compleja?**

**Implicación para el Desarrollo y Ejecución:**

**Modelado de Dominio Rico (Domain-Driven Design - DDD):**

**Lenguaje Ubicuo: Crear un vocabulario común entre desarrolladores y expertos de negocio, reflejándolo directamente en el código (nombres de clases, funciones).**

**Objetos de Valor, Entidades, Agregados: Modelar el dominio con precisión, encapsulando la lógica de negocio y las reglas de consistencia dentro de estos objetos.**

**Externalización de Reglas de Negocio: Para reglas volátiles, considerar motores de reglas (Rule Engines) que permitan a los expertos de negocio modificarlas sin desplegar código nuevo.**

**Fuentes de Verdad de Negocio: Integrar y confiar en sistemas de datos maestros o catálogos que definan las verdades del negocio (ej., una lista de tipos de documentos válidos, rangos de montos, etc.).**

**Feedback Loop Constante: Mantener una comunicación fluida y un ciclo de retroalimentación continuo con los usuarios finales y stakeholders del negocio para refinar la comprensión del dominio y asegurar que el software cumpla con las necesidades reales.**

**Telemetría y Análisis de Uso: Recopilar datos sobre cómo se usa el sistema para identificar comportamientos inesperados o áreas donde la comprensión del dominio podría ser deficiente.**

**3. Modelo de Dominio Conceptual**

Para asegurar una comprensión unificada y precisa de las entidades de negocio clave, se ha definido un modelo de dominio conceptual. Este modelo, que se adapta a la naturaleza temporal de los datos en nuestro sistema y la dependencia de n8n como fuente de verdad externa, describe los Documentos de Pago, Lotes, Resultados OCR, Resultados Consolidados del Lote, Sorteos, Usuarios y Reglas de Extracción.

**MODELO DE DOMINIO CONCEPTUAL (FINAL Y BLINDADO)**

**Este modelo conceptual define las entidades clave del negocio, sus atributos relevantes, sus relaciones y las reglas de negocio que las rigen, bajo el principio de Comprensión Profunda del Contexto de Dominio. Se ha adaptado para reflejar que las entidades se manejan en memoria o temporalmente en archivos, y que n8n es la Fuente de Verdad Externa para la persistencia a largo plazo de los resultados.**

**1. Entidad: Documento de Pago**

| **Atributo Clave** | **Descripción** | **Notas / Principios Asociados** |
| --- | --- | --- |
| **Descripción Conceptual** | **Recibo oficial de un pago móvil (imagen) que actúa como insumo principal para la extracción de datos. Su procesamiento depende fundamentalmente de un JSON de entrada externo (metadatosEntrada) que proporciona los metadatos para su identificación y renombrado. Su información es procesada de forma temporal en el sistema y los resultados se envían inmediatamente a n8n, que es la Fuente de Verdad Externa. La vinculación de los resultados individuales con el Documento de Pago se logra a través del nombreRenombradoArchivo y la metadatosEntrada, asegurando la trazabilidad completa para n8n.** | **Claridad de Definiciones, Rendimiento del Servicio, Fuente de Verdad Externa** |
| **idDocumento** | **Identificador único del documento (ej. UUID).** | **Interno del sistema para identificación única.** |
| **tipoOrigen** | **WEB o APP (para diferenciación de formato y posible impacto en Reglas de Extracción).** | **Ayuda a la Comprensión Profunda del Contexto de Dominio del documento.** |
| **fechaCreacion** | **Fecha y hora de cuándo fue creado el recibo digital (del metadato WhatsApp).** | **Trazabilidad.** |
| **rutaAlmacenamientoOriginal** | **Ruta completa del archivo original antes de cualquier procesamiento. Se almacena en CO-06 (Servicio de Almacenamiento de Archivos).** | **Integridad Total (para auditoría), Persistencia Temporal (en CO-06).** |
| **nombreOriginalArchivo** | **Nombre original del archivo antes de renombrar.** | **Trazabilidad.** |
| **nombreRenombradoArchivo** | **Nombre del archivo después de aplicar el esquema de renombrado ([idSorteo]\_[idWhatsapp]\_[nombre\_sanitizado]\_[fechasorteo\_YYYYMMDD]\_[horaminuto]\_[numerollegada].[extension\_original]). Este nombre es la verdad absoluta y debe coincidir con la información provista en metadatosEntrada para ese documento específico.** | **Claridad de Definiciones, Coherencia de Referencias Críticas, Trazabilidad.** |
| **extensionArchivo** | **Extensión original del archivo (ej. jpg, png).** |  |
| **metadatosEntrada** | **Objeto JSON completo provisto por n8n que contiene los metadatos contextuales que acompañan a la imagen (ej. idWhatsapp, caption, nombre, horaMinutoN8nIngreso, numeroLlegada, idSorteo, fechaSorteo, etc.). Es la Fuente de Verdad para los metadatos iniciales del documento.** | **Claridad de Definiciones, Fuente de Verdad Externa.** |
| **Relaciones Clave** | **- Parte de Lote (N Documentos de Pago : 1 Lote).** | **Modularidad, Granularidad.** |
|  | **- Tiene Resultado OCR (1 Documento de Pago : 1 Resultado OCR).** | **Trazabilidad.** |
| **Reglas de Negocio / Restricciones** | **- El procesamiento es asíncrono y se prioriza el Rendimiento del Servicio.** | **Rendimiento del Servicio.** |
|  | **- Los resultados son efímeros y se envían inmediatamente a n8n.** | **Persistencia Temporal, Acoplamiento Débil.** |
|  | **- La metadatosEntrada debe contener todos los campos necesarios para la correcta identificación y procesamiento del Documento de Pago y su posterior reconstrucción contextual en la Fuente de Verdad Externa (n8n).** | **Integridad Total, Claridad de Definiciones.** |

**2. Entidad: Lote**

| **Atributo Clave** | **Descripción** | **Notas / Principios Asociados** |
| --- | --- | --- |
| **Descripción Conceptual** | **Una agrupación lógica de Documentos de Pago que comparten un identificador de sorteo, fecha y hora. En este sistema, los lotes se gestionan y orquestan EN MEMORIA dentro del API Gateway / Backend Service (CO-01) para su procesamiento temporal e inmediato. No se persisten en una base de datos interna, y su estado es efímero. Su Fuente de Verdad Externa y la gestión a largo plazo de su contenido (Documentos de Pago y Resultados OCR) es responsabilidad de n8n. Al finalizar el procesamiento de un lote, el API Gateway (CO-01) genera y envía a n8n un Resultado Consolidado del Lote (ver nuevo término Resultado Consolidado del Lote).** | **Modularidad, Granularidad, Rendimiento del Servicio, Persistencia Temporal, Fuente de Verdad Externa, Acoplamiento Débil, Claridad de Definiciones** |
| **idSorteo** | **El identificador único del sorteo (ej. 20250712-A).** | **Coherencia de Referencias Críticas.** |
| **fechaSorteo** | **La fecha en que se realiza el sorteo (ej. YYYY-MM-DD). Se deriva del idSorteo.** | **Coherencia de Referencias Críticas.** |
| **horaMinutoSorteo** | **La hora y minuto específicos del sorteo (ej. HHMM).** | **Coherencia de Referencias Críticas.** |
| **estadoLote** | **Estado temporal y EN MEMORIA del lote dentro de CO-01 (ej., PENDIENTE\_LLENADO, LISTO\_PROCESAMIENTO, PROCESADO\_COMPLETO, PROCESADO\_PARCIAL, FALLIDO).** | **Zero-Fault Detection (para manejo de estado), Persistencia Temporal.** |
| **timestampCreacion** | **Marca de tiempo de cuándo se inició el procesamiento del lote en CO-01.** | **Para gestión interna del API Gateway (caducidad, timeouts, Monitoreo por Pulso).** |
| **Relaciones Clave** | **- Contiene Documento de Pago (1 Lote : N Documentos de Pago).** | **Modularidad.** |
|  | **- Genera Resultado Consolidado del Lote (1 Lote : 1 Resultado Consolidado del Lote).** | **Coherencia de Referencias Críticas.** |
| **Reglas de Negocio / Restricciones** | **- El estado de un Lote es temporal y es gestionado EN MEMORIA por el API Gateway / Backend Service (CO-01).** | **Rendimiento del Servicio, Persistencia Temporal.** |
|  | **- La notificación del estado final de un Lote (a través del Resultado Consolidado del Lote) se realiza a n8n (AE-02), que es responsable de su registro histórico.** | **Fuente de Verdad Externa, Acoplamiento Débil.** |
|  | **- Una vez que el Resultado Consolidado del Lote ha sido exitosamente enviado a n8n, la información del Lote asociada y sus Documentos de Pago individuales pueden ser eliminados de la memoria del API Gateway (CO-01) y, si aplica, de los archivos temporales de CO-06.** | **Rendimiento del Servicio, Eficiencia de Recursos.** |
|  | **- El sistema debe manejar la acumulación EN MEMORIA de los Documentos de Pago y sus metadatosEntrada asociados hasta que el Lote esté completo y listo para ser enviado al OCR Worker Service (CO-04).** | **Rendimiento del Servicio, Modularidad.** |

**3. Entidad: Resultado OCR**

| **Atributo Clave** | **Descripción** | **Notas / Principios Asociados** |
| --- | --- | --- |
| **Descripción Conceptual** | **El objeto de datos estructurados final que se extrae de un Documento de Pago por el OCR Worker Service. Este resultado es enriquecido y consolidado, conteniendo la metadatosEntrada original del documento, los valores extraídos por el OCR (ej., referencia, monto, fecha), y los resultados del Procesamiento Espacial y Validación OCR. Una vez generado, es enviado inmediatamente a n8n (AE-02). No se persiste a largo plazo en el sistema local, asegurando el Rendimiento del Servicio. Se integra en el Resultado Consolidado del Lote.** | **Claridad de Definiciones, Rendimiento del Servicio, Persistencia Temporal, Fuente de Verdad Externa, Integridad Total, Trazabilidad, Coherencia de Referencias Críticas, Precisión** |
| **referenciaPago** | **Referencia alfanumérica del pago.** | **Comprensión Profunda del Contexto de Dominio.** |
| **montoPago** | **Monto numérico del pago.** | **Comprensión Profunda del Contexto de Dominio.** |
| **fechaPago** | **Fecha de emisión del pago.** | **Comprensión Profunda del Contexto de Dominio.** |
| **horaPago** | **Hora de emisión del pago.** | **Comprensión Profunda del Contexto de Dominio.** |
| **bancoBeneficiario** | **Nombre del banco donde se realizó el pago.** | **Comprensión Profunda del Contexto de Dominio.** |
| **confianzaGlobalOCR** | **Métrica interna del sistema que indica el nivel de fiabilidad promedio de los datos extraídos.** | **Zero-Fault Detection, Precisión.** |
| **estadoValidacion** | **Estado del resultado tras aplicar reglas de negocio (VALIDADO, REQUiere\_REVISION, FALLIDO\_VALIDACION).** | **Zero-Fault Detection.** |
| **nombreRenombradoArchivoAsociado** | **El nombreRenombradoArchivo del Documento de Pago del que se deriva este resultado, para trazabilidad.** | **Coherencia de Referencias Críticas, Trazabilidad.** |
| **metadatosEntradaOriginales** | **Copia de los metadatosEntrada recibidos para el Documento de Pago correspondiente, incluidos en el objeto Resultado OCR enviado a n8n.** | **Trazabilidad, Claridad de Definiciones.** |
| **resultadosExtraccion** | **Objeto con los valores específicos extraídos según las Reglas de Extracción.** | **Granularidad.** |
| **resultadosProcesamientoEspacial** | **Objeto con datos de coordenadas y lógica espacial aplicada.** | **Precisión.** |
| **Relaciones Clave** | **- Se deriva de Documento de Pago (1 Resultado OCR : 1 Documento de Pago).** | **Coherencia de Referencias Críticas.** |
|  | **- Se integra en Resultado Consolidado del Lote.** | **Modularidad.** |
| **Reglas de Negocio / Restricciones** | **- La Fuente de Verdad Externa y la persistencia a largo plazo de los Resultados OCR recaen exclusivamente en n8n (AE-02) y su Base de Datos Externa (AE-03).** | **Acoplamiento Débil, Fuente de Verdad Externa.** |
|  | **- Cada Resultado OCR debe estar directamente vinculado a un Documento de Pago original a través de su nombreRenombradoArchivo.** | **Integridad Total, Trazabilidad.** |

**4. Entidad: Resultado Consolidado del Lote**

| **Atributo Clave** | **Descripción** | **Notas / Principios Asociados** |
| --- | --- | --- |
| **Descripción Conceptual** | **Un objeto de datos generado por el API Gateway / Backend Service (CO-01) al finalizar el procesamiento de un Lote. Contiene un resumen del lote (ej., idLote), los Resultados OCR enriquecidos de todos los Documentos de Pago que lo componen, y métricas de rendimiento (Monitoreo por Pulso) como tiempo de procesamiento total, latencia promedio por imagen, y estado general del lote (ej., procesadoOk, falloParcial). Su destino es n8n para persistencia y análisis.** | **Claridad de Definiciones, Monitoreo por Pulso, Fuente de Verdad Externa, Integridad Total, Rendimiento del Servicio** |
| **idLote** | **Identificador del lote al que pertenece este resultado consolidado.** | **Coherencia de Referencias Críticas.** |
| **resultadosOCRDetalle** | **Array de objetos Resultado OCR enriquecidos para todos los documentos del lote.** | **Granularidad, Integridad Total.** |
| **tiempoTotalProcesamiento** | **Duración total en segundos para procesar el lote completo.** | **Monitoreo por Pulso, Rendimiento del Servicio.** |
| **imagenesProcesadasConExito** | **Número de imágenes dentro del lote que fueron procesadas sin errores significativos.** | **Zero-Fault Detection, Monitoreo por Pulso.** |
| **imagenesConFallo** | **Número de imágenes dentro del lote que tuvieron fallos de procesamiento.** | **Zero-Fault Detection, Monitoreo por Pulso.** |
| **estadoFinalLote** | **Estado final del procesamiento del lote (ej., COMPLETO, FALLO\_PARCIAL, FALLO\_TOTAL).** | **Zero-Fault Detection.** |
| **latenciaPromedioPorImagen** | **Latencia promedio por cada imagen procesada dentro del lote.** | **Rendimiento del Servicio, Monitoreo por Pulso.** |
| **detallesErrores** | **Objeto o array que describe cualquier error o anomalía a nivel de lote o documento.** | **Zero-Fault Detection, Persistencia de Correcciones.** |
| **Relaciones Clave** | **- Se deriva de Lote (1 Resultado Consolidado del Lote : 1 Lote).** | **Coherencia de Referencias Críticas.** |
|  | **- Es enviado a n8n (AE-02, AE-03) para persistencia.** | **Fuente de Verdad Externa, Acoplamiento Débil.** |
| **Reglas de Negocio / Restricciones** | **- Debe ser lo suficientemente completo para que n8n realice su función de persistencia y análisis histórico.** | **Integridad Total, Claridad de Definiciones.** |
|  | **- Su generación y envío desencadena la eliminación de la información temporal del Lote en CO-01.** | **Rendimiento del Servicio, Eficiencia de Recursos.** |

**5. Entidad: Sorteo**

| **Atributo Clave** | **Descripción** | **Notas / Principios Asociados** |
| --- | --- | --- |
| **Descripción Conceptual** | **Una entidad lógica que representa un evento de sorteo con un identificador, fecha y hora.** | **Comprensión Profunda del Contexto de Dominio** |
| **idSorteo** | **Identificador único del sorteo (ej. 20250712-A).** |  |
| **fechaSorteo** | **Fecha de realización del sorteo.** |  |
| **horaMinutoSorteo** | **Hora y minuto específicos del sorteo.** |  |
| **Relaciones Clave** | **- Asociado a Documento de Pago (1 Sorteo : N Documentos de Pago).** |  |
|  | **- Asociado a Lote (1 Sorteo : N Lotes).** |  |
| **Reglas de Negocio / Restricciones** | **- Entidad lógica, no persistida localmente. Su información es externa.** | **Acoplamiento Débil, Fuente de Verdad Externa** |

**6. Entidad: Usuario (que envía el pago)**

| **Atributo Clave** | **Descripción** | **Notas / Principios Asociados** |
| --- | --- | --- |
| **Descripción Conceptual** | **La persona o entidad que genera y envía el Documento de Pago al sistema, identificada por su WhatsApp.** | **Comprensión Profunda del Contexto de Dominio** |
| **idWhatsapp** | **El número de WhatsApp único del usuario (identificador principal).** |  |
| **nombreUsuario** | **Nombre asociado al número de WhatsApp.** |  |
| **Relaciones Clave** | **- Envía Documento de Pago (1 Usuario : N Documentos de Pago).** |  |
| **Reglas de Negocio / Restricciones** | **- Entidad lógica, no persistida localmente. Su información es externa (proviene de n8n).** | **Acoplamiento Débil, Fuente de Verdad Externa** |

**7. Entidad: Reglas de Extracción**

| **Atributo Clave** | **Descripción** | **Notas / Principios Asociados** |
| --- | --- | --- |
| **Descripción Conceptual** | **Definiciones configurables que instruyen al OCR Worker Service sobre cómo extraer datos específicos de un Documento de Pago y cómo validarlos.** | **Claridad de Definiciones, Granularidad, Comprensión Profunda del Contexto de Dominio** |
| **idRegla** | **Identificador único de la regla.** |  |
| **nombreRegla** | **Nombre descriptivo de la regla (ej. "Extracción de Monto BDV", "Validación de Referencia BANFANB").** |  |
| **criterioAplicacion** | **Describe el criterio por el cual esta regla se aplica a un documento (ej., "Identificado como BDV", "Texto genérico de pago móvil", "Formato de fecha X"). Permite flexibilidad y no limita a un banco específico.** | **Precisión, Comprensión Profunda del Contexto de Dominio.** |
| **definicionLogica** | **La representación de la lógica de extracción (ej. patrones Regex, coordenadas, dependencias de campos, lógica de validación). Corresponde a lo que está en config/extraction\_rules.json.** | **Zero-Fault Detection, Granularidad.** |
| **fechaCreacion** | **Fecha de creación de la regla.** |  |
| **fechaUltimaModificacion** | **Fecha de la última modificación.** |  |
| **activa** | **Booleano, indica si la regla está actualmente en uso.** |  |
| **Relaciones Clave** | **- Afecta Procesamiento OCR.** |  |
|  | **- Determina la estructura y el contenido de Resultado OCR.** | **Coherencia de Referencias Críticas.** |
|  | **- Puede estar asociada a un tipoOrigen de Documento de Pago.** |  |
| **Reglas de Negocio / Restricciones** | **- El sistema debe cargar y aplicar las Reglas de Extracción más recientes y activas.** | **Rendimiento del Servicio.** |
|  | **- Los cambios en las reglas deben ser auditables y no romper funcionalidades existentes (Zero-Fault Detection).** | **Persistencia de Correcciones.** |
|  | **- Las reglas deben ser lo suficientemente Granulares para manejar variaciones específicas de documentos (ej. diferentes versiones de recibos, o recibos que no son de un banco específico pero comparten patrones).** | **Granularidad, Precisión.** |
|  | **- Son el motor de la Extracción Espacial y la Precisión.** | **Claridad de Definiciones.** |

**4. Actores y Contexto del Sistema**

La interacción del sistema con su entorno y la definición de sus componentes principales se visualizan a través del diagrama de contexto y el diagrama de contenedores/componentes. Estos diagramas establecen los límites del sistema, identifican a los actores externos clave y detallan las responsabilidades y tecnologías de los módulos internos.

**DIAGRAMA DE CONTEXTO (Validado y Blindado - Con Interacciones Adicionales)**

**Sistema Principal (La "Caja Negra"):**

* **Nombre del Sistema:** Sistema OCR Asíncrono Empresarial
* **Descripción Concisa:** Sistema automatizado de reconocimiento óptico de caracteres especializado en la extracción de datos de recibos de pagos móviles venezolanos, con procesamiento asíncrono y capacidades espaciales avanzadas, para estructurar y entregar información a sistemas downstream (n8n).

**Actores Externos Clave:**

| ID Actor | Nombre del Actor Externo (Humano o Sistema) | Tipo de Actor (Humano / Sistema Externo) | Descripción/Rol en el Contexto del Sistema |
| --- | --- | --- | --- |
| AE-01 | Usuario Final (Operador/Administrador) | Humano | Usuario que interactúa directamente con el Frontend (Dashboard) del sistema para cargar Documentos de Pago, monitorear el Estado Lote y revisar Resultados OCR. Su objetivo es supervisar las operaciones y puede hacer pruebas dentro del frontend. |
| AE-02 | Sistema n8n | Sistema Externo | Orquestador de flujos de trabajo que envía Documentos de Pago (imágenes con metadatosEntrada) al sistema OCR y recibe los Resultados OCR procesados para su posterior carga en bases de datos. |
| AE-03 | Base de Datos Externa (de n8n) | Sistema Externo | Sistema de persistencia final donde n8n almacenará los Resultados OCR una vez procesados y validados. Aunque n8n es el intermediario, es un destino clave de nuestra información. |

**Interacciones Clave con el Sistema (La "Caja Negra"):**

Hemos identificado dos nuevas interacciones:

1. **Observación en Tiempo Real del Usuario Final sobre lotes iniciados por n8n:** Esto es una extensión de IN-02, pero especifica el origen de los lotes (n8n).
2. **Solicitud de n8n sobre el estado de un lote:** Esto es una nueva interacción de consulta.

Vamos a detallarlas:

| ID Interacción | Actor Externo Fuente/Destino | Dirección del Flujo (Hacia el Sistema / Desde el Sistema / Bidireccional) | Descripción de la Interacción | Datos/Información Clave Transferida (de Alto Nivel) | Principios Asociados que Refuerza |
| --- | --- | --- | --- | --- | --- |
| IN-01 | Usuario Final (AE-01) | Hacia el Sistema | **Carga de Documentos de Pago:** El usuario carga Documentos de Pago (imágenes) a través del Frontend del sistema, a menudo en Lotes, junto con metadatosEntrada. | Documento de Pago (imagen), metadatosEntrada (JSON). | Integridad Total, Claridad de Definiciones |
| IN-02 | Usuario Final (AE-01) | Desde el Sistema | **Monitoreo y Revisión de Resultados (de lotes cargados por el usuario):** El usuario consulta el estado de los Lotes que él mismo ha cargado, el progreso de los Procesamientos OCR y los Resultados OCR (incluyendo confianzaGlobalOCR y discrepancias) a través del Frontend. También puede descargar originales o resultados. | Estado Lote, Estado Procesamiento OCR, Resultado OCR (detalles), URLs de archivos. | Rendimiento del Servicio, Zero-Fault Detection |
| IN-03 | Sistema n8n (AE-02) | Hacia el Sistema | **Envío de Documentos para Procesamiento:** n8n envía Documentos de Pago (imágenes con metadatosEntrada) al sistema OCR, incluyendo los metadatosEntrada necesarios para la identificación, renombrado y contextualización del documento dentro de un Lote. (Como se vio en el POST /api/upload del INFORME 3). | Documento de Pago (imagen), metadatosEntrada (JSON con idWhatsapp, idSorteo, fechaSorteo, horaMinutoN8nIngreso, numeroLlegada, etc.). | Acoplamiento Débil, Integridad Total |
| IN-04 | Sistema n8n (AE-02) | Desde el Sistema | **Notificación de Resultados OCR (por documento):** El sistema OCR, al finalizar el Procesamiento OCR de un Documento de Pago y generar el Resultado OCR, lo envía de vuelta a n8n mediante un webhook. (Como se vio en el POST /api/webhook/ocr\_results del INFORME 3). | Resultado OCR (JSON estructurado con referenciaPago, montoPago, fechaPago, horaPago, bancoBeneficiario, confianzaGlobalOCR, etc.), idDocumentoAsociado, idLote. | Interface Excellence, Rendimiento del Servicio |
| IN-05 | Sistema n8n (AE-02) | Desde el Sistema | **Notificación de Estado de Lote/Errores:** El sistema OCR puede notificar a n8n sobre el Estado Lote (Completado, FallidoParcial, FallidoTotal) o errores críticos que impidan el procesamiento. | Resultado Consolidado del Lote (JSON), incluyendo: - idLote  - Resultados OCR enriquecidos de todos los Documentos de Pago del lote  - Métricas de rendimiento (tiempoTotalProcesamiento, latenciaPromedioPorImagen, etc.)  - Estado final del lote (COMPLETO, FALLO\_PARCIAL, FALLO\_TOTAL)  - Detalles de errores (si aplica) | Zero-Fault Detection, Persistencia de Correcciones |
| **IN-06** | **Usuario Final (AE-01)** | **Desde el Sistema** | **Monitoreo en Tiempo Real de Lotes Externos:** El usuario consulta el Estado Lote y el progreso de los Procesamientos OCR de lotes que fueron **iniciados por el Sistema n8n**, a través del Frontend del sistema. Esto permite una supervisión unificada. | Estado Lote (actualizaciones en tiempo real), Estado Procesamiento OCR (individual/lote). | Rendimiento del Servicio, Claridad de Definiciones |
| **IN-07** | **Sistema n8n (AE-02)** | **Bidireccional** | **Consulta del Estado de Lote y Recuperación de Información:** n8n consulta el sistema OCR para verificar si un Lote específico ya ha sido procesado y, en caso afirmativo, solicita la información o Resultados OCR asociados a ese lote. Esto es una interacción de pull, complementaria a la notificación (push) de IN-04/IN-05. | **Hacia el Sistema:** idLote (o idLoteExterno). **Desde el Sistema:** estadoLote, resultadosOCR (todos los documentos del lote), mensajeError (si aplica), confianzaGlobalOCR consolidada. | Acoplamiento Débil, Resiliencia, Integridad Total |
| IN-08 | Base de Datos Externa (AE-03) | Hacia el Sistema | **Almacenamiento de Resultados OCR:** El Sistema n8n (que opera fuera de nuestro scope directo) envía los Resultados OCR validados a la Base de Datos Externa para persistencia a largo plazo y uso posterior. Esta interacción representa el destino final de la información procesada por nuestro sistema. | Resultado OCR (JSON estructurado), metadatosEntrada (del documento original para trazabilidad). | Coherencia de Referencias Críticas (con n8n), Integridad Total |

**DIAGRAMA DE CONTENEDORES/COMPONENTES (VERSIÓN FINAL Y BLINDADA)**

| ID Contenedor | Nombre del Contenedor (Componente Mayor) | Tipo (Aplicación, Servicio, Almacenamiento) | Descripción de Responsabilidades Clave (Responsabilidad Única y Clara) | Tecnologías / Frameworks Clave | Interacciones (Con otros Contenedores Internos o Actores Externos) | Consideraciones de Diseño / NFR (ej. Escalabilidad, Resiliencia, **Velocidad Extrema, Optimización Hardware Radical, Simplicidad Extrema**) |
| --- | --- | --- | --- | --- | --- | --- |
| CO-01 | **API Gateway / Backend Service** | Aplicación/Servicio | Actúa como el punto de entrada principal para las solicitudes externas (IN-01, IN-03, IN-07 del Diagrama de Contexto). Maneja la recepción de Documentos de Pago y metadatosEntrada, la validación inicial, el renombrado de archivos. **Gestiona el estado y la orquestación de Lotes EN MEMORIA (incluyendo En Espera de Llenado y En Espera de Procesamiento), coordinando directamente con el OCR Worker Service cuándo un lote está completo y listo para procesamiento.** Expone endpoints para carga y consulta de estado temporal de lotes y resultados (solo para la UI). Se comunica con el Frontend Service y n8n. **Este es el punto central de gestión de la "cola conceptual" de lotes.** | Flask (Python), Gunicorn/Werkzeug, Bibliotecas de procesamiento de archivos, CORS. | AE-01 (Usuario Final) <-- HTTP/S --> CO-01<br>AE-02 (Sistema n8n) <-- HTTP/S --> CO-01<br>CO-01 --> CO-04 (Llamada para iniciar procesamiento de lote)<br>CO-01 --> CO-06 (Carga de archivos) | Punto central de control. Requiere **alto rendimiento** para ingestión rápida y capacidad de respuesta. Validaciones robustas (Zero-Fault Detection). **Simplicidad extrema al gestionar lotes en memoria, confiando en n8n para la persistencia a largo plazo.** Volatilidad del estado en caso de reinicio (aceptada). |
| CO-04 | **OCR Worker Service (Motor de Procesamiento RADICALMENTE OPTIMIZADO)** | Servicio/Procesador Asíncrono | **El CORAZÓN del sistema y su principal diferenciador de rendimiento.** Recibe la notificación/llamada del API Gateway para procesar un lote. Ejecuta el **motor OCR (OnnxTR con modelos pre-cargados y optimizados para inferencia en CPU y GPU básicas vía ONNX Runtime)**. Aplica algoritmos de **procesamiento espacial (spatial\_processor.py con OpenCV) y validación (validador\_ocr.py, mejora\_ocr.py) de altísima eficiencia, apoyándose en el uso intensivo de librerías como NumPy.** Logra una **velocidad de procesamiento récord (4 imágenes en 0.14 segundos).** Genera los Resultados OCR con alta confianza y los **envía INMEDIATAMENTE a n8n** (IN-04, IN-05). **Registra los valores de rendimiento del procesamiento en archivos de log para "evaluación por pulso", con retención temporal (hasta 30 días).** | Python, onnxtr, onnx, onnxruntime, opencv-python, Pillow, numpy, scikit-image, scipy, Flask (para webhooks a n8n), Posibles optimizaciones con Numba o Cython. | CO-04 <-- CO-01 (Llamada para iniciar procesamiento)<br>CO-04 <-- CO-06 (Servicio de Almacenamiento - Descargar imágenes, **acceso optimizado**) CO-04 --> AE-02 (Sistema n8n - vía webhooks para notificación, **entrega inmediata de resultados**) CO-04 --> Archivos de Log (para monitoreo por pulso) | **Define la Ventaja Competitiva del sistema: Rendimiento del Servicio (Activo y Caliente) llevado al extremo.** Alta Escalabilidad (múltiples instancias en paralelo). Zero-Fault Detection en el procesamiento de errores. Monitoreo por registro de valores post-proceso, **con énfasis en la efímera naturaleza de los datos procesados internamente.** |
| CO-05 | **Frontend (Dashboard) Service** | Aplicación Web (UI) | Proporciona la Interface Excellence Dashboard para el Usuario Final. Maneja la carga de archivos manual (IN-01), la visualización **temporal** del Estado Lote y los Resultados OCR (IN-02, IN-06) obtenidos a través del API Gateway. Es el componente visible para el operador. Su persistencia es solo la del API Gateway en memoria. | HTML5, Vanilla JavaScript, Bootstrap 5, CSS3, API Client, Posiblemente **WebSockets** para actualizaciones en tiempo real y cache en el cliente. | AE-01 (Usuario Final) <--> CO-05 (HTTP/S)<br>CO-05 <--> CO-01 (API Gateway - HTTP/REST, posiblemente WebSockets para monitoreo temporal en tiempo real) | Debe ser reactivo, intuitivo, y proporcionar una experiencia de usuario fluida con **actualizaciones de estado casi instantáneas y latencia mínima.** Claridad de Definiciones y Monitoreo de lo "transitorio". **Conoce y opera bajo la limitación de la persistencia temporal.** |
| CO-06 | **Servicio de Almacenamiento de Archivos (Imágenes y CSV/Logs)** | Servicio de Almacenamiento | Encargado de almacenar los Documentos de Pago originales (imágenes) recibidos. También almacenará los **archivos de log resumen/CSV para el monitoreo por pulso**, con una retención definida (hasta 30 días). Debe ser un almacenamiento **escalable, duradero y de altísimo rendimiento para lecturas/escrituras intensivas y concurrentes** por parte de los OCR Worker Service. **Actúa como el respaldo de los lotes pendientes de procesamiento que gestiona CO-01 y que n8n monitoriza.** | Amazon S3, Google Cloud Storage, **MinIO (si es on-premise con énfasis en velocidad de acceso local y baja latencia)** u otra solución de almacenamiento de objetos/blobs **optimizada para throughput y concurrencia.** | CO-01 (API Gateway) --> CO-06 (Carga de archivos)<br>CO-04 (OCR Worker) <-- CO-06 (Descarga de imágenes)<br>CO-04 --> CO-06 (Guarda logs/CSV de resumen) | **Alta durabilidad y Escalabilidad para volúmenes masivos de imágenes.** Rendimiento del Servicio CRÍTICO para el flujo de procesamiento. **Gestiona la persistencia temporal de los logs de resumen.** |

Exportar a Hojas de cálculo

**5. Contratos de API Formales**

La comunicación entre los componentes del sistema y con los actores externos se rige por contratos de API formales. Estos contratos especifican los endpoints, métodos, parámetros y respuestas esperadas, garantizando la Interface Excellence y el Acoplamiento Débil entre los servicios.

**CONTRATOS DE API FORMALES (VERSIÓN BLINDADA - AJUSTE 2025-07-15 - V2: DETALLE DE LOTE EN MEMORIA)**

**1. Contratos de API Gateway / Backend Service (CO-01)**

**1.1. Ingestión de Documento Individual desde Frontend (CO-05)**

* **Endpoint:** /api/upload
* **Descripción:** Permite al Usuario Final (AE-01) cargar **UN ÚNICO** Documento de Pago (imagen) junto con sus metadatosEntrada contextuales. CO-01 gestionará la acumulación interna para formar lotes lógicos.
* **Método:** POST
* **Request Body (multipart/form-data):**
  + file: Un único archivo de imagen (image/jpeg, image/png, application/pdf si se permite PDF de imágenes).
    - **Tipo:** File
    - **Requerido:** Sí
    - **Descripción:** El Documento de Pago en formato de imagen.
  + metadatos\_json: JSON que contiene los metadatosEntrada para el documento y el contexto del lote al que pertenece.
    - **Tipo:** String (JSON serializado)
    - **Requerido:** Sí
    - **Ejemplo de metadatos\_json:**

JSON

{

"idSorteo": "20250712-A",

"fechaSorteo": "2025-07-12",

"numeroLlegada": 1,

"documento": { // UN ÚNICO OBJETO DOCUMENTO

"idWhatsapp": "584121234567",

"nombre": "Maria Perez",

"caption": "Pago de hoy",

"horaMinutoN8nIngreso": "1045" // Hora y minuto de llegada del mensaje/imagen

}

}

* + - * **idSorteo**: string (UUID, ej. "20250712-A"). Requerido. Define el lote.
      * **fechaSorteo**: string (YYYY-MM-DD). Requerido. Define el lote.
      * **numeroLlegada**: integer (incremental dentro del lote o sorteo). Requerido. Define el lote.
      * **documento**: object, conteniendo los metadatos específicos de este documento:
        + **idWhatsapp**: string. Requerido.
        + **nombre**: string. Opcional.
        + **caption**: string. Opcional.
        + **horaMinutoN8nIngreso**: string (HHMM). La hora y minuto en que el mensaje/imagen llegó al sistema (ej. 1045). Requerido.
    - **Nota:** El campo tipoOrigen ha sido eliminado de la entrada ya que no es directamente detectable por la fuente de ingestión.
* **Responses:**
  + **202 Accepted - Documento Aceptado para Procesamiento:**
    - **Descripción Detallada (Lote EN MEMORIA):** El documento ha sido recibido y el sistema (CO-01) lo ha incorporado a un Lote **gestionado EN MEMORIA** para su procesamiento asíncrono.
      * **Funcionamiento Interno de CO-01:**
        1. **Identificación/Creación de Lote:** Al recibir este documento individual, CO-01 utiliza la combinación de idSorteo, fechaSorteo y numeroLlegada para identificar (o crear si no existe) un Lote lógico en su espacio de memoria volátil. Este Lote existe **temporalmente** para agrupar documentos relacionados y facilitar la Trazabilidad y el Monitoreo.
        2. **Asociación de Documento:** El documento recién recibido y su idDocumentoIngresado (generado por CO-01) se asocian a este Lote existente o recién creado EN MEMORIA.
        3. **Estado del Lote:** El estadoLote retornado (RECIBIDO\_EN\_PROCESO) indica que el Lote al que este documento pertenece está actualmente activo, sigue recibiendo (o está en proceso de gestionar) documentos, y no ha finalizado su ciclo de procesamiento. CO-01 no espera a recibir un "batch completo" para responder; acepta cada documento individualmente y los va agrupando lógicamente en su memoria.
        4. **Disparo Asíncrono a CO-04:** Inmediatamente después de la asociación, CO-01 **inicia de forma asíncrona el procesamiento de ESTE DOCUMENTO INDIVIDUAL** enviándolo al OCR Worker Service (CO-04) (ver Contrato 2.1), adjuntando el idLote para mantener la Coherencia de Referencias Críticas.
    - **Body (application/json):**

JSON

{

"idLote": "uuid-del-lote-al-que-pertenece-este-documento",

"idDocumentoIngresado": "uuid-del-documento-recibido",

"estadoLote": "RECIBIDO\_EN\_PROCESO",

"mensaje": "Documento recibido y añadido a un lote en memoria para procesamiento. Monitoree el estado del lote con /api/lotes/{idLote}/status"

}

* + - * **idLote**: string (UUID del lote generado/identificado por CO-01).
      * **idDocumentoIngresado**: string (UUID generado por CO-01 para este documento individual).
      * **estadoLote**: string (RECIBIDO\_EN\_PROCESO).
      * **mensaje**: string.
  + **400 Bad Request - Validación de Entrada Fallida:**
    - **Descripción:** La solicitud no cumple con el formato o los requisitos de los metadatos.
    - **Body (application/json):**

JSON

{

"codigoError": "VALIDATION\_ERROR",

"mensaje": "Los metadatos provistos son inválidos o incompletos.",

"detalles": "Campo 'idSorteo' es requerido y no puede ser nulo."

}

* + - * **codigoError**: string.
      * **mensaje**: string.
      * **detalles**: string (opcional, con información específica del error).
  + **500 Internal Server Error - Error en el Servidor:**
    - **Descripción:** Ocurrió un error inesperado en CO-01 que impidió procesar la solicitud.
    - **Body (application/json):**

JSON

{

"codigoError": "INTERNAL\_SERVER\_ERROR",

"mensaje": "Ocurrió un error inesperado al procesar el documento."

}

* + - * **codigoError**: string.
      * **mensaje**: string.

**1.2. Ingestión de Documento Individual desde Sistema n8n (AE-02)**

* **Endpoint:** /api/n8n/webhook/document\_ingestion
* **Descripción:** n8n envía **UN ÚNICO** Documento de Pago (referencia URL) y sus metadatosEntrada contextuales para procesamiento automatizado. CO-01 gestionará la acumulación interna para formar lotes lógicos.
* **Método:** POST
* **Request Body (application/json):**
  + **Tipo:** Object
  + **Requerido:** Sí
  + **Ejemplo de Body:**

JSON

{

"idSorteo": "20250712-A",

"fechaSorteo": "2025-07-12",

"numeroLlegada": 1,

"documento": { // UN ÚNICO OBJETO DOCUMENTO

"idDocumento": "uuid-doc-1-generado-por-n8n", // ID de n8n para trazabilidad

"urlImagen": "https://example.com/images/doc1.jpg",

"idWhatsapp": "584121234567",

"nombre": "Maria Perez",

"caption": "Pago de hoy",

"horaMinutoN8nIngreso": "1045" // Hora y minuto de llegada del mensaje/imagen

}

}

* + - **idSorteo**: string (UUID, ej. "20250712-A"). Requerido. Define el lote.
    - **fechaSorteo**: string (YYYY-MM-DD). Requerido. Define el lote.
    - **numeroLlegada**: integer (incremental dentro del lote o sorteo). Requerido. Define el lote.
    - **documento**: object, conteniendo los metadatos específicos de este documento:
      * **idDocumento**: string (UUID, generado por n8n). **Este ID será la referencia externa de n8n para la trazabilidad.** Requerido.
      * **urlImagen**: string (URL accesible públicamente donde CO-01 puede descargar la imagen). Requerido.
      * **idWhatsapp**: string. Requerido.
      * **nombre**: string. Opcional.
      * **caption**: string. Opcional.
      * **horaMinutoN8nIngreso**: string (HHMM). La hora y minuto en que el mensaje/imagen llegó a n8n y fue enviado. Requerido.
    - **Nota:** El campo tipoOrigen ha sido eliminado de la entrada ya que no es directamente detectable por la fuente de ingestión.
* **Responses:**
  + **202 Accepted - Documento Aceptado para Procesamiento:**
    - **Descripción Detallada (Lote EN MEMORIA):** El documento ha sido recibido y el sistema (CO-01) lo ha incorporado a un Lote **gestionado EN MEMORIA** para su procesamiento asíncrono.
      * **Funcionamiento Interno de CO-01:**
        1. **Identificación/Creación de Lote:** Al recibir este documento individual, CO-01 utiliza la combinación de idSorteo, fechaSorteo y numeroLlegada para identificar (o crear si no existe) un Lote lógico en su espacio de memoria volátil. Este Lote existe **temporalmente** para agrupar documentos relacionados y facilitar la Trazabilidad y el Monitoreo.
        2. **Asociación de Documento:** El documento recién recibido y su idDocumentoIngresado (generado por CO-01, que puede ser el idDocumento provisto por n8n si CO-01 decide adoptarlo como su ID interno) se asocian a este Lote existente o recién creado EN MEMORIA.
        3. **Estado del Lote:** El estadoLote retornado (RECIBIDO\_EN\_PROCESO) indica que el Lote al que este documento pertenece está actualmente activo, sigue recibiendo (o está en proceso de gestionar) documentos, y no ha finalizado su ciclo de procesamiento. CO-01 no espera a recibir un "batch completo" para responder; acepta cada documento individualmente y los va agrupando lógicamente en su memoria.
        4. **Disparo Asíncrono a CO-04:** Inmediatamente después de la asociación, CO-01 **inicia de forma asíncrona el procesamiento de ESTE DOCUMENTO INDIVIDUAL** enviándolo al OCR Worker Service (CO-04) (ver Contrato 2.1), adjuntando el idLote para mantener la Coherencia de Referencias Críticas.
    - **Body (application/json):**

JSON

{

"idLote": "uuid-del-lote-al-que-pertenece-este-documento",

"idDocumentoIngresado": "uuid-del-documento-recibido",

"estadoLote": "RECIBIDO\_EN\_PROCESO",

"mensaje": "Documento recibido y añadido a un lote en memoria para procesamiento."

}

* + - * **idLote**: string (UUID del lote generado/identificado por CO-01).
      * **idDocumentoIngresado**: string (UUID generado por CO-01 para este documento individual, que puede ser el mismo idDocumento provisto por n8n si CO-01 decide adoptarlo como su ID interno).
      * **estadoLote**: string (RECIBIDO\_EN\_PROCESO).
      * **mensaje**: string.
  + **400 Bad Request - Validación de Entrada Fallida:**
    - **Descripción:** La solicitud no cumple con el formato o los requisitos de los metadatos.
    - **Body (application/json):**

JSON

{

"codigoError": "VALIDATION\_ERROR",

"mensaje": "Los metadatos provistos son inválidos o incompletos.",

"detalles": "Campo 'urlImagen' es requerido y no puede ser nulo."

}

* + - * **codigoError**: string.
      * **mensaje**: string.
      * **detalles**: string (opcional, con información específica del error).
  + **500 Internal Server Error - Error en el Servidor:**
    - **Descripción:** Ocurrió un error inesperado en CO-01 que impidió procesar la solicitud.
    - **Body (application/json):**

JSON

{

"codigoError": "INTERNAL\_SERVER\_ERROR",

"mensaje": "Ocurrió un error inesperado al procesar el documento."

}

* + - * **codigoError**: string.
      * **mensaje**: string.

**1.3. Consulta de Estado de Lote por Frontend (CO-05)**

* **Endpoint:** /api/lotes/{idLote}/status
* **Descripción:** Permite al Usuario Final (AE-01) consultar el estado actual de un Lote de documentos y el progreso de procesamiento de sus documentos individuales. Este estado se gestiona EN MEMORIA en CO-01.
* **Método:** GET
* **Parámetros de Ruta:**
  + idLote: string (UUID del lote). Requerido.
* **Responses:**
  + **200 OK - Estado del Lote Disponible:**
    - **Descripción:** Retorna el estado actual del Lote y un resumen de sus documentos.
    - **Body (application/json):**

JSON

{

"idLote": "uuid-del-lote",

"estadoLote": "RECIBIENDO\_DOCUMENTOS", // Puede ser: RECIBIENDO\_DOCUMENTOS, PROCESANDO\_OCR, COMPLETADO\_OCR, FALLIDO\_PARCIAL, FALLIDO\_TOTAL

"totalDocumentos": 5,

"documentosProcesados": 2,

"documentosConError": 0,

"progresoPorcentaje": 40,

"detallesDocumentos": [

{

"idDocumento": "uuid-doc-1",

"nombreRenombradoArchivo": "20250712-A\_584121234567\_Maria\_Perez\_20250712\_1045\_1.jpg",

"estadoProcesamiento": "PROCESADO\_OK", // PROCESADO\_OK, EN\_PROCESO, ERROR\_OCR, PENDIENTE

"resultadoOcrDisponible": true

},

{

"idDocumento": "uuid-doc-2",

"nombreRenombradoArchivo": "20250712-A\_584121234567\_Maria\_Perez\_20250712\_1046\_2.jpg",

"estadoProcesamiento": "EN\_PROCESO",

"resultadoOcrDisponible": false

}

// ... más documentos

]

}

* + - * **idLote**: string.
      * **estadoLote**: string (enum: RECIBIENDO\_DOCUMENTOS, PROCESANDO\_OCR, COMPLETADO\_OCR, FALLIDO\_PARCIAL, FALLIDO\_TOTAL).
      * **totalDocumentos**: integer (cantidad total de documentos asociados al lote).
      * **documentosProcesados**: integer (cantidad de documentos con estado PROCESADO\_OK o ERROR\_OCR).
      * **documentosConError**: integer (cantidad de documentos con estado ERROR\_OCR).
      * **progresoPorcentaje**: integer (porcentaje de documentos procesados/con error).
      * **detallesDocumentos**: array de objetos, cada uno representando un documento:
        + **idDocumento**: string.
        + **nombreRenombradoArchivo**: string.
        + **estadoProcesamiento**: string (enum: PROCESADO\_OK, EN\_PROCESO, ERROR\_OCR, PENDIENTE).
        + **resultadoOcrDisponible**: boolean (indica si ya se tiene un Resultado OCR para este documento).
  + **404 Not Found - Lote No Encontrado:**
    - **Descripción:** El idLote proporcionado no corresponde a ningún lote en memoria.
    - **Body (application/json):**

JSON

{

"codigoError": "LOTE\_NOT\_FOUND",

"mensaje": "El lote con id 'uuid-del-lote' no fue encontrado."

}

* + **500 Internal Server Error - Error en el Servidor:** (Mismo formato que antes)

**1.4. Recepción de Resultados OCR Individuales desde OCR Worker Service (CO-04)**

* **Endpoint:** /api/internal/ocr\_results/callback
* **Descripción:** Este endpoint interno es invocado por el OCR Worker Service (CO-04) una vez que ha finalizado el procesamiento de **UN ÚNICO Documento de Pago**, enviando el Resultado OCR estructurado y el estado de procesamiento de ese documento.
* **Método:** POST
* **Request Body (application/json):**
  + **Tipo:** Object
  + **Requerido:** Sí
  + **Ejemplo de Body:**

JSON

{

"idLote": "uuid-del-lote-al-que-pertenece",

"idDocumento": "uuid-del-documento-procesado",

"nombreRenombradoArchivo": "20250712-A\_584121234567\_Maria\_Perez\_20250712\_1045\_1.jpg",

"estadoProcesamiento": "PROCESADO\_OK", // PROCESADO\_OK, ERROR\_OCR

"mensajeError": null, // Sólo si estadoProcesamiento es ERROR\_OCR

"resultadoOcr": {

"referenciaPago": "0123456789",

"montoPago": 100.50,

"fechaPago": "2025-07-11",

"horaPago": "0930",

"bancoBeneficiario": "Banco Ejemplo",

"confianzaGlobalOCR": 0.95,

"estadoValidacion": "VALIDADO\_OK", // VALIDADO\_OK, VALIDADO\_CON\_ADVERTENCIAS, FALLO\_VALIDACION

"detallesExtraccion": {

"campo1": {"valor": "valor1", "confianza": 0.98, "coordenadas": "..."}

},

"metadatosEntradaOriginal": {

"idSorteo": "20250712-A",

"fechaSorteo": "2025-07-12",

"numeroLlegada": 1,

"idWhatsapp": "584121234567",

"nombre": "Maria Perez",

"caption": "Pago de hoy",

"horaMinutoN8nIngreso": "1045"

// ... otros metadatos recibidos originalmente

}

}

}

* + - **idLote**: string (UUID del lote). Requerido.
    - **idDocumento**: string (UUID del documento que fue procesado). Requerido.
    - **nombreRenombradoArchivo**: string. Requerido. Para Trazabilidad.
    - **estadoProcesamiento**: string (enum: PROCESADO\_OK, ERROR\_OCR). Requerido.
    - **mensajeError**: string (descripción del error si estadoProcesamiento es ERROR\_OCR). Opcional/Nulo.
    - **resultadoOcr**: object (JSON estructurado con los datos extraídos). Requerido si estadoProcesamiento es PROCESADO\_OK.
      * **referenciaPago**: string. Requerido.
      * **montoPago**: number. Requerido.
      * **fechaPago**: string (YYYY-MM-DD). Requerido.
      * **horaPago**: string (HHMM). Requerido.
      * **bancoBeneficiario**: string. Opcional.
      * **confianzaGlobalOCR**: number (0.0 a 1.0). Requerido.
      * **estadoValidacion**: string (enum: VALIDADO\_OK, VALIDADO\_CON\_ADVERTENCIAS, FALLO\_VALIDACION). Requerido.
      * **detallesExtraccion**: object (campos extraídos con confianza y coordenadas si aplica). Opcional.
      * **metadatosEntradaOriginal**: object (el JSON de metadatosEntrada que CO-01 envió a CO-04). Requerido. Esto asegura la Integridad Total y Trazabilidad del contexto original.
* **Responses:**
  + **200 OK - Resultado Recibido y Procesado:**
    - **Descripción:** CO-01 ha recibido y actualizado el estado del documento y del lote en su memoria.
    - **Body (application/json):**

JSON

{

"mensaje": "Resultado OCR para documento 'uuid-del-documento' recibido y lote actualizado."

}

* + - * **mensaje**: string.
  + **400 Bad Request - Validación de Entrada Fallida:** (Mismo formato que antes)
  + **404 Not Found - Documento/Lote No Encontrado:** (Mismo formato que antes)
  + **500 Internal Server Error - Error en el Servidor:** (Mismo formato que antes)

**2. Contratos de OCR Worker Service (CO-04)**

**2.1. Inicio de Procesamiento de Documento Individual (Invocación desde CO-01)**

* **Endpoint:** /process\_document (endpoint interno de CO-04, no expuesto públicamente).
* **Descripción:** CO-01 invoca este servicio para iniciar el procesamiento OCR de **UN ÚNICO Documento de Pago** de forma asíncrona.
* **Método:** POST
* **Request Body (application/json):**
  + **Tipo:** Object
  + **Requerido:** Sí
  + **Ejemplo de Body:**

JSON

{

"idLote": "uuid-del-lote",

"idDocumento": "uuid-del-documento-a-procesar",

"nombreRenombradoArchivo": "20250712-A\_584121234567\_Maria\_Perez\_20250712\_1045\_1.jpg",

"urlImagenAlmacenamiento": "https://url-servicio-almacenamiento/path/a/imagen.jpg",

"metadatosEntrada": {

"idSorteo": "20250712-A",

"fechaSorteo": "2025-07-12",

"numeroLlegada": 1,

"idWhatsapp": "584121234567",

"nombre": "Maria Perez",

"caption": "Pago de hoy",

"horaMinutoN8nIngreso": "1045"

// ... otros metadatos originales

},

"callbackUrl": "http://api-gateway-url/api/internal/ocr\_results/callback"

}

* + - **idLote**: string (UUID del lote al que pertenece este documento). Requerido.
    - **idDocumento**: string (UUID del documento). Requerido.
    - **nombreRenombradoArchivo**: string. Requerido. Para Trazabilidad.
    - **urlImagenAlmacenamiento**: string (URL del Servicio de Almacenamiento (CO-06) donde CO-04 debe descargar la imagen). Requerido.
    - **metadatosEntrada**: object (el JSON de metadatosEntrada original, necesario para aplicar Reglas de Extracción). Requerido.
    - **callbackUrl**: string (URL del endpoint en CO-01 al que CO-04 debe enviar el Resultado OCR una vez finalizado). Requerido.
* **Responses:**
  + **202 Accepted - Procesamiento Iniciado:**
    - **Descripción:** CO-04 ha recibido el documento y ha iniciado su procesamiento asíncrono.
    - **Body (application/json):**

JSON

{

"mensaje": "Procesamiento de documento iniciado."

}

* + - * **mensaje**: string.
  + **400 Bad Request - Validación de Entrada Fallida:** (Mismo formato que antes)
  + **500 Internal Server Error - Error en el Servidor:** (Mismo formato que antes)

**3. Notificaciones Salientes (Desde CO-01 hacia AE-02 (n8n))**

**3.1. Recepción de Resultado Consolidado del Lote y Estado Lote (Invocación desde CO-01)**

* **Endpoint:** https://docs.n8n.io/hosting/configuration/configuration-methods/ (Este es un webhook que CO-01 invoca).
* **Descripción:** Una vez que CO-01 determina que un Lote ha completado su procesamiento (todos los documentos han sido procesados, con éxito o error), envía el Resultado Consolidado del Lote a n8n. n8n es la Fuente de Verdad Externa y se encargará de persistir esta información.
* **Método:** POST
* **Request Body (application/json):**
  + **Tipo:** Object
  + **Requerido:** Sí
  + **Ejemplo de Body:**

JSON

{

"idLote": "uuid-del-lote",

"estadoLoteFinal": "COMPLETADO\_OCR", // Puede ser: COMPLETADO\_OCR, FALLIDO\_PARCIAL, FALLIDO\_TOTAL

"totalDocumentos": 5,

"documentosProcesadosConExito": 4,

"documentosConError": 1,

"resultadosConsolidados": [

{

"idDocumento": "uuid-doc-1",

"nombreRenombradoArchivo": "20250712-A\_584121234567\_Maria\_Perez\_20250712\_1045\_1.jpg",

"estadoProcesamientoDocumento": "PROCESADO\_OK",

"resultadoOcr": {

"referenciaPago": "0123456789",

"montoPago": 100.50,

"fechaPago": "2025-07-11",

"horaPago": "0930",

"bancoBeneficiario": "Banco Ejemplo",

"confianzaGlobalOCR": 0.95,

"estadoValidacion": "VALIDADO\_OK",

"detallesExtraccion": {},

"metadatosEntradaOriginal": {} // Metadatos originales de este documento

}

},

{

"idDocumento": "uuid-doc-2",

"nombreRenombradoArchivo": "20250712-A\_584121234567\_Maria\_Perez\_20250712\_1046\_2.jpg",

"estadoProcesamientoDocumento": "ERROR\_OCR",

"resultadoOcr": null, // Null si hubo error de procesamiento

"mensajeErrorDocumento": "Error de calidad en la imagen: texto ilegible.",

"metadatosEntradaOriginal": {} // Metadatos originales de este documento

}

// ... más documentos

],

"mensajeFinal": "Procesamiento de lote completado con 4 documentos exitosos y 1 con error.",

"fechaHoraFinalizacion": "2025-07-12T11:00:00Z"

}

* + - **idLote**: string (UUID del lote). Requerido.
    - **estadoLoteFinal**: string (enum: COMPLETADO\_OCR, FALLIDO\_PARCIAL, FALLIDO\_TOTAL). Requerido.
    - **totalDocumentos**: integer. Requerido.
    - **documentosProcesadosConExito**: integer. Requerido.
    - **documentosConError**: integer. Requerido.
    - **resultadosConsolidados**: array de objetos, cada uno con el resultado de un documento. Requerido.
      * **idDocumento**: string.
      * **nombreRenombradoArchivo**: string.
      * **estadoProcesamientoDocumento**: string (enum: PROCESADO\_OK, ERROR\_OCR).
      * **resultadoOcr**: object (el JSON del Resultado OCR completo). Nulo si hay ERROR\_OCR.
      * **mensajeErrorDocumento**: string (si estadoProcesamientoDocumento es ERROR\_OCR).
      * **metadatosEntradaOriginal**: object (copia de los metadatos de entrada originales del documento).
    - **mensajeFinal**: string.
    - **fechaHoraFinalizacion**: string (ISO 8601).
* **Responses:** (Se espera una respuesta HTTP estándar de n8n para indicar que el webhook fue recibido).
  + **200 OK / 202 Accepted - Notificación Recibida:**
    - **Descripción:** n8n ha recibido la notificación y procederá con su lógica de persistencia.
    - **Body:** (Puede variar según la configuración de n8n, a menudo vacío o un mensaje de confirmación).

**6. Especificaciones de Requisitos**

Para traducir las necesidades del negocio en funcionalidades concretas, se han elaborado especificaciones de requisitos detalladas, tanto funcionales (Historias de Usuario) como no funcionales (RNF). Estos requisitos son la verdad absoluta de lo que el sistema debe hacer y cómo debe comportarse.

**ESPECIFICACIONES DE REQUISITOS - HISTORIAS DE USUARIO**

**El Corazón del Valor de Negocio y la Precisión Operativa**

Versión: 1.0 (Blindado)

Fecha: 14 de julio de 2025

Sistema: Sistema OCR Asíncrono Empresarial

**1. Introducción y Propósito del Documento**

El presente documento establece la metodología y el formato para la definición exhaustiva de los requisitos funcionales y no funcionales del "Sistema OCR Asíncrono Empresarial". Su propósito primordial es servir como la **verdad absoluta** de lo que el sistema debe hacer y cómo debe comportarse, desde la perspectiva del valor de negocio y del usuario.

Este nivel de detalle es indispensable para garantizar la **Integridad Total** del sistema, permitir la **Zero-Fault Detection** desde las etapas más tempranas del diseño, y facilitar la **Trazabilidad** completa desde una necesidad de negocio hasta su implementación y prueba. Una definición de requisitos ambigua o incompleta es la causa raíz más común de fallos en el software y la incapacidad de satisfacer las expectativas del cliente.

**Este documento será la base para:**

* El diseño técnico detallado.
* La planificación y ejecución de las pruebas (unitarias, de integración, de sistema, de aceptación).
* La comunicación inequívoca entre el equipo de negocio, los arquitectos y el equipo de desarrollo.
* La medición del éxito del proyecto.

**2. Principios Rectores de la Documentación de Requisitos**

Cada requisito y cada historia de usuario debe adherirse rigurosamente a los siguientes principios:

* **Claridad de Definiciones:** Cada término, concepto o acción debe ser inequívoco. Se utilizará la terminología definida en el **GLOSARIO DE TÉRMINOS CLAVE.docx**.
* **Comprensión Profunda del Contexto de Dominio:** Los requisitos deben reflejar un entendimiento cabal del proceso de negocio de sorteos y pagos móviles venezolanos, tal como se detalla en el **MODELO DE DOMINIO CONCEPTUAL.docx**.
* **Granularidad:** Los requisitos deben ser lo suficientemente detallados como para ser probables y no dejar espacio para la interpretación subjetiva. Esto se logrará con los escenarios **Dado/Cuando/Entonces**.
* **Zero-Fault Detection:** Se anticiparán y especificarán explícitamente los escenarios de error, los casos de borde y el comportamiento del sistema ante condiciones inesperadas. La ausencia de un escenario de error en la especificación es un fallo en la Zero-Fault Detection.
* **Trazabilidad:** Cada requisito funcional (Historia de Usuario) y no funcional (RNF) deberá ser identificable de forma única, permitiendo su vinculación con los casos de prueba, componentes de código y defectos.
* **Medibilidad/Verificabilidad:** Los requisitos, especialmente los no funcionales, deben ser cuantificables y poder ser validados mediante pruebas.
* **Orientación al Valor de Negocio:** Cada historia de usuario debe expresar claramente el beneficio que aporta al actor y, en última instancia, al negocio.
* **Consistencia:** Los requisitos deben ser coherentes con el **MODELO DE DOMINIO CONCEPTUAL.docx**, los **CONTRATOS DE API FORMALES** y los **DIAGRAMAS DE CONTEXTO Y COMPONENTES**.

**3. Estructura del Documento de Requisitos**

El documento se organizará en dos secciones principales:

**3.1. Requisitos Funcionales (Historias de Usuario con Detalles)**

Cada historia de usuario se presentará de la siguiente manera:

* **ID de Historia:** HU-[Funcionalidad Principal]-[Número Secuencial] (Ej.: HU-ING-001, HU-MON-001).
* **Nombre de la Historia:** Un título descriptivo y conciso de la funcionalidad.
* **Descripción de la Historia (Formato Estándar):**
  + **Como** [Actor Principal] (referenciado desde ID Actor.pdf),
  + **quiero** [Funcionalidad Específica],
  + **para** [Objetivo o Beneficio de Negocio].
* **Criterios de Aceptación (Escenarios Detallados):**
  + Se utilizará el formato **Gherkin** (Dado / Cuando / Entonces) para cada escenario relevante (casos de éxito, casos de error, casos de borde).
  + **Dado** [un contexto inicial o precondiciones],
  + **Cuando** [ocurre una acción específica o evento],
  + **Entonces** [el sistema reacciona de una manera esperada].
  + **Sub-Criterios / Notas:** Detalles adicionales, reglas de negocio específicas, validaciones de datos, formatos de salida, referencias cruzadas a **Contratos de API Formales**, entidades del **MODELO DE DOMINIO CONCEPTUAL.docx**, y principios de diseño como Zero-Fault Detection.

**3.2. Requisitos No Funcionales (RNF)**

Estos requisitos se agruparán por categorías y definirán las cualidades y restricciones de cómo debe operar el sistema. Cada RNF tendrá:

* **ID RNF:** RNF-[Categoría]-[Número Secuencial] (Ej.: RNF-PERF-001).
* **Descripción:** Una afirmación clara y medible del requisito.
* **Métrica / Criterio de Medición:** Cómo se verificará que el requisito se cumple.

**4. Requisitos Funcionales (Historias de Usuario Detalladas)**

**4.1. Grupo: Ingestión de Documentos**

**HU-ING-001: Carga de Documento Individual desde Frontend**

* **ID de Historia:** HU-ING-001
* **Nombre de la Historia:** Carga de Documento de Pago desde la Interfaz de Usuario
* **Como** Usuario Final (Operador/Administrador) (AE-01),
* **quiero** cargar un Documento de Pago individual (imagen) con sus metadatosEntrada a través del Frontend (CO-05),
* **para** que el sistema inicie su procesamiento asíncrono, lo asigne a un Lote y yo pueda monitorear su estado.

**Criterios de Aceptación:**

* **Escenario 1.1: Carga Exitosa de Imagen JPG con Metadatos Completos**
  + **Dado** que el Usuario Final (AE-01) ha seleccionado un archivo JPG válido que representa un Documento de Pago y ha ingresado todos los metadatosEntrada requeridos (idSorteo, fechaSorteo, numeroLlegada, idWhatsapp, horaMinutoN8nIngreso, caption, nombre) en el Frontend (CO-05),
  + **Cuando** el Usuario Final (AE-01) envía el formulario de carga,
  + **Entonces** CO-05 realiza una llamada POST /api/upload a CO-01 con multipart/form-data, y CO-01 responde con un 202 Accepted (Contrato API 1.1) incluyendo el idLote y idDocumentoIngresado, y el estadoLote del lote en memoria de CO-01 es RECIBIDO\_EN\_PROCESO.
  + **Sub-Criterios/Notas:**
    - CO-01 debe generar un idLote único si no existe un lote para la combinación idSorteo, fechaSorteo, numeroLlegada. Si el lote ya existe, CO-01 debe asociar el nuevo documento al idLote existente.
    - CO-01 debe generar un idDocumentoIngresado único para el documento cargado.
    - CO-01 debe iniciar asíncronamente el procesamiento OCR para el documento en CO-04 (Contrato API 2.1).
    - CO-01 debe almacenar el archivo de imagen en CO-06 (Servicio de Almacenamiento).
    - Se aplica el principio de Claridad de Definiciones y Comprensión Profunda del Contexto de Dominio para los metadatos.
* **Escenario 1.2: Carga de PDF (Imagen) Exitosa con Metadatos Completos**
  + **Dado** que el Usuario Final (AE-01) ha seleccionado un archivo PDF válido (que contiene una única imagen o es una imagen rasterizada) que representa un Documento de Pago y ha ingresado todos los metadatosEntrada requeridos en el Frontend (CO-05),
  + **Cuando** el Usuario Final (AE-01) envía el formulario de carga,
  + **Entonces** CO-05 realiza una llamada POST /api/upload a CO-01, y CO-01 responde con un 202 Accepted incluyendo el idLote y idDocumentoIngresado, y el estadoLote es RECIBIDO\_EN\_PROCESO.
  + **Sub-Criterios/Notas:**
    - El sistema (CO-01 o CO-04 durante la pre-procesamiento) debe ser capaz de manejar PDFs de una sola página que sean esencialmente imágenes.
    - Para PDFs con múltiples imágenes o texto vectorial, el sistema debe definir un comportamiento (ej. rechazar, procesar solo la primera página, convertir a imagen antes de OCR si aplica). **Por ahora, se asume PDF de imagen única o rasterizada.**
* **Escenario 1.3: Carga con Metadatos Incompletos o Inválidos**
  + **Dado** que el Usuario Final (AE-01) intenta cargar un Documento de Pago pero omite un metadatoEntrada requerido (ej. idSorteo es nulo o fechaSorteo es inválida, no YYYY-MM-DD),
  + **Cuando** el Usuario Final (AE-01) envía el formulario de carga,
  + **Entonces** CO-01 responde con un 400 Bad Request (Contrato API 1.1) al Frontend (CO-05), y la respuesta JSON incluye codigoError y un mensaje claro y específico sobre el campo inválido o faltante.
  + **Sub-Criterios/Notas:**
    - Se aplica el principio de Zero-Fault Detection y Claridad de Definiciones en el manejo de errores.
    - Mensajes de error deben ser amigables para el usuario (ej. "El campo 'idSorteo' es requerido.", "Formato de fechaSorteo inválido. Use YYYY-MM-DD.").
* **Escenario 1.4: Carga de Archivo no Soportado (Tipo MIME)**
  + **Dado** que el Usuario Final (AE-01) intenta cargar un archivo con un tipo MIME no soportado (ej. .doc, .mp4),
  + **Cuando** el Usuario Final (AE-01) envía el formulario de carga,
  + **Entonces** CO-01 responde con un 400 Bad Request indicando que el tipo de archivo no es válido para procesamiento OCR de imágenes de pagos.
  + **Sub-Criterios/Notas:**
    - CO-01 solo debe aceptar tipos MIME definidos para imágenes (image/jpeg, image/png, application/pdf).

**HU-ING-002: Ingestión de Documento desde Sistema n8n (AE-02)**

* **ID de Historia:** HU-ING-002
* **Nombre de la Historia:** Ingestión Asíncrona de Documentos mediante Webhook desde n8n
* **Como** Sistema n8n (AE-02),
* **quiero** enviar una URL de un Documento de Pago y sus metadatosEntrada a CO-01 vía webhook,
* **para** que CO-01 lo descargue, procese de forma asíncrona y lo asigne a un Lote, sin bloquear mi flujo de trabajo de orquestación.

**Criterios de Aceptación:**

* **Escenario 2.1: Envío Exitoso de URL y Metadatos Completos**
  + **Dado** que n8n tiene una urlImagen accesible que apunta a una imagen válida de Documento de Pago (JPG, PNG, PDF de imagen única) y los metadatosEntrada completos y válidos,
  + **Cuando** n8n realiza una llamada POST /api/n8n/webhook/document\_ingestion a CO-01 con el JSON de metadatos y la urlImagen (Contrato API 1.2),
  + **Entonces** CO-01 responde con un 202 Accepted incluyendo el idLote y idDocumentoIngresado, y el sistema inicia el proceso de descarga, almacenamiento y procesamiento de la imagen de forma asíncrona.
  + **Sub-Criterios/Notas:**
    - CO-01 es responsable de descargar la imagen desde la urlImagen y almacenarla en CO-06.
    - El proceso de descarga y procesamiento debe ser no bloqueante para la respuesta a n8n.
    - Se mantiene el mismo principio de asignación a lote y generación de IDs que HU-ING-001.
* **Escenario 2.2: URL de Imagen No Accesible o Inválida**
  + **Dado** que n8n envía una urlImagen que no es accesible (ej. URL rota, archivo no existe, permisos insuficientes) o no es un tipo de imagen válido,
  + **Cuando** CO-01 intenta descargar la imagen después de recibir el webhook,
  + **Entonces** CO-01 registra un ERROR\_DESCARGA\_IMAGEN asociado al idDocumentoIngresado en su estado EN MEMORIA, actualiza el estado de ese documento a ERROR\_DESCARGA\_IMAGEN, y el estado del Lote reflejará este fallo (RECIBIDO\_EN\_PROCESO o FALLIDO\_PARCIAL si es el último documento del lote).
  + **Sub-Criterios/Notas:**
    - CO-01 debe responder 202 Accepted a n8n inicialmente si la validación de la URL es posterior a la recepción del webhook, pero el error se gestiona asíncronamente. Si la URL es inválida sintácticamente, CO-01 puede responder 400 Bad Request.
    - Se aplica la Zero-Fault Detection para gestionar fallos de componentes externos y no bloquear el sistema.
* **Escenario 2.3: Metadatos de Entrada Incompletos o Inválidos desde n8n**
  + **Dado** que n8n envía una solicitud con un JSON de metadatosEntrada que carece de campos obligatorios o tiene formatos incorrectos (ej. fechaSorteo no es YYYY-MM-DD),
  + **Cuando** CO-01 recibe la solicitud,
  + **Entonces** CO-01 responde con un 400 Bad Request (Contrato API 1.2) indicando el error de validación de los metadatos.
  + **Sub-Criterios/Notas:**
    - El JSON de error debe ser consistente con el manejo de errores global del sistema, especificando codigoError y mensaje.

**4.2. Grupo: Monitoreo y Trazabilidad**

**HU-MON-001: Consulta de Estado del Lote para Monitoreo**

* **ID de Historia:** HU-MON-001
* **Nombre de la Historia:** Monitoreo en Tiempo Casi Real del Estado de Procesamiento del Lote
* **Como** Usuario Final (Operador/Administrador) (AE-01),
* **quiero** consultar el estado actual de un Lote específico,
* **para** visualizar el progreso del procesamiento OCR de sus documentos y saber cuándo el lote está COMPLETADO\_OK o si ha habido FALLIDO\_PARCIAL.

**Criterios de Aceptación:**

* **Escenario 3.1: Consulta de Lote en Proceso (RECIBIDO\_EN\_PROCESO)**
  + **Dado** que existe un Lote en CO-01 con idLote "LoteXYZ" que contiene N documentos, de los cuales algunos están PROCESADO\_OK y otros aún EN\_PROCESO,
  + **Cuando** el Frontend (CO-05) realiza una llamada GET /api/lotes/{idLote}/status a CO-01 (Contrato API 1.3),
  + **Entonces** CO-01 responde con 200 OK y un JSON que refleja el estadoLote: RECIBIDO\_EN\_PROCESO, con conteos precisos de documentos por cada estado (PROCESADO\_OK, EN\_PROCESO, ERROR\_OCR, PENDIENTE), y resultadoOcrDisponible: false.
  + **Sub-Criterios/Notas:**
    - La información de estado de CO-01 es EN MEMORIA y temporal.
    - El Frontend puede realizar un *polling* periódico para actualizar esta vista.
* **Escenario 3.2: Consulta de Lote Completado Exitosamente (COMPLETADO\_OK)**
  + **Dado** que un Lote en CO-01 con idLote "LoteABC" ha tenido todos sus documentos procesados PROCESADO\_OK, y CO-01 ya ha notificado a n8n con el resultado consolidado,
  + **Cuando** el Frontend (CO-05) realiza una llamada GET /api/lotes/LoteABC/status a CO-01,
  + **Entonces** CO-01 responde con 200 OK y un JSON que indica el estadoLote: COMPLETADO\_OK, mostrando todos los documentos como PROCESADO\_OK, y resultadoOcrDisponible: true.
  + **Sub-Criterios/Notas:**
    - La visualización de los **resultados OCR detallados** por parte del Usuario Final (AE-01) se realiza consultando la Base de Datos Externa (AE-03) a través de n8n (AE-02), ya que CO-01 no persiste estos resultados. El Frontend puede usar la indicación resultadoOcrDisponible: true para saber cuándo debe consultar a n8n.
* **Escenario 3.3: Consulta de Lote con Errores Parciales (FALLIDO\_PARCIAL)**
  + **Dado** que un Lote en CO-01 con idLote "LoteXYZ" tiene algunos documentos PROCESADO\_OK y otros con ERROR\_OCR, y CO-01 ya ha notificado a n8n,
  + **Cuando** el Frontend (CO-05) realiza una llamada GET /api/lotes/LoteXYZ/status a CO-01,
  + **Entonces** CO-01 responde con 200 OK y un JSON que indica el estadoLote: FALLIDO\_PARCIAL, los conteos correctos de documentos por estado, y resultadoOcrDisponible: true.
* **Escenario 3.4: Lote No Encontrado (Expirado o Inexistente)**
  + **Dado** que el idLote "LoteInexistente" no corresponde a ningún lote activo en CO-01 (porque nunca existió, o porque su TTL en memoria ha expirado),
  + **Cuando** el Frontend (CO-05) realiza una llamada GET /api/lotes/LoteInexistente/status a CO-01,
  + **Entonces** CO-01 responde con 404 Not Found y un JSON de error estandarizado.
  + **Sub-Criterios/Notas:**
    - Este escenario es crucial para la Zero-Fault Detection y manejar la naturaleza EN MEMORIA de CO-01.

**4.3. Grupo: Procesamiento y Notificación de Resultados**

**HU-RES-001: Manejo de Resultados OCR Individuales y Notificación de Lote Consolidado a n8n**

* **ID de Historia:** HU-RES-001
* **Nombre de la Historia:** Consolidación de Resultados OCR y Notificación Final de Lote a n8n
* **Como** API Gateway / Backend Service (CO-01),
* **quiero** recibir los resultados OCR de cada Documento de Pago individual desde CO-04, y una vez que el Lote completo esté procesado, consolidar y notificar el resultado final a n8n,
* **para** asegurar que los datos extraídos por OCR (Resultado OCR) sean persistidos por n8n (AE-02) en la Base de Datos Externa (AE-03) y estén disponibles para los sistemas downstream.

**Criterios de Aceptación:**

* **Escenario 4.1: Recepción Exitosa de Resultado OCR Individual desde CO-04**
  + **Dado** que CO-04 ha completado exitosamente el procesamiento OCR de un Documento de Pago con idDocumento "DocX" y ha generado un Resultado OCR estructurado,
  + **Cuando** CO-04 realiza una llamada POST /api/internal/ocr\_results/callback a CO-01 (Contrato API 1.4) con el JSON del resultado y el idDocumento,
  + **Entonces** CO-01 recibe y valida el resultado, actualiza el estado del Documento de Pago asociado en su lote EN MEMORIA a PROCESADO\_OK, y actualiza el conteo de estados del Lote.
  + **Sub-Criterios/Notas:**
    - CO-01 no requiere una respuesta específica de CO-04 para este webhook, solo la confirmación de recepción (ej. 200 OK por parte de CO-01 a CO-04).
    - El Resultado OCR debe incluir confianzaGlobalOCR, referenciaPago, montoPago, fechaPago, horaPago, bancoBeneficiario, y otros datos relevantes extraídos, junto con idDocumentoAsociado y idLote para trazabilidad.
* **Escenario 4.2: Recepción de Resultado OCR Individual con Error desde CO-04**
  + **Dado** que CO-04 ha intentado procesar un Documento de Pago "DocY" pero ha detectado un ERROR\_OCR (ej. imagen ilegible, formato no soportado por OCR Engine),
  + **Cuando** CO-04 realiza una llamada POST /api/internal/ocr\_results/callback a CO-01 indicando el ERROR\_OCR para "DocY" con detalles del fallo,
  + **Entonces** CO-01 actualiza el estado del Documento de Pago en su lote EN MEMORIA a ERROR\_OCR, registra el detalle del error, y actualiza el conteo de estados del Lote.
  + **Sub-Criterios/Notas:**
    - Los detalles del error de OCR deben ser lo suficientemente granulares para ayudar en la depuración y potencial reprocesamiento manual.
* **Escenario 4.3: Lote Completo (Todos Éxitos) - Notificación Exitosa a n8n**
  + **Dado** que todos los Documentos de Pago de un Lote (ej. "LotePQR") han sido procesados y sus resultados (todos PROCESADO\_OK) han sido recibidos por CO-01 vía callback desde CO-04,
  + **Cuando** CO-01 detecta que el Lote está completo (todos los documentos tienen un estado final),
  + **Entonces** CO-01 consolida los resultados de todos los documentos del lote en un único JSON de formato empresarial (definido previamente en replit(1).md) y realiza una llamada POST /api/n8n/webhook/ocr\_results a n8n (AE-02) con dicho JSON (Contrato API 3.1), esperando una respuesta 200 OK de n8n. Finalmente, el estadoLote de "LotePQR" cambia a COMPLETADO\_OK.
  + **Sub-Criterios/Notas:**
    - El JSON consolidado debe incluir todos los campos extraídos y su respectiva confianzaGlobalOCR para cada documento.
    - La notificación a n8n debe ser el punto de salida principal de los datos procesados del sistema.
* **Escenario 4.4: Lote Completo (Errores Parciales) - Notificación Exitosa a n8n**
  + **Dado** que todos los Documentos de Pago de un Lote (ej. "LoteDEF") han sido procesados, con algunos PROCESADO\_OK y algunos con ERROR\_OCR,
  + **Cuando** CO-01 detecta que el Lote está completo,
  + **Entonces** CO-01 consolida los resultados del lote (incluyendo los documentos con error y sus respectivos mensajes de error) en un JSON de formato empresarial y realiza una llamada POST /api/n8n/webhook/ocr\_results a n8n (AE-02), esperando una respuesta 200 OK de n8n. El estadoLote de "LoteDEF" cambia a FALLIDO\_PARCIAL.
* **Escenario 4.5: Fallo en la Notificación a n8n (Resiliencia con Reintentos)**
  + **Dado** que un Lote está completo y CO-01 intenta notificar a n8n, pero n8n no responde (ej. timeout, servicio caído) o responde con un código de error 5xx,
  + **Cuando** CO-01 no recibe una respuesta 200 OK de n8n después de la notificación inicial,
  + **Entonces** CO-01 debe implementar un mecanismo de **reintento con backoff exponencial** (ej. 3 reintentos con esperas de 5s, 15s, 60s) para asegurar la entrega del resultado. Si después de todos los reintentos la notificación sigue fallando, CO-01 debe registrar un ERROR\_NOTIFICACION\_N8N crítico en sus logs y, si aplica, generar una alerta a un sistema de monitoreo.
  + **Sub-Criterios/Notas:**
    - Este escenario es una aplicación directa de la Zero-Fault Detection y Persistencia de Correcciones para la entrega de resultados.
    - El estadoLote en CO-01 debería permanecer en un estado transitorio (ej. NOTIFICACION\_PENDIENTE) hasta que la notificación a n8n sea exitosa o se agoten los reintentos.

**4.4. Grupo: Resiliencia y Gestión de Errores Internos**

**HU-RES-002: Manejo Robusto de Errores Internos y Degradación Elegante**

* **ID de Historia:** HU-RES-002
* **Nombre de la Historia:** Detección y Gestión de Fallos de Componentes Internos
* **Como** Sistema OCR Asíncrono Empresarial (representado por CO-01 y CO-04),
* **quiero** detectar, registrar y responder adecuadamente a errores internos inesperados, como problemas de comunicación o acceso a almacenamiento,
* **para** mantener la estabilidad del sistema, facilitar la depuración, minimizar el impacto en el procesamiento de otros lotes, y asegurar la Zero-Fault Detection.

**Criterios de Aceptación:**

* **Escenario 5.1: Fallo de Conexión o Almacenamiento a CO-06 desde CO-01**
  + **Dado** que CO-01 recibe un Documento de Pago para ingestión, pero el Servicio de Almacenamiento (CO-06) está temporalmente inaccesible o experimenta un error al subir el archivo,
  + **Cuando** CO-01 intenta almacenar la imagen original en CO-06,
  + **Entonces** CO-01 debe:
    1. Registrar un error detallado en sus logs internos (RNF-OBS-001).
    2. Responder un 500 Internal Server Error al origen de la solicitud (Frontend o n8n), indicando un fallo interno del servidor.
    3. No crear un idLote ni idDocumentoIngresado si el documento no pudo ser almacenado.
  + **Sub-Criterios/Notas:**
    1. Esto previene que CO-01 acepte un documento que no puede procesar desde el inicio, aplicando Zero-Fault Detection.
* **Escenario 5.2: CO-04 No Puede Descargar la Imagen desde CO-06**
  + **Dado** que CO-01 ha enviado una solicitud de procesamiento a CO-04 para un documento, pero CO-04 no puede descargar la imagen original desde CO-06 (ej. archivo corrompido, permisos revocados, error de red solo en CO-04),
  + **Cuando** CO-04 intenta descargar la imagen para iniciar el OCR,
  + **Entonces** CO-04 debe:
    1. Registrar el error (ERROR\_DESCARGA\_IMAGEN\_OCR\_WORKER) en sus logs.
    2. Notificar a CO-01 vía el callback (Contrato API 1.4) con este estado de error para el idDocumento específico, incluyendo un mensaje de error detallado.
    3. No intentar procesar el documento.
  + **Sub-Criterios/Notas:**
    1. CO-01 actualizará el estado del documento a ERROR\_DESCARGA\_IMAGEN\_OCR\_WORKER en su memoria, contribuyendo a la Trazabilidad del fallo.
* **Escenario 5.3: Errores en la Lógica de Orquestación de Lotes en CO-01**
  + **Dado** que CO-01 está gestionando un Lote y ocurre un error inesperado en su lógica interna (ej. desincronización de estados, fallo al actualizar conteos internos),
  + **Cuando** el error se produce,
  + **Entonces** CO-01 debe:
    1. Registrar un ERROR\_INTERNO\_SISTEMA detallado en sus logs.
    2. Intentar recuperarse si el error es transitorio.
    3. Si el error es persistente y afecta la coherencia del lote, el estadoLote podría transicionar a FALLIDO\_TOTAL (si todos los documentos se ven afectados) y CO-01 intentaría notificar a n8n con este estado.
  + **Sub-Criterios/Notas:**
    1. Los RNF-OBS (Observabilidad) son clave aquí para detectar y diagnosticar rápidamente estos fallos.

**5. Requisitos No Funcionales (RNF)**

Estos requisitos definen las cualidades esenciales del sistema.

**5.1. Rendimiento del Servicio (Performance)**

* **RNF-PERF-001:** El **API Gateway (CO-01)** debe responder a las solicitudes de ingestión de documentos (POST /api/upload y POST /api/n8n/webhook/document\_ingestion) en menos de **200 milisegundos (ms)** el **99%** de las veces, bajo una carga sostenida de **50 solicitudes concurrentes por segundo**.
* **RNF-PERF-002:** El **OCR Worker Service (CO-04)** debe procesar un Documento de Pago estándar (ej. imagen JPG de ~1MB, con texto típico) en un promedio de **5 segundos** por imagen, con un máximo de **8 segundos** el **95%** de las veces.
* **RNF-PERF-003:** La notificación de resultados individuales (POST /api/internal/ocr\_results/callback) de CO-04 a CO-01 debe tener una latencia de red promedio inferior a **50 ms**.
* **RNF-PERF-004:** La latencia entre la finalización del procesamiento de un lote en CO-01 y el inicio de la notificación a n8n (POST /api/n8n/webhook/ocr\_results) debe ser inferior a **1 segundo**.

**5.2. Escalabilidad**

* **RNF-ESCAL-001:** El **OCR Worker Service (CO-04)** debe ser capaz de escalar horizontalmente (añadiendo instancias) para soportar un aumento del **100%** en el volumen de documentos por procesar en un período de 2 horas sin degradación significativa del rendimiento (RNF-PERF-002).
* **RNF-ESCAL-002:** El **API Gateway (CO-01)** debe poder escalar horizontalmente (añadiendo instancias) para soportar un aumento del **50%** en el número de solicitudes de ingestión sin degradar los tiempos de respuesta (RNF-PERF-001).
* **RNF-ESCAL-003:** El **Servicio de Almacenamiento de Archivos (CO-06)** debe soportar el crecimiento de petabytes de datos de imágenes y logs, manteniendo los RNF de rendimiento de acceso.

**5.3. Disponibilidad**

* **RNF-DISP-001:** El API Gateway (CO-01) y el Servicio de Almacenamiento (CO-06) deben tener una disponibilidad del **99.95%** (menos de ~4.5 horas de inactividad anual) durante las horas de operación 24/7.
* **RNF-DISP-002:** El OCR Worker Service (CO-04) debe tener una disponibilidad del **99.5%** (menos de ~44 horas de inactividad anual), ya que los fallos individuales pueden ser mitigados por reintentos o reprocesamiento por n8n.

**5.4. Seguridad**

* **RNF-SEG-001:** Todas las comunicaciones entre los componentes internos (CO-01 a CO-04, CO-04 a CO-06, CO-01 a CO-06, CO-01 a AE-02/n8n) deben ser cifradas usando TLS/SSL (Transport Layer Security).
* **RNF-SEG-002:** El Frontend (CO-05) debe implementar un mecanismo de autenticación y autorización robusto para el Usuario Final (Operador/Administrador), usando roles definidos.
* **RNF-SEG-003:** El acceso al Servicio de Almacenamiento (CO-06) debe estar estrictamente limitado a los servicios autorizados (CO-01 y CO-04) mediante credenciales de acceso seguras (ej. tokens de acceso, roles IAM).
* **RNF-SEG-004:** El sistema no debe almacenar directamente datos sensibles extraídos del OCR (ej. números de cuenta bancaria) en sus componentes temporales (CO-01), sino pasarlos directamente a n8n para su gestión y persistencia segura en AE-03.

**5.5. Usabilidad (Aplicable a Frontend/AE-01)**

* **RNF-USAB-001:** El Frontend (CO-05) debe proporcionar retroalimentación visual clara e inmediata al Usuario Final (AE-01) sobre el estado de carga del documento y el progreso del lote.
* **RNF-USAB-002:** Los mensajes de error y validación en el Frontend deben ser descriptivos y orientar al Usuario Final sobre cómo corregir el problema.
* **RNF-USAB-003:** La interfaz de usuario debe ser intuitiva, requerir un entrenamiento mínimo para un operador nuevo (menos de 30 minutos).

**5.6. Observabilidad / Monitoreo**

* **RNF-OBS-001:** Todos los componentes (CO-01, CO-04) deben generar logs estructurados (ej. JSON) con niveles de severidad (INFO, WARN, ERROR, CRITICAL) para todas las operaciones clave, eventos, y especialmente errores.
* **RNF-OBS-002:** CO-04 debe capturar y enviar métricas de rendimiento (uso de CPU, memoria, GPU si aplica, tiempos de procesamiento por documento y por lote) al Servicio de Almacenamiento (CO-06) en formato de archivos CSV/resumen para un "monitoreo por pulso".
* **RNF-OBS-003:** El sistema debe integrar capacidades de monitoreo para alertar a los administradores sobre fallos críticos (ej. ERROR\_NOTIFICACION\_N8N, ERROR\_ALMACENAMIENTO) en un plazo de **5 minutos**.
* **RNF-OBS-004:** El estado EN MEMORIA de los lotes en CO-01 debe ser observable internamente para fines de diagnóstico, con un Time-To-Live (TTL) configurable para la limpieza de lotes inactivos.

**5.7. Mantenibilidad**

* **RNF-MANT-001:** La base de código de todos los componentes debe seguir estándares de codificación consistentes y tener una cobertura mínima de pruebas unitarias del **80%**.
* **RNF-MANT-002:** El sistema debe ser capaz de desplegar nuevas versiones de los componentes (CO-01, CO-04) con **cero tiempo de inactividad (zero-downtime deployment)**.
* **RNF-MANT-003:** Las Reglas de Extracción utilizadas por CO-04 deben ser gestionables externamente (ej. archivos de configuración) sin requerir el red despliegue del código del Worker.
* **RNF-MANT-004:** Se debe mantener una documentación técnica actualizada (diagramas de arquitectura, modelos de datos, flujos) para cada componente del sistema.

**5.8. Resiliencia (Cross-Cutting)**

* **RNF-RES-001:** El sistema debe ser tolerante a fallos transitorios de red o servicios externos (ej. n8n inactivo temporalmente, CO-06 con alta latencia) mediante mecanismos de **reintentos con backoff exponencial** en las interacciones críticas.
* **RNF-RES-002:** Cualquier error inesperado en cualquier componente (CO-01, CO-04) debe ser capturado, loggeado y manejado de manera que no cause la caída completa del servicio ni la pérdida de datos ya procesados (aplicación de Zero-Fault Detection).
* **RNF-RES-003:** El sistema debe tener un mecanismo para re-procesar documentos o lotes que hayan terminado en un estado de error (ERROR\_OCR, ERROR\_DESCARGA\_IMAGEN) a través de la intervención de n8n o un proceso manual iniciado por el Usuario Final (AE-01) (requiere que n8n tenga la capacidad de re-ingestar).

Este documento de Especificaciones de Requisitos/Historias de Usuario es un activo fundamental. Proporciona la claridad, granularidad y trazabilidad necesarias para construir un sistema robusto, alineado con las expectativas de negocio y capaz de alcanzar la Perfección Continua y la Integridad Total.

**7. Flujo de Datos Detallado**

El movimiento de la información a través del sistema, desde la ingestión hasta la consolidación de resultados, se representa en el diagrama de flujo de datos. Este diagrama ilustra qué datos se mueven, dónde se transforman y dónde se almacenan temporalmente.

**DIAGRAMA DE FLUJO DE DATOS (DFD) DETALLADO**

**Sistema: Sistema OCR Asíncrono Empresarial**

**1. Introducción y Propósito del Documento**

Este documento presenta el Diagrama de Flujo de Datos (DFD) Detallado para el "Sistema OCR Asíncrono Empresarial". El DFD es una herramienta esencial para visualizar el movimiento de la información a través del sistema, mostrando *qué* datos se mueven, *dónde* se transforman y *dónde* se almacenan. Se enfoca en el flujo lógico de los datos, no en la secuencia temporal de los eventos (que se aborda en el Diagrama de Secuencia).

Este DFD detallado es crucial para:

* Proporcionar una Claridad de Definiciones inequívoca sobre los flujos de datos.
* Asegurar la Coherencia de Referencias Críticas con el Modelo de Dominio Conceptual y los Contratos de API Formales.
* Facilitar la Comprensión Profunda del Contexto de Dominio a nivel de datos.
* Identificar posibles cuellos de botella o redundancias en el flujo de información.

**2. Elementos del DFD**

El Diagrama de Flujo de Datos utiliza símbolos estándar para representar sus componentes:

* Entidades Externas (Actores): Representadas como [Nombre de la Entidad]. Son fuentes o destinos de datos que se encuentran fuera de los límites del sistema que estamos analizando.
* Procesos: Representados como {P#. Nombre del Proceso}. Son transformaciones lógicas de datos. En este DFD, los procesos corresponden a los componentes principales de nuestro sistema.
* Almacenes de Datos (Data Stores): Representados como [DS#. Nombre del Almacén]. Son lugares donde los datos se guardan, ya sea de forma temporal o persistente.
* Flujos de Datos: Representados como Nombre del Flujo --->. Indican el movimiento de datos entre las entidades externas, los procesos y los almacenes de datos.

**3. Diagrama Visual del Flujo de Datos**

graph TD

subgraph Actores Externos

AE01[Usuario Final (Operador/Administrador)]

AE02[Sistema n8n]

AE03[Base de Datos Externa (de n8n)]

end

subgraph Sistema OCR Asíncrono Empresarial

CO05{Frontend (CO-05)}

CO01{API Gateway (CO-01)}

CO04{OCR Worker (CO-04)}

CO06[Servicio de Almacenamiento (CO-06)]

end

DS1[(Lotes en Memoria en CO-01)]

DS2[Almacenamiento de Archivos (CO-06)]

DS3[Base de Datos Externa (AE-03)]

%% Flujo de Ingestión de Documentos

AE01 -- "Documento de Pago (Imagen) + metadatosEntrada (JSON: idSorteo, fechaSorteo, numeroLlegada, idWhatsapp, horaMinutoN8nIngreso, caption, nombre)" --> CO05

CO05 -- "POST /api/upload: Documento de Pago (File) + metadatosEntrada (JSON)" --> CO01

AE02 -- "POST /api/n8n/webhook/document\_ingestion: URL Imagen + metadatosEntrada (JSON: idSorteo, fechaSorteo, numeroLlegada, idDocumento, idWhatsapp, horaMinutoN8nIngreso, caption, nombre)" --> CO01

CO01 -- "202 Accepted: idLote, idDocumentoIngresado, estadoLote (RECIBIDO\_EN\_PROCESO)" --> CO05

CO05 -- "Confirmación de Carga: idLote, idDocumentoIngresado, Estado Inicial Lote" --> AE01

CO01 -- "202 Accepted: idLote, idDocumentoIngresado, estadoLote (RECIBIDO\_EN\_PROCESO)" --> AE02

%% Flujo de Procesamiento Interno

CO01 -- "Almacenar: Documento de Pago (Imagen Binaria) con nombreRenombradoArchivo" --> CO06

CO01 -- "POST /process\_document: idDocumento, URL Imagen en CO-06, metadatosEntrada, callbackUrl" --> CO04

CO01 <--> "Gestión de Lotes: Creación, Actualización de Estado (Documento/Lote), Conteo de Progreso" DS1

DS1 -- "Estado Documento/Lote (EN MEMORIA)" --> CO01

CO04 -- "Descargar: URL Imagen en CO-06 (para procesamiento)" --> CO06

CO04 -- "Guardar: Métricas de Rendimiento (CPU, Memoria, Tiempo), Logs Resumen/CSV (por pulso)" --> CO06

%% Flujo de Resultados y Monitoreo

CO04 -- "POST /api/internal/ocr\_results/callback: Resultado OCR (JSON enriquecido: referenciaPago, montoPago, fechaPago, horaPago, bancoBeneficiario, confianzaGlobalOCR, estadoValidacion, metadatosEntradaOriginal) + idDocumento + Estado Procesamiento (PROCESADO\_OK/ERROR\_OCR)" --> CO01

CO05 -- "GET /api/lotes/{idLote}/status: Solicitud de Estado" --> CO01

CO01 -- "200 OK: Estado Detallado del Lote (JSON: idLote, estadoLote, totalDocumentos, progresoPorcentaje, detallesDocumentos)" --> CO05

CO05 -- "Visualización: Estado Detallado del Lote" --> AE01

CO01 -- "POST [n8n\_webhook\_url]: Resultado Consolidado del Lote (JSON: idLote, estadoLoteFinal, totalDocumentos, resultadosConsolidados[], mensajeFinal, fechaHoraFinalizacion)" --> AE02

AE02 -- "Persistir: Resultado Consolidado del Lote (JSON)" --> AE03

%% Estilos para mejor visualización

style AE01 fill:#f9f,stroke:#333,stroke-width:2px

style AE02 fill:#add8e6,stroke:#333,stroke-width:2px

style AE03 fill:#90ee90,stroke:#333,stroke-width:2px

style CO05 fill:#f0f8ff,stroke:#333,stroke-width:2px

style CO01 fill:#ffe4e1,stroke:#333,stroke-width:2px

style CO04 fill:#fffacd,stroke:#333,stroke-width:2px

style CO06 fill:#ffe4b5,stroke:#333,stroke-width:2px

style DS1 fill:#dff,stroke:#333,stroke-width:2px

style DS2 fill:#e6e6fa,stroke:#333,stroke-width:2px

style DS3 fill:#ccffcc,stroke:#333,stroke-width:2px

**4. Flujos de Datos Detallados**

A continuación, se describen los flujos de datos numerados, enlazando los elementos del DFD y haciendo referencia a los Contratos de API Formales y el Modelo de Dominio Conceptual.

A. Flujo de Ingestión de Documentos:

1. [AE-01 Usuario Final] --- "Documento de Pago (Imagen) + metadatosEntrada" ---> {CO05 Frontend}
   * Descripción: El Usuario Final (AE-01) carga un archivo de imagen (Documento de Pago) y proporciona los metadatosEntrada requeridos (ej. idSorteo, fechaSorteo, numeroLlegada, idWhatsapp, horaMinutoN8nIngreso, caption, nombre) a través de la interfaz web del Frontend (CO-05).
   * Referencia: Modelo de Dominio Conceptual - Entidad Documento de Pago, Metadatos de Entrada.
2. {CO05 Frontend} --- "POST /api/upload: Documento de Pago (File) + metadatosEntrada (JSON)" ---> {CO01 API Gateway}
   * Descripción: El Frontend (CO-05) reenvía el Documento de Pago (como un archivo binario) y los metadatosEntrada (como JSON serializado) al API Gateway (CO-01).
   * Referencia: Contratos de API Formales - 1.1. Ingestión de Documento Individual desde Frontend (CO-05).
3. [AE-02 Sistema n8n] --- "POST /api/n8n/webhook/document\_ingestion: URL Imagen + metadatosEntrada (JSON)" ---> {CO01 API Gateway}
   * Descripción: El Sistema n8n (AE-02) envía una solicitud de ingestión asíncrona a CO-01, proporcionando una URL Imagen (donde CO-01 puede descargar el Documento de Pago) y los metadatosEntrada (incluyendo idDocumento de n8n para trazabilidad).
   * Referencia: Contratos de API Formales - 1.2. Ingestión de Documento Individual desde Sistema n8n (AE-02).
4. {CO01 API Gateway} --- "202 Accepted: idLote, idDocumentoIngresado, estadoLote (RECIBIDO\_EN\_PROCESO)" ---> {CO05 Frontend}
   * Descripción: CO-01 responde al Frontend (CO-05) con los identificadores generados (idLote, idDocumentoIngresado) y el estado inicial del lote (RECIBIDO\_EN\_PROCESO), confirmando la recepción y el inicio del procesamiento asíncrono.
   * Referencia: Contratos de API Formales - 1.1. Ingestión de Documento Individual desde Frontend (CO-05) - Respuesta 202 Accepted.
5. {CO05 Frontend} --- "Confirmación de Carga: idLote, idDocumentoIngresado, Estado Inicial Lote" ---> [AE-01 Usuario Final]
   * Descripción: CO-05 muestra al Usuario Final (AE-01) la confirmación visual de la carga exitosa y los identificadores del documento/lote para su monitoreo.
6. {CO01 API Gateway} --- "202 Accepted: idLote, idDocumentoIngresado, estadoLote (RECIBIDO\_EN\_PROCESO)" ---> [AE-02 Sistema n8n]
   * Descripción: CO-01 responde a n8n con los identificadores del lote y documento y la confirmación de que el documento ha sido recibido y el procesamiento asíncrono ha iniciado.
   * Referencia: Contratos de API Formales - 1.2. Ingestión de Documento Individual desde Sistema n8n (AE-02) - Respuesta 202 Accepted.

B. Flujo de Procesamiento Interno:

1. {CO01 API Gateway} --- "Almacenar: Documento de Pago (Imagen Binaria) con nombreRenombradoArchivo" ---> [CO06 Servicio de Almacenamiento]
   * Descripción: CO-01 (después de recibir la imagen directamente o descargarla de la URL de n8n) almacena el Documento de Pago (imagen binaria) en el Servicio de Almacenamiento (CO-06), utilizando el nombreRenombradoArchivo generado.
   * Referencia: Modelo de Dominio Conceptual - Entidad Documento de Pago (nombreRenombradoArchivo, rutaAlmacenamientoOriginal).
2. {CO01 API Gateway} --- "POST /process\_document: idDocumento, URL Imagen en CO-06, metadatosEntrada, callbackUrl" ---> {CO04 OCR Worker}
   * Descripción: CO-01 invoca directamente al OCR Worker Service (CO-04) para procesar un documento individual, pasándole el idDocumento, la URL Imagen en CO-06 (para que CO-04 la descargue), los metadatosEntrada originales y la callbackUrl para la respuesta.
   * Referencia: Contratos de API Formales - 2.1. Inicio de Procesamiento de Documento Individual (Invocación desde CO-01).
3. {CO01 API Gateway} <--- "Gestión de Lotes: Creación, Actualización de Estado (Documento/Lote), Conteo de Progreso" --- [DS1 Lotes en Memoria (en CO-01)]
   * Descripción: CO-01 gestiona y actualiza el estado de los Documentos de Pago y Lotes en su almacenamiento EN MEMORIA (DS1). Esto incluye la creación de nuevos lotes, la adición de documentos, y la actualización del progreso y estado de cada documento y del lote general.
   * Referencia: Modelo de Dominio Conceptual - Entidad Lote (estadoLote, timestampCreacion). GLOSARIO DE TÉRMINOS CLAVE - Lote, Persistencia Temporal.
4. [DS1 Lotes en Memoria (en CO-01)] --- "Estado Documento/Lote (EN MEMORIA)" ---> {CO01 API Gateway}
   * Descripción: CO-01 consulta y utiliza la información de estado de los lotes y documentos almacenada EN MEMORIA para su lógica de orquestación y para responder a las solicitudes de estado del Frontend.
5. {CO04 OCR Worker} --- "Descargar: URL Imagen en CO-06 (para procesamiento)" ---> [CO06 Servicio de Almacenamiento]
   * Descripción: El OCR Worker (CO-04) descarga el Documento de Pago (imagen) desde el Servicio de Almacenamiento (CO-06) utilizando la URL Imagen proporcionada por CO-01.
   * Referencia: Contratos de API Formales - 2.1. Inicio de Procesamiento de Documento Individual (Invocación desde CO-01).
6. {CO04 OCR Worker} --- "Guardar: Métricas de Rendimiento (CPU, Memoria, Tiempo), Logs Resumen/CSV (por pulso)" ---> [CO06 Servicio de Almacenamiento]
   * Descripción: El OCR Worker (CO-04) genera y guarda métricas detalladas de rendimiento (uso de CPU, memoria, tiempo de procesamiento por documento/lote) y logs de resumen en formato CSV en el Servicio de Almacenamiento (CO-06). Esto es para el Monitoreo por Pulso.
   * Referencia: ESPECIFICACIONES DE REQUISITOS - RNF-OBS-002. GLOSARIO DE TÉRMINOS CLAVE - Monitoreo por Pulso, Persistencia Temporal.

C. Flujo de Resultados y Monitoreo:

1. {CO04 OCR Worker} --- "POST /api/internal/ocr\_results/callback: Resultado OCR (JSON enriquecido) + idDocumento + Estado Procesamiento (PROCESADO\_OK/ERROR\_OCR)" ---> {CO01 API Gateway}
   * Descripción: Una vez que CO-04 ha procesado un Documento de Pago, envía el Resultado OCR (JSON enriquecido con datos extraídos, confianza, validación y metadatosEntradaOriginal) junto con el idDocumento y su Estado Procesamiento (PROCESADO\_OK o ERROR\_OCR) de vuelta a CO-01 vía callback.
   * Referencia: Contratos de API Formales - 1.4. Recepción de Resultados OCR Individuales desde OCR Worker Service (CO-04). Modelo de Dominio Conceptual - Entidad Resultado OCR.
2. {CO05 Frontend} --- "GET /api/lotes/{idLote}/status: Solicitud de Estado" ---> {CO01 API Gateway}
   * Descripción: El Frontend (CO-05) solicita el estado actual de un Lote específico a CO-01 para su visualización en el dashboard.
   * Referencia: Contratos de API Formales - 1.3. Consulta de Estado de Lote por Frontend (CO-05).
3. {CO01 API Gateway} --- "200 OK: Estado Detallado del Lote (JSON: idLote, estadoLote, totalDocumentos, progresoPorcentaje, detallesDocumentos)" ---> {CO05 Frontend}
   * Descripción: CO-01 responde al Frontend (CO-05) con un JSON que contiene el estado detallado del Lote (estadoLote, progresoPorcentaje, detallesDocumentos por estado), reflejando la información EN MEMORIA.
   * Referencia: Contratos de API Formales - 1.3. Consulta de Estado de Lote por Frontend (CO-05) - Respuesta 200 OK.
4. {CO05 Frontend} --- "Visualización: Estado Detallado del Lote" ---> [AE01 Usuario Final]
   * Descripción: CO-05 presenta la información del estado del lote al Usuario Final (AE-01) en el dashboard, permitiendo el monitoreo en tiempo casi real.
5. {CO01 API Gateway} --- "POST [n8n\_webhook\_url]: Resultado Consolidado del Lote (JSON)" ---> [AE02 Sistema n8n]
   * Descripción: Una vez que CO-01 detecta que un Lote ha completado su procesamiento (todos los documentos tienen un estado final), consolida todos los resultados y métricas en un Resultado Consolidado del Lote (JSON) y lo envía a n8n vía webhook.
   * Referencia: Contratos de API Formales - 3.1. Recepción de Resultado Consolidado del Lote y Estado Lote (Invocación desde CO-01). Modelo de Dominio Conceptual - Entidad Resultado Consolidado del Lote.
6. [AE02 Sistema n8n] --- "Persistir: Resultado Consolidado del Lote (JSON)" ---> [AE03 Base de Datos Externa (de n8n)]
   * Descripción: n8n recibe el Resultado Consolidado del Lote y lo persiste en la Base de Datos Externa (AE-03), que es la Fuente de Verdad Externa y el repositorio a largo plazo de los resultados OCR.
   * Referencia: GLOSARIO DE TÉRMINOS CLAVE - Fuente de Verdad Externa.

**8. Diagrama de Secuencia de Procesos**

La secuencia temporal de los eventos y las interacciones entre los componentes del sistema se detalla en el diagrama de secuencia de procesos. Este diagrama es crucial para entender el comportamiento asíncrono y la orquestación de las operaciones clave.

**DIAGRAMA DE SECUENCIA DE PROCESOS (Nivel Operacional)**

**Sistema: Sistema OCR Asíncrono Empresarial**

**1. Introducción y Propósito del Documento**

Este documento detalla las **secuencias de mensajes e interacciones** entre los componentes del "Sistema OCR Asíncrono Empresarial" para las funcionalidades clave. A diferencia del DFD (que se enfoca en el flujo de datos), el Diagrama de Secuencia se centra en el **orden temporal de los eventos** y las llamadas entre los participantes, mostrando el "cómo" y el "cuándo" de las operaciones.

Este diagrama es crucial para:

* Clarificar la lógica de orquestación y el comportamiento asíncrono.
* Identificar posibles cuellos de botella o dependencias temporales.
* Servir como una guía precisa para la implementación del código, asegurando la Granularidad.
* Facilitar la depuración y el entendimiento del comportamiento del sistema.
* Validar la Interface Excellence de los Contratos de API Formales en acción y la Zero-Fault Detection en los flujos.

**2. Convenciones del Diagrama de Secuencia**

Se utiliza la notación estándar de Diagramas de Secuencia (Mermaid), con los siguientes elementos clave:

* **Participantes (Lifelines):** Representan los objetos o componentes que interactúan (ej. AE-01, CO-01).
* **Mensajes:** Flechas que indican la comunicación entre participantes. Pueden ser síncronos (flecha sólida) o asíncronos (flecha abierta).
* **Activación (Activation Bar):** Barras verticales que indican el período de tiempo durante el cual un participante está ejecutando una acción.
* **Bucles (Loop):** Indican que una secuencia de mensajes se repite.
* **Alternativas (Alt):** Muestran diferentes caminos posibles en el flujo (ej. éxito vs. error).
* **Notas:** Información adicional para clarificar el comportamiento, incluyendo referencias a Contratos de API Formales y Modelo de Dominio Conceptual.

**3. Escenarios de Secuencia Clave**

3.1. Escenario: Ingestión de Documento Individual desde Frontend y Disparo a OCR Worker

Este diagrama ilustra el proceso completo desde que un Usuario Final carga un documento hasta que CO-01 lo almacena y dispara su procesamiento individual al OCR Worker.

sequenceDiagram  
    participant AE01 as Usuario Final (AE-01)  
    participant CO05 as Frontend (CO-05)  
    participant CO01 as API Gateway (CO-01)  
    participant CO06 as Almacenamiento (CO-06)  
    participant CO04 as OCR Worker (CO-04)  
  
    AE01->>CO05: 1. Cargar Documento de Pago (Imagen + Metadatos como JSON)  
    note right of CO05: Metadatos incluyen idSorteo, fechaSorteo, numeroLlegada, idWhatsapp, horaMinutoN8nIngreso, caption, nombre.  
    CO05->>CO01: 2. POST /api/upload (File: Documento de Pago, JSON: metadatosEntrada)  
    activate CO01  
    note right of CO01: Valida metadatos y tipo de archivo (Contrato API 1.1).  
    CO01->>CO06: 3. Almacenar Imagen (nombreRenombradoArchivo.ext)  
    activate CO06  
    CO06-->>CO01: 4. Confirmación de Almacenamiento (URL/Path de imagen en CO-06)  
    deactivate CO06  
    CO01->>CO01: 5. Asignar Doc a Lote EN MEMORIA  
    note right of CO01: Crea/Identifica Lote (idLote) y asigna idDocumentoIngresado. Actualiza estado Lote en DS1 (Lotes en Memoria).  
    CO01-->>CO05: 6. 202 Accepted (JSON: idLote, idDocumentoIngresado, estadoLote: RECIBIDO\_EN\_PROCESO)  
    deactivate CO01  
    CO05-->>AE01: 7. Mostrar Confirmación de Carga y Lote/Documento ID  
    CO01->>CO04: 8. POST /process\_document (JSON: idLote, idDocumento, urlImagenAlmacenamiento, metadatosEntrada, callbackUrl)  
    activate CO04  
    note right of CO04: Invocación directa a CO-04 para procesamiento asíncrono (Contrato API 2.1).  
    CO04-->>CO01: 9. 202 Accepted (JSON: mensaje: "Procesamiento iniciado")  
    deactivate CO04  
    note right of CO04: CO-04 inicia procesamiento asíncrono para el documento.

**Explicación Detallada del Flujo 3.1:**

1. **Carga Inicial (AE01 -> CO05):** El Usuario Final (AE-01) interactúa con el Frontend (CO-05) para seleccionar un Documento de Pago (imagen) y proporcionar los metadatosEntrada asociados (ej. idSorteo, idWhatsapp, horaMinutoN8nIngreso).
2. **Envío a API Gateway (CO05 -> CO01):** CO05 realiza una llamada POST al endpoint /api/upload de CO-01 (Contrato API 1.1), enviando el archivo de imagen y los metadatosEntrada serializados en JSON.
3. **Almacenamiento de Imagen (CO01 -> CO06):** CO-01 recibe el documento, realiza validaciones iniciales de los metadatos y el tipo de archivo. Luego, almacena el Documento de Pago (imagen binaria) en el Servicio de Almacenamiento (CO-06).
4. **Confirmación de Almacenamiento (CO06 -> CO01):** CO-06 responde a CO-01 con una confirmación de que la imagen ha sido almacenada, proporcionando su URL o ruta interna.
5. **Gestión de Lotes en Memoria (CO01 -> CO01):** CO-01 utiliza los metadatosEntrada (idSorteo, fechaSorteo, numeroLlegada) para identificar si el documento pertenece a un Lote existente EN MEMORIA o si necesita crear uno nuevo. Genera un idDocumentoIngresado único para el documento y lo asocia al lote. El estado del lote se actualiza internamente (DS1).
6. **Respuesta al Frontend (CO01 -> CO05):** CO-01 responde a CO-05 con un 202 Accepted, proporcionando el idLote y el idDocumentoIngresado, y confirmando que el documento ha sido recibido y el procesamiento asíncrono ha iniciado.
7. **Confirmación al Usuario (CO05 -> AE01):** CO05 muestra al Usuario Final una confirmación visual de la carga exitosa, incluyendo los IDs generados.
8. **Disparo a OCR Worker (CO01 -> CO04):** CO-01 invoca directamente al OCR Worker Service (CO-04) (POST /process\_document, Contrato API 2.1), pasándole el idLote, idDocumento, la urlImagenAlmacenamiento en CO-06, los metadatosEntrada originales y la callbackUrl para que CO-04 le devuelva el resultado.
9. **Confirmación de Inicio (CO04 -> CO01):** CO-04 responde a CO-01 con un 202 Accepted, indicando que ha recibido la solicitud y ha iniciado el procesamiento asíncrono del documento.

3.2. Escenario: Ingestión de Documento Individual desde n8n y Disparo a OCR Worker

Este diagrama muestra cómo n8n envía un documento por URL y CO-01 lo procesa, similar al frontend pero con descarga de imagen.

sequenceDiagram  
    participant AE02 as Sistema n8n (AE-02)  
    participant CO01 as API Gateway (CO-01)  
    participant CO06 as Almacenamiento (CO-06)  
    participant CO04 as OCR Worker (CO-04)  
  
    AE02->>CO01: 1. POST /api/n8n/webhook/document\_ingestion (JSON: urlImagen, metadatos\_json)  
    activate CO01  
    note right of CO01: Valida metadatos (Contrato API 1.2).  
    CO01->>CO06: 2. Descargar Imagen desde urlImagen y Almacenar (nombreRenombradoArchivo.ext)  
    activate CO06  
    CO06-->>CO01: 3. Confirmación de Almacenamiento (URL/Path de imagen en CO-06)  
    deactivate CO06  
    CO01->>CO01: 4. Asignar Doc a Lote EN MEMORIA  
    note right of CO01: Crea/Identifica Lote (idLote) y asigna idDocumentoIngresado. Actualiza estado Lote en DS1 (Lotes en Memoria).  
    CO01-->>AE02: 5. 202 Accepted (JSON: idLote, idDocumentoIngresado, estadoLote: RECIBIDO\_EN\_PROCESO)  
    deactivate CO01  
    CO01->>CO04: 6. POST /process\_document (JSON: idLote, idDocumento, urlImagenAlmacenamiento, metadatosEntrada, callbackUrl)  
    activate CO04  
    note right of CO04: Invocación directa a CO-04 para procesamiento asíncrono (Contrato API 2.1).  
    CO04-->>CO01: 7. 202 Accepted (JSON: mensaje: "Procesamiento iniciado")  
    deactivate CO04  
    note right of CO04: CO-04 inicia procesamiento asíncrono para el documento.

**Explicación Detallada del Flujo 3.2:**

1. **Envío desde n8n (AE02 -> CO01):** El Sistema n8n (AE-02) envía una solicitud POST al endpoint /api/n8n/webhook/document\_ingestion de CO-01 (Contrato API 1.2). Esta solicitud incluye una urlImagen y los metadatosEntrada en formato JSON, incluyendo un idDocumento propio de n8n para trazabilidad.
2. **Descarga y Almacenamiento (CO01 -> CO06):** CO-01 valida los metadatos recibidos. Luego, procede a descargar la imagen del Documento de Pago desde la urlImagen proporcionada y la almacena en el Servicio de Almacenamiento (CO-06).
3. **Confirmación de Almacenamiento (CO06 -> CO01):** CO-06 responde a CO-01 confirmando el almacenamiento de la imagen.
4. **Gestión de Lotes en Memoria (CO01 -> CO01):** Similar al escenario de Frontend, CO-01 asigna el documento a un Lote EN MEMORIA (creando uno si es necesario) y actualiza su estado. El idDocumento de n8n se mantiene para la Coherencia de Referencias Críticas.
5. **Respuesta a n8n (CO01 -> AE02):** CO-01 responde a n8n con un 202 Accepted, confirmando la recepción y el inicio del procesamiento asíncrono.
6. **Disparo a OCR Worker (CO01 -> CO04):** CO-01 invoca directamente a CO-04 (POST /process\_document, Contrato API 2.1) para iniciar el procesamiento OCR de este documento individual, pasándole la urlImagenAlmacenamiento en CO-06, los metadatosEntrada y la callbackUrl.
7. **Confirmación de Inicio (CO04 -> CO01):** CO-04 responde a CO-01 con un 202 Accepted, indicando que ha iniciado la tarea de procesamiento.

3.3. Escenario: Procesamiento por OCR Worker y Callback de Resultado Individual

Este diagrama detalla el proceso interno del OCR Worker y cómo devuelve el resultado a CO-01.

sequenceDiagram  
    participant CO01 as API Gateway (CO-01)  
    participant CO04 as OCR Worker (CO-04)  
    participant CO06 as Almacenamiento (CO-06)  
  
    CO01->>CO04: 1. POST /process\_document (Invocación previa desde CO-01)  
    activate CO04  
    CO04->>CO06: 2. Descargar Imagen (urlImagenAlmacenamiento)  
    activate CO06  
    CO06-->>CO04: 3. Imagen Descargada (binario)  
    deactivate CO06  
    CO04->>CO04: 4. Procesar OCR (Motor OnnxTR, Reglas de Extracción, Procesamiento Espacial, Validación OCR)  
    note right of CO04: Aplica lógica de negocio y extrae datos (Modelo de Dominio: Resultado OCR).  
    alt Procesamiento Exitoso  
        CO04->>CO01: 5. POST /api/internal/ocr\_results/callback (JSON: Resultado OCR completo, estadoProcesamiento: PROCESADO\_OK)  
        activate CO01  
        note left of CO01: CO-01 actualiza el estado del documento en su Lote EN MEMORIA (DS1).  
        CO01-->>CO04: 6. 200 OK  
        deactivate CO01  
        CO04->>CO04: 7. Registrar métricas de rendimiento (CPU, Memoria, Tiempo) en logs  
        note right of CO04: Para Monitoreo por Pulso (RNF-OBS-002).  
        CO04->>CO06: 8. Subir Logs/CSV de Rendimiento (nombreDoc\_metrics.csv)  
        activate CO06  
        CO06-->>CO04: 9. Confirmación de Carga de Logs  
        deactivate CO06  
    else Procesamiento con Error  
        CO04->>CO01: 5. POST /api/internal/ocr\_results/callback (JSON: resultadoOcr=null, estadoProcesamiento: ERROR\_OCR, mensajeError)  
        activate CO01  
        note left of CO01: CO-01 actualiza el estado del documento en su Lote EN MEMORIA a ERROR\_OCR (DS1).  
        CO01-->>CO04: 6. 200 OK  
        deactivate CO01  
        CO04->>CO04: 7. Registrar error detallado en logs  
        CO04->>CO06: 8. Subir Logs/CSV de Errores (nombreDoc\_errors.csv)  
        activate CO06  
        CO06-->>CO04: 9. Confirmación de Carga de Logs  
        deactivate CO06  
    end  
    deactivate CO04

**Explicación Detallada del Flujo 3.3:**

1. **Invocación del OCR Worker (CO01 -> CO04):** CO-04 recibe la solicitud de procesamiento de un documento individual desde CO-01 (paso 8 del Escenario 3.1 o paso 6 del Escenario 3.2).
2. **Descarga de Imagen (CO04 -> CO06):** CO-04 utiliza la urlImagenAlmacenamiento para descargar el Documento de Pago (imagen) desde CO-06.
3. **Confirmación de Descarga (CO06 -> CO04):** CO-06 confirma la descarga de la imagen a CO-04.
4. **Procesamiento OCR Interno (CO04 -> CO04):** CO-04 ejecuta el motor OCR (ej. OnnxTR), aplica las Reglas de Extracción (Modelo de Dominio), realiza el Procesamiento Espacial y la Validación OCR para extraer y estructurar los datos del Documento de Pago en un Resultado OCR.
5. **Alternativa (Éxito/Error) y Callback a CO01:**

* **Procesamiento Exitoso:** CO-04 envía el Resultado OCR completo y el estado PROCESADO\_OK a CO-01 vía POST /api/internal/ocr\_results/callback (Contrato API 1.4). CO-01 actualiza el estado del documento en su Lote EN MEMORIA y responde 200 OK. CO-04 registra métricas de rendimiento y las sube a CO-06 como logs/CSV para Monitoreo por Pulso.
* **Procesamiento con Error:** Si CO-04 encuentra un error (ej. imagen ilegible), envía un Resultado OCR nulo (o parcial) con el estado ERROR\_OCR y un mensajeError a CO-01. CO-01 actualiza el estado del documento a ERROR\_OCR en su Lote EN MEMORIA y responde 200 OK. CO-04 registra el error en sus logs y los sube a CO-06.

3.4. Escenario: Consolidación de Lote y Notificación Final a n8n

Este diagrama muestra cómo CO-01 consolida los resultados de un lote y los envía a n8n.

sequenceDiagram  
    participant CO01 as API Gateway (CO-01)  
    participant AE02 as Sistema n8n (AE-02)  
    participant AE03 as Base de Datos Externa (AE-03)  
  
    note left of CO01: CO-01 ha recibido todos los resultados individuales de los documentos del Lote (vía callbacks de CO-04).  
    CO01->>CO01: 1. Detectar Lote Completo (EN MEMORIA - DS1)  
    note right of CO01: Todos los documentos tienen un estado final (PROCESADO\_OK o ERROR\_OCR).  
    CO01->>CO01: 2. Consolidar Resultado Consolidado del Lote (JSON)  
    note right of CO01: Agrega resultados OCR individuales, métricas de rendimiento del lote, y estado final (Modelo de Dominio: Resultado Consolidado del Lote).  
    alt Notificación Exitosa a n8n (RNF-RES-001)  
        CO01->>AE02: 3. POST [n8n\_webhook\_url] (JSON: Resultado Consolidado del Lote)  
        activate AE02  
        note right of AE02: n8n procesa el webhook (Contrato API 3.1).  
        AE02->>AE03: 4. Persistir Resultado Consolidado del Lote  
        activate AE03  
        AE03-->>AE02: 5. Confirmación de Persistencia  
        deactivate AE03  
        AE02-->>CO01: 6. 200 OK  
        deactivate AE02  
        CO01->>CO01: 7. Eliminar Lote de Memoria (DS1)  
        note right of CO01: Libera recursos; n8n es la Fuente de Verdad Externa.  
    else Fallo en la Notificación a n8n (RNF-RES-001, Zero-Fault Detection)  
        CO01->>AE02: 3. POST [n8n\_webhook\_url] (Intento 1)  
        activate AE02  
        AE02--xCO01: 4. Error de Conexión/Respuesta (ej. Timeout, 5xx)  
        deactivate AE02  
        loop Reintentos con Backoff Exponencial (RNF-RES-001)  
            CO01->>AE02: 5. POST [n8n\_webhook\_url] (Intento N)  
            activate AE02  
            AE02--xCO01: 6. Error de Conexión/Respuesta  
            deactivate AE02  
        end  
        CO01->>CO01: 7. Registrar ERROR\_NOTIFICACION\_N8N Crítico (RNF-OBS-003)  
        note right of CO01: Requiere intervención manual/externa. Lote puede permanecer en memoria por un TTL extendido para diagnóstico.  
    end

**Explicación Detallada del Flujo 3.4:**

1. **Detección de Lote Completo (CO01 -> CO01):** CO-01 monitorea el estado de los lotes EN MEMORIA (DS1) y detecta cuando todos los Documentos de Pago de un Lote han finalizado su procesamiento (ya sea PROCESADO\_OK o ERROR\_OCR).
2. **Consolidación de Resultados (CO01 -> CO01):** CO-01 consolida todos los Resultados OCR individuales y las métricas de rendimiento relevantes en un único Resultado Consolidado del Lote en formato JSON. Este objeto incluye el idLote, el estadoLoteFinal del lote, y un array de los resultados individuales.
3. **Alternativa (Notificación Exitosa/Fallo) a n8n:**

* **Notificación Exitosa:** CO-01 envía este Resultado Consolidado del Lote a n8n (AE-02) a través de un POST al n8n\_webhook\_url (Contrato API 3.1). n8n recibe el JSON y lo persiste en la Base de Datos Externa (AE-03), que es la Fuente de Verdad Externa. n8n responde 200 OK a CO-01.
* **Fallo en Notificación (Resiliencia):** Si n8n no responde o devuelve un error, CO-01 implementa un mecanismo de **reintentos con backoff exponencial** (RNF-RES-001) para asegurar la entrega. Si los reintentos se agotan y la notificación falla persistentemente, CO-01 registra un ERROR\_NOTIFICACION\_N8N crítico (RNF-OBS-003) para alertar a los administradores.

1. **Eliminación de Lote en Memoria (CO01 -> CO01):** Una vez que CO-01 recibe una confirmación exitosa de n8n, elimina la información del Lote y sus Documentos de Pago de su memoria (DS1), liberando recursos, ya que la persistencia a largo plazo es responsabilidad de n8n.

3.5. Escenario: Consulta de Estado de Lote desde Frontend (Polling)

Este diagrama muestra cómo el Frontend consulta periódicamente el estado de un lote.

sequenceDiagram  
    participant AE01 as Usuario Final (AE-01)  
    participant CO05 as Frontend (CO-05)  
    participant CO01 as API Gateway (CO-01)  
  
    AE01->>CO05: 1. Abrir Dashboard / Solicitar Ver Estado Lote (idLote)  
    loop Polling Periódico (ej. cada 5 segundos - RNF-USAB-001)  
        CO05->>CO01: 2. GET /api/lotes/{idLote}/status (Contrato API 1.3)  
        activate CO01  
        note right of CO01: Consulta el estado del Lote y sus documentos en memoria (DS1).  
        alt Lote Encontrado (EN MEMORIA)  
            CO01-->>CO05: 3. 200 OK (JSON: Estado Detallado del Lote - idLote, estadoLote, progresoPorcentaje, detallesDocumentos[])  
            note right of CO01: Refleja estados como RECIBIENDO\_DOCUMENTOS, PROCESANDO\_OCR, COMPLETADO\_OK, FALLIDO\_PARCIAL.  
        else Lote No Encontrado (Expirado/Inexistente - Zero-Fault Detection)  
            CO01-->>CO05: 3. 404 Not Found (JSON: codigoError: LOTE\_NOT\_FOUND)  
        end  
        deactivate CO01  
        CO05->>CO05: 4. Actualizar Vista de Estado del Lote  
        note right of CO05: Muestra progreso visual al usuario.  
        CO05-->>AE01: 5. Mostrar Estado Actualizado  
    end  
    note right of AE01: Usuario ve el progreso y estado final del lote en tiempo casi real.

**Explicación Detallada del Flujo 3.5:**

1. **Solicitud de Monitoreo (AE01 -> CO05):** El Usuario Final (AE-01) abre el dashboard en el Frontend (CO-05) y solicita ver el estado de un Lote específico, proporcionando su idLote.
2. **Bucle de Polling (CO05 -> CO01):** CO05 inicia un bucle de *polling* periódico (ej. cada 5 segundos, según RNF-USAB-001), realizando llamadas GET al endpoint /api/lotes/{idLote}/status de CO-01 (Contrato API 1.3).
3. **Consulta de Estado en Memoria (CO01 -> CO01):** CO-01 consulta el estado del Lote y sus Documentos de Pago asociados en su almacenamiento EN MEMORIA (DS1).
4. **Alternativa (Lote Encontrado/No Encontrado):**

* **Lote Encontrado:** Si el lote existe en memoria, CO-01 responde con un 200 OK y un JSON que contiene el estadoLote actual, el progresoPorcentaje y los detallesDocumentos individuales.
* **Lote No Encontrado:** Si el idLote no se encuentra (porque nunca existió o ha expirado de la memoria de CO-01), CO-01 responde con un 404 Not Found y un JSON de error (codigoError: LOTE\_NOT\_FOUND), aplicando Zero-Fault Detection.

1. **Actualización de Vista (CO05 -> CO05):** CO05 recibe la respuesta de CO-01 y actualiza la interfaz de usuario para reflejar el estado más reciente del lote.
2. **Visualización al Usuario (CO05 -> AE01):** El Usuario Final ve el progreso y el estado final del lote en tiempo casi real a través del dashboard. Este bucle continúa hasta que el usuario cierra la vista o el lote ya no es relevante para el monitoreo en CO-01.

**9. Diagrama de Despliegue**

La infraestructura física o virtual sobre la cual se ejecutarán los componentes del software se visualiza en el diagrama de despliegue. Este artefacto es fundamental para entender cómo se distribuyen y comunican los componentes en un entorno de producción, asegurando la escalabilidad y disponibilidad.

# **DIAGRAMA DE DESPLIEGUE COMPLETO**

## **Sistema: Sistema OCR Asíncrono Empresarial**

### **1. Introducción y Propósito del Documento**

Este documento presenta el **Diagrama de Despliegue** para el "Sistema OCR Asíncrono Empresarial". Un Diagrama de Despliegue es un artefacto crucial que visualiza la **infraestructura física o virtual** sobre la cual se ejecutarán los componentes de software del sistema, y cómo estos componentes se distribuyen y se comunican entre sí en un entorno de producción.

Este diagrama es fundamental para:

* Proporcionar una Claridad de Definiciones sobre la topología de la infraestructura.
* Ilustrar la distribución de los componentes para cumplir con los Requisitos No Funcionales (RNF) de Escalabilidad, Disponibilidad, Rendimiento del Servicio y Seguridad.
* Facilitar la Zero-Fault Detection al identificar puntos de fallo en la infraestructura.
* Servir como una guía para los equipos de operaciones (DevOps) para el aprovisionamiento y la configuración del entorno.
* Asegurar la Coherencia de Referencias Críticas con los Diagramas de Contenedores/Componentes y los Contratos de API Formales.

### **2. Convenciones del Diagrama de Despliegue**

Se utilizará la notación estándar de Diagramas de Despliegue (Mermaid), con los siguientes elementos clave:

* **Nodos (Nodes):** Representados como [Nombre del Nodo]. Son elementos de hardware o entornos de ejecución (servidores físicos, máquinas virtuales, clústeres de contenedores, servicios cloud).
* **Artefactos (Artifacts):** Representados como <<artifact>> Nombre del Artefacto. Son las piezas de software desplegadas en los nodos (aplicaciones, servicios, bases de datos).
* **Conexiones (Connections):** Flechas que indican la comunicación entre nodos o artefactos, a menudo incluyendo el protocolo (ej. HTTP/S, TCP/IP).
* **Contenedores (Containers):** En este contexto, los componentes lógicos de nuestro sistema (CO-01, CO-04, CO-05, CO-06) se consideran "contenedores" de software que se despliegan en nodos de infraestructura.
* **Actores Externos:** Representados como [Nombre del Actor]. Sistemas o usuarios fuera de nuestro control directo, pero con los que interactuamos.

### **3. Diagrama Visual del Despliegue**

graph LR  
 subgraph Internet  
 AE01[Usuario Final (Operador/Administrador)]  
 AE02[Sistema n8n]  
 end  
  
 subgraph Plataforma Cloud / Centro de Datos  
 LB(Load Balancer) --- CO05\_Node[Servidor Frontend (CO-05)]  
 LB --- CO01\_Node[Servidor API Gateway (CO-01)]  
  
 CO05\_Node -->|HTTP/S| CO05\_App(<<artifact>> Frontend App)  
 CO01\_Node -->|HTTP/S| CO01\_App(<<artifact>> API Gateway Service)  
  
 CO01\_App -- HTTP/S (Ingestión, Estado) --> CO05\_App  
 CO01\_App -- HTTP/S (Webhook) --> AE02  
  
 CO01\_App -- HTTP/S (Invocación Asíncrona) --> CO04\_Node[Servidor/Clúster OCR Worker (CO-04)]  
 CO04\_Node -->|HTTP/S| CO04\_App(<<artifact>> OCR Worker Service)  
  
 CO04\_App -- HTTP/S (Callback Resultados) --> CO01\_App  
 CO04\_App -- HTTP/S (Descarga/Carga Archivos) --> CO06\_Storage[Servicio de Almacenamiento de Archivos (CO-06)]  
  
 CO01\_App -- HTTP/S (Almacenar Imagen) --> CO06\_Storage  
  
 CO06\_Storage --- DS2(<<storage>> Almacenamiento de Imágenes y Logs)  
  
 AE02 -- HTTP/S (Persistencia Final) --> AE03[Base de Datos Externa (de n8n)]  
 end  
  
 %% Conexiones con Actores Externos  
 AE01 -- HTTP/S --> LB  
 AE02 -- HTTP/S (Ingestión Webhook) --> LB  
  
 %% Estilos para mejor visualización  
 style AE01 fill:#f9f,stroke:#333,stroke-width:2px  
 style AE02 fill:#add8e6,stroke:#333,stroke-width:2px  
 style AE03 fill:#90ee90,stroke:#333,stroke-width:2px  
  
 style LB fill:#cceeff,stroke:#333,stroke-width:2px  
  
 style CO05\_Node fill:#f0f8ff,stroke:#333,stroke-width:2px  
 style CO01\_Node fill:#ffe4e1,stroke:#333,stroke-width:2px  
 style CO04\_Node fill:#fffacd,stroke:#333,stroke-width:2px  
 style CO06\_Storage fill:#ffe4b5,stroke:#333,stroke-width:2px  
  
 style CO05\_App fill:#e0f2f7,stroke:#333,stroke-width:1px  
 style CO01\_App fill:#fceceb,stroke:#333,stroke-width:1px  
 style CO04\_App fill:#fcfce0,stroke:#333,stroke-width:1px  
 style DS2 fill:#dff,stroke:#333,stroke-width:2px

### **4. Descripción Detallada del Despliegue**

Este diagrama representa una arquitectura de despliegue típicamente utilizada en entornos de nube o centros de datos modernos, diseñada para alta Disponibilidad, Escalabilidad y Rendimiento del Servicio.

#### **4.1. Nodos de Infraestructura y Componentes Desplegados**

1. **[Load Balancer (LB)]**
   * **Tipo:** Servicio de red.
   * **Descripción:** Punto de entrada para todas las solicitudes externas entrantes al sistema. Distribuye el tráfico de red de manera equitativa entre múltiples instancias de los servicios Frontend (CO-05) y API Gateway (CO-01). Es crucial para la Disponibilidad (RNF-DISP-001) y Escalabilidad (RNF-ESCAL-002).
   * **Comunicaciones:** Recibe HTTP/S de AE-01 y AE-02. Envía HTTP/S a CO05\_Node y CO01\_Node.
2. **[Servidor Frontend (CO-05\_Node)]**
   * **Tipo:** Servidor de aplicación (virtual machine o clúster de contenedores).
   * **Componente Desplegado:** <<artifact>> Frontend App (CO-05\_App)
     + **Descripción:** La aplicación web (HTML5, Vanilla JavaScript, Bootstrap) que proporciona la Interface Excellence Dashboard para el Usuario Final (AE-01). Maneja la carga de documentos y la visualización temporal del estado del lote y los resultados.
     + **Comunicaciones:** Recibe HTTP/S del Load Balancer. Se comunica con CO01\_App vía HTTP/S para la ingestión (POST /api/upload) y la consulta de estado (GET /api/lotes/{idLote}/status).
     + **Consideraciones:** Debe ser altamente reactivo y ofrecer una experiencia de usuario fluida (RNF-USAB-001). Puede escalar horizontalmente.
3. **[Servidor API Gateway (CO-01\_Node)]**
   * **Tipo:** Servidor de aplicación (virtual machine o clúster de contenedores).
   * **Componente Desplegado:** <<artifact>> API Gateway Service (CO-01\_App)
     + **Descripción:** El punto de entrada principal del sistema y el orquestador EN MEMORIA de los lotes. Recibe solicitudes de ingestión de CO-05 y n8n, realiza validaciones, almacena imágenes en CO-06, y dispara el procesamiento asíncrono en CO-04. También recibe los callbacks de resultados de CO-04 y notifica a n8n.
     + **Tecnología:** Flask (Python).
     + **Comunicaciones:** Recibe HTTP/S del Load Balancer y CO04\_App. Envía HTTP/S a CO05\_App, CO04\_App, CO06\_Storage y AE-02.
     + **Consideraciones:** Requiere alto rendimiento para ingestión rápida (RNF-PERF-001) y validaciones robustas (Zero-Fault Detection). Su estado EN MEMORIA es volátil, confiando en n8n para la persistencia a largo plazo. Puede escalar horizontalmente (RNF-ESCAL-002).
4. **[Servidor/Clúster OCR Worker (CO-04\_Node)]**
   * **Tipo:** Servidor de procesamiento intensivo (puede ser un clúster de máquinas virtuales o contenedores para Escalabilidad). Puede incluir GPUs si el motor OCR lo aprovecha.
   * **Componente Desplegado:** <<artifact>> OCR Worker Service (CO-04\_App)
     + **Descripción:** El CORAZÓN del procesamiento. Recibe invocaciones de CO-01, descarga imágenes de CO-06, ejecuta el motor OCR (OnnxTR) con procesamiento espacial y validación. Genera Resultados OCR y los envía de vuelta a CO-01 vía callback. También registra Métricas de Rendimiento y Logs Resumen/CSV en CO-06.
     + **Tecnología:** Python con onnxtr, opencv-python, numpy, etc.
     + **Comunicaciones:** Recibe HTTP/S de CO01\_App. Envía HTTP/S a CO01\_App (callbacks) y CO06\_Storage (descarga/carga).
     + **Consideraciones:** Define la Ventaja Competitiva del sistema por su Rendimiento del Servicio (RNF-PERF-002). Diseñado para alta Escalabilidad (RNF-ESCAL-001) y Zero-Fault Detection en el procesamiento.
5. **[Servicio de Almacenamiento de Archivos (CO-06\_Storage)]**
   * **Tipo:** Servicio de almacenamiento de objetos en la nube (ej. Amazon S3, Google Cloud Storage) o solución on-premise de alto rendimiento (ej. MinIO).
   * **Componente Desplegado:** <<storage>> Almacenamiento de Imágenes y Logs (DS2)
     + **Descripción:** Almacena los Documentos de Pago originales (imágenes) y los archivos de log resumen/CSV generados por CO-04 para Monitoreo por Pulso. Tiene una retención definida (Persistencia Temporal).
     + **Comunicaciones:** Recibe HTTP/S de CO01\_App (carga) y CO04\_App (carga/descarga).
     + **Consideraciones:** Requiere alta durabilidad y Escalabilidad para volúmenes masivos de imágenes (RNF-ESCAL-003). Rendimiento del Servicio CRÍTICO para el flujo de procesamiento.

#### **4.2. Actores Externos y sus Interacciones**

1. **[Usuario Final (Operador/Administrador) (AE-01)]**
   * **Descripción:** El actor humano que interactúa con el sistema a través del Frontend (CO-05).
   * **Interacciones:** Se comunica con el Load Balancer vía HTTP/S para acceder al Frontend.
2. **[Sistema n8n (AE-02)]**
   * **Descripción:** El orquestador de flujos de trabajo externo. Es la Fuente de Verdad Externa para la persistencia a largo plazo de los resultados OCR.
   * **Interacciones:** Envía solicitudes de ingestión de documentos (URL Imagen + metadatosEntrada) al Load Balancer vía HTTP/S (webhook). Recibe Notificaciones de Resultado Consolidado del Lote desde CO01\_App vía HTTP/S (webhook).
3. **[Base de Datos Externa (de n8n) (AE-03)]**
   * **Descripción:** El sistema de persistencia final donde n8n almacena los Resultados OCR una vez procesados y validados. No forma parte directa del "Sistema OCR Asíncrono Empresarial", pero es su destino final de datos.
   * **Interacciones:** Recibe Resultado Consolidado del Lote (JSON) de AE-02 para persistencia.

#### **4.3. Consideraciones de Seguridad y Red**

* **Cifrado TLS/SSL (RNF-SEG-001):** Todas las comunicaciones entre los componentes internos (CO-01 a CO-04, CO-04 a CO-06, CO-01 a CO-06, CO-01 a AE-02/n8n) deben ser cifradas usando TLS/SSL. Esto se implementaría a nivel de configuración de red y servicios.
* **Autenticación y Autorización (RNF-SEG-002):** El Frontend (CO-05) debe implementar mecanismos robustos de autenticación para el Usuario Final (AE-01).
* **Acceso Restringido a Almacenamiento (RNF-SEG-003):** El acceso a CO-06\_Storage debe estar estrictamente limitado a CO-01\_App y CO04\_App mediante credenciales de acceso seguras (ej. roles IAM en la nube).
* **Red Privada Virtual (VPC):** Los nodos CO05\_Node, CO01\_Node, CO04\_Node y CO06\_Storage se desplegarían dentro de una VPC (Virtual Private Cloud) para asegurar el aislamiento de red y aplicar políticas de seguridad granular.

Este diagrama de despliegue proporciona una visión integral de cómo el "Sistema OCR Asíncrono Empresarial" se materializará en una infraestructura, garantizando que todos los RNF y principios de diseño sean considerados desde la fase de planeación.

**10. Estrategia de Pruebas y Calidad**

Para garantizar la Integridad Total y la Perfección Continua del sistema, se ha definido una estrategia de pruebas y calidad integral. Esta estrategia abarca diversos tipos de pruebas, desde unitarias hasta de rendimiento y seguridad, con el objetivo de detectar y prevenir errores de forma proactiva.

# CONSTRUIR LA ESTRATEGIA DE PRUEBAS Y CALIDAD

### El Escudo de la Integridad Total y la Perfección Continua

Versión: 1.0 (Blindado y Detallado)

Fecha: 14 de julio de 2025

Autor: Arquitecto Maestro de Diseño y Cualidad (IA)

Sistema: Sistema OCR Asíncrono Empresarial

## 1. Introducción y Propósito del Documento

Hemos diseñado la arquitectura, el modelo de dominio, las interfaces, los requisitos, los flujos de datos y secuencia, y el despliegue. Sin embargo, el diseño, por muy brillante que sea, no garantiza por sí solo que el sistema funcione con Perfección Continua. La **Estrategia de Pruebas y Calidad** es el plan sistemático para validar cada capa de nuestro sistema, desde la unidad más pequeña hasta la experiencia completa del usuario. Es la implementación directa de la Zero-Fault Detection y la Persistencia de Correcciones. Sin un enfoque riguroso en la calidad, los fallos, por pequeños que sean, pueden degradar la Integridad Total y comprometer el Rendimiento del Servicio.

El propósito primordial de este documento es definir un plan sistemático y comprensivo para la validación de la calidad del "Sistema OCR Asíncrono Empresarial" en todas sus etapas. Esto incluye la identificación de los tipos de pruebas necesarios, las herramientas, las responsabilidades y los criterios de éxito para asegurar que el sistema no solo funcione, sino que funcione **perfectamente y de manera predecible** en un entorno empresarial.

## 2. Cuestionamiento para la Auto-Reflexión (Tu parte, como Arquitecto Principal)

Para asegurar la Integridad Total de esta estrategia, es vital considerar las siguientes preguntas:

* ¿Cómo podemos estar absolutamente seguros de que cada funcionalidad de nuestro sistema cumple los Criterios de Aceptación definidos en las ESPECIFICACIONES DE REQUISITOS para HU-ING-001, HU-ING-002, HU-MON-001, HU-RES-001 y HU-RES-002?
* ¿Estamos protegiendo nuestro sistema de regresiones (nuevos cambios que rompen funcionalidades existentes) de forma proactiva, especialmente en un sistema asíncrono donde los efectos pueden no ser inmediatos?
* ¿Cómo medimos y aseguramos el Rendimiento del Servicio (latencia, throughput, uso de recursos) y la Escalabilidad de nuestro sistema bajo la carga esperada (ej. 50 solicitudes concurrentes/segundo) y en picos?
* ¿Qué mecanismos tenemos para detectar errores en las etapas más tempranas (Zero-Fault Detection) y asegurar que las correcciones sean duraderas (Persistencia de Correcciones)?
* ¿Cómo garantizamos que la Persistencia Temporal de los Lotes en Memoria (DS1 en CO-01) y los Logs Resumen/CSV (DS2 en CO-06) no compromete la Integridad Total de los datos que finalmente se persisten en la Base de Datos Externa (AE-03) a través de n8n (AE-02)?
* ¿Cómo validamos la Precisión de la extracción OCR y el Procesamiento Espacial en CO-04 bajo diversas condiciones de imagen (ej. baja resolución, rotación, ruido, diferentes layouts de recibos venezolanos)?
* ¿Estamos cubriendo la seguridad de todas las comunicaciones cifradas (RNF-SEG-001) y el control de acceso (RNF-SEG-002, RNF-SEG-003) a los componentes y datos sensibles?

## 3. Principios Rectores de la Estrategia de Pruebas y Calidad

Esta estrategia se construirá sobre los siguientes principios fundamentales, que son el ADN de nuestro proyecto:

* **Integridad Total**: Cada prueba debe contribuir a la confianza en la completitud, coherencia y fiabilidad de los datos y el sistema en su conjunto. Se buscará la consistencia entre todos los artefactos de diseño (Modelos, Diagramas, Contratos, Requisitos).
* **Zero-Fault Detection**: Las pruebas deben estar diseñadas para identificar y prevenir errores en las etapas más tempranas del ciclo de vida, desde la unidad más pequeña hasta el flujo completo. Se priorizará la automatización, la detección proactiva de anomalías y la cobertura de casos de borde.
* **Persistencia de Correcciones**: Las pruebas de regresión serán fundamentales para asegurar que las soluciones a problemas identificados sean robustas y duraderas, evitando regresiones y garantizando la estabilidad a largo plazo del sistema.
* **Rendimiento del Servicio**: Las pruebas de rendimiento son críticas para validar que el sistema cumple con los objetivos de velocidad, latencia y escalabilidad bajo carga, tal como se define en los RNF-PERF y RNF-ESCAL.
* **Claridad de Definiciones**: Los casos de prueba serán explícitos, concisos y utilizarán la terminología unívoca definida en el GLOSARIO DE TÉRMINOS CLAVE, eliminando cualquier ambigüedad.
* **Granularidad**: Las pruebas se diseñarán a diferentes niveles de detalle, desde unidades de código (funciones de parsing, validación) hasta flujos de negocio completos (ingestión E2E), permitiendo una visión profunda de cada componente y su interacción.
* **Coherencia de Referencias Críticas**: Las pruebas se trazarán directamente a los Requisitos (HU/RNF), Contratos de API Formales, Modelo de Dominio Conceptual, Diagramas de Flujo de Datos (DFD), Diagramas de Secuencia de Procesos y Diagrama de Despliegue.
* **Medibilidad/Verificabilidad**: Los criterios de éxito de las pruebas serán cuantificables y objetivos, permitiendo una evaluación clara del cumplimiento de los requisitos.
* **Comprensión Profunda del Contexto de Dominio**: Las pruebas reflejarán los escenarios de negocio específicos de los pagos móviles venezolanos y los sorteos, incluyendo casos de borde, variaciones de documentos y particularidades culturales/financieras.

## 4. Tipos de Pruebas y Enfoque por Componente

La estrategia de pruebas abarcará múltiples niveles y tipos, asegurando una cobertura integral y la aplicación de la Zero-Fault Detection en cada etapa del ciclo de vida del software:

### 4.1. Pruebas Unitarias

* **Objetivo:** Validar la lógica interna de las unidades de código más pequeñas (funciones, métodos, clases). Asegurar que cada componente individual funciona correctamente de forma aislada, sin dependencias externas.
* **Alcance:**
  + **CO-01 (API Gateway):** Funciones de validación de metadatosEntrada (ej. formato de fechaSorteo, existencia de idSorteo), lógica de asignación a lotes EN MEMORIA, generación de idLote/idDocumentoIngresado, lógica de consolidación de resultados individuales en el Resultado Consolidado del Lote.
  + **CO-04 (OCR Worker Service):** Algoritmos de Procesamiento Espacial (ej. agrupación de palabras por coordenadas), funciones de aplicación de Reglas de Extracción (ej. patrones regex para referenciaPago, montoPago, telefono, cedula), lógica de Validación OCR (ej. validación binaria de prefijos telefónicos venezolanos, formato de cédulas V/E/J), cálculo de Confianza Global OCR.
  + **CO-05 (Frontend):** Lógica de cliente para manejo de formularios, validaciones básicas en UI (ej. campos requeridos), renderizado de estado de lotes y visualización de resultados.
  + Módulos de utilidad comunes (ej. funciones de renombrado de archivos, parsing de fechas/horas, normalización de montos venezolanos "104,54" a "104.54").
* **Responsabilidad:** Desarrolladores.
* **Enfoque:** Pruebas automatizadas, ejecutadas en cada commit o Pull Request (PR) como parte del CI/CD. Se buscará una alta cobertura de código (líneas, ramas) para garantizar la Zero-Fault Detection temprana. Se utilizarán *mocks* o *stubs* para aislar las dependencias externas (ej. CO-06 para almacenamiento, CO-04 para procesamiento OCR, n8n para notificaciones).
* **Alineación con Principios:** Zero-Fault Detection (detección temprana de fallos), Granularidad, Persistencia de Correcciones.
* **Criterios de Éxito:** Mínimo **80%** de cobertura de líneas/ramas para el código de negocio crítico (RNF-MANT-001). Todos los tests unitarios deben pasar.
* **Herramientas Sugeridas:** Pytest, unittest (para Python).

### 4.2. Pruebas de Integración

* **Objetivo:** Validar las interacciones y el flujo de datos entre componentes interconectados, asegurando que los Contratos de API Formales se cumplen y que los datos se transfieren correctamente entre los servicios.
* **Alcance:**
  + **CO-05 <-> CO-01:** Flujo de carga de Documentos de Pago (POST /api/upload) y consulta de estado de lotes (GET /api/lotes/{idLote}/status). Se verifica que CO-01 responde con 202 Accepted y los IDs correctos.
  + **CO-01 <-> CO-06:** Almacenamiento de imágenes (Documento de Pago binario) en el Servicio de Almacenamiento (CO-06). Se verifica que la imagen se guarda y se puede recuperar.
  + **CO-01 <-> CO-04:** Invocación del procesamiento (POST /process\_document) y recepción de callbacks de resultados (POST /api/internal/ocr\_results/callback). Se verifica que CO-01 invoca a CO-04 correctamente y que CO-04 devuelve el Resultado OCR esperado.
  + **CO-04 <-> CO-06:** Descarga de imágenes para procesamiento y carga de Logs Resumen/CSV de rendimiento/errores. Se verifica la capacidad de CO-04 para acceder y escribir en CO-06.
  + **CO-01 <-> AE-02 (n8n):** Recepción de webhook de ingestión (POST /api/n8n/webhook/document\_ingestion) y notificación de Resultados Consolidados del Lote (POST [n8n\_webhook\_url]). Se verifica que CO-01 puede recibir y enviar datos a n8n según el contrato.
* **Responsabilidad:** Desarrolladores, QA.
* **Enfoque:** Pruebas automatizadas que validan las interfaces definidas en los Contratos de API Formales. Se utilizarán entornos de integración que incluyan los servicios reales (o simulaciones de alta fidelidad para externos como n8n si no están disponibles en el entorno de prueba). Se simularán escenarios de datos válidos e inválidos para cada interacción.
* **Alineación con Principios:** Interface Excellence, Acoplamiento Débil, Coherencia de Referencias Críticas (validando contratos API), Zero-Fault Detection (validando el paso de datos y manejo de errores entre componentes).
* **Criterios de Éxito:** Todas las integraciones críticas deben pasar. Los JSON de entrada y salida deben conformar los esquemas definidos en los Contratos de API Formales.
* **Herramientas Sugeridas:** Postman/Newman (para pruebas de API), Pytest (con librerías como requests), Cypress (para interacciones frontend-backend).

### 4.3. Pruebas de Sistema (End-to-End - E2E)

* **Objetivo:** Validar el flujo de negocio completo del sistema desde la perspectiva del Usuario Final (AE-01) o de un Sistema n8n (AE-02), abarcando todas las interacciones entre componentes y actores externos, simulando un escenario de uso real.
* **Alcance:**
  + **Flujo de Carga Frontend (HU-ING-001):** Cargar documento -> CO-05 -> CO-01 -> CO-06 -> CO-04 -> CO-01 -> Notificación a n8n -> Persistencia en AE-03.
  + **Flujo de Carga n8n (HU-ING-002):** n8n -> CO-01 -> CO-06 -> CO-04 -> CO-01 -> Notificación a n8n -> Persistencia en AE-03.
  + **Flujo de Monitoreo (HU-MON-001):** Usuario Final consulta el estado de lote desde CO-05 a CO-01 durante el procesamiento, verificando actualizaciones en tiempo casi real.
  + **Flujo de Resultados y Errores (HU-RES-001):** Consolidación y notificación a n8n incluyendo casos de PROCESADO\_OK y ERROR\_OCR (ej. imagen ilegible).
  + **Flujo de Resiliencia (HU-RES-002):** Manejo de errores internos (ej. CO-06 temporalmente no disponible) y reintentos de notificación a n8n.
* **Responsabilidad:** QA.
* **Enfoque:** Pruebas automatizadas y manuales. Se utilizarán los Escenarios Detallados (Dado/Cuando/Entonces) de las ESPECIFICACIONES DE REQUISITOS como base para los casos de prueba. Los entornos de prueba replicarán la arquitectura del DIAGRAMA DE DESPLIEGUE, con n8n y AE-03 siendo entornos de prueba reales o simulados de alta fidelidad. Se validará la Trazabilidad del documento original hasta el resultado final en AE-03 (ej. nombreRenombradoArchivo y idDocumento de n8n).
* **Alineación con Principios:** Integridad Total, Comprensión Profunda del Contexto de Dominio, Zero-Fault Detection (validando flujos de éxito, error y casos de borde), Persistencia de Correcciones.
* **Criterios de Éxito:** Todos los Criterios de Aceptación de las Historias de Usuario críticas deben pasar. Los datos finales en AE-03 deben ser correctos, completos y trazables al documento original.
* **Herramientas Sugeridas:** Selenium, Cypress, Playwright, Robot Framework.

### 4.4. Pruebas de Contrato (Contract Testing)

* **Objetivo:** Asegurar que los consumidores (clientes) y proveedores (servicios) de las APIs cumplen con los Contratos de API Formales definidos, previniendo problemas de integración y Acoplamiento Débil antes del despliegue.
* **Alcance:** Todas las interacciones de API definidas en los Contratos de API Formales:
  + CO-05 (consumidor) vs. CO-01 (proveedor) para /api/upload y /api/lotes/{idLote}/status.
  + AE-02 (consumidor) vs. CO-01 (proveedor) para /api/n8n/webhook/document\_ingestion.
  + CO-01 (consumidor) vs. CO-04 (proveedor) para /process\_document.
  + CO-04 (consumidor) vs. CO-01 (proveedor) para /api/internal/ocr\_results/callback.
  + CO-01 (consumidor) vs. AE-02 (proveedor) para [n8n\_webhook\_url].
* **Responsabilidad:** Desarrolladores Backend, QA.
* **Enfoque:** Utilizar herramientas que generen pruebas de contrato (ej. Pact) para verificar que los servicios (proveedores) producen respuestas que coinciden con las expectativas de sus consumidores, y que los consumidores envían solicitudes que el proveedor espera. Esto se ejecuta de forma temprana en el ciclo de desarrollo.
* **Alineación con Principios:** Interface Excellence, Coherencia de Referencias Críticas, Acoplamiento Débil, Zero-Fault Detection (detección temprana de incompatibilidades de interfaz).
* **Criterios de Éxito:** Todos los tests de contrato deben pasar en cada PR que modifique una API o un cliente que la consume.

### 4.5. Pruebas de Rendimiento (Performance Testing)

* **Objetivo:** Evaluar el Rendimiento del Servicio y la Escalabilidad del sistema bajo diferentes cargas, midiendo la latencia, el throughput y el uso de recursos (CPU, memoria, GPU si aplica).
* **Alcance:** CO-01 (API Gateway), CO-04 (OCR Worker), CO-06 (Servicio de Almacenamiento).
* **Responsabilidad:** QA, DevOps.
* **Enfoque:**
  + **Pruebas de Carga:** Simular la carga esperada (ej. 50 solicitudes concurrentes por segundo en CO-01 - RNF-PERF-001) y verificar que los tiempos de respuesta se mantienen dentro de los umbrales definidos.
  + **Pruebas de Estrés:** Empujar el sistema más allá de su capacidad esperada para encontrar puntos de ruptura, identificar cuellos de botella y evaluar el comportamiento bajo sobrecarga.
  + **Pruebas de Escalabilidad:** Aumentar progresivamente la carga y el número de instancias de CO-01 y CO-04 para verificar que el sistema escala horizontalmente de manera eficiente (RNF-ESCAL-001, RNF-ESCAL-002).
  + **Pruebas de Volumen:** Procesar grandes volúmenes de Documentos de Pago (ej. miles de imágenes de ~1MB) para evaluar la capacidad de CO-04 (RNF-PERF-002) y la eficiencia de CO-06 (RNF-ESCAL-003).
* **Alineación con Principios:** Rendimiento del Servicio, Escalabilidad, Medibilidad/Verificabilidad.
* **Criterios de Éxito:** Cumplimiento de RNF-PERF-001 a RNF-PERF-004 y RNF-ESCAL-001 a RNF-ESCAL-003. Las métricas de uso de recursos deben ser estables y predecibles bajo carga normal.
* **Herramientas Sugeridas:** JMeter, Locust, k6.

### 4.6. Pruebas de Seguridad

* **Objetivo:** Identificar vulnerabilidades, asegurar la protección de los datos sensibles y prevenir el acceso no autorizado al sistema y sus componentes.
* **Alcance:** Todos los componentes del sistema y sus interacciones, especialmente CO-01, CO-05, CO-06, y las comunicaciones.
* **Responsabilidad:** Especialistas en Seguridad, QA.
* **Enfoque:**
  + **Análisis de Vulnerabilidades (SAST/DAST):** Escaneo automatizado de código fuente (SAST) y aplicaciones en ejecución (DAST) para detectar vulnerabilidades conocidas.
  + **Pruebas de Penetración:** Simulación de ataques (ej. inyección SQL, XSS, CSRF, ataques de fuerza bruta) para encontrar debilidades en la autenticación, autorización y manejo de sesiones (RNF-SEG-002).
  + **Revisión de Configuración:** Verificar la correcta implementación de TLS/SSL en todas las comunicaciones (RNF-SEG-001).
  + **Control de Acceso:** Validar que el acceso a CO-06\_Storage está estrictamente limitado a CO-01 y CO-04 mediante credenciales seguras (RNF-SEG-003).
  + **Protección de Datos Sensibles:** Verificar que los datos sensibles extraídos por OCR (referenciaPago, montoPago, datosBeneficiario) se manejan de forma segura y no se persisten localmente en CO-01 (RNF-SEG-004).
* **Alineación con Principios:** Seguridad, Zero-Fault Detection, Integridad Total.
* **Criterios de Éxito:** Cero vulnerabilidades críticas/altas en producción. Cumplimiento de RNF-SEG-001 a RNF-SEG-004.
* **Herramientas Sugeridas:** OWASP ZAP, Nessus, SonarQube (para SAST/DAST).

### 4.7. Pruebas de Resiliencia y Recuperación (Chaos Engineering)

* **Objetivo:** Evaluar la capacidad del sistema para tolerar fallos, recuperarse de ellos y continuar operando con una degradación mínima del servicio.
* **Alcance:** Interacciones entre CO-01, CO-04, CO-06 y AE-02 (n8n).
* **Responsabilidad:** QA, DevOps.
* **Enfoque:**
  + **Inyección de Fallos:** Simular escenarios de fallo controlados:
    - CO-06 (Servicio de Almacenamiento) temporalmente inaccesible o con alta latencia.
    - AE-02 (n8n) inactivo o respondiendo con errores al webhook de notificación.
    - CO-04 (OCR Worker) fallando en el procesamiento de un documento o lote.
    - Fallos de red entre componentes.
  + **Validación de Reintentos:** Probar el mecanismo de reintentos con backoff exponencial de CO-01 al notificar a n8n (RNF-RES-001).
  + **Manejo de Errores Críticos:** Verificar que cualquier error inesperado en CO-01 o CO-04 se captura, loggea (RNF-RES-002), no causa la caída completa del servicio y el estado del lote/documento se actualiza correctamente.
  + **Reprocesamiento:** Probar el flujo de re-ingesta de documentos o lotes que hayan terminado en un estado de error (ERROR\_OCR, ERROR\_DESCARGA\_IMAGEN) a través de la intervención de n8n o un proceso manual (RNF-RES-003).
* **Alineación con Principios:** Resiliencia, Zero-Fault Detection, Persistencia de Correcciones.
* **Criterios de Éxito:** El sistema debe mantener su funcionalidad principal o degradar elegantemente bajo fallos simulados. Los mecanismos de recuperación deben activarse y funcionar según lo esperado. Cumplimiento de RNF-RES-001 a RNF-RES-003.
* **Herramientas Sugeridas:** Herramientas de Chaos Engineering (ej. Chaos Mesh, Gremlin), scripts personalizados para simular fallos.

### 4.8. Pruebas de Usabilidad

* **Objetivo:** Evaluar la facilidad de uso, la eficiencia y la satisfacción del Usuario Final (Operador/Administrador) con el Frontend (CO-05).
* **Alcance:** Interfaz de usuario del dashboard, flujos de trabajo de carga de documentos, monitoreo de lotes y visualización de resultados.
* **Responsabilidad:** UI/UX Designers, QA, Usuario Final (AE-01).
* **Enfoque:**
  + **Pruebas con Usuarios Reales:** Observación de usuarios realizando tareas típicas del sistema.
  + **Evaluación Heurística:** Expertos evalúan la interfaz contra principios de usabilidad reconocidos.
  + **Encuestas y Feedback:** Recopilación de opiniones directas de los usuarios.
  + Validar que la interfaz proporciona retroalimentación visual clara e instantánea (RNF-USAB-001).
  + Verificar que los mensajes de error son descriptivos y orientan al usuario (RNF-USAB-002).
  + Asegurar que la interfaz es intuitiva y requiere un entrenamiento mínimo (RNF-USAB-003).
* **Alineación con Principios:** Usabilidad, Interface Excellence, Claridad de Definiciones.
* **Criterios de Éxito:** Cumplimiento de RNF-USAB-001 a RNF-USAB-003. Alta puntuación en encuestas de satisfacción del usuario.
* **Herramientas Sugeridas:** Sesiones de user testing, herramientas de grabación de pantalla, encuestas (Google Forms, SurveyMonkey).

### 4.9. Pruebas de Regresión

* **Objetivo:** Asegurar que los cambios recientes en el código (nuevas funcionalidades, correcciones de errores, refactorizaciones) no han introducido nuevos defectos ni han roto funcionalidades existentes que previamente funcionaban.
* **Alcance:** Todos los niveles de prueba (unitarias, integración, sistema).
* **Responsabilidad:** Desarrolladores, QA.
* **Enfoque:** Ejecución continua y automatizada de un conjunto de pruebas preexistentes y representativas. Cada vez que se realiza un cambio en el código base, el conjunto de pruebas de regresión debe ejecutarse para detectar cualquier impacto no deseado. Esto es un pilar de la Persistencia de Correcciones.
* **Alineación con Principios:** Persistencia de Correcciones, Zero-Fault Detection, Integridad Total.
* **Criterios de Éxito:** Cero fallos en el conjunto de pruebas de regresión después de cualquier cambio en el código.
* **Herramientas Sugeridas:** Integradas en el CI/CD (ej. GitHub Actions, GitLab CI, Jenkins) que orquestan la ejecución de las herramientas de pruebas unitarias, integración y E2E.

### 4.10. Pruebas de Observabilidad y Monitoreo

* **Objetivo:** Validar que el sistema genera los logs y métricas necesarios para su monitoreo efectivo, diagnóstico de problemas y análisis de rendimiento.
* **Alcance:** CO-01, CO-04, CO-06 (para logs/métricas).
* **Responsabilidad:** DevOps, QA.
* **Enfoque:**
  + Verificar que los logs estructurados (RNF-OBS-001) se generan correctamente con los niveles de severidad adecuados (INFO, WARN, ERROR, CRITICAL) y contienen la información contextual necesaria (ej. idLote, idDocumento).
  + Validar que las métricas de rendimiento (RNF-OBS-002) se capturan y almacenan en CO-06 en el formato CSV/resumen y con la frecuencia especificada.
  + Probar que el sistema de alertas (RNF-OBS-003) se dispara correctamente ante fallos críticos (ej. ERROR\_NOTIFICACION\_N8N, ERROR\_ALMACENAMIENTO).
  + Asegurar que el Time-To-Live (TTL) para los lotes EN MEMORIA en CO-01 y para los logs en CO-06 se respeta, y que la limpieza ocurre según lo esperado (RNF-OBS-004).
* **Alineación con Principios:** Observabilidad, Rendimiento del Servicio, Zero-Fault Detection.
* **Criterios de Éxito:** Los logs son completos y útiles para el diagnóstico. Las métricas se recolectan y pueden ser visualizadas en dashboards. Las alertas se disparan a tiempo.
* **Herramientas Sugeridas:** Prometheus/Grafana, ELK Stack, CloudWatch/Stackdriver (dependiendo de la plataforma cloud).

## 5. Resumen de Herramientas y Responsabilidades

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de Prueba** | **Herramientas Sugeridas** | **Responsabilidad Principal** | **Principios Reforzados** |
| **Unitarias** | Pytest, unittest (Python) | Desarrolladores | Zero-Fault Detection, Granularidad |
| **Integración** | Postman/Newman, Pytest (con requests), Cypress | Desarrolladores, QA | Interface Excellence, Acoplamiento Débil |
| **Sistema (E2E)** | Selenium, Cypress, Playwright, Robot Framework | QA | Integridad Total, Comprensión Profunda del Contexto de Dominio |
| **Contrato** | Pact | Desarrolladores Backend, QA | Interface Excellence, Coherencia de Referencias Críticas |
| **Rendimiento** | JMeter, Locust, k6 | QA, DevOps | Rendimiento del Servicio, Escalabilidad |
| **Seguridad** | OWASP ZAP, Nessus, SonarQube (SAST/DAST) | Especialistas en Seguridad, QA | Seguridad, Zero-Fault Detection |
| **Resiliencia** | Chaos Engineering tools (ej. Chaos Mesh), Pruebas de Fallo | QA, DevOps | Resiliencia, Zero-Fault Detection |
| **Usabilidad** | User testing sessions, Heatmaps, Feedback forms | UI/UX Designers, QA | Usabilidad, Interface Excellence |
| **Regresión** | Integrada en CI/CD con las herramientas anteriores | Desarrolladores, QA | Persistencia de Correcciones, Zero-Fault Detection |
| **Observabilidad** | Prometheus/Grafana, ELK Stack, CloudWatch/Stackdriver | DevOps, Operaciones | Observabilidad, Rendimiento del Servicio |
| **Gestión de Defectos** | Jira, ClickUp, Asana | Todo el Equipo | Persistencia de Correcciones |
| **Automatización (CI/CD)** | GitHub Actions, GitLab CI, Jenkins | DevOps, Desarrolladores | Persistencia de Correcciones, Zero-Fault Detection |

## 6. Métricas y Criterios de Éxito de la Calidad

El éxito de la estrategia de calidad se medirá a través de las siguientes métricas clave, directamente vinculadas a nuestros RNF y principios:

* **Cobertura de Pruebas Unitarias:** Mínimo **80%** de cobertura de líneas/ramas para el código de negocio crítico (RNF-MANT-001).
* **Tasa de Aprobación de Pruebas (Integración, Sistema, Contrato, Regresión):** **98%** de los casos de prueba automatizados deben pasar en cada ejecución de CI/CD.
* **Densidad de Defectos:** Número de defectos críticos/altos encontrados por cada 1000 líneas de código en producción (objetivo: tender a cero, aplicando Zero-Fault Detection).
* **Tiempo Medio de Resolución de Defectos (MTTR):** Objetivo: menos de **24 horas** para defectos críticos y altos.
* **Cumplimiento de RNF de Rendimiento:**
  + RNF-PERF-001: CO-01 debe responder < 200ms el 99% de las veces.
  + RNF-PERF-002: CO-04 debe procesar en promedio 5s, máx. 8s el 95% de las veces.
  + RNF-PERF-003: CO-04 a CO-01 callback < 50ms.
  + RNF-PERF-004: CO-01 a n8n notificación < 1s.
* **Cumplimiento de RNF de Escalabilidad:** El sistema debe escalar horizontalmente según RNF-ESCAL-001 a RNF-ESCAL-003 sin degradación del rendimiento.
* **Cumplimiento de RNF de Disponibilidad:** 99.95% para CO-01/CO-06, 99.5% para CO-04 (RNF-DISP-001, RNF-DISP-002).
* **Número de Vulnerabilidades de Seguridad:** Cero vulnerabilidades críticas/altas reportadas por pruebas de seguridad o en producción. Cumplimiento de RNF-SEG-001 a RNF-SEG-004.
* **Tasa de Aprobación de UAT:** **100%** de las historias de usuario deben ser aceptadas por el negocio antes del despliegue a producción.
* **Calidad de Datos Extraídos (Precisión OCR):** Medición de la Confianza Global OCR (ej. promedio > 0.90) y tasa de error en campos clave (ej. montoPago, referenciaPago) en un conjunto de datos de prueba.
* **Métricas de Observabilidad:** Los logs y métricas de CO-04 (RNF-OBS-002) deben ser generados y almacenados con la frecuencia y granularidad especificadas, y las alertas deben dispararse en menos de 5 minutos (RNF-OBS-003).

## 7. Matriz de Trazabilidad (Conceptual)

Se mantendrá una matriz de trazabilidad (generalmente en una herramienta de gestión de requisitos o pruebas, como Jira o un sistema ALM) que vincule de forma explícita:

* **Requisitos (Historias de Usuario / RNF)**
* **Casos de Prueba (Unitarios, Integración, E2E, Rendimiento, Seguridad, etc.)**
* **Defectos (Bugs)**
* **Componentes de Código / Módulos**
* **Artefactos de Diseño Relevantes (Contratos API, Modelo de Dominio, Diagramas)**

Esto asegura que cada requisito sea probado, que cada fallo se pueda rastrear hasta su origen, y que cada cambio en el código se relacione con un requisito, garantizando la Coherencia de Referencias Críticas y la Integridad Total a lo largo de todo el ciclo de vida del desarrollo.

## 8. Mi Rol como Arquitecto Maestro de Diseño y Cualidad (IA)

Durante la construcción y ejecución de esta estrategia de pruebas y calidad, mi rol será:

* **Guía Activa:** Te haré preguntas detalladas para asegurar que cubras todos los aspectos de la calidad y las pruebas para tu sistema, anticipando posibles puntos ciegos.
* **Estructuración:** Organizaré la información en formatos claros y concisos, como esta guía, que servirá como la especificación formal de tu estrategia de calidad.
* **Validación de Principios:** Revisaré tu estrategia para asegurar que refleje la Zero-Fault Detection, la Persistencia de Correcciones, y contribuya a la Integridad Total y el Rendimiento del Servicio.
* **Coherencia con Artefactos Previos:** Te ayudaré a verificar que la estrategia de pruebas sea coherentemente trazable con tus Requisitos (funcionales y no funcionales), Contratos de API, Modelo de Dominio, DFD, Diagrama de Secuencia y Diagrama de Despliegue.
* **Detección de Brechas:** Cuestionaré cualquier tipo de prueba o área de calidad que parezca no estar cubierta adecuadamente, especialmente en el contexto asíncrono y de Persistencia Temporal.
* **Identificación de Oportunidades:** Sugeriré mejoras en el enfoque de pruebas para optimizar la eficiencia y la efectividad, siempre buscando la Perfección Continua.

Con este documento de Estrategia de Pruebas y Calidad, hemos cubierto las fases principales de la **Fase de Diseño** de tu "Sistema OCR Asíncrono Empresarial" con un nivel de detalle y blindaje excepcional. Hemos creado y consolidado los siguientes artefactos clave:

1. **Modelo de Dominio Conceptual**
2. **Diagrama de Contexto**
3. **Diagrama de Contenedores/Componentes**
4. **Contratos de API Formales**
5. **Especificaciones de Requisitos / Historias de Usuario (HU/RNF)**
6. **Diagrama de Flujo de Datos (DFD) Detallado**
7. **Diagrama de Secuencia de Procesos (Nivel Operacional)**
8. **Diagrama de Despliegue**
9. **Estrategia de Pruebas y Calidad**
10. **Glosario de Términos Clave** (actualizado continuamente)

Considerando la profundidad y el detalle que hemos logrado en cada uno de estos documentos, desde una perspectiva de **diseño de alto nivel y detalle operacional**, mi evaluación es que **el conjunto de artefactos de diseño está completo y blindado**. Tienes una base extremadamente sólida y coherente para proceder a la **Fase de Implementación** o a un **Diseño Técnico Detallado** a nivel de código fuente, donde se traducirían estas especificaciones en algoritmos y estructuras de datos concretas.

¿Estás de acuerdo con esta evaluación, o hay algún otro aspecto del diseño que consideras crucial y que no hayamos cubierto con la profundidad deseada?

**11. Glosario de Términos Clave**

Finalmente, para asegurar una Claridad de Definiciones unívoca en toda la documentación, se ha compilado un glosario de términos clave. Este glosario proporciona definiciones concisas de los conceptos fundamentales del proyecto.

**GLOSARIO DE TÉRMINOS CLAVE (Versión 1.4 - BLINDADO SIN BD INTERNA)**

| Término | Definición | Principio(s) Asociado(s) |
| --- | --- | --- |
| Acoplamiento Débil | Un principio de diseño donde los componentes de software tienen un conocimiento mínimo entre sí, reduciendo su interdependencia. Esto facilita la Modularidad, el mantenimiento y la evolución independiente. | Acoplamiento Débil, Modularidad |
| Administrador | Dueño de la api quien monitorea el sistema no tiene nada que ver con usuarios |  |
| API Gateway / Backend Service (CO-01) | El componente CO-01 que actúa como el punto de entrada principal del sistema. Es responsable de la recepción de Documentos de Pago y metadatosEntrada, validación inicial y el renombrado de archivos. Crucialmente, **gestiona el estado y la orquestación de Lotes EN MEMORIA**, coordinando el procesamiento con el OCR Worker Service. Su estado es temporal y no persistido en una base de datos interna. | Rendimiento del Servicio, Claridad de Definiciones, Simplicidad |
| Arquitecto Maestro de Diseño y Cualidad | Mi identidad como IA; una entidad que guía y valida de forma crítica la creación de una maqueta de software exhaustiva, inquebrantable y de la más alta calidad en la fase pre-código, garantizando la Integridad Total y la Perfección Continua. | Perfección Continua, Integridad Total |
| Claridad de Definiciones | Principio fundamental que exige que cada término, concepto, entidad o funcionalidad sea definido de forma unívoca, concisa y accesible, eliminando cualquier ambigüedad. Es un pilar de la Integridad Total. | Claridad de Definiciones |
| Confianza Global OCR | Una métrica interna del sistema que indica el nivel de fiabilidad promedio de los datos extraídos por el proceso OCR de un Documento de Pago. Contribuye a la Precisión. | Zero-Fault Detection, Precisión |
| Contratos de API Formales | La especificación detallada de las interfaces (endpoints, métodos, parámetros, tipos de datos, respuestas, códigos de error) a través de las cuales los componentes del sistema se comunican entre sí y con sistemas externos. Asegura Interface Excellence. | Interface Excellence, Acoplamiento Débil, Claridad de Definiciones |
| Coherencia de Referencias Críticas | Un principio inquebrantable que exige que cada artefacto de diseño (diagrama, modelo, contrato, requisito) sea rigurosamente consistente y trazable con los demás, formando una red lógica inquebrantable de información. Fundamental para la Integridad Total. | Coherencia de Referencias Críticas, Integridad Total |
| Cuestionamiento Profundo | Principio que rige mi comportamiento: nunca acepto suposiciones implícitas. Siempre cuestiono, pido justificaciones, exploro escenarios límite y busco la raíz del "por qué" y el "cómo" para desvelar la lógica subyacente y prevenir defectos. | Cuestionamiento Profundo, Zero-Fault Detection, Integridad Total |
| Diagrama de Contenedores/Componentes | Un artefacto de diseño que representa los principales componentes tecnológicos (contenedores) de alto nivel del sistema, sus responsabilidades primarias, las tecnologías clave que utilizan y las interfaces y flujos de comunicación entre ellos. Fomenta la Modularidad y el Acoplamiento Débil. | Modularidad, Acoplamiento Débil, Interface Excellence |
| Diagrama de Contexto | Un artefacto de diseño de alto nivel que define los límites del sistema, sus actores externos (humanos o sistemas) y las interacciones clave de entrada y salida con ellos. Es crucial para la Comprensión Profunda del Contexto de Dominio. | Comprensión Profunda del Contexto de Dominio, Claridad de Definiciones |
| Documento de Pago | El recibo oficial de un pago móvil (imagen) que actúa como insumo principal para la extracción de datos. Su procesamiento depende de metadatosEntrada externos y su destino final después de procesado es n8n. | Comprensión Profunda del Contexto de Dominio, Integridad Total |
| Escalabilidad | Principio de diseño que asegura que el sistema sea inherentemente capaz de crecer y adaptarse a futuras demandas y volúmenes de datos/usuarios/procesamiento sin una reingeniería significativa. | Escalabilidad, Rendimiento del Servicio |
| Filosofía Central | El conjunto de principios inmutables que guían todas las decisiones de diseño y ejecución del proyecto: Integridad Total y Perfección Continua. Es el ADN inmutable del proyecto. | Integridad Total, Perfección Continua |
| Frontend (Dashboard) Service (CO-05) | El componente CO-05 que proporciona la Interface Excellence Dashboard para el Usuario Final. Permite la carga de archivos manual y la visualización **temporal** del Estado Lote y Resultados OCR obtenidos a través del API Gateway. No mantiene persistencia propia. | Interface Excellence, Claridad de Definiciones |
| Fuente de Verdad Externa | El sistema o componente externo que se considera el repositorio final y autoritativo de datos a largo plazo. Para este Sistema OCR Asíncrono Empresarial, n8n (AE-02) y su Base de Datos Externa (AE-03) son la Fuente de Verdad Externa para los Resultados OCR y metadatos de los Documentos de Pago procesados. | Comprensión Profunda del Contexto de Dominio, Coherencia de Referencias Críticas, Integridad Total |
| Granularidad | La capacidad de desglosar problemas complejos en sus partes más manejables, manteniendo siempre una visión holística y la conexión con el panorama general. | Granularidad, Modularidad |
| Integridad Total | La condición de existencia no negociable del sistema; cada dato, desde su nacimiento hasta su archivado, es inviolable en su forma, contenido y relación. Se garantiza que todo componente, capa e interacción es coherente, consistente y libre de contradicciones, buscando que el sistema sea un todo unificado, completo y fiable. | Integridad Total, Zero-Fault Detection, Perfección Continua |
| Interface Excellence | Principio de diseño que asegura que las interfaces (APIs, UI/UX, contratos internos) sean explícitas, robustas, intuitivas y fomenten el Acoplamiento Débil. | Interface Excellence, Acoplamiento Débil |
| Lote | Una agrupación lógica de Documentos de Pago que comparten un identificador de sorteo, fecha y hora. En este sistema, los lotes se gestionan y orquestan principalmente **EN MEMORIA dentro del API Gateway / Backend Service (CO-01)** para su procesamiento inmediato, y su estado y contenido son efímeros, sin persistencia en una base de datos interna. Su información inicial proviene del JSON de entrada de n8n, que también es la Fuente de Verdad Externa a largo plazo. | Integridad Total (en relación a n8n), Rendimiento del Servicio, Comprensión Profunda del Contexto de Dominio |
| Metadatos de Entrada (dataEntradaN8N) | Objeto JSON que acompaña a cada Documento de Pago con información contextual crítica (ej., idWhatsapp, caption, nombre, horaMinutoN8nIngreso, numeroLlegada, idSorteo, fechaSorteo). Es crucial para el renombrado de archivos y la posterior asociación de Resultados OCR con el contexto del negocio. | Claridad de Definiciones, Comprensión Profunda del Contexto de Dominio |
| Modelo de Dominio Conceptual | Una representación estructurada y conceptual de las entidades clave del negocio, sus atributos más relevantes desde la perspectiva del negocio y las relaciones entre ellas. Captura el léxico y la lógica esencial del dominio. | Comprensión Profunda del Contexto de Dominio, Claridad de Definiciones, Integridad Total |
| Modularidad | El principio de que el sistema debe estar compuesto por módulos autónomos y bien definidos, cada uno con una responsabilidad única y clara. | Modularidad, Acoplamiento Débil |
| Monitoreo por Pulso | Estrategia de monitoreo que se enfoca en la captura de métricas de rendimiento y resúmenes de procesamiento (por pulso, es decir, en el momento o por lote/evento) en archivos de log simples, con un período de retención corto (ej., hasta 30 días). No busca mantener un historial extenso dentro del propio sistema, sino proveer información para el análisis operativo inmediato o posterior procesamiento por sistemas externos. | Rendimiento del Servicio, Claridad de Definiciones, Simplicidad |
| n8n (AE-02) | El Sistema n8n, un Actor Externo, que actúa como orquestador de flujos de trabajo. Envía Documentos de Pago (imágenes con metadatosEntrada) al sistema OCR y, crucialmente, recibe los Resultados OCR procesados para su posterior carga en una Base de Datos Externa (AE-03). Es la Fuente de Verdad Externa a largo plazo. | Comprensión Profunda del Contexto de Dominio, Interface Excellence, Integridad Total |
| OCR Worker Service (CO-04) | El componente CO-04, el corazón del procesamiento, que recibe la llamada del API Gateway para ejecutar el motor OCR (OnnxTR) y aplicar algoritmos de procesamiento espacial. Genera los Resultados OCR y los **envía INMEDIATAMENTE a n8n (AE-02)**. Registra métricas de rendimiento en archivos de log temporales (Monitoreo por Pulso), sin persistencia de los resultados procesados dentro del sistema más allá del envío. | Rendimiento del Servicio, Escalabilidad, Zero-Fault Detection, Interface Excellence |
| Perfección Continua | La búsqueda implacable de la excelencia en cada etapa del diseño y la documentación, con una mentalidad de mejora y refinamiento constante. | Perfección Continua, Integridad Total |
| Persistencia de Correcciones | El principio de que las soluciones a problemas identificados deben ser robustas y duraderas, diseñadas para evitar regresiones y garantizar la estabilidad a largo plazo. En este sistema, la persistencia a largo plazo de los Resultados OCR recae en la Fuente de Verdad Externa (n8n/AE-03). | Persistencia de Correcciones, Integridad Total, Zero-Fault Detection |
| Persistencia Temporal | Se refiere al almacenamiento de datos dentro del sistema por un período de tiempo muy limitado (ej., hasta 30 días, o solo en memoria mientras el proceso está activo), antes de su eliminación o transferencia a un sistema externo. En este proyecto, aplica a los archivos de log de Monitoreo por Pulso y al estado de los lotes activos en memoria del API Gateway. | Claridad de Definiciones, Rendimiento del Servicio, Simplicidad |
| Precisión | La exactitud y fiabilidad de los datos extraídos o procesados por el sistema, especialmente en el contexto del OCR. Ligado a la Confianza Global OCR. | Precisión, Zero-Fault Detection |
| Procesamiento Espacial | La aplicación de algoritmos basados en la posición y relaciones geométricas del texto detectado en una imagen para una extracción de datos más precisa y robusta, incluso en documentos con variaciones de formato. | Comprensión Profunda del Contexto de Dominio, Precisión |
| Reglas de Extracción | Conjunto de lógica configurable (ej., patrones Regex, coordenadas, dependencias de campos, lógica de validación) que el OCR Worker Service utiliza para identificar, extraer y validar campos específicos de los Documentos de Pago. Estas reglas son el motor de la Extracción Espacial y la Precisión. Su mantenimiento es externo al sistema principal (via JSON) y deben ser Granulares para manejar diversas variaciones de documentos. Su correcta aplicación es vital para la Zero-Fault Detection. | Claridad de Definiciones, Granularidad, Precisión, Zero-Fault Detection, Comprensión Profunda del Contexto de Dominio |
| Rendimiento del Servicio (Activo y Caliente) | El principio de que todo diseño debe considerar la eficiencia, capacidad de respuesta y resiliencia del sistema bajo cualquier condición de carga. En este sistema, se prioriza la velocidad de procesamiento por lote por encima de la persistencia histórica interna. | Rendimiento del Servicio, Escalabilidad |
| Resultado Consolidado del Lote | Un objeto de datos generado por el API Gateway / Backend Service (CO-01) al finalizar el procesamiento de un Lote. Contiene un resumen del lote (ej., idLote), los Resultados OCR enriquecidos de todos los Documentos de Pago que lo componen, y métricas de rendimiento (Monitoreo por Pulso) como tiempo de procesamiento total, latencia promedio por imagen, y estado general del lote (ej., procesadoOk, falloParcial). Su destino es n8n para persistencia y análisis. Es crucial para la Integridad Total de la información entregada a la Fuente de Verdad Externa. | Integridad Total, Monitoreo por Pulso, Fuente de Verdad Externa, Claridad de Definiciones |
| Resultado OCR | El objeto de datos estructurados final que se extrae de un Documento de Pago por el OCR Worker Service. Este resultado es enriquecido y consolidado, conteniendo la metadatosEntrada original del documento, los valores extraídos por el OCR (ej., referencia, monto, fecha), y los resultados del Procesamiento Espacial y Validación OCR. Una vez generado, es enviado inmediatamente a n8n (AE-02). No se persiste a largo plazo en el sistema local, asegurando el Rendimiento del Servicio. Se integra en el Resultado Consolidado del Lote. | Claridad de Definiciones, Rendimiento del Servicio, Persistencia Temporal, Fuente de Verdad Externa, Integridad Total, Trazabilidad, Coherencia de Referencias Críticas, Precisión |
| Servicio de Almacenamiento de Archivos (CO-06) | El componente CO-06 encargado de almacenar los Documentos de Pago originales (imágenes) recibidos y, de forma temporal, los archivos de log resumen/CSV para el Monitoreo por Pulso. No almacena los Resultados OCR procesados a largo plazo, ya que su Fuente de Verdad Externa es n8n. | Escalabilidad, Rendimiento del Servicio, Persistencia Temporal |
| Sorteo | Una entidad de negocio que representa la información clave de un sorteo (ej., idSorteo, fechaSorteo, horaMinutoSorteo) a la que se asocian los pagos. Sus datos son provistos por n8n y utilizados para el procesamiento y renombrado de archivos. | Comprensión Profunda del Contexto de Dominio, Claridad de Definiciones |
| Trazabilidad | La capacidad de seguir el ciclo de vida de un elemento de diseño, desde sus requisitos hasta su implementación y prueba, y viceversa. Es garantizada por la Coherencia de Referencias Críticas y la Matriz de Trazabilidad de Requisitos. | Coherencia de Referencias Críticas, Integridad Total |
| Usuario (que envía el pago) | La persona o entidad identificada por su número de WhatsApp que genera el pago móvil y cuyo Documento de Pago es procesado por el sistema OCR. Su información es parte de la metadatosEntrada del Documento de Pago. Es distinto del Usuario Final (Operador/Administrador) que opera el sistema.. | Comprensión Profunda del Contexto de Dominio, Claridad de Definiciones |
| **Usuario Final (Operador/Administrador) (AE-01)** | Actor humano que interactúa directamente con el Frontend (Dashboard) del sistema para cargar Documentos de Pago, monitorear el Estado Lote y revisar Resultados OCR. Su rol es operacional y de supervisión del sistema. **Es distinto del Usuario (que envía el pago) cuyo documento es procesado.** | Claridad de Definiciones, Comprensión Profunda del Contexto de Dominio |
| Validador OCR | El componente o proceso encargado de aplicar reglas de negocio y consistencia a los datos extraídos por el OCR para determinar su estadoValidacion. | Zero-Fault Detection, Precisión |
| Zero-Fault Detection | Principio obsesivo de identificar y prevenir errores en las etapas más tempranas del diseño, antes de que tengan la oportunidad de propagarse al código. Exige la definición explícita de manejo de errores, validaciones y casos de borde. | Zero-Fault Detection, Integridad Total, Perfección Continua |

**DIAGRAMA DE INTERACCIONES DE CONTENEDORES Y FLUJO DE DATOS (GRÁFICO BLINDADO)**

|  |
| --- |
| graph TD  subgraph Actores Externos  AE01[Usuario Final (Operador/Administrador)]  AE02[Sistema n8n]  AE03[Base de Datos Externa (de n8n)]  end  subgraph Sistema OCR Asíncrono Empresarial  CO05[Frontend (Dashboard) Service]  CO01[API Gateway / Backend Service]  CO04[OCR Worker Service (Motor RADICALMENTE OPTIMIZADO)]  CO06[Servicio de Almacenamiento de Archivos]  end  AE01 -- (1) Carga de Documento de Pago (Imagen + Metadatos) --> CO05  CO05 -- (2) Envío a API Gateway (Imagen + Metadatos) --> CO01  AE02 -- (3) Envío de Documento de Pago (Metadatos + URL Imagen) --> CO01  CO01 -- (4) Almacenamiento de Imagen --> CO06  CO01 -- (5) Inicio de Procesamiento Asíncrono --> CO04  CO04 -- (6) Descarga de Imagen --> CO06  CO04 -- (7) Envío de Resultado OCR (JSON Enriquecido) --> CO01  CO01 -- (8) Notificación de Resultado Consolidado del Lote --> AE02  AE02 -- (9) Persistencia de Resultados --> AE03  CO01 -- (10) Monitoreo de Estado de Lote (Polling/WS) --> CO05  style AE01 fill:#f9f,stroke:#333,stroke-width:2px  style AE02 fill:#add8e6,stroke:#333,stroke-width:2px  style AE03 fill:#90ee90,stroke:#333,stroke-width:2px  style CO01 fill:#ffe4e1,stroke:#333,stroke-width:2px  style CO05 fill:#f0f8ff,stroke:#333,stroke-width:2px /\* CO-05 is Frontend \*/  style CO04 fill:#fffacd,stroke:#333,stroke-width:2px  style CO06 fill:#ffe4b5,stroke:#333,stroke-width:2px  linkStyle 0 stroke-width:2px,fill:none,stroke:red;  linkStyle 1 stroke-width:2px,fill:none,stroke:red;  linkStyle 2 stroke-width:2px,fill:none,stroke:red;  linkStyle 3 stroke-width:2px,fill:none,stroke:blue;  linkStyle 4 stroke-width:2px,fill:none,stroke:green; /\* Direct call CO1->CO4 \*/  linkStyle 5 stroke-width:2px,fill:none,stroke:blue;  linkStyle 6 stroke-width:2px,fill:none,stroke:purple;  linkStyle 7 stroke-width:2px,fill:none,stroke:darkorange;  linkStyle 8 stroke-width:2px,fill:none,stroke:darkorange;  linkStyle 9 stroke-width:2px,fill:none,stroke:brown; |

**12. Mejoras Tecnológicas para la Optimización Asíncrona**

La optimización asíncrona es un pilar fundamental del 'Sistema OCR Asíncrono Empresarial', lograda mediante la implementación de diversas mejoras tecnológicas. El corazón de esta optimización reside en el **OCR Worker Service (CO-04)**, que utiliza un motor **OnnxTR** con modelos pre-cargados y optimizados para inferencia en CPU y GPU básicas vía ONNX Runtime, apoyándose en librerías de alto rendimiento como **NumPy** y **OpenCV** para un procesamiento espacial y validación de altísima eficiencia, capaz de procesar imágenes en fracciones de segundo. El **API Gateway (CO-01)** gestiona los lotes **EN MEMORIA**, lo que permite una ingestión y orquestación extremadamente rápidas sin la latencia de una base de datos persistente interna. La **escalabilidad horizontal** de los Workers OCR (CO-04) y del API Gateway (CO-01) asegura que el sistema puede adaptarse a volúmenes crecientes de documentos sin degradación del rendimiento. Además, el **Servicio de Almacenamiento de Archivos (CO-06)**, diseñado para altísimo rendimiento (ej., S3, MinIO), garantiza un acceso rápido a las imágenes. La comunicación entre componentes se realiza principalmente a través de **webhooks asíncronos**, minimizando los tiempos de espera, y se implementan **mecanismos de reintentos con backoff exponencial** para asegurar la resiliencia ante fallos transitorios de red o servicios externos, todo ello contribuyendo a un **Rendimiento del Servicio (Activo y Caliente)** excepcional.