

## Integrantes:

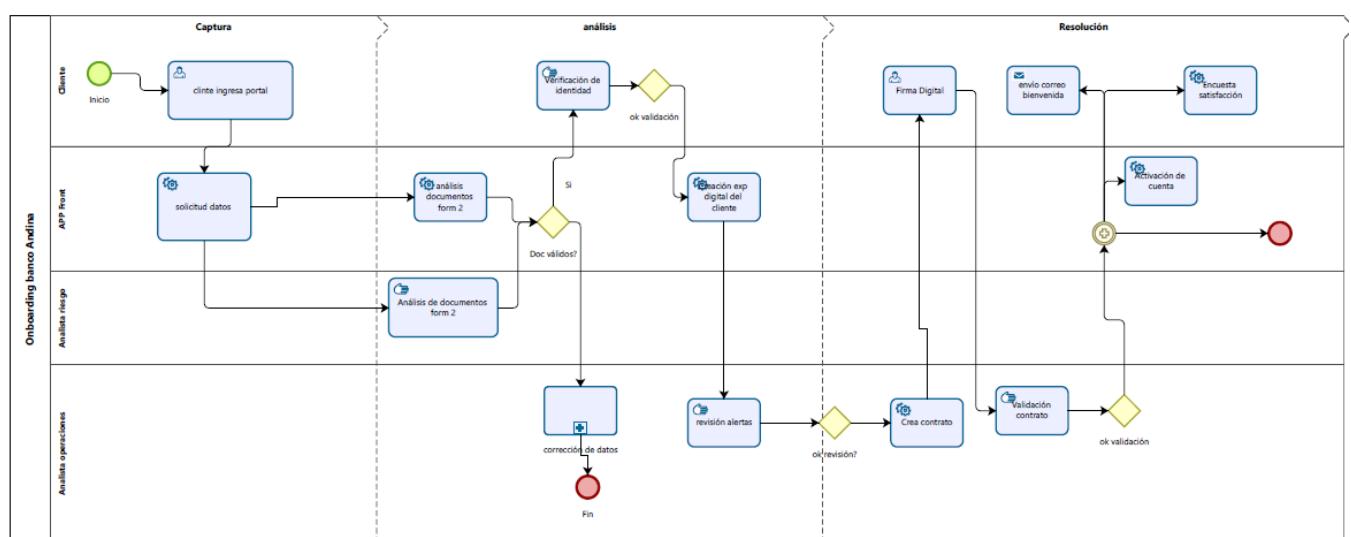
Diana Avelino  
 Johana Correa  
 Ricardo Rivera  
 Adriana Hernandez

## Taller 1 - Definiendo iniciativa GenAI

### Actividad 1 – Entendimiento del proceso

Seleccionamos el caso del Banco Andino para el proceso Onboarding de clientes digitales (personas naturales)

#### Modelo del proceso As Is



### Actividad 2 – Análisis del proceso y sus actividades

Esta es una clasificación detallada de las actividades del proceso. Incluimos una nueva actividad para mejorar la eficiencia del proceso

Categoría 1: Actividad automatizable con motor de procesos o integración entre sistemas.

Categoría 2: Actividad automatizable con RPA.

Categoría 3: Actividad que requiere algún proceso cognitivo potencialmente delegable a la IA Generativa

Categoría 4: Actividad que requiere obligatoriamente de un humano para llevarse a cabo.

Cod	Proceso	Actividad	Categoría	Salida
Pr-01	Solicitud datos	Cliente accede al sitio web o app y diligencia formulario	Categoría 1	Puede integrarse con sistemas de CRM
Pr-02	Solicitud datos	Validación automática de algunos campos del formulario	Categoría 1	Se realiza mediante reglas de negocio y validaciones en tiempo real integradas al sistema.
Pr-03	Análisis Documentos	Revisión manual de campos y documentos por analistas de riesgos y cumplimiento	Categoría 2	La revisión realizada por los analistas de riesgo puede ser automatizada a partir de bots a los que se les configura los criterios de evaluación
Pr-04	Verificación identidad	Llamada o videollamada para verificar identidad del cliente	Categoría 3	Agente de IA generativa que realice las llamadas automatizadas para validar la identidad del cliente, realice

				preguntas de seguridad y complete el proceso de forma autónoma
Pr-05	Creación Expediente	Envío del expediente digital al comité operativo	Categoría 1	Puede automatizarse con integración entre sistemas documentales
Pr-06	Verificación de alertas	Revisión de alertas, inconsistencias o antecedentes financieros	Categoría 1	Puede automatizarse con la integración entre sistemas documentales como FATC, listas restrictivas, PEP
Pr-07	Aprobación	Decisión de aprobación por el comité operativo	Categoría 3	A partir de la definición de los criterios de aceptación y de una muestra de casos ya resueltos se pueden identificar patrones que pueden ser entrenados en una IA con el objetivo de que pueda asumir el rol de decisión en la aprobación
Pr-08	Creación contrato	Generación del contrato digital	Categoría 1	Puede automatizarse con sistemas de gestión documental y firma digital.
Pr-09	Firma digital contrato	Firma digital del contrato por parte del cliente	Categoría 1	Integración con plataformas de firma digital.
Pr-10	Validación contrato	Validación manual posterior (confirmación telefónica de recepción y uso)	Categoría 2	Puede automatizarse con RPA que realice llamadas automáticas o envíe mensajes de confirmación
Pr-11	Activación de cuenta	Activación de la cuenta	Categoría 1	Integración directa con sistemas bancarios tras aprobación.
Pr-12	Envío correo de bienvenida	Envío correo de bienvenida	Categoría 1	Automatizable desde CRM o motor de procesos tras activación de cuenta.
Pr-13	Envío encuesta de satisfacción	Envío encuesta de satisfacción	Categoría 1	Automatizable con otras herramientas.

**Proponga una actividad con procesos cognitivos delegables a la IA que podría hacer más eficiente el proceso.**

	<b>Cod</b>	<b>Proceso</b>	<b>Actividad</b>	<b>Categoría</b>
Actual	Pr-03	Análisis Documentos	Revisión manual de campos y documentos por analistas de riesgos y cumplimiento	Categoría 2
Eficiencia	Pr-03	Análisis Documentos	Análisis inteligente de inconsistencias en formularios y Documentos de onboarding	Categoría 3

#### **Descripción Eficiencia**

Para lograr esta eficiencia se requiere una IA generativa que analice los formularios completados por clientes, detecta inconsistencias entre datos personales, laborales y de origen de fondos, detecta inconsistencias o documentos faltantes, clasifica tipo de documentos y genera alertas explicativas para el analista de cumplimiento.

#### **Salidas esperadas del proceso cognitivo**

- Estado del expediente (completo, incompleto, inconsistente).
- Lista de inconsistencias detectadas con explicación contextual.
- Nivel de riesgo preliminar.
- Recomendaciones automatizadas para el analista.
- Campos faltantes o contradictorios.
- Sugerencia de escalamiento o aprobación directa

#### **Beneficios**

Esto reduce el tiempo de revisión manual, mayor precisión en la detección de errores o documentos incompletos, escalabilidad del proceso sin necesidad de aumentar el personal, mejora la calidad del expediente, mejora en tiempos de respuesta y todo esto en conjunto mejora la experiencia del cliente.

Las tecnologías que sugerimos para lograr las salidas que esperamos:

- OCR avanzado para extracción de texto desde imágenes y PDFs
- Clasificadores para identificar tipo de documento entrenados con ejemplos de documentos reales
- Modelos de lenguaje como GPT para inferencia semántica, detección de inconsistencias y generación de alertas explicativas
- Integración con sistemas de gestión documental (DMS) para trazabilidad y almacenamiento

- Modelos de lenguaje como GPT integrados con OCR

JSON- Análisis de formulario	JSON- Análisis de Documentos
<pre>{   "cliente_id": "C0012456789",   "estado_expediente": "Inconsistente",   "nivel_riesgo": "Medio",   "inconsistencias_detectadas": [     {       "campo": "ocupacion",       "tipo": "Contradicción semántica",       "descripcion": "El cliente declara ser freelance en diseño gráfico, pero el certificado laboral indica contrato a término indefinido con empresa de tecnología.",       "accion_recomendada": "Solicitar aclaración sobre doble fuente de ingresos."     },     {       "campo": "origen_fondos",       "tipo": "Omisión documental",       "descripcion": "Se menciona ingresos por NFTs en la declaración, pero no se adjunta soporte ni historial de transacciones.",       "accion_recomendada": "Solicitar evidencia complementaria si el monto supera umbral de riesgo."     }   ],   "documentos_faltantes": [     {       "tipo_documento": "Comprobante de residencia",       "descripcion": "No se encontró documento que valide dirección actual del cliente."     }   ],   "recomendaciones": [     "Solicitar comprobante de residencia.",     "Validar ingresos por NFTs con soporte adicional.",     "Escalar expediente si no se recibe documentación en 48 horas."   ],   "resumen_general": "Se detectaron 2 inconsistencias relevantes y 1 documento faltante. Requiere revisión parcial por analista AML." } }</pre>	<pre>{   "cliente_id": "C0012456789",   "documentos_procesados": [     {       "nombre_documento": "cedula_frontal.jpg",       "tipo_documento": "Documento de identidad",       "datos_extraidos": {         "nombre_completo": "Laura Gómez Pérez",         "numero_documento": "1023456789",         "fecha_nacimiento": "1990-05-12",         "fecha_emision": "2018-06-01",         "lugar_emision": "Bogotá D.C."       },       "estado_documento": "Completo",       "alertas": []     },     {       "nombre_documento": "certificado_laboral.pdf",       "tipo_documento": "Certificado laboral",       "datos_extraidos": {         "empresa": "TechNova S.A.S.",         "cargo": "Analista de datos",         "fecha_ingreso": "2021-03-15",         "salario_mensual": "4.500.000 COP"       },       "estado_documento": "Completo",       "alertas": []     },     {       "nombre_documento": "declaracion_origen_fondos.docx",       "tipo_documento": "Declaración de origen de fondos",       "datos_extraidos": {         "fuente_principal": "Salario",         "fuente_secundaria": "Inversiones en criptomonedas",         "detalle": "Opera en plataformas como Binance y Bitso desde 2022"       },       "estado_documento": "Inconsistencia detectada",       "alertas": [         {           "tipo_alerta": "Inconsistencia con certificado laboral",           "descripcion": "El cliente declara ingresos por criptomonedas, pero no se menciona en el certificado laboral ni en otros documentos.",           "nivel_riesgo": "Medio",           "accion_recomendada": "Solicitar soporte adicional sobre ingresos por inversión."         }       ]     },     {       "tipo_documento": "Comprobante de residencia",       "descripcion": "No se encontró documento que valide dirección actual del cliente."     }   ],   "resumen_general": {     "estado_expediente": "Pendiente por aclaración",     "documentos_completos": 2,     "documentos_con_alertas": 1,     "documentos_faltantes": 1,     "recomendaciones": [       "Solicitar comprobante de residencia.",       "Validar ingresos por criptomonedas con soporte adicional.",       "Escalar expediente si no se recibe documentación en 48 horas."     ]   } }</pre>

		[ } }
--	--	-------------

Para las actividades de la categoría 3 describa cómo sería la estructura de salida del proceso cognitivo imaginando que le pide la respuesta al LLM en un JSON

Cod	Proceso	Actividad	Categoría	
Actual	Pr-04	Verificación identidad	Llamada o videollamada para verificar identidad del cliente	Categoría 3

#### Descripción

Integrar un agente virtual de voz que realice la llamada al cliente para la validación de identidad partiendo del guion definido y que aprenda secuencialmente de los datos. El agente debería:

- Permitir la modificación del tono de voz para customizar con la región del consumidor
- Realizar las llamadas en los horarios definidos por ley
- Responder las dudas del cliente en el proceso con lenguaje natural
- Retroalimentar los sistemas internos del banco para trazabilidad y auditoria
- Reintentar en caso de problemas de señal (llamada intermitente), Buzón de voz, o por solicitud del cliente.

#### Salidas esperadas del proceso cognitivo:

- Validación de coincidencia entre datos declarados y respuestas verbales.
- Detección de evasivas, contradicciones o incoherencias.
- Análisis de tono emocional (seguridad, evasión, ansiedad).
- Nivel de confianza en la verificación.
- Recomendaciones para escalamiento, aclaración o aprobación directa.
- Registro estructurado de la interacción para trazabilidad.

#### Beneficios

- Escalabilidad: El agente puede realizar llamadas simultáneas, a diferencia de los agentes humanos que solo pueden atender una llamada a la vez. En caso de una proyección de crecimiento o campañas, no existe una dependencia de la capacidad operativa para soportar el proceso
- Optimización del tiempo de verificación: de 8-12 min agente humano a 2-3 min agente virtual
- Eliminación de colas de espera: Procesamiento inmediato de las solicitudes.
- Consistencia: 100% de adherencia al guion y protocolo de verificación.
- Productividad: Incremento de la productividad de 400%. Un agente humano puede atender entre 30 y 40 llamadas por día. Un agente virtual puede atender entre 160 y 200 llamadas por día. (en el modelo no se tienen en cuenta las llamadas simultaneas)

Las tecnologías que sugerimos para lograr las salidas se requiere la implementación de modelos de lenguaje como GPT o Claude para la comprensión del contexto y la intención

#### JSON

```
{
  "cliente_id": "C0098765432",
  "verificacion_ia": {
    "preguntas_clave": [
      {
        "pregunta": "¿Cuál es su número de documento?",
        "respuesta_cliente": "1023456789",
        "coincide_con_registro": true
      },
      {
        "pregunta": "¿Cuál es su dirección actual?",
        "respuesta_cliente": "Carrera 15 #45-23, Bogotá",
        "coincide_con_registro": false,
        "alerta": "Dirección verbalizada no coincide con la registrada en el formulario."
      },
      {
        "pregunta": "¿Cuál es su fuente principal de ingresos?",
        "respuesta_cliente": "Trabajo como freelance en diseño gráfico",
        "coincide_con_registro": true
      }
    ]
  }
}
```

```

        "coincide_con_registro": true
    },
],
"tono_detectado": "Ansiedad leve",
"nivel_confianza": "Moderado",
"inconsistencias_detectadas": [
{
    "tipo": "Dirección no coincidente",
    "descripcion": "La dirección verbalizada difiere de la registrada. Puede ser error o cambio reciente.",
    "nivel_riesgo": "Bajo",
    "accion_recomendada": "Solicitar comprobante actualizado de residencia."
},
],
"recomendaciones": [
    "Validar dirección con documento adicional.",
    "No se requiere escalamiento inmediato.",
    "Aprobación posible si se corrige inconsistencia menor."
],
"estado_verificacion": "Parcialmente aprobada"
}
}

```

Cod	Proceso	Actividad	Categoría
Actual	Pr-07	Aprobación	Decisión de aprobación por el comité operativo

#### Descripción

Asistir al comité operativo en la toma de decisiones sobre la aprobación o rechazo de expedientes complejos, mediante un análisis argumentado de la información disponible. La IA generativa actúa como asistente cognitivo que:

- Resume el expediente.
- Identifica riesgos, inconsistencias o faltantes.
- Evalúa criterios de aprobación definidos por la organización.
- Propone una recomendación argumentada.
- Genera una salida estructurada para trazabilidad y auditoría.

#### Salidas esperadas del proceso cognitivo:

- Resumen ejecutivo del expediente.
- Identificación de riesgos clave (AML, operativos, reputacionales).
- Evaluación de cumplimiento de criterios de aprobación.
- Recomendación argumentada (aprobar, rechazar, escalar).
- Nivel de confianza de la IA en su recomendación.
- Justificación textual y estructurada.
- Formato JSON para integración con sistemas de decisión o actas del comité.

JSON
{     "expediente_id": "EXP-2025-0456",     "cliente_id": "C0012456789",     "resumen_expediente": {         "tipo_cliente": "Persona natural",         "producto_solicitado": "Cuenta corriente + tarjeta de crédito",         "ingresos_mensuales": "4.500.000 COP",         "origen_fondos": "Servicios de consultoría",         "documentacion_completa": true,         "verificacion_identidad": "Aprobada con observación leve"     },     "riesgos_identificados": [         {             "tipo": "AML",             "descripcion": "Declaración de ingresos incluye actividades en criptoactivos sin soporte documental.",             "nivel_riesgo": "Medio",             "accion_recomendada": "Solicitar soporte adicional o escalar a oficial de cumplimiento."         },         {             "tipo": "Reputacional",             "descripcion": "Cliente figura como socio en empresa recientemente sancionada por la SIC.",             "nivel_riesgo": "Alto"         }     ] }

```

        "accion_recomendada": "Escalar a comité de cumplimiento para revisión adicional."
    },
    "evaluacion_criterios_aprobacion": {
        "criterio_ingresos_minimos": true,
        "criterio_documentacion": true,
        "criterio_riesgo_aceptable": false,
        "criterio_verificacion_identidad": true
    },
    "recomendacion_ia": {
        "decision_sugerida": "No aprobar",
        "nivel_confianza": "Alto",
        "justificacion": "Aunque el cliente cumple con ingresos y documentación, los riesgos AML y reputacionales superan el umbral de tolerancia definido por la política interna. Se recomienda escalar a comité de cumplimiento."
    },
    "proxima_accion": "Escalar a comité de cumplimiento para decisión final"
}

```

### Actividad 3 – Selección de la actividad para la PoC (Proof of Concept-Prueba de Concepto)

Cada miembro del grupo deberá votar por las actividades identificadas en la categoría 3 y la nueva actividad de tal manera que el grupo seleccione al final una actividad para trabajar una PoC en el siguiente taller. Asegúrense de documentar su proceso de selección.

Para realizar la votación tenga en cuenta los siguientes criterios:

- Posibilidad de hacer uso de LLMs para realizar la actividad.
- Posibilidad de hacer pruebas con LLMs específicos.
- Viabilidad de que generando como resultado un JSON o algún formato estructurado esta salida se pueda conectar con el siguiente proceso de negocio.

	Cod	Proceso	Actividad	Uso y prueba con LLMs	Generacion de JSON de salida	Voto
Eficiencia	Pr-03	Análisis Documentos	Análisis inteligente de inconsistencias en formularios y Documentos de onboarding	<p><b>GPT-4 (OpenAI)</b></p> <p><u>Ventajas:</u> Excelente comprensión contextual, generación de resúmenes, extracción de datos, razonamiento complejo.</p> <p><u>Ideal para:</u> Procesar documentos variados, detectar inconsistencias, generar respuestas estructuradas en JSON.</p> <p><u>Acceso:</u> API de OpenAI.</p> <p><b>Claude Sonnet 3.7 (Anthropic)</b></p> <p><u>Ventajas:</u> Gran capacidad de análisis semántico, enfoque en seguridad y precisión.</p> <p><u>Ideal para:</u> Procesos que requieren interpretación cuidadosa de texto.</p> <p><u>Acceso:</u> Amazon Bedrock, Google Vertex AI.</p> <p><b>Gemini 2.5 Pro (Google DeepMind)</b></p> <p><u>Ventajas:</u> Multimodal, muy buena en tareas de comprensión y extracción de información.</p> <p><u>Ideal para:</u> Documentos con texto e imagen (como escaneos).</p> <p><u>Acceso:</u> Google Cloud.</p>	<input checked="" type="checkbox"/> Sí Con las plataformas es totalmente posible generar salidas estructuradas en formato JSON que se puedan conectar directamente con el siguiente proceso de negocio	<input checked="" type="checkbox"/> Diana <input checked="" type="checkbox"/> Ricardo <input checked="" type="checkbox"/> Adriana
Actual	Pr-04	Verificación identidad	Llamada o videollamada para verificar identidad del cliente	<p><b>Google Cloud Speech-to-Text (STT)</b> reconocimiento de voz en tiempo real</p> <p><b>GPT 4 ó Claude</b> para la comprensión del contexto y la intención. Gestión contextual</p> <p><b>Google Cloud Speech-to-Text (STT) ó amazon polly:</b> estandarización con voces neutrales con personalización regional.</p>	Parcial	<input checked="" type="checkbox"/> Johana

**Google Cloud Speech-to-Text (STT) ó amazon polly:**  
estandarización con voces neutrales con personalización regional.

Actual	Pr-07	Aprobación	Decisión de aprobación por el comité operativo	<p><b>Twilio Voice Api / Amazon connect:</b> para telefonía y gestión del flujo conversacional basado en el guion</p> <p><b>DeepSeek-R1 (DeepSeek AI)</b></p> <p><u>Ventajas:</u> Razonamiento legal avanzado, ventana de contexto de 164K tokens.</p> <p><u>Ideal para:</u> Revisión de contratos, análisis de cláusulas, detección de riesgos legales.</p> <p><u>Acceso:</u> SiliconFlow.</p> <p><b>Qwen3-235B-A22B (Alibaba)</b></p> <p><u>Ventajas:</u> Precisión jurídica, entrenamiento en corpus legal.</p> <p><u>Ideal para:</u> Validación de contratos complejos y cumplimiento normativo.</p> <p><u>Acceso:</u> Plataformas de código abierto.</p> <p><b>GPT-OSS-120B (OpenAI, versión open source)</b></p> <p><u>Ventajas:</u> Adaptable a entornos legales, buena capacidad de personalización.</p> <p><u>Ideal para:</u> Firmas legales que requieren control total sobre datos.</p> <p><u>Acceso:</u> Requiere infraestructura propia.</p>	⚠️ Parcial
--------	-------	------------	--	---	------------

#### Actividad elegida: Pr-03 – Análisis inteligente de inconsistencias en formularios y documentos de onboarding

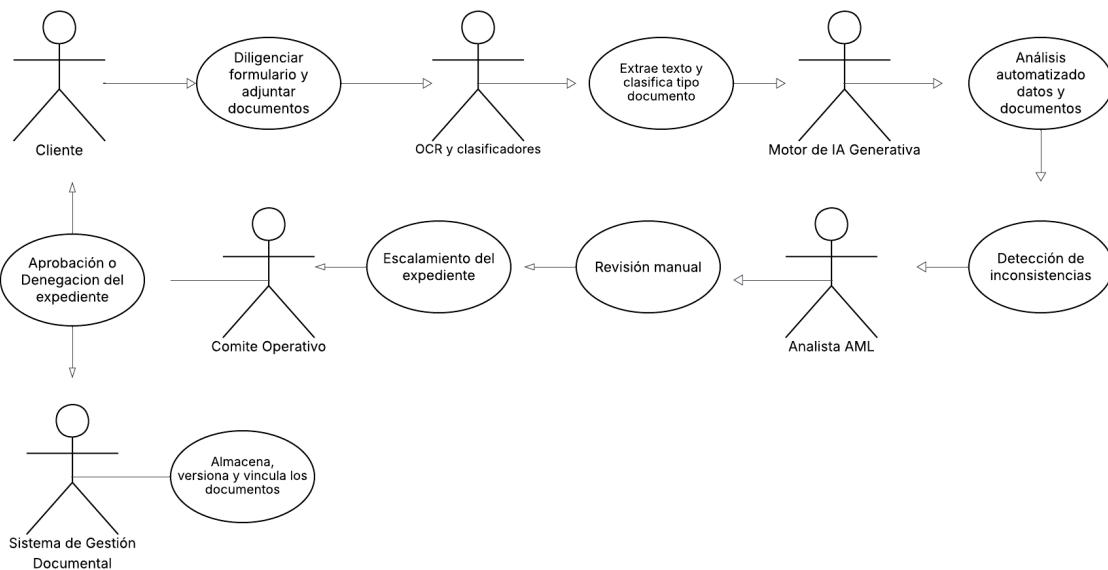
##### Justificación:

- ✓ Alta viabilidad técnica: Compatible con múltiples LLMs ya disponibles.
- ✓ Pruebas inmediatas posibles: Acceso a modelos vía API (OpenAI, Google, Anthropic).
- ✓ Salida estructurada clara: JSON con campos clave que se conectan directamente con el proceso de revisión aprobación.
- ✓ Escalabilidad institucional: Aplicable a múltiples procesos de onboarding, cumplimiento y gestión documental.

##### Roles multiagente:

Rol	Responsabilidad Principal	Interacción con IA Generativa	Tipo de Salida Esperada
Cliente	Diligencia formulario y adjunta documentos.	Indirecta	Datos + documentos
Motor de IA Generativa	Analiza semánticamente datos y documentos.	Directa	JSON con inconsistencias y recomendaciones
OCR + Clasificadores	Extrae texto y clasifica tipo documental.	Integrado	Texto limpio + etiquetas
Sistema de Gestión Documental	Almacena y vincula expediente digital.	Integrado	Expediente enriquecido
Analista AML / Cumplimiento	Revisa alertas generadas por IA.	Directa	Validación + retroalimentación
Comité Operativo	Decide aprobación o escalamiento.	Indirecta	Informe consolidado

## Diagrama Caso de uso principal



**Enlace para editar:** [https://lucid.app/lucidchart/7e176651-f981-4a6a-820a-6cbd9f2a12c0/edit?viewport\\_loc=-1651%2C-754%2C3327%2C1407%2C0\\_0&invitationId=inv\\_06891bda-3568-4756-9ab1-d1168ecd49ff](https://lucid.app/lucidchart/7e176651-f981-4a6a-820a-6cbd9f2a12c0/edit?viewport_loc=-1651%2C-754%2C3327%2C1407%2C0_0&invitationId=inv_06891bda-3568-4756-9ab1-d1168ecd49ff)

## Proyecto Hito 1

### Parte I: Actividad 1 – Integrando el taller 1

#### Correcciones:

En respuesta a las observaciones recibidas, se realizó una revisión integral del documento, corrigiendo la redacción y ajustando los contenidos según las recomendaciones planteadas.

#### Definición a alto nivel de la solución (PoC):

#### Reto identificado:

El Banco Andino enfrenta inconsistencias entre la información ingresada en los formularios digitales y los documentos cargados por los clientes durante el onboarding de personas naturales.

Estas validaciones manuales generan demoras, errores y riesgos de cumplimiento (KYC, AML, SARLAFT), afectando la experiencia del cliente digital.

#### Objetivo de la PoC:

Desarrollar una solución inteligente que automatice la detección de inconsistencias entre los formularios y documentos del proceso de vinculación, mejorando la calidad de la información, la eficiencia operativa y la seguridad del onboarding digital.

#### Descripción general de la solución:

La PoC implementará un módulo de análisis inteligente que:

- ✓ Extrae datos de documentos mediante OCR y procesamiento inteligente (IDP).
- ✓ Compara la información con los formularios digitales.
- ✓ Identifica y clasifica inconsistencias o alertas usando IA y reglas de negocio.
- ✓ Genera un reporte automático para revisión o aprobación.

#### Valor esperado:

- ✓ Reducción de tiempos de validación y errores manuales.
- ✓ Mayor detección de fraudes y cumplimiento normativo.
- ✓ Experiencia digital más ágil y confiable para el cliente.

## Parte II: Maduración del MVP

Link del repositorio: <https://github.com/adrianafernandezbejarano/BancoAndino-onboarding-PoC>

### Actividad 1: Construyendo diagramas con GenAI (ChatGPT y Claude)

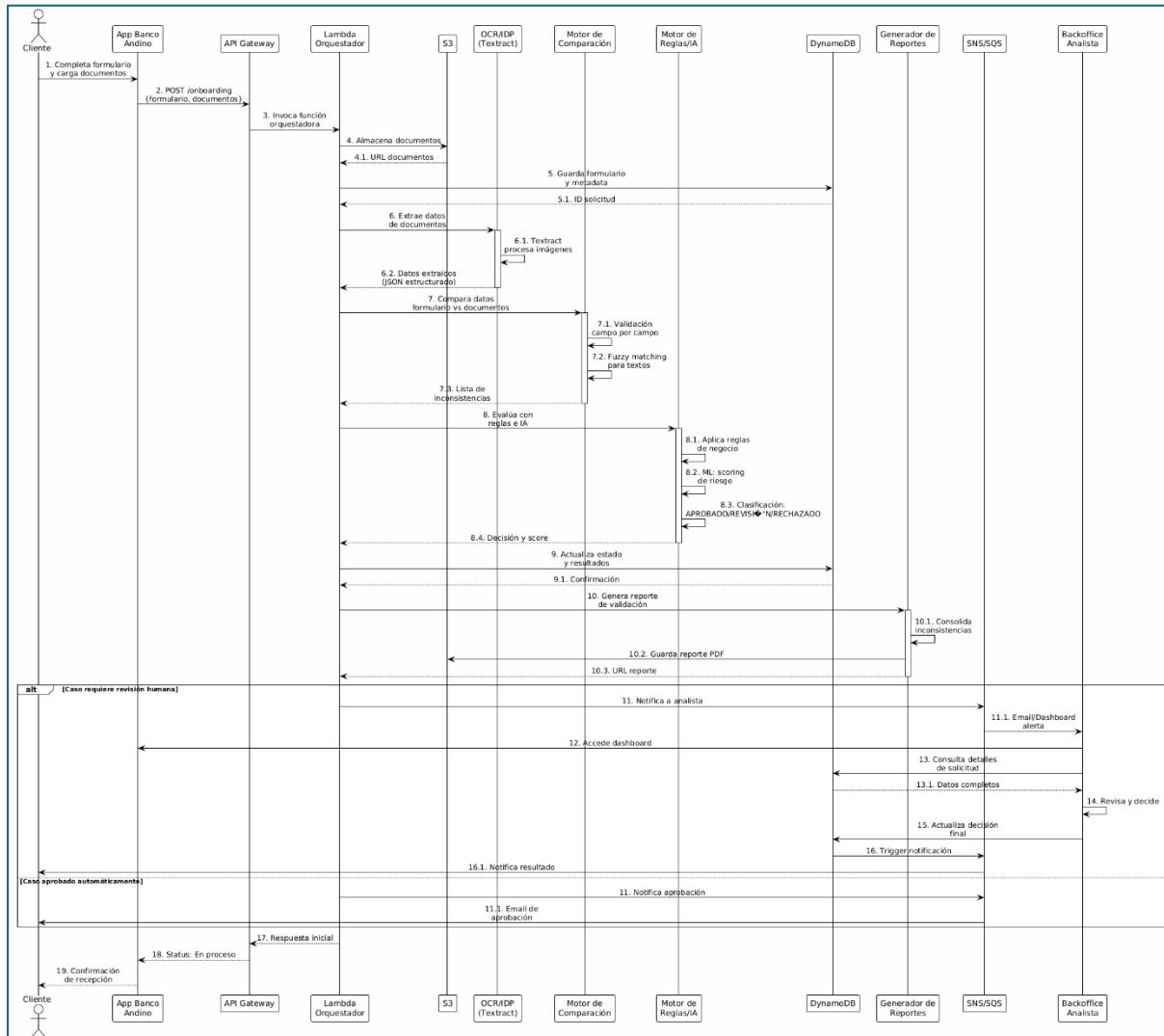
Uso de GenAI para idear, diseñar y representar la solución: Se aprovechan las herramientas GenAI de forma activa para construir y analizar diagramas, arquitectura y entidades, comparando resultados entre modelos.

Para adaptar los diagramas a nuestro proyecto Análisis inteligente de inconsistencias en formularios y Documentos de onboarding creamos este prompt, siendo específicos en componentes de arquitectura:

PROMPT – Diagramas Análisis inteligente de inconsistencias en formularios y Documentos de onboarding	
<p>Diseña una solución inteligente para el Banco Andino que automatice la detección de inconsistencias entre formularios digitales y documentos cargados por clientes durante el proceso de onboarding de personas naturales. La solución debe abordar los siguientes aspectos:</p> <p>Objetivo:</p> <ul style="list-style-type: none"><li>- Mejorar la calidad de la información, la eficiencia operativa y la seguridad del onboarding digital.</li><li>- Reducir errores manuales, tiempos de validación y riesgos de cumplimiento (KYC, AML, SARLAFT).</li></ul> <p>Componentes clave a nivel arquitectónico:</p> <ul style="list-style-type: none"><li>- Capa de Cliente: App de onboarding digital, autenticación segura (Cognito).</li><li>- Capa de Entrada: API Gateway para recepción de formularios y documentos.</li><li>- Capa de Almacenamiento: Amazon S3 para documentos, DynamoDB para formularios y resultados.</li><li>- Capa de Procesamiento Inteligente:<ul style="list-style-type: none"><li>- OCR con Amazon Rekognition o Textract.</li><li>- IDP/NLP con Amazon Comprehend o SageMaker.</li></ul></li><li>- Comparador de datos (Lambda).</li><li>- Motor de reglas e IA (Lambda + SageMaker).</li><li>- Capa de Reportes y Alertas:</li><li>- Generador de reportes automáticos (Lambda).</li><li>- Notificación mediante SQS/SNS.</li><li>- Capa de Revisión Humana: Dashboard para analistas, flujo de revisión manual.</li><li>- Capa de Seguridad y Auditoría: IAM, KMS, CloudWatch, AWS Config.</li></ul>	<p>Diagrama de Secuencia (PlantUML): Genera un diagrama que muestre el flujo entre:</p> <ul style="list-style-type: none"><li>- Cliente</li><li>- App Banco Andino</li><li>- Módulo OCR/IDP</li><li>- Motor de Comparación</li><li>- Motor de Reglas/IA</li><li>- Generador de Reportes</li><li>- Backoffice/Analista</li></ul> <p>Diagrama de Interacción de Entidades: Muestra cómo interactúan los siguientes elementos:</p> <ul style="list-style-type: none"><li>- Formulario digital</li><li>- Documento cargado</li><li>- OCR/IDP</li><li>- Comparador</li><li>- Motor de reglas</li><li>- Reporte</li><li>- Analista</li></ul> <p>Arquitectura Tentativa en AWS (usando librería Diagrams en Python): Genera un diagrama que incluya:</p> <ul style="list-style-type: none"><li>- Cognito, API Gateway, Lambda, S3, DynamoDB</li><li>- Rekognition, Comprehend, SageMaker</li><li>- SQS/SNS, CloudWatch, IAM, KMS</li><li>- Agrupación por capas funcionales</li></ul> <p>El diseño debe ser modular, escalable y trazable, alineado con buenas prácticas de seguridad y cumplimiento normativo. incorpore la respuesta en un archivo markdown en un proyecto nuevo en Cursor</p>

## GPT: Archivo markdown: banco\_andino\_onboarding.md (Ver ruta Github)

### Diagrama de secuencia



### Diagrama de interacción de entidades

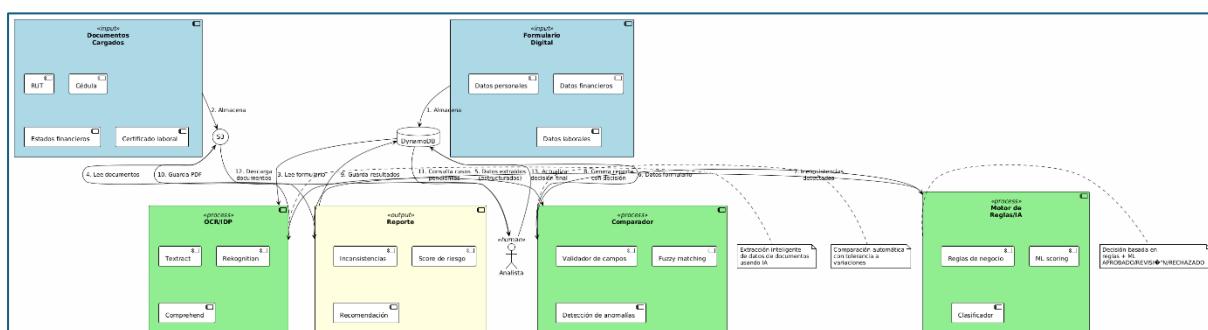
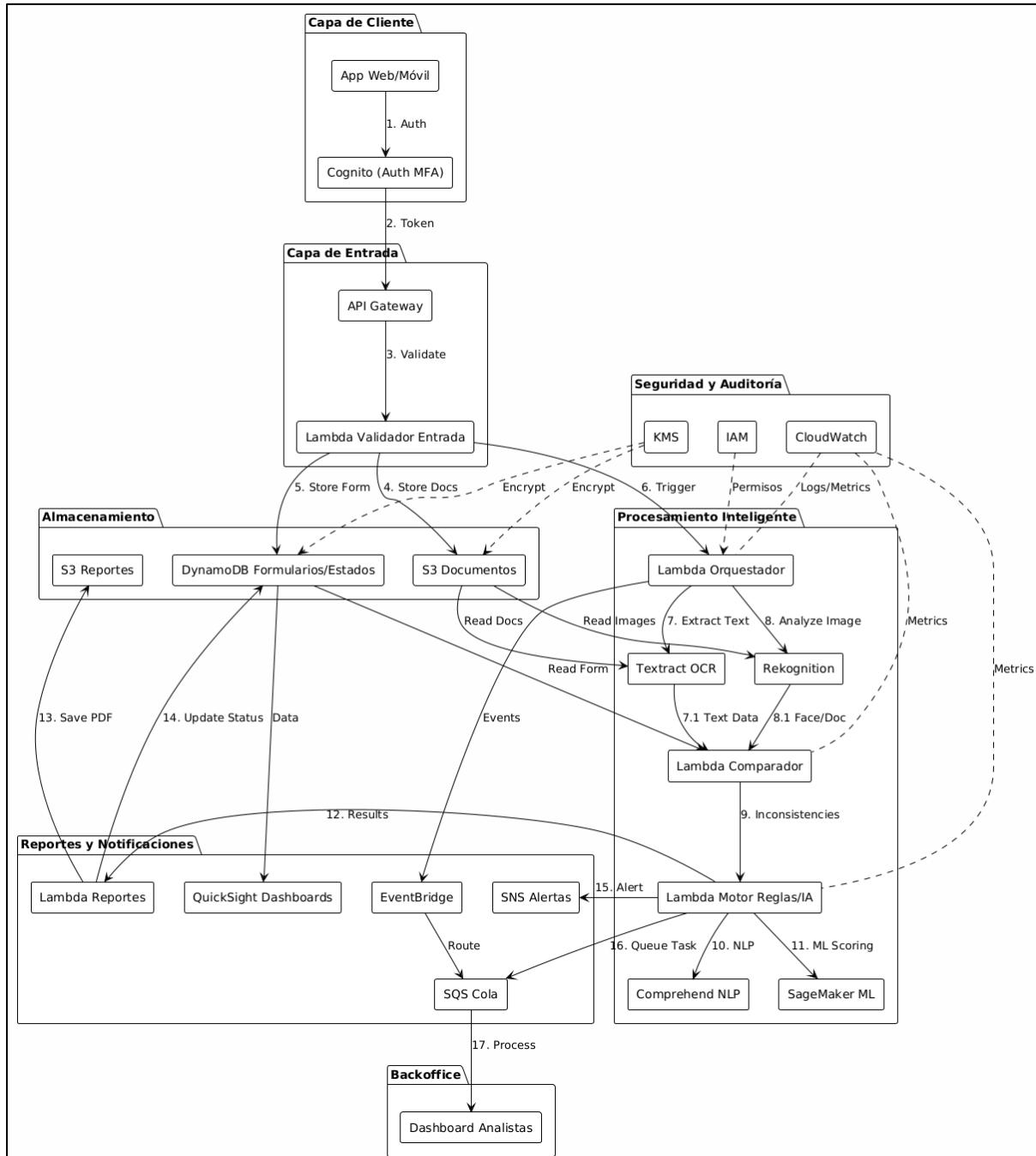
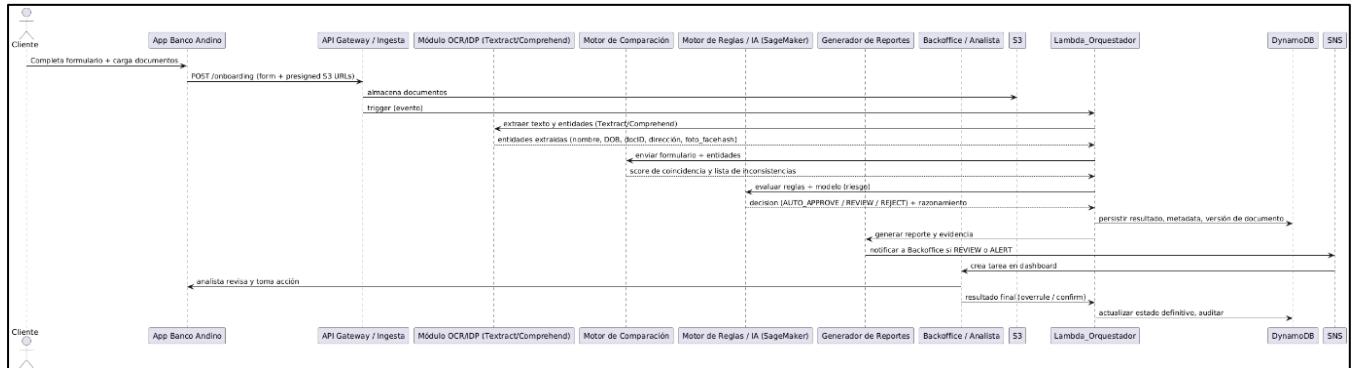


Diagrama arquitectura tentativa para AWS

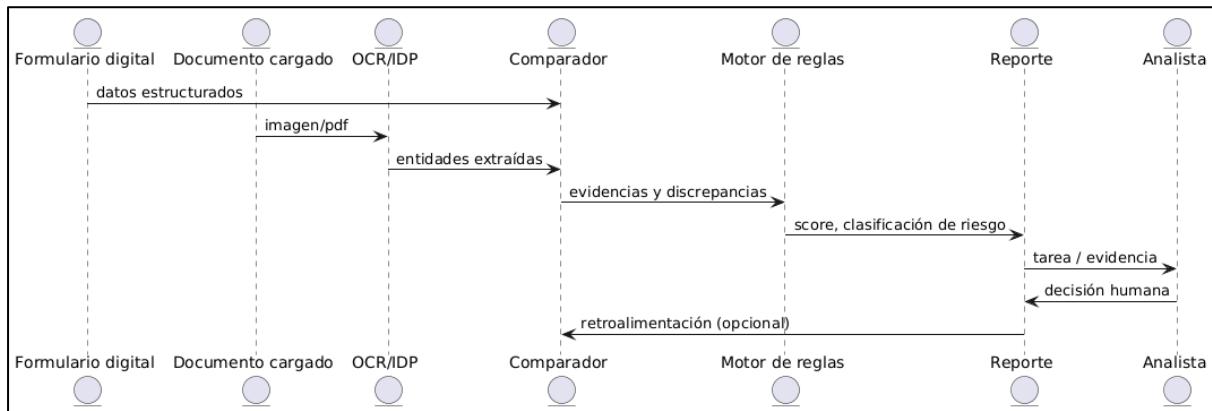


Claude: Archivo markdown: **banco\_andino\_onboarding\_solucion.md** (Ver ruta Github)

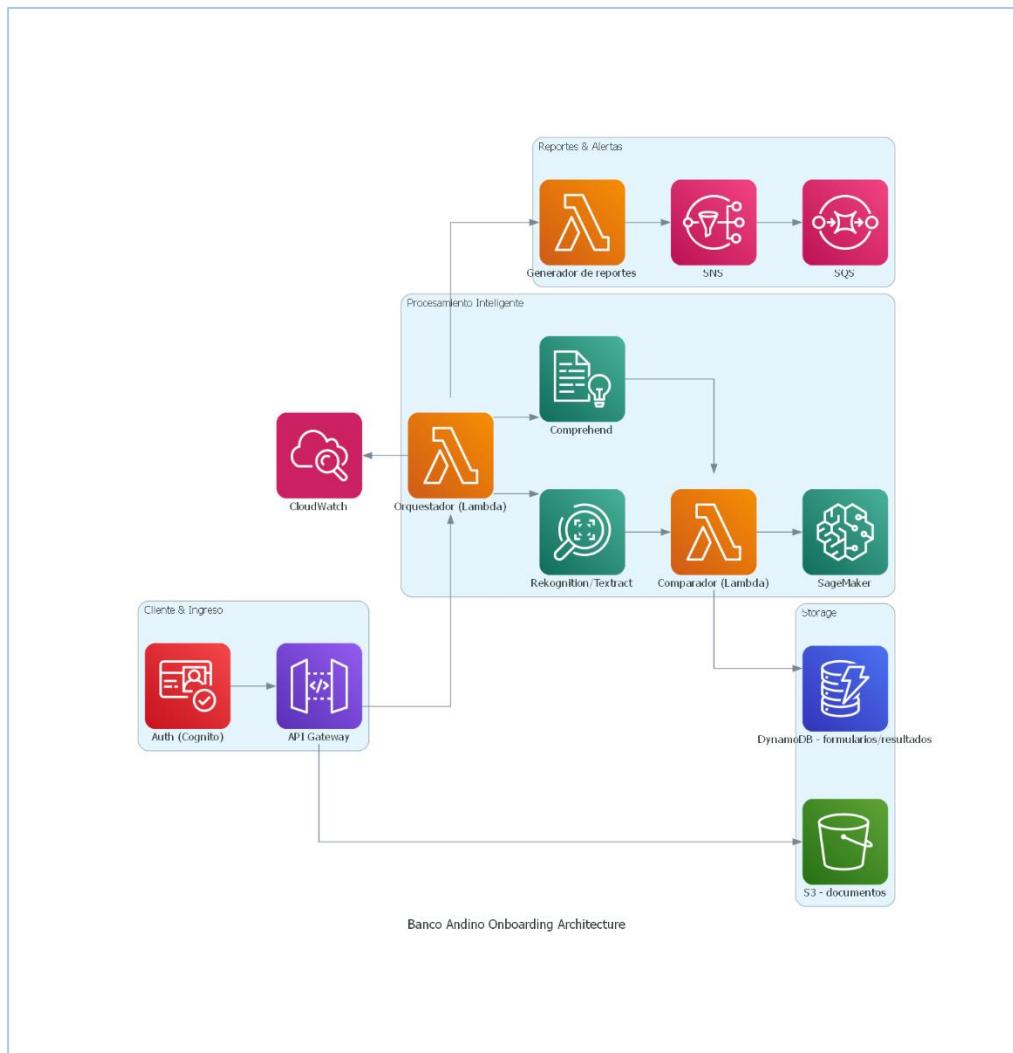
**Diagrama de secuencia**



**Diagrama de interacción de entidades**



## Diagrama arquitectura tentativa para AWS



## Análisis comparativo

	Aspecto	GPT	Claude	Análisis
Enfoque General	Propuesta ejecutiva y arquitectónica, orientada a negocio y cumplimiento	Guía técnica y práctica, enfocada en implementación, código y despliegue.	Ambos buscan automatizar la detección de inconsistencias en el onboarding digital del Banco Andino usando AWS	
Diferencias clave	Propósito	Diseño conceptual y estratégico.	Implementación técnica lista.	
	Estilo	Formal, descriptivo y con KPIs.	Técnico, con código, JSON y diagramas.	
	Público	Ejecutivos, arquitectos, auditores.	Desarrolladores y DevOps.	
	Diagramas	Más detallados (PlantUML + AWS completo).	Más simples y ejecutables (diagrams Python).	
	Seguridad	Incluye TLS, VPC, Config, IAM avanzado.	IAM, KMS, CloudWatch básicos.	
	Código	Solo diagramas.	Handlers Lambda y contratos JSON.	
KPIs y fases	Incluye métricas y plan por etapas.	-	Incluye pruebas y dataset técnico.	

<b>Diagramas</b>				
<b>Secuencia (PlantUML)</b>	- Completo, con pasos numerados del 1 al 19. - Ideal para documentación formal y auditoría	- Más simple, sin numeración, enfocado en flujo lógico. - Ideal para presentaciones técnicas o documentación de código		Usar el diagrama de secuencia de GPT (para documentación)
<b>Interacción de Entidades (PlantUML)</b>	- Detallado con notas, colores y roles (OCR, DB, Analista). - Más explicativo y pedagógico	- Minimalista, centrado en entidades funcionales y sus flujos. - Más limpio y reutilizable en código.		Usar el diagrama de entidades de GPT (para documentación)
<b>Arquitectura AWS (Python / Diagrams)</b>	- Extenso, con EventBridge, QuickSight y control de seguridad. - Más realista y completo	- Compacto, solo los servicios esenciales (Cognito, API, S3, DynamoDB, Lambdas, ML). - Más ágil, portable y ejecutable		Usar el diagrama de arquitectura AWS de Cloud (para implementación y demos)
<b>Coincidencias</b>	Mismo objetivo y flujo funcional: formulario → OCR → comparación → IA → reporte → revisión humana Uso de AWS Textract, Comprehend, SageMaker, Lambda, DynamoDB, S3, Cognito Cumplimiento con KYC/AML/SARLAFT y enfoque modular			
<b>Conclusiones</b>	Destaca en visión estratégica, cumplimiento y documentación ejecutiva	Sobresale en detalle técnico, código y reproducibilidad		Integrar ambos produciría un documento ideal: GPT aporta la estrategia y seguridad; Cloud, la implementación práctica

## Actividad 2: Planeando un Software con GenAI- Claude

Promt
Eres un experto en producto, con experiencia en Sistema Inteligente de Validación Documental (SIVD) con enfoque para el Banco Andino que automatice la detección de inconsistencias entre formularios digitales y documentos cargados por clientes durante el proceso de onboarding de personas naturales
¿Qué funcionalidades básicas tiene un Sistema Inteligente de Validación Documental (SIVD)? Describelas en un listado, ordenado de mayor a menor prioridad
¿Qué beneficios obtiene el cliente de un Sistema Inteligente de Validación Documental (SIVD) para considerar su uso?
¿Cómo es el customer journey normal de un cliente que usa un Sistema Inteligente de Validación Documental (SIVD)? Describeme paso a paso todas las interacciones
Eres un analista de software experto. Estoy construyendo un MVP de un Sistema Inteligente de Validación Documental (SIVD) que haga detección de inconsistencias entre formularios digitales y documentos cargados por clientes durante el proceso de onboarding de personas naturales. Enumera y describe brevemente los casos de uso más importantes a implementar para lograr una funcionalidad básica
Representa estos casos de uso en el tipo de diagrama más adecuado usando el formato plantUML. Diferencia entre usuarios visitantes y usuarios logueados. Acorde a la sintaxis y buenas prácticas UML, define y describe lo que sea necesario.
Eres un arquitecto de software experto. Cuales son las 3 entidades de modelo de datos esenciales en un sistema de Sistema Inteligente de Validación Documental (SIVD)? Dame algunos campos esenciales de cada una y cómo se relacionan
Eres un arquitecto de software. Eres capaz de diseñar, explicar y diagramar los diferentes aspectos de un sistema de software.
Estoy construyendo un sistema de Sistema Inteligente de Validación Documental (SIVD). He definido las entidades Cliente, Formulario digital, Documento cargado, OCR/IDP, Entidades extraídas, Motor de comparación, Motor de reglas / IA, Resultado de validación, Reporte, Analista / Auditoría con sus campos y relaciones.
Generame un diagrama Mermaid de este sistema
Una arquitectura de microservicios para un sistema Sistema Inteligente de Validación Documental (SIVD) como Expedia. Cada microservicio tiene su propia base de datos y una interfaz de usuario que se comunica mediante API. El proveedor de nube es AWS; utilice los servicios adecuados, incluyendo balanceo de carga y CDN.
Generar el diagrama de arquitectura con DiagramGPT

## 1. Funcionalidades Básicas del SIVD (Ordenadas por Prioridad)

1. **Extracción automática de datos (OCR/IDP):** Captura de información de documentos de identidad, comprobantes de ingresos y otros documentos requeridos
2. **Motor de comparación inteligente:** Análisis y cotejo de datos entre formularios digitales y documentos cargados
3. **Detección de inconsistencias en tiempo real:** Identificación automática de discrepancias en nombres, direcciones, fechas, montos, etc.
4. **Sistema de reglas de validación:** Motor configurable para establecer criterios de aceptación y umbrales de similitud
5. **Gestión de documentos:** Almacenamiento seguro, versionamiento y trazabilidad de documentos cargados
6. **Dashboard de analista:** Interface para revisión manual de casos con inconsistencias detectadas
7. **Generación de reportes:** Informes de validación con evidencias y nivel de confianza
8. **Sistema de alertas:** Notificaciones automáticas sobre inconsistencias críticas
9. **Integración con sistemas bancarios:** Conexión con core bancario y sistemas de onboarding existentes
10. **Auditología y trazabilidad:** Registro completo de todas las validaciones y decisiones tomadas

## 2. Beneficios para el Cliente

### Para el Banco:

- Reducción de tiempos de onboarding de días a minutos
- Disminución de errores humanos en validación documental
- Menor carga operativa en equipos de back-office
- Cumplimiento normativo automatizado (KYC, AML)
- Reducción de costos operativos hasta 70%
- Mejora en la experiencia del cliente
- Trazabilidad completa para auditorías
- Detección temprana de posibles fraudes

### Para el Cliente Final:

- Proceso de apertura de cuenta más rápido
- Menor cantidad de visitas a sucursal
- Respuesta inmediata sobre el estado de su solicitud
- Menos documentación rechazada por errores menores

## 3. Customer Journey del Cliente

### Fase 1: Inicio del Proceso

1. Cliente accede a la plataforma de onboarding del banco (web/móvil)
2. Completa registro inicial con email y número de teléfono
3. Recibe verificación OTP

**Fase 2: Llenado de Formulario Digital** 4. Completa formulario con datos personales (nombre, identificación, dirección, ingresos, ocupación) 5. Acepta términos y condiciones 6. El sistema valida formato de datos en tiempo real

**Fase 3: Carga de Documentos** 7. Recibe instrucciones sobre documentos requeridos 8. Captura/sube documento de identidad (cédula, pasaporte) 9. Captura/sube comprobante de domicilio 10. Captura/sube comprobante de ingresos 11. El sistema confirma recepción y calidad de cada documento

**Fase 4: Validación Automática** 12. El SIVD procesa documentos con OCR/IDP 13. El sistema compara datos extraídos vs. formulario 14. Cliente ve indicador de "procesando documentación"

**Fase 5: Resultado** 15. **Escenario A (Sin inconsistencias):** Cliente recibe aprobación inmediata y puede continuar con apertura de cuenta 16. **Escenario B (Inconsistencias menores):** Cliente recibe notificación de correcciones necesarias con indicaciones específicas 17. **Escenario C (Inconsistencias mayores):** Solicitud pasa a revisión manual, cliente recibe tiempo estimado de respuesta

**Fase 6: Seguimiento** 18. Cliente puede consultar estado de su solicitud en tiempo real 19. Recibe notificaciones por email/SMS de avances 20. Una vez aprobado, procede con activación de cuenta

#### **4. Casos de Uso Principales para MVP**

##### **CU-01: Registrar Cliente**

- Actor: Cliente
- Descripción: El cliente crea su cuenta en el sistema proporcionando datos básicos de contacto

##### **CU-02: Completar Formulario Digital**

- Actor: Cliente
- Descripción: El cliente ingresa su información personal, laboral y financiera en el formulario estructurado

##### **CU-03: Cargar Documentos**

- Actor: Cliente
- Descripción: El cliente sube fotografías o archivos de sus documentos de identidad, domicilio e ingresos

##### **CU-04: Extraer Datos de Documentos (OCR)**

- Actor: Sistema
- Descripción: El sistema procesa los documentos cargados y extrae información estructurada mediante OCR/IDP

##### **CU-05: Validar Inconsistencias**

- Actor: Sistema
- Descripción: El motor de comparación coteja los datos del formulario vs. los extraídos de documentos y detecta discrepancias

##### **CU-06: Aplicar Reglas de Negocio**

- Actor: Sistema
- Descripción: El sistema evalúa las inconsistencias encontradas según reglas predefinidas y umbrales de aceptación

##### **CU-07: Generar Reporte de Validación**

- Actor: Sistema
- Descripción: El sistema crea un reporte con resultados, inconsistencias detectadas y nivel de confianza

##### **CU-08: Revisar Casos Manualmente**

- Actor: Analista
- Descripción: El analista revisa solicitudes con inconsistencias que requieren validación humana

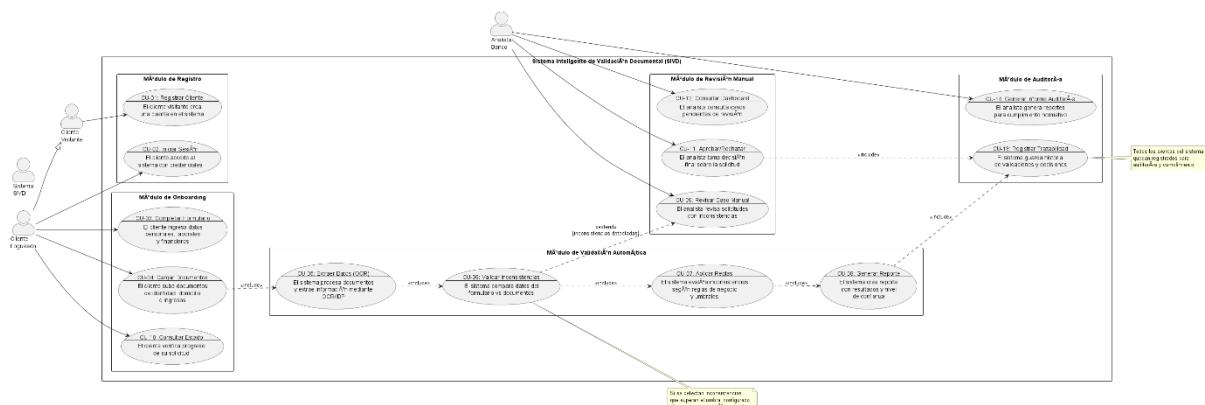
##### **CU-09: Aprobar/Rechazar Solicitud**

- Actor: Analista
- Descripción: El analista toma decisión final sobre casos en revisión manual

##### **CU-10: Consultar Estado de Solicitud**

- Actor: Cliente
- Descripción: El cliente verifica el progreso de su solicitud de onboarding

#### 4. Diagrama de Casos de Uso (PlantUML)



<pre> @startuml !define RECTANGLE class  skinparam packageStyle rectangle skinparam actorStyle awesome  left to right direction  actor "Cliente\nVisitante" as ClienteVisitante actor "Cliente\nLogueado" as ClienteLogueado actor "Sistema\nSIVD" as Sistema actor "Analista\nBanco" as Analista  rectangle "Sistema Inteligente de Validación Documental (SIVD)" {      package "Módulo de Registro" {         usecase UC01 as "CU-01: Registrar Cliente"         --         El cliente visitante crea         una cuenta en el sistema"     }      usecase UC02 as "CU-02: Iniciar Sesión"     --     El cliente accede al     sistema con credenciales" }  package "Módulo de Onboarding" {     usecase UC03 as "CU-03: Completar Formulario"     --     El cliente ingresa datos     personales, laborales     y financieros"      usecase UC04 as "CU-04: Cargar Documentos"     --     El cliente sube documentos     de identidad, domicilio     e ingresos"      usecase UC10 as "CU-10: Consultar Estado"     --     El cliente verifica progreso     de su solicitud" }  package "Módulo de Validación Automática" {     usecase UC05 as "CU-05: Extraer Datos (OCR)"     --     El sistema procesa documentos     y extrae información mediante     OCR/IDP"      usecase UC06 as "CU-06: Validar Inconsistencias" } </pre>	<p>usecase UC09 as "CU-09: Revisar Caso Manual -- El analista revisa solicitudes con inconsistencias"</p> <p>usecase UC11 as "CU-11: Aprobar/Rechazar -- El analista toma decisión final sobre la solicitud"</p> <p>usecase UC12 as "CU-12: Consultar Dashboard -- El analista consulta casos pendientes de revisión" }</p> <p>package "Módulo de Auditoría" {     usecase UC13 as "CU-13: Registrar Trazabilidad -- El sistema guarda historial de validaciones y decisiones"</p> <p>usecase UC14 as "CU-14: Generar Informe Auditoría -- El analista genera reportes para cumplimiento normativo" }</p> <p>' Relaciones Cliente Visitante ClienteVisitante --&gt; UC01</p> <p>' Relaciones Cliente Logueado ClienteLogueado --&gt; UC02 ClienteLogueado --&gt; UC03 ClienteLogueado --&gt; UC04 ClienteLogueado --&gt; UC10</p> <p>' Relaciones Sistema UC04 ..&gt; UC05 : &lt;&lt;include&gt;&gt; UC05 ..&gt; UC06 : &lt;&lt;include&gt;&gt; UC06 ..&gt; UC07 : &lt;&lt;include&gt;&gt; UC07 ..&gt; UC08 : &lt;&lt;include&gt;&gt; UC08 ..&gt; UC13 : &lt;&lt;include&gt;&gt;</p> <p>' Relaciones Analista Analista --&gt; UC09 Analista --&gt; UC11 Analista --&gt; UC12 Analista --&gt; UC14</p> <p>' Extensiones y dependencias UC06 ..&gt; UC09 : &lt;&lt;extend&gt;&gt;\n[inconsistencias detectadas] UC11 ..&gt; UC13 : &lt;&lt;include&gt;&gt;</p>
---	--

<pre>-- El sistema compara datos del formulario vs documentos" usecase UC07 as "CU-07: Aplicar Reglas -- El sistema evalúa inconsistencias según reglas de negocio y umbrales" usecase UC08 as "CU-08: Generar Reporte -- El sistema crea reporte con resultados y nivel de confianza" }  package "Módulo de Revisión Manual" {</pre>	<p>' Herencia ClienteLogueado --&gt; ClienteVisitante</p> <p>note right of UC06 Si se detectan inconsistencias que superan el umbral configurado, el caso pasa a revisión manual end note</p> <p>note right of UC13 Todos los eventos del sistema quedan registrados para auditoría y cumplimiento end note</p> <p>@enduml</p>
---	--

## 6. Tres Entidades Esenciales del Modelo de Datos

### Entidad 1: Solicitud de Onboarding

#### Campos esenciales:

- id\_solicitud (PK)
- id\_cliente (FK)
- fecha\_creacion
- estado (pendiente, en\_validacion, aprobada, rechazada)
- datos\_formulario (JSON con información ingresada)
- score\_validacion
- observaciones

### Entidad 2: Documento

#### Campos esenciales:

- id\_documento (PK)
- id\_solicitud (FK)
- tipo\_documento (cedula, pasaporte, comprobante\_domicilio, comprobante\_ingresos)
- archivo\_url
- datos\_extraidos (JSON con información del OCR)
- fecha\_carga
- estado\_procesamiento
- nivel\_confianza\_ocr

### Entidad 3: Inconsistencia

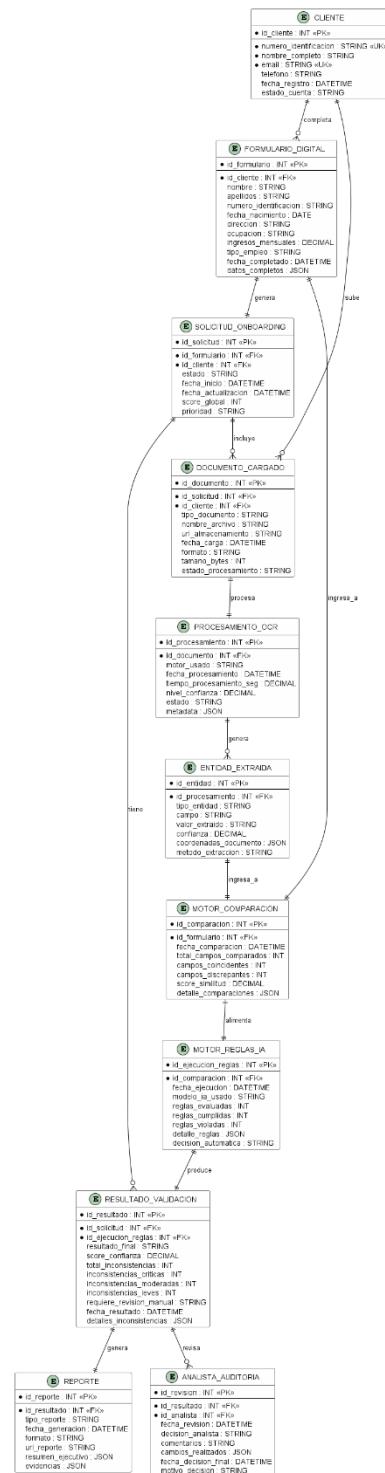
#### Campos esenciales:

- id\_inconsistencia (PK)
- id\_solicitud (FK)
- id\_documento (FK)
- campo\_afectado
- valor\_formulario
- valor\_documento
- tipo\_inconsistencia (diferencia\_texto, diferencia\_fecha, diferencia\_numerica)
- severidad (baja, media, alta, critica)
- estado\_revision (pendiente, resuelta, ignorada)
- resolucion\_analista

## Relaciones:

- Una Solicitud tiene muchos Documentos (1:N)
- Una Solicitud tiene muchas Inconsistencias (1:N)
- Un Documento puede tener muchas Inconsistencias (1:N)

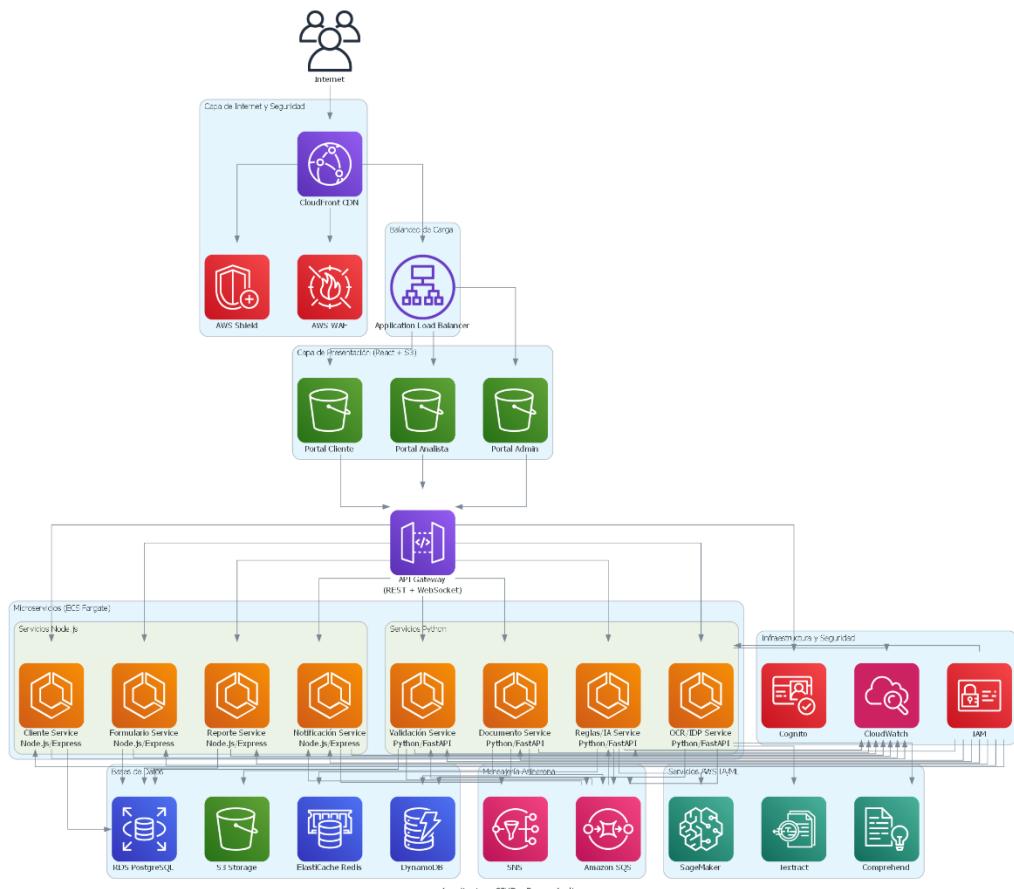
## 7. Diagrama Mermaid del Sistema



<pre> erDiagram     CLIENTE   --o{ FORMULARIO_DIGITAL : completa     CLIENTE   --o{ DOCUMENTO_CARGADO : sube     FORMULARIO_DIGITAL   --      SOLICITUD_ONBOARDING : genera     SOLICITUD_ONBOARDING   --o{     DOCUMENTO_CARGADO : incluye     DOCUMENTO_CARGADO   --      PROCESAMIENTO_OCR : procesa     PROCESAMIENTO_OCR   --o{ ENTIDAD_EXTRAIDA :     genera     FORMULARIO_DIGITAL   --   MOTOR_COMPARACION     : ingresa_a     ENTIDAD_EXTRAIDA   --   MOTOR_COMPARACION :     ingresa_a     MOTOR_COMPARACION   --   MOTOR_REGLAS_IA :     alimenta     MOTOR_REGLAS_IA   --   RESULTADO_VALIDACION :     produce     RESULTADO_VALIDACION   --   REPORTE : genera     RESULTADO_VALIDACION   --o{     ANALISTA_AUDITORIA : revisa     SOLICITUD_ONBOARDING   --o{     RESULTADO_VALIDACION : tiene      CLIENTE {         int id_cliente PK         string numero_identificacion UK         string nombre_completo         string email UK         string telefono         datetime fecha_registro         string estado_cuenta     }      FORMULARIO_DIGITAL {         int id_formulario PK         int id_cliente FK         string nombre         string apellidos         string numero_identificacion         date fecha_nacimiento         string direccion         string ocupacion         decimal ingresos_mensuales         string tipo_empleo         datetime fecha_completado         json datos_completos     }      SOLICITUD_ONBOARDING {         int id_solicitud PK         int id_formulario FK         int id_cliente FK         string estado         datetime fecha_inicio         datetime fecha_actualizacion         int score_global         string prioridad     }      DOCUMENTO_CARGADO {         int id_documento PK         int id_solicitud FK         int id_cliente FK         string tipo_documento         string nombre_archivo         string url_almacenamiento         datetime fecha_carga         string formato         int tamano_bytes         string estado_procesamiento     } } </pre>	<pre> PROCESAMIENTO_OCR {     int id_procesamiento PK     int id_documento FK     string motor_usado     datetime fecha_procesamiento     decimal tiempo_procesamiento_seg     decimal nivel_confianza     string estado     json metadata }  ENTIDAD_EXTRAIDA {     int id_entidad PK     int id_procesamiento FK     string tipo_entidad     string campo     string valor_extraido     decimal confianza     json coordenadas_documento     string metodo_extraccion }  MOTOR_COMPARACION {     int id_comparacion PK     int id_formulario FK     datetime fecha_comparacion     int total_campos_comparados     int campos_coincidentes     int campos_discrepantes     decimal score_similitud     json detalle_comparaciones }  MOTOR_REGLAS_IA {     int id_ejecucion_reglas PK     int id_comparacion FK     datetime fecha_ejecucion     string modelo_ia_usado     int reglas_evaluadas     int reglas_cumplidas     int reglas_violadas     json detalle_reglas     string decisionAutomatica }  RESULTADO_VALIDACION {     int id_resultado PK     int id_solicitud FK     int id_ejecucion_reglas FK     string resultado_final     decimal score_confianza     int total_inconsistencias     int inconsistencias_criticas     int inconsistencias_moderadas     int inconsistencias_leves     string requiere_revision_manual     datetime fecha_resultado     json detalles_inconsistencias }  REPORTE {     int id_reporte PK     int id_resultado FK     string tipo_reporte     datetime fecha_generacion     string formato     string url_reporte     json resumen_ejecutivo     json evidencias }  ANALISTA_AUDITORIA {     int id_revision PK     int id_resultado FK     int id_analista FK } </pre>
--	--

	<pre>         datetime fecha_revision         string decision_analista         string comentarios         json cambios_realizados         datetime fecha_decision_final         string motivo_decision     } </pre>
--	---

## 8. Arquitectura de Microservicios en AWS



<pre> import React from 'react';  import { Cloud, Database, Shield, Zap, FileText, CheckCircle, AlertCircle, Users, BarChart3, Lock, Globe } from 'lucide-react';  export default function SIVDArchitecture() {     return ( </pre>	<pre> /* API Gateway */  &lt;div className="bg-gradient-to-r from-yellow-500/20 to-orange-500/20 rounded-lg p-4 mb-6 border border-yellow-400/30"&gt;      &lt;div className="flex items-center justify-center gap-3"&gt;          &lt;Cloud className="text-yellow-300" size={24}&gt;     /&gt; </pre>
---	---

<pre> &lt;div className="w-full h-full bg-gradient-to-br from-state-900 via-blue-900 to-slate-900 p-8 overflow-auto"&gt;    &lt;div className="max-w-7xl mx-auto"&gt;      {/* Header */}      &lt;div className="text-center mb-8"&gt;        &lt;h1 className="text-4xl font-bold text-white mb-2"&gt;         Arquitectura SIVD - Banco Andino       &lt;/h1&gt;        &lt;p className="text-blue-200 text-lg"&gt;         Sistema Inteligente de Validación Documental en AWS       &lt;/p&gt;     &lt;/div&gt;      {/* Internet Layer */}      &lt;div className="bg-gradient-to-r from-purple-500/20 to-pink-500/20 rounded-lg p-6 mb-6 border border-purple-400/30"&gt;        &lt;div className="flex items-center gap-3 mb-4"&gt;         &lt;Globe className="text-purple-300" size={28}&gt;           &lt;h2 className="text-2xl font-bold text-white"&gt;Capa de Internet&lt;/h2&gt;         &lt;/div&gt;         &lt;div className="flex gap-4 justify-center"&gt;           &lt;div className="bg-purple-600/30 p-4 rounded-lg border border-purple-400/50 text-center"&gt;             &lt;Shield className="text-purple-300 mx-auto mb-2" size={32}&gt;               &lt;p className="text-white font-semibold"&gt;AWS WAF&lt;/p&gt;             &lt;Shield className="text-purple-300 mx-auto mb-2" size={32}&gt;               &lt;p className="text-purple-200 text-sm"&gt;Firewall de Aplicaciones&lt;/p&gt;             &lt;/div&gt;           &lt;/div&gt;         &lt;/div&gt;       &lt;/div&gt;     &lt;/div&gt;   &lt;/div&gt; </pre>	<pre> &lt;h3 className="text-xl font-bold text-white"&gt;Amazon API Gateway&lt;/h3&gt;  &lt;p className="text-yellow-200 text-sm"&gt;(REST + WebSocket APIs)&lt;/p&gt;  &lt;/div&gt; &lt;/div&gt;  {/* Microservices Layer */}  &lt;div className="bg-gradient-to-r from-indigo-500/20 to-blue-500/20 rounded-lg p-6 mb-6 border border-indigo-400/30"&gt;    &lt;div className="flex items-center gap-3 mb-4"&gt;     &lt;Cloud className="text-indigo-300" size={28}&gt;       &lt;h2 className="text-2xl font-bold text-white"&gt;Capa de Microservicios (ECS Fargate)&lt;/h2&gt;     &lt;/div&gt;   &lt;/div&gt;    &lt;div className="grid grid-cols-2 lg:grid-cols-4 gap-4"&gt;     {/* Cliente Service */}     &lt;div className="bg-indigo-600/30 p-4 rounded-lg border border-indigo-400/50"&gt;       &lt;Users className="text-indigo-300 mb-2" size={28}&gt;         &lt;p className="text-white font-semibold mb-1"&gt;Cliente Service&lt;/p&gt;         &lt;p className="text-indigo-200 text-xs mb-2"&gt;Node.js/Express&lt;/p&gt;         &lt;div className="bg-indigo-700/40 p-2 rounded mt-2"&gt;           &lt;Database className="text-indigo-300 inline mr-1" size={16}&gt;             &lt;span className="text-indigo-200 text-xs"&gt;RDS PostgreSQL&lt;/span&gt;           &lt;/div&gt;         &lt;/div&gt;       &lt;/div&gt;     &lt;/div&gt;   &lt;/div&gt; </pre>
--	--

<pre> &lt;div className="bg-purple-600/30 p-4 rounded-lg border border-purple-400/50 text-center"&gt;      &lt;Zap className="text-purple-300 mx-auto mb-2" size={32} /&gt;      &lt;p className="text-white font- semibold"&gt;Amazon CloudFront&lt;/p&gt;      &lt;p className="text-purple-200 text- sm"&gt;CDN Global&lt;/p&gt;  &lt;/div&gt;  &lt;div className="bg-purple-600/30 p-4 rounded-lg border border-purple-400/50 text-center"&gt;      &lt;Lock className="text-purple-300 mx-auto mb-2" size={32} /&gt;      &lt;p className="text-white font- semibold"&gt;AWS Shield&lt;/p&gt;      &lt;p className="text-purple-200 text- sm"&gt;Protección DDoS&lt;/p&gt;  &lt;/div&gt;  &lt;/div&gt;  &lt;/div&gt;  /* Load Balancer Layer */  &lt;div className="bg-gradient-to-r from-blue- 500/20 to-cyan-500/20 rounded-lg p-6 mb-6 border border-blue-400/30"&gt;      &lt;div className="flex items-center gap-3 mb- 4"&gt;          &lt;Zap className="text-blue-300" size={28} /&gt;          &lt;h2 className="text-2xl font-bold text- white"&gt;Balanceo de Carga&lt;/h2&gt;      &lt;/div&gt;      &lt;div className="flex gap-4 justify-center"&gt;          &lt;div className="bg-blue-600/30 p-4 rounded- lg border border-blue-400/50 text-center"&gt;              &lt;div className="text-white font-semibold text-lg mb-2"&gt;Application Load Balancer&lt;/div&gt;              &lt;p className="text-blue-200 text- sm"&gt;Distribución de tráfico entre microservicios&lt;/p&gt;          &lt;/div&gt;     </pre>	<pre> /* Formulario Service */  &lt;div className="bg-indigo-600/30 p-4 rounded-lg border border-indigo-400/50"&gt;      &lt;FileText className="text-indigo-300 mb-2" size={28} /&gt;      &lt;p className="text-white font-semibold mb- 1"&gt;Formulario Service&lt;/p&gt;      &lt;p className="text-indigo-200 text-xs mb- 2"&gt;Node.js/Express&lt;/p&gt;      &lt;div className="bg-indigo-700/40 p-2 rounded mt-2"&gt;          &lt;Database className="text-indigo-300 inline mr-1" size={16} /&gt;          &lt;span className="text-indigo-200 text- xs"&gt;RDS PostgreSQL&lt;/span&gt;      &lt;/div&gt;  &lt;/div&gt;  /* Documento Service */  &lt;div className="bg-indigo-600/30 p-4 rounded-lg border border-indigo-400/50"&gt;      &lt;FileText className="text-indigo-300 mb-2" size={28} /&gt;      &lt;p className="text-white font-semibold mb- 1"&gt;Documento Service&lt;/p&gt;      &lt;p className="text-indigo-200 text-xs mb- 2"&gt;Python/FastAPI&lt;/p&gt;      &lt;div className="bg-indigo-700/40 p-2 rounded mt-2"&gt;          &lt;Database className="text-indigo-300 inline mr-1" size={16} /&gt;          &lt;span className="text-indigo-200 text- xs"&gt;DynamoDB&lt;/span&gt;      &lt;/div&gt;      &lt;div className="bg-indigo-700/40 p-2 rounded mt-2"&gt;          &lt;Cloud className="text-indigo-300 inline mr-1" size={16} /&gt;          &lt;span className="text-indigo-200 text- xs"&gt;S3 (archivos)&lt;/span&gt;      &lt;/div&gt; </pre>
--	---

<pre>         &lt;/div&gt;         &lt;/div&gt;          /* Frontend Layer */          &lt;div className="bg-gradient-to-r from-green-500/20 to-emerald-500/20 rounded-lg p-6 mb-6 border border-green-400/30"&gt;           &lt;div className="flex items-center gap-3 mb-4"&gt;             &lt;Globe className="text-green-300" size={28} /&gt;             &lt;h2 className="text-2xl font-bold text-white"&gt;Capa de Presentación&lt;/h2&gt;           &lt;/div&gt;           &lt;div className="grid grid-cols-3 gap-4"&gt;             &lt;div className="bg-green-600/30 p-4 rounded-lg border border-green-400/50"&gt;               &lt;Users className="text-green-300 mx-auto mb-2" size={32} /&gt;               &lt;p className="text-white font-semibold text-center"&gt;Portal Cliente&lt;/p&gt;               &lt;p className="text-green-200 text-sm text-center mt-2"&gt;React + S3&lt;/p&gt;               &lt;p className="text-green-300 text-xs text-center mt-1"&gt;Onboarding digital&lt;/p&gt;             &lt;/div&gt;             &lt;div className="bg-green-600/30 p-4 rounded-lg border border-green-400/50"&gt;               &lt;BarChart3 className="text-green-300 mx-auto mb-2" size={32} /&gt;               &lt;p className="text-white font-semibold text-center"&gt;Portal Analista&lt;/p&gt;               &lt;p className="text-green-200 text-sm text-center mt-2"&gt;React + S3&lt;/p&gt;               &lt;p className="text-green-300 text-xs text-center mt-1"&gt;Revisión y gestión&lt;/p&gt;             &lt;/div&gt;             &lt;div className="bg-green-600/30 p-4 rounded-lg border border-green-400/50"&gt; </pre>	<pre>         &lt;/div&gt;         &lt;/div&gt;          /* OCR/IDP Service */          &lt;div className="bg-indigo-600/30 p-4 rounded-lg border border-indigo-400/50"&gt;           &lt;AlertCircle className="text-indigo-300 mb-2" size={28} /&gt;           &lt;p className="text-white font-semibold mb-1"&gt;OCR/IDP Service&lt;/p&gt;           &lt;p className="text-indigo-200 text-xs mb-2"&gt;Python/FastAPI&lt;/p&gt;           &lt;div className="bg-indigo-700/40 p-2 rounded mt-2"&gt;             &lt;Cloud className="text-indigo-300 inline mr-1" size={16} /&gt;             &lt;span className="text-indigo-200 text-xs"&gt;Textract + Comprehend&lt;/span&gt;           &lt;/div&gt;           &lt;div className="bg-indigo-700/40 p-2 rounded mt-2"&gt;             &lt;Database className="text-indigo-300 inline mr-1" size={16} /&gt;             &lt;span className="text-indigo-200 text-xs"&gt;DynamoDB&lt;/span&gt;           &lt;/div&gt;           &lt;div className="bg-indigo-600/30 p-4 rounded-lg border border-indigo-400/50"&gt;             /* Validación Service */             &lt;div className="bg-indigo-600/30 p-4 rounded-lg border border-indigo-400/50"&gt;               &lt;CheckCircle className="text-indigo-300 mb-2" size={28} /&gt;               &lt;p className="text-white font-semibold mb-1"&gt;Validación Service&lt;/p&gt;               &lt;p className="text-indigo-200 text-xs mb-2"&gt;Python/FastAPI&lt;/p&gt;               &lt;div className="bg-indigo-700/40 p-2 rounded mt-2"&gt; </pre>
--	--

<pre> &lt;Shield className="text-green-300 mx-auto mb-2" size={32} /&gt;  &lt;p className="text-white font-semibold text-center"&gt;Portal Admin&lt;/p&gt;  &lt;p className="text-green-200 text-sm text-center mt-2"&gt;React + S3&lt;/p&gt;  &lt;p className="text-green-300 text-xs text-center mt-1"&gt;Configuración&lt;/p&gt;  &lt;/div&gt; &lt;/div&gt; &lt;/div&gt; </pre>	<pre> &lt;Database className="text-indigo-300 inline mr-1" size={16} /&gt;  &lt;span className="text-indigo-200 text-xs"&gt;RDS PostgreSQL&lt;/span&gt;  &lt;/div&gt; &lt;/div&gt;  /* Reglas/IA Service */  &lt;div className="bg-indigo-600/30 p-4 rounded-lg border border-indigo-400/50"&gt;  &lt;Zap className="text-indigo-300 mb-2" size={28} /&gt;  &lt;p className="text-white font-semibold mb-1"&gt;Reglas/IA Service&lt;/p&gt;  &lt;p className="text-indigo-200 text-xs mb-2"&gt;Python/FastAPI&lt;/p&gt;  &lt;div className="bg-indigo-700/40 p-2 rounded mt-2"&gt;  &lt;Cloud className="text-indigo-300 inline mr-1" size={16} /&gt;  &lt;span className="text-indigo-200 text-xs"&gt;SageMaker&lt;/span&gt;  &lt;/div&gt;  &lt;div className="bg-indigo-700/40 p-2 rounded mt-2"&gt;  &lt;Database className="text-indigo-300 inline mr-1" size={16} /&gt;  &lt;span className="text-indigo-200 text-xs"&gt;ElastiCache Redis&lt;/span&gt;  &lt;/div&gt; &lt;/div&gt;  /* Reporte Service */  &lt;div className="bg-indigo-600/30 p-4 rounded-lg border border-indigo-400/50"&gt;  &lt;BarChart3 className="text-indigo-300 mb-2" size={28} /&gt;  &lt;p className="text-white font-semibold mb-1"&gt;Reporte Service&lt;/p&gt; </pre>
--	---

```
<p className="text-indigo-200 text-xs mb-2">Node.js/Express</p>

<div className="bg-indigo-700/40 p-2 rounded mt-2">

    <Database className="text-indigo-300 inline mr-1" size={16} />

    <span className="text-indigo-200 text-xs">RDS PostgreSQL</span>

</div>

<div className="bg-indigo-700/40 p-2 rounded mt-2">

    <Cloud className="text-indigo-300 inline mr-1" size={16} />

    <span className="text-indigo-200 text-xs">QuickSight</span>

</div>

</div>

/* Notificación Service */

<div className="bg-indigo-600/30 p-4 rounded-lg border border-indigo-400/50">

    <AlertCircle className="text-indigo-300 mb-2" size={28} />

    <p className="text-white font-semibold mb-1">Notificación Service</p>

    <p className="text-indigo-200 text-xs mb-2">Node.js/Express</p>

    <div className="bg-indigo-700/40 p-2 rounded mt-2">

        <Cloud className="text-indigo-300 inline mr-1" size={16} />

        <span className="text-indigo-200 text-xs">SNS + SES</span>

    </div>

</div>

</div>

</div>
```

```
/* Messaging Layer */

<div className="bg-gradient-to-r from-pink-500/20 to-rose-500/20 rounded-lg p-4 mb-6 border border-pink-400/30">

  <div className="flex items-center justify-center gap-3">

    <Zap className="text-pink-300" size={24} />

    <h3 className="text-xl font-bold text-white">Amazon SQS / EventBridge</h3>

    <p className="text-pink-200 text-sm">(Comunicación asíncrona entre servicios)</p>

  </div>

</div>

</div>

/* Infrastructure Services */

<div className="bg-gradient-to-r from-slate-500/20 to-gray-500/20 rounded-lg p-6 border border-slate-400/30">

  <div className="flex items-center gap-3 mb-4">

    <Shield className="text-slate-300" size={28} />

    <h2 className="text-2xl font-bold text-white">Servicios de Infraestructura</h2>

  </div>

</div>

<div className="grid grid-cols-2 lg:grid-cols-4 gap-4">

  <div className="bg-slate-600/30 p-3 rounded-lg border border-slate-400/50 text-center">

    <Lock className="text-slate-300 mx-auto mb-1" size={24} />

    <p className="text-white font-semibold text-sm">AWS Cognito</p>

    <p className="text-slate-300 text-xs">Autenticación</p>

  </div>

  <div className="bg-slate-600/30 p-3 rounded-lg border border-slate-400/50 text-center">
```

```
<Shield className="text-slate-300 mx-auto mb-1" size={24} />

<p className="text-white font-semibold text-sm">AWS Secrets Manager</p>

<p className="text-slate-300 text-xs">Gestión secretos</p>

</div>

<div className="bg-slate-600/30 p-3 rounded-lg border border-slate-400/50 text-center">

  <BarChart3 className="text-slate-300 mx-auto mb-1" size={24} />

  <p className="text-white font-semibold text-sm">CloudWatch</p>

  <p className="text-slate-300 text-xs">Monitoreo/Logs</p>

  </div>

  <div className="bg-slate-600/30 p-3 rounded-lg border border-slate-400/50 text-center">

    <AlertCircle className="text-slate-300 mx-auto mb-1" size={24} />

    <p className="text-white font-semibold text-sm">X-Ray</p>

    <p className="text-slate-300 text-xs">Trazabilidad</p>

    </div>

    </div>

    </div>

  /* Architecture Notes */

  <div className="mt-6 bg-blue-500/10 border border-blue-400/30 rounded-lg p-4">

    <h3 className="text-white font-bold mb-2 flex items-center gap-2">

      <AlertCircle className="text-blue-300" size={20} />

      Características Clave de la Arquitectura

    </h3>

    <ul className="text-blue-200 text-sm space-y-1">
```

	<ul style="list-style-type: none"> <li>&lt;li&gt;• &lt;strong&gt;Alta Disponibilidad:&lt;/strong&gt; Multi-AZ deployment con Auto Scaling&lt;/li&gt;</li> <li>&lt;li&gt;• &lt;strong&gt;Escalabilidad:&lt;/strong&gt; ECS Fargate con escalado automático basado en métricas&lt;/li&gt;</li> <li>&lt;li&gt;• &lt;strong&gt;Seguridad:&lt;/strong&gt; VPC aislada, Cognito, Secrets Manager, WAF + Shield&lt;/li&gt;</li> <li>&lt;li&gt;• &lt;strong&gt;Rendimiento:&lt;/strong&gt; CloudFront CDN, ElastiCache Redis, balanceo de carga&lt;/li&gt;</li> <li>&lt;li&gt;• &lt;strong&gt;Observabilidad:&lt;/strong&gt; CloudWatch + X-Ray para monitoreo completo&lt;/li&gt;</li> <li>&lt;li&gt;• &lt;strong&gt;Desacoplamiento:&lt;/strong&gt; SQS/EventBridge para comunicación asíncrona&lt;/li&gt;</li> </ul> <p>&lt;/ul&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;/div&gt;</p> <p>);</p> <p>}</p>
--	---

## Descripción de la Arquitectura de Microservicios

### Componentes Principales:

#### 1. Frontend Layer:

- Aplicaciones web (React/Angular) y móviles servidas a través de CloudFront CDN
- Route 53 para gestión de DNS

#### 2. Security & Gateway:

- AWS WAF protege contra ataques comunes
- Application Load Balancer distribuye tráfico
- API Gateway gestiona todas las APIs REST
- Cognito maneja autenticación y autorización

#### 3. Microservicios (ECS Fargate & Lambda):

- **Client Service:** Gestión de clientes y perfiles
- **Onboarding Service:** Orquestación del proceso de onboarding
- **Document Service:** Almacenamiento y gestión de documentos
- **OCR Service:** Procesamiento con Textract y Rekognition
- **Validation Engine:** Motor de comparación y detección de inconsistencias
- **AI Rules Engine:** Evaluación inteligente con SageMaker
- **Report Service:** Generación de reportes
- **Notification Service:** Envío de alertas (SNS/SES)
- **Audit Service:** Trazabilidad completa

#### **4. Bases de Datos:**

- RDS PostgreSQL para datos transaccionales
- DynamoDB para reglas de validación y auditoría
- S3 para almacenamiento de documentos
- ElastiCache Redis para caché distribuido

#### **5. Procesamiento Asíncrono:**

1. SQS para colas de mensajes
2. EventBridge como bus de eventos

#### **6. Monitoreo & Analytics:**

- CloudWatch para logs y métricas
- X-Ray para trazabilidad distribuida
- S3 Data Lake + Athena + QuickSight para analytics

Esta arquitectura garantiza escalabilidad, alta disponibilidad, seguridad y cumplimiento normativo para el SIVD del Banco Andino.

#### **Parte II - Actividad 3: Trabajando como Tech Lead con GenAI**

Prompt	
Ten en cuenta el siguiente documento de especificación de requerimientos. Responde: "entiendo el documento", pero no lo expliques.  Actuando como un Analista de Software, construye un listado de las principales User Stories para completar el MVP de que haga detección de inconsistencias entre formularios digitales y documentos cargados por clientes durante el proceso de onboarding de personas naturales del Sistema Inteligente de Validación Documental (SIVD)  Actuando como un Product Owner, describeme la User Story US-001: Iniciar proceso de onboarding de forma detallada.	Actuando como un Software Architect y un Tech Lead, genérame los tickets de trabajo (tasks) de Jira para realizar la user story US-001: Iniciar proceso de onboarding. Quiero que los tickets tengan:  - ID del Ticket: - Título del Ticket: - Descripción: - Criterios de aceptación: - Prioridad: - Estimación de esfuerzo (en horas): - Tareas Técnicas: - Notas

#### **3. User Stories MVP - Sistema Inteligente de Validación Documental (SIVD)**

##### **Banco Andino - Onboarding Personas Naturales**

---

##### **ÉPICA 1: REGISTRO Y FORMULARIO DIGITAL**

##### **US-001: Iniciar proceso de onboarding**

**Como** cliente nuevo del Banco Andino  
**Quiero** acceder al portal de onboarding digital  
**Para** iniciar mi proceso de apertura de cuenta sin ir a una sucursal

##### **Criterios de Aceptación:**

- El cliente puede acceder al portal desde web y móvil
- Se muestra información clara sobre requisitos y documentos necesarios
- El cliente puede crear una cuenta con email y contraseña
- Se envía un email de verificación
- El sistema registra la fecha/hora de inicio del proceso

**Prioridad:** ALTA  
**Estimación:** 5 puntos  
**Dependencias:** Ninguna

---

#### **US-002: Completar datos personales en formulario**

**Como** cliente  
**Quiero** ingresar mis datos personales básicos en el formulario digital  
**Para** proporcionar la información requerida para mi cuenta

**Criterios de Aceptación:**

- Formulario incluye campos: tipo de identificación, número, nombres, apellidos, fecha de nacimiento, nacionalidad
- Validación en tiempo real de formato de campos (sin caracteres especiales en nombres, formato de cédula)
- El sistema detecta si el cliente ya existe en la base de datos
- Se puede guardar progreso y continuar después
- Todos los campos obligatorios están marcados claramente

**Prioridad:** ALTA  
**Estimación:** 8 puntos  
**Dependencias:** US-001

---

#### **US-003: Completar datos de contacto y residencia**

**Como** cliente  
**Quiero** ingresar mi información de contacto y dirección  
**Para** que el banco pueda comunicarse conmigo y verificar mi residencia

**Criterios de Aceptación:**

- Formulario incluye: dirección completa, ciudad, código postal, teléfono, email
- Validación de formato de teléfono y email
- Autocompletado de dirección con sugerencias
- Campos separados para tipo de vivienda (propia, arrendada, familiar)
- Tiempo de residencia en dirección actual

**Prioridad:** ALTA  
**Estimación:** 5 puntos  
**Dependencias:** US-002

---

#### **US-004: Completar información financiera**

**Como** cliente  
**Quiero** declarar mi información financiera y laboral  
**Para** que el banco evalúe mi perfil económico

**Criterios de Aceptación:**

- Campos para: ocupación, tipo de contrato, nombre empresa, ingresos mensuales, otros ingresos
- Validación de rangos razonables de ingresos
- Lista desplegable de profesiones/ocupaciones
- Campo para origen de fondos
- Declaración de persona expuesta políticamente (PEP)

**Prioridad:** ALTA  
**Estimación:** 8 puntos  
**Dependencias:** US-003

---

#### **US-005: Revisar y confirmar información del formulario**

**Como** cliente  
**Quiero** revisar todos los datos ingresados antes de enviar  
**Para** asegurarme que la información es correcta

##### **Criterios de Aceptación:**

- Vista resumen de todas las secciones completadas
- Opción de editar cualquier sección
- Indicador visual de secciones completas/incompletas
- Checkbox de aceptación de términos y condiciones
- Botón "Enviar formulario" solo habilitado si todo está completo

**Prioridad:** ALTA  
**Estimación:** 5 puntos  
**Dependencias:** US-004

---

### **ÉPICA 2: CARGA DE DOCUMENTOS**

#### **US-006: Cargar documento de identidad (cédula/pasaporte)**

**Como** cliente  
**Quiero** cargar mi documento de identidad  
**Para** verificar mi identidad ante el banco

##### **Criterios de Aceptación:**

- Soporta carga por foto (cámara) o archivo (PDF, JPG, PNG)
- Detecta y rechaza imágenes borrosas o con poca luz
- Guía visual para posicionar correctamente el documento
- Validación de tamaño máximo (5MB)
- Previsualización antes de confirmar carga
- Permite cargar anverso y reverso por separado
- Barra de progreso durante la carga

**Prioridad:** CRÍTICA  
**Estimación:** 13 puntos  
**Dependencias:** US-005

---

#### **US-007: Cargar comprobante de ingresos**

**Como** cliente  
**Quiero** cargar mi comprobante de ingresos  
**Para** validar la información financiera declarada

##### **Criterios de Aceptación:**

- Acepta: colilla de pago, certificado laboral, declaración de renta
- Formatos: PDF, JPG, PNG, DOCX
- Validación de que el documento tiene fecha reciente (últimos 3 meses)
- Sistema detecta tipo de documento automáticamente

- Permite cargar hasta 3 documentos de ingresos

**Prioridad:** CRÍTICA

**Estimación:** 8 puntos

**Dependencias:** US-006

---

#### **US-008: Cargar comprobante de domicilio**

**Como** cliente

**Quiero** cargar un comprobante de mi dirección de residencia

**Para** verificar que vivo donde declaré en el formulario

##### **Criterios de Aceptación:**

- Acepta: recibo de servicios públicos, extracto bancario, contrato arriendo
- Documento debe tener fecha de emisión no mayor a 3 meses
- Sistema verifica que el nombre en el documento coincida con el del cliente
- Dirección en documento debe ser legible
- Mensaje claro si el documento es rechazado

**Prioridad:** ALTA

**Estimación:** 8 puntos

**Dependencias:** US-006

---

#### **US-009: Ver estado de carga de documentos**

**Como** cliente

**Quiero** ver qué documentos he cargado y cuáles faltan

**Para** completar todos los requisitos del onboarding

##### **Criterios de Aceptación:**

- Lista visual de documentos requeridos
- Indicadores: "Cargado", "Pendiente", "Rechazado", "En validación"
- Opción de reemplazar documentos rechazados
- Notificación si algún documento es rechazado con motivo específico
- Botón para continuar solo si documentos obligatorios están cargados

**Prioridad:** ALTA

**Estimación:** 5 puntos

**Dependencias:** US-006, US-007, US-008

---

### **ÉPICA 3: EXTRACCIÓN Y PROCESAMIENTO OCR**

#### **US-010: Extraer datos de documento de identidad**

**Como** sistema SIVD

**Quiero** extraer automáticamente los datos del documento de identidad

**Para** comparar con la información del formulario digital

##### **Criterios de Aceptación:**

- Extrae: número de identificación, nombres, apellidos, fecha de nacimiento, fecha de expedición, fecha de vencimiento
- Nivel de confianza mínimo del 85% por campo
- Procesa documentos en menos de 10 segundos

- Si confianza < 85%, marca campo para revisión manual
- Almacena coordenadas de cada campo extraído en el documento

**Prioridad:** CRÍTICA

**Estimación:** 13 puntos

**Dependencias:** US-006

---

#### **US-011: Extraer datos de comprobante de ingresos**

**Como** sistema SIVD

**Quiero** extraer la información financiera de los comprobantes de ingresos

**Para** validar contra lo declarado en el formulario

##### **Criterios de Aceptación:**

- Extrae: nombre del empleado, monto del salario, nombre de la empresa, período, fecha de emisión
- Identifica automáticamente el tipo de comprobante
- Maneja diferentes formatos de colillas de pago
- Convierte montos en texto a números
- Detecta moneda del documento

**Prioridad:** CRÍTICA

**Estimación:** 13 puntos

**Dependencias:** US-007

---

#### **US-012: Extraer datos de comprobante de domicilio**

**Como** sistema SIVD

**Quiero** extraer la dirección y datos del comprobante de domicilio

**Para** validar la residencia declarada

##### **Criterios de Aceptación:**

- Extrae: nombre del titular, dirección completa, fecha de emisión, tipo de servicio
- Normaliza formatos de direcciones (calle, carrera, número, etc.)
- Identifica automáticamente el tipo de comprobante
- Valida fecha de emisión no mayor a 3 meses

**Prioridad:** ALTA

**Estimación:** 8 puntos

**Dependencias:** US-008

---

### **ÉPICA 4: VALIDACIÓN Y COMPARACIÓN**

#### **US-013: Comparar datos personales (Formulario vs Documento ID)**

**Como** sistema SIVD

**Quiero** comparar los datos personales del formulario contra el documento de identidad

**Para** detectar inconsistencias en información básica

##### **Criterios de Aceptación:**

- Compara: número de identificación, nombres completos, apellidos, fecha de nacimiento
- Calcula score de similitud usando algoritmos de distancia de Levenshtein
- Tolera pequeñas diferencias ortográficas (tildes, espacios)

- Genera inconsistencia CRÍTICA si número ID no coincide exactamente
- Genera inconsistencia ALTA si nombres difieren más del 15%

**Prioridad:** CRÍTICA

**Estimación:** 13 puntos

**Dependencias:** US-010

---

#### **US-014: Validar vigencia del documento de identidad**

**Como** sistema SIVD

**Quiero** verificar que el documento de identidad esté vigente

**Para** asegurar que el cliente presenta documentos válidos

##### **Criterios de Aceptación:**

- Compara fecha de vencimiento del documento contra fecha actual
- Genera alerta CRÍTICA si documento está vencido
- Alerta MEDIA si documento vence en menos de 3 meses
- Valida que fecha de expedición sea anterior a fecha actual
- Valida coherencia entre fecha de nacimiento y edad aparente del documento

**Prioridad:** CRÍTICA

**Estimación:** 5 puntos

**Dependencias:** US-010

---

#### **US-015: Comparar información de ingresos**

**Como** sistema SIVD

**Quiero** comparar los ingresos declarados vs los documentados

**Para** detectar discrepancias en información financiera

##### **Criterios de Aceptación:**

- Compara monto declarado en formulario vs monto en comprobante
- Tolera diferencia de  $\pm 10\%$  (variaciones por bonos, deducciones)
- Genera inconsistencia ALTA si diferencia > 20%
- Genera inconsistencia MEDIA si diferencia entre 10-20%
- Verifica que el comprobante sea reciente (< 3 meses)

**Prioridad:** CRÍTICA

**Estimación:** 8 puntos

**Dependencias:** US-011

---

#### **US-016: Comparar información de domicilio**

**Como** sistema SIVD

**Quiero** validar que la dirección declarada coincida con el comprobante

**Para** verificar residencia del cliente

##### **Criterios de Aceptación:**

- Normaliza ambas direcciones antes de comparar
- Compara componentes: tipo de vía, número, barrio, ciudad
- Tolera pequeñas diferencias en formato (Calle vs Cl., # vs Nro)
- Genera inconsistencia ALTA si ciudad no coincide

- Genera inconsistencia MEDIA si dirección específica difiere

**Prioridad:** ALTA

**Estimación:** 8 puntos

**Dependencias:** US-012

---

#### **US-017: Validar coherencia del nombre en todos los documentos**

**Como** sistema SIVD

**Quiero** verificar que el nombre del cliente sea consistente en todos los documentos

**Para** detectar posibles fraudes o errores

##### **Criterios de Aceptación:**

- Compara nombre en: formulario, documento ID, comprobante ingresos, comprobante domicilio
- Identifica variaciones: nombre completo vs nombre corto, uso de segundo nombre
- Genera inconsistencia ALTA si nombres difieren significativamente entre documentos
- Permite variaciones comunes (María vs Ma., José vs J.)

**Prioridad:** ALTA

**Estimación:** 8 puntos

**Dependencias:** US-010, US-011, US-012

---

### **ÉPICA 5: MOTOR DE REGLAS Y DECISIÓN**

#### **US-018: Aplicar reglas de negocio del banco**

**Como** sistema SIVD

**Quiero** aplicar las políticas de validación del Banco Andino

**Para** asegurar cumplimiento normativo y políticas internas

##### **Criterios de Aceptación:**

- Valida edad mínima del cliente (18 años)
- Verifica que ingresos mínimos cumplan umbral del banco
- Aplica reglas específicas según tipo de cuenta solicitada
- Valida que cliente no esté en listas restrictivas (mock para MVP)
- Cada regla tiene severidad asignada (Crítica, Alta, Media, Baja)

**Prioridad:** CRÍTICA

**Estimación:** 13 puntos

**Dependencias:** US-013, US-014, US-015, US-016

---

#### **US-019: Calcular score de validación general**

**Como** sistema SIVD

**Quiero** calcular un score general de validación

**Para** determinar si el caso requiere revisión manual

##### **Criterios de Aceptación:**

- Score de 0-100 basado en todas las validaciones
- Ponderación: inconsistencias críticas (-30), altas (-20), medias (-10), bajas (-5)
- Score > 80: Aprobación automática
- Score 60-80: Revisión manual recomendada

- Score < 60: Rechazo o revisión manual obligatoria
- Score se recalcula si cliente corrige información

**Prioridad:** CRÍTICA

**Estimación:** 8 puntos

**Dependencias:** US-018

---

#### **US-020: Generar resultado de validación**

**Como** sistema SIVD

**Quiero** consolidar todos los resultados en un reporte estructurado

**Para** informar la decisión sobre el onboarding

##### **Criterios de Aceptación:**

- Estado final: APROBADO, EN\_REVISIÓN, RECHAZADO, CORRECCIÓN\_REQURIDA
- Lista detallada de inconsistencias encontradas
- Score general de validación
- Campos específicos que requieren atención
- Timestamp de la validación
- Recomendación de acción (aprobar, revisar, rechazar)

**Prioridad:** CRÍTICA

**Estimación:** 8 puntos

**Dependencias:** US-019

---

### **ÉPICA 6: NOTIFICACIONES Y COMUNICACIÓN CON CLIENTE**

#### **US-021: Notificar resultado de validación al cliente**

**Como** cliente

**Quiero** recibir notificación del resultado de mi validación

**Para** saber si mi onboarding fue exitoso o requiere acciones

##### **Criterios de Aceptación:**

- Email automático con resultado (aprobado/en revisión/requiere corrección)
- Notificación in-app en el portal
- Si aprobado: próximos pasos para completar apertura
- Si requiere corrección: lista específica de qué corregir
- Si en revisión: tiempo estimado de respuesta (24-48 horas)

**Prioridad:** ALTA

**Estimación:** 8 puntos

**Dependencias:** US-020

---

#### **US-022: Solicitar corrección de inconsistencias al cliente**

**Como** cliente

**Quiero** saber exactamente qué inconsistencias se encontraron

**Para** corregir la información de manera eficiente

##### **Criterios de Aceptación:**

- Lista clara de campos con inconsistencias

- Comparación lado a lado: "Dijiste X en formulario, pero documento dice Y"
- Instrucciones específicas de cómo corregir
- Opción de corregir en formulario o recargar documento
- Plazo para realizar correcciones (7 días)

**Prioridad:** ALTA

**Estimación:** 8 puntos

**Dependencias:** US-021

---

#### **US-023: Ver estado del proceso de validación en tiempo real**

**Como** cliente

**Quiero** ver el progreso de mi validación

**Para** saber en qué etapa está mi solicitud

##### **Criterios de Aceptación:**

- Dashboard con estados: "Documentos cargados", "En validación", "Validación completada"
- Barra de progreso visual
- Indicador de tiempo estimado restante
- Historial de acciones realizadas
- Actualización en tiempo real sin necesidad de refrescar

**Prioridad:** MEDIA

**Estimación:** 5 puntos

**Dependencias:** US-009

---

#### **ÉPICA 7: PORTAL DE ANALISTA**

##### **US-024: Ver cola de casos para revisión manual**

**Como** analista bancario

**Quiero** ver todos los casos que requieren revisión manual

**Para** priorizar y gestionar mi trabajo

##### **Criterios de Aceptación:**

- Lista de casos ordenados por prioridad (score más bajo primero)
- Filtros por: estado, rango de fechas, tipo de inconsistencia, score
- Búsqueda por número de identificación o nombre
- Indicador de tiempo en cola
- Contador de casos pendientes por analista

**Prioridad:** ALTA

**Estimación:** 8 puntos

**Dependencias:** US-020

---

##### **US-025: Revisar detalles de un caso específico**

**Como** analista bancario

**Quiero** ver toda la información de un caso en una sola pantalla

**Para** tomar una decisión informada

##### **Criterios de Aceptación:**

- Vista unificada: formulario, documentos cargados, datos extraídos
- Comparación visual lado a lado de inconsistencias
- Score de validación y detalle de cada regla aplicada
- Imágenes de documentos originales con zoom
- Historial de acciones del cliente
- Campos marcados en documentos (coordenadas OCR)

**Prioridad:** ALTA

**Estimación:** 13 puntos

**Dependencias:** US-024

---

#### **US-026: Aprobar o rechazar caso manualmente**

**Como** analista bancario

**Quiero** tomar una decisión sobre un caso en revisión

**Para** continuar o detener el proceso de onboarding

#### **Criterios de Aceptación:**

- Opciones: Aprobar, Rechazar, Solicitar más información
- Campo obligatorio de observaciones/justificación
- Si rechaza: debe seleccionar motivo de lista predefinida
- Si solicita información: especificar qué documentos/datos adicionales
- Decisión queda registrada con timestamp y usuario
- Cliente es notificado automáticamente de la decisión

**Prioridad:** ALTA

**Estimación:** 8 puntos

**Dependencias:** US-025

---

#### **US-027: Solicitar documentos adicionales**

**Como** analista bancario

**Quiero** solicitar documentos complementarios al cliente

**Para** resolver dudas sobre inconsistencias

#### **Criterios de Aceptación:**

- Lista de tipos de documentos que puede solicitar
- Campo de texto para instrucciones específicas
- Sistema notifica al cliente vía email y in-app
- Cliente puede cargar documentos solicitados en su portal
- Analista recibe notificación cuando cliente carga documentos
- Caso vuelve a la cola del analista automáticamente

**Prioridad:** MEDIA

**Estimación:** 8 puntos

**Dependencias:** US-025

---

### **ÉPICA 8: REPORTERÍA Y AUDITORÍA**

#### **US-028: Generar reporte de validación para el cliente**

**Como** sistema SIVD

**Quiero** generar un reporte en PDF del resultado de validación

**Para** que el cliente tenga registro oficial del proceso

**Criterios de Aceptación:**

- PDF con logo del banco y formato profesional
- Resumen ejecutivo del resultado
- Lista de documentos validados con estado
- Inconsistencias encontradas (si aplica)
- Fecha y hora de validación
- Número de referencia único
- Descargable desde el portal del cliente

**Prioridad:** MEDIA

**Estimación:** 8 puntos

**Dependencias:** US-020

---

**US-029: Registrar trazabilidad completa del proceso**

**Como** oficial de cumplimiento

**Quiero** tener registro auditável de todas las acciones

**Para** cumplir con requisitos regulatorios

**Criterios de Aceptación:**

- Log de todas las acciones: carga documentos, validaciones, decisiones
- Registro con: timestamp, usuario, acción, antes/después
- Almacenamiento inmutable (no se puede editar logs)
- Incluye IP y dispositivo de cada acción del cliente
- Registro de cada regla aplicada y su resultado
- Mantener logs por mínimo 5 años

**Prioridad:** ALTA

**Estimación:** 8 puntos

**Dependencias:** Todas las US anteriores

---

**US-030: Dashboard de métricas operativas**

**Como** gerente de operaciones

**Quiero** ver métricas del desempeño del SIVD

**Para** identificar cuellos de botella y oportunidades de mejora

**Criterios de Aceptación:**

- KPIs: casos procesados/día, tiempo promedio de validación, tasa de aprobación automática
- Tasa de inconsistencias por tipo de documento
- Casos en revisión manual vs casos totales
- Tiempo promedio de resolución por analista
- Gráficos de tendencias (últimos 7, 30, 90 días)
- Exportable a Excel/PDF

**Prioridad:** BAJA

**Estimación:** 13 puntos

**Dependencias:** US-029

---

**RESUMEN DE PRIORIZACIÓN**

**Críticas (Implementar Primero - Sprint 1-2):**

- US-001, US-002, US-003, US-004, US-005 (Formulario digital completo)
- US-006, US-007 (Carga de documentos críticos)
- US-010, US-011 (Extracción OCR básica)
- US-013, US-014, US-015 (Validaciones básicas)
- US-018, US-019, US-020 (Motor de decisión)

**Altas (Implementar Segundo - Sprint 3-4):**

- US-008, US-009 (Comprobante domicilio y gestión documentos)
- US-012, US-016, US-017 (Validaciones adicionales)
- US-021, US-022 (Notificaciones al cliente)
- US-024, US-025, US-026 (Portal analista básico)
- US-029 (Auditoría)

**Medias (Implementar Tercero - Sprint 5):**

- US-023 (Estado en tiempo real)
- US-027 (Solicitud documentos adicionales)
- US-028 (Reporte PDF)

**Bajas (Mejoras futuras):**

- US-030 (Dashboard métricas)
- 

**Total User Stories:** 30

**Estimación Total:** 265 puntos

**Sprints Estimados:** 5-6 sprints de 2 semanas (10-12 semanas)

**2. User Story US-001: Iniciar proceso de onboarding**

**Descripción Detallada como Product Owner**

---

**INFORMACIÓN GENERAL**

**ID:** US-001

**Título:** Iniciar proceso de onboarding

**Épica:** Registro y Formulario Digital

**Prioridad:** CRÍTICA

**Estimación:** 5 puntos

**Sprint Propuesto:** Sprint 1

**Dependencias:** Ninguna

**Stakeholders:** Clientes potenciales, Área de Onboarding Digital, Área de Experiencia de Usuario

---

**NARRATIVA**

**Como** cliente potencial del Banco Andino

**Quiero** acceder fácilmente al portal de onboarding digital desde cualquier dispositivo

**Para** iniciar mi proceso de apertura de cuenta sin necesidad de visitar una sucursal física

---

**OBJETIVO DE NEGOCIO**

Reducir la fricción en el primer contacto del cliente con el banco, aumentando la tasa de conversión de visitantes a clientes registrados. Actualmente, el 40% de los clientes abandonan el proceso en la etapa inicial debido a complejidad o falta de claridad sobre requisitos.

#### KPIs de Éxito:

- Tasa de registro completado > 70%
  - Tiempo promedio para completar registro < 3 minutos
  - Tasa de rebote en landing page < 30%
  - NPS del proceso de registro > 8/10
- 

#### PERSONAS INVOLUCRADAS

##### Persona Principal: María, 28 años

- Profesional independiente
- Busca abrir cuenta de ahorros
- Prefiere procesos 100% digitales
- Usa principalmente smartphone
- Valora claridad y rapidez

##### Persona Secundaria: Carlos, 45 años

- Empleado formal
  - Necesita cuenta nómina
  - Poco familiarizado con procesos digitales
  - Usa computador de escritorio
  - Requiere instrucciones claras
- 

#### CRITERIOS DE ACEPTACIÓN DETALLADOS

##### CA-001: Acceso Multiplataforma

DADO que soy un cliente potencial

CUANDO accedo a la URL del onboarding ([www.bancoandino.com/abrir cuenta](http://www.bancoandino.com/abrir cuenta))

ENTONCES el portal se carga correctamente en:

- ✓ Navegadores desktop (Chrome, Firefox, Safari, Edge - últimas 2 versiones)
- ✓ Navegadores móviles (iOS Safari, Chrome Android)
- ✓ Tablets (iPad, Android tablets)
- ✓ Tiempo de carga < 3 segundos en conexión 4G

##### Validación:

- Test de compatibilidad en BrowserStack
  - Google Lighthouse Score > 90
  - Responsive design validado en al menos 5 tamaños de pantalla
- 

##### CA-002: Landing Page Informativa

DADO que accedo por primera vez al portal

CUANDO llego a la landing page

ENTONCES debo ver claramente:

#### **Elementos Obligatorios:**

1. **Título principal:** "Abre tu cuenta 100% digital en minutos"
2. **Beneficios destacados:**
  - Sin filas ni papeleos
  - Validación en menos de 24 horas
  - Cuenta activa desde tu celular
3. **Requisitos claros:**
  - Ser mayor de 18 años
  - Tener cédula de ciudadanía colombiana vigente
  - Contar con comprobante de ingresos reciente
  - Tener comprobante de domicilio actualizado
4. **Tiempo estimado:** "El proceso toma aproximadamente 10-15 minutos"
5. **Botón CTA principal:** "Comenzar ahora" (color primario del banco)
6. **Enlace secundario:** "¿Ya iniciaste tu proceso? Ingresar"

#### **Diseño:**

- Jerarquía visual clara (título > beneficios > requisitos > CTA)
  - Imágenes/iconos que representen cada requisito
  - Copy simple y directo (máximo 8vo grado de lecturabilidad)
  - Footer con enlaces: Términos y condiciones, Políticas de privacidad, Ayuda
- 

#### **CA-003: Registro de Nueva Cuenta**

**DADO** que decidí iniciar el proceso

**CUANDO** hago clic en "Comenzar ahora"

**ENTONCES** se muestra un formulario de registro con:

#### **Campos del Formulario:**

1. **Email** (obligatorio)
  - Validación de formato en tiempo real
  - Mensaje: "Usaremos este email para notificaciones importantes"
  - Verificación de que no esté registrado previamente
2. **Contraseña** (obligatorio)
  - Mínimo 8 caracteres
  - Al menos 1 mayúscula, 1 minúscula, 1 número
  - Indicador visual de fortaleza (débil/media/fuerte)
  - Campo "Confirmar contraseña" con validación de coincidencia
3. **Número de celular** (obligatorio)
  - Formato: +57 XXX XXX XXXX
  - Validación de formato colombiano
  - Mensaje: "Recibirás un código de verificación"
4. **Checkbox:** "Acepto términos y condiciones y políticas de privacidad" (obligatorio)
  - Enlaces abren en modal o nueva pestaña
  - No se puede continuar sin aceptar

#### **Validaciones:**

- Todos los campos son obligatorios
  - Mensajes de error claros y específicos por campo
  - Validación en tiempo real (no solo al enviar)
  - Botón "Registrarse" deshabilitado hasta completar todo correctamente
- 

#### **CA-004: Verificación de Email**

**DADO** que completé el formulario de registro

**CUANDO** hago clic en "Registrarse"

**ENTONCES:**

1. **Confirmación Inmediata:**
  - Mensaje en pantalla: "¡Registro exitoso! Revisa tu email"
  - Indicador visual de email enviado
2. **Email Recibido (< 2 minutos):**
  - Asunto: "Verifica tu cuenta - Banco Andino"
  - Remitente: [noreply@bancoandino.com](mailto:noreply@bancoandino.com)
  - Contenido:
    - Saludo personalizado con email
    - Explicación clara: "Para continuar, verifica tu email"
    - Botón CTA: "Verificar mi cuenta"
    - Link alternativo si botón no funciona
    - Link válido por 24 horas
    - Mensaje: "¿No solicitaste esto? Ignora este email"
3. **Verificación Exitosa:**
  - Al hacer clic en link/botón:
  - Redirección a portal con mensaje: "¡Email verificado! Ahora completa tu información"
  - Sesión automáticamente iniciada
  - Redirección a formulario de datos personales (US-002)

#### Casos de Error:

- Link expirado: Opción de reenviar email
  - Link inválido: Mensaje de error con contacto de soporte
  - Email no recibido: Botón "Reenviar email" disponible después de 2 minutos
- 

#### CA-005: Validación de Cliente Existente

DADO que intento registrarme con un email ya existente  
CUANDO el sistema detecta que el email ya está registrado  
ENTONCES:

##### Si cuenta no verificada:

- Mensaje: "Este email ya está registrado pero no verificado"
- Opciones:
  1. "Reenviar email de verificación"
  2. "Usar otro email"

##### Si cuenta verificada:

1. Mensaje: "Ya tienes una cuenta con este email"
2. Opciones:
  1. Botón: "Iniciar sesión"
  2. Link: "¿Olvidaste tu contraseña?"

##### Si proceso de onboarding incompleto:

- Mensaje: "Ya iniciaste tu proceso de apertura de cuenta"
  - Botón: "Continuar mi solicitud"
  - Muestra progreso: "Último paso completado: [nombre del paso]"
- 

#### CA-006: Registro de Datos en Sistema

DADO que el cliente completa el registro  
CUANDO se verifica exitosamente  
ENTONCES el sistema debe:

##### Crear registro en base de datos con:

- id\_cliente (UUID generado automáticamente)
- email (normalizado: lowercase, trimmed)
- telefono (formato E.164)
- password\_hash (bcrypt, salt rounds = 10)
- estado\_onboarding = "REGISTRO\_COMPLETADO"
- email\_verificado = true
- fecha\_registro (timestamp UTC)
- ip\_registro (para auditoría)
- dispositivo\_registro (user agent)
- origen\_registro (web/mobile/tablet)

#### Triggers automáticos:

- Evento enviado a sistema de analytics: "user\_registered"
  - Notificación a sistema CRM: nuevo lead
  - Inicio de sesión automático del cliente
- 

### CA-007: Inicio de Sesión para Usuarios Registrados

**DADO** que soy un cliente que ya se registró previamente  
**CUANDO** accedo al portal y hago clic en "¿Ya iniciaste tu proceso? Ingresar"  
**ENTONCES** se muestra formulario de login con:

#### Campos:

1. Email
2. Contraseña
3. Checkbox: "Recordar mi sesión" (opcional)
4. Link: "¿Olvidaste tu contraseña?"

#### Después del login exitoso:

- Redirección inteligente según estado\_onboarding:
  - "REGISTRO\_COMPLETADO" → Formulario datos personales (US-002)
  - "FORMULARIO\_COMPLETADO" → Carga de documentos (US-006)
  - "DOCUMENTOS\_CARGADOS" → Estado de validación (US-023)
  - "VALIDACION\_COMPLETADA" → Dashboard final

#### Seguridad:

- Máximo 5 intentos fallidos → bloqueo temporal (15 minutos)
  - Sesión válida por 2 horas de inactividad
  - Token JWT con refresh token
- 

### CRITERIOS DE RECHAZO (Out of Scope)

Lo siguiente NO está incluido en esta User Story:

- ✗ Registro mediante redes sociales (Google, Facebook)
- ✗ Autenticación biométrica (huella, facial)
- ✗ Chatbot de asistencia en landing page
- ✗ Validación de celular con SMS (se hará en US posterior)
- ✗ Opciones de idioma (solo español)
- ✗ Proceso para extranjeros o empresas (solo personas naturales colombianas)
- ✗ Integración con sistemas core del banco en esta fase
- ✗ Analytics avanzado de comportamiento de usuario

---

## DISEÑO Y UX

### Wireframes Requeridos:

1. Landing page desktop
2. Landing page mobile
3. Formulario de registro desktop
4. Formulario de registro mobile
5. Pantalla de confirmación de email enviado
6. Email de verificación (template HTML)
7. Pantalla de email verificado exitosamente

### Guía de Estilo:

- Seguir Design System del Banco Andino v2.3
  - Colores corporativos: Azul #0052CC (primario), Verde #00875A (éxito)
  - Tipografía: Roboto (títulos), Open Sans (cuerpo)
  - Espaciado: múltiplos de 8px
  - Botones: altura mínima 48px (mobile), 40px (desktop)
  - Accesibilidad: WCAG 2.1 nivel AA mínimo
- 

## CONSIDERACIONES TÉCNICAS

### Frontend:

- Framework: React 18+
- Librería de formularios: React Hook Form + Yup (validación)
- Gestión de estado: React Context o Redux Toolkit
- Responsive: Mobile-first approach
- Accesibilidad: Usar semantic HTML, ARIA labels donde sea necesario

### Backend:

- API REST: Node.js + Express
- Base de datos: PostgreSQL
- Endpoints necesarios:
  - POST /api/v1/auth/register - Crear cuenta
  - POST /api/v1/auth/verify-email - Verificar email
  - POST /api/v1/auth/login - Iniciar sesión
  - POST /api/v1/auth/resend-verification - Reenviar verificación
  - GET /api/v1/auth/check-email - Verificar si email existe

### Seguridad:

- HTTPS obligatorio
- Rate limiting: 10 requests/minuto por IP
- Sanitización de inputs (prevenir XSS)
- Headers de seguridad: CSP, HSTS, X-Frame-Options
- Passwords: bcrypt con salt rounds = 10
- Tokens: JWT con expiración de 2 horas

### Servicios Externos:

- Email: Amazon SES o SendGrid
- Template engine: Handlebars para emails
- CDN: CloudFront para assets estáticos

---

## MÉTRICAS Y ANALÍTICA

### Eventos a Trackear:

1. page\_view\_landing - Vista de landing page
2. click\_start\_onboarding - Click en "Comenzar ahora"
3. form\_started - Usuario comienza a llenar formulario
4. form\_error - Error en validación de campo (incluir campo)
5. registration\_completed - Registro exitoso
6. email\_verification\_sent - Email de verificación enviado
7. email\_verified - Email verificado exitosamente
8. registration\_abandoned - Usuario abandona en formulario (> 5 min sin interacción)

### Dashboards Requeridos:

- Funnel de conversión (landing → registro → verificación)
  - Tasa de abandono por paso
  - Errores más comunes por campo
  - Tiempo promedio por paso
  - Dispositivos más usados
- 

## TESTING Y QA

### Tests Unitarios (Mínimo 80% cobertura):

- Validaciones de formulario
- Funciones de sanitización de inputs
- Generación de tokens
- Hash de passwords

### Tests de Integración:

- Flujo completo registro → verificación → login
- Manejo de errores de API
- Timeout de requests

### Tests E2E (Cypress o Playwright):

- Happy path completo desde landing hasta login
- Registro con email duplicado
- Verificación de email expirado
- Login con credenciales incorrectas
- Responsive en 3 breakpoints principales

### Tests de Seguridad:

- Intentos de SQL injection
- XSS attacks en campos de texto
- CSRF protection
- Rate limiting funcionando

### Tests Manuales:

- ✓ Accesibilidad con lector de pantalla
- ✓ Navegación solo con teclado (tab order lógico)
- ✓ Textos legibles y claros para usuario no técnico

- ✓ Errores comprensibles en lenguaje humano
- 

## ESCENARIOS DE USO REALES

### Escenario 1: María - Primera vez (Happy Path)

1. María ve publicidad en Instagram del Banco Andino
2. Hace clic y llega a landing page desde su iPhone
3. Lee requisitos y ve que los cumple todos
4. Hace clic en "Comenzar ahora"
5. Ingresa su email personal, crea contraseña fuerte
6. Acepta términos y condiciones después de leerlos rápidamente
7. Recibe email de verificación en 30 segundos
8. Abre email desde Gmail app en su celular
9. Hace clic en botón de verificación
10. Es redirigida al portal con sesión iniciada
11. Ve mensaje de bienvenida y continúa con datos personales

**Tiempo total:** 2 minutos 45 segundos

**Resultado:** ✓ Registro exitoso

---

### Escenario 2: Carlos - Abandono y Retorno

1. Carlos accede desde computador del trabajo
2. Comienza registro pero se distrae con una llamada
3. Cierra el navegador sin completar
4. Al día siguiente, vuelve a la URL desde su casa
5. Hace clic en "¿Ya iniciaste tu proceso? Ingresar"
6. Intenta ingresar pero no recuerda contraseña
7. Usa "¿Olvidaste tu contraseña?"
8. Recibe email de recuperación
9. Crea nueva contraseña
10. Ingresa exitosamente
11. Sistema lo lleva exactamente donde se quedó

**Tiempo total:** 5 minutos (distribuido en 2 sesiones)

**Resultado:** ✓ Recuperación exitosa

---

### Escenario 3: Andrea - Email Duplicado

1. Andrea intenta registrarse con email corporativo
2. Email ya existe de un intento previo hace 1 semana
3. Sistema le muestra: "Ya tienes una cuenta"
4. Hace clic en "Iniciar sesión"
5. Ingresa sus credenciales correctamente
6. Sistema la lleva a donde se quedó (carga de documentos)
7. Continúa su proceso sin problemas

**Tiempo total:** 1 minuto

**Resultado:** ✓ Experiencia fluida sin pérdida de progreso

---

## **MANEJO DE ERRORES**

### **Error 1: Email no recibido**

**Síntoma:** Usuario no recibe email de verificación  
**Causas posibles:**

- Email en spam
- Servidor de email caído
- Email mal escrito

#### **Solución:**

- Botón "Reenviar email" visible después de 2 minutos
  - Mensaje: "Revisa tu carpeta de spam"
  - Opción de cambiar email si fue mal escrito
  - Soporte en línea si persiste
- 

### **Error 2: Link de verificación expirado**

**Síntoma:** Usuario hace clic después de 24 horas  
**Solución:**

- Mensaje claro: "Este link expiró por seguridad"
  - Botón: "Enviar nuevo link de verificación"
  - Nuevo email enviado inmediatamente
- 

### **Error 3: Problemas de conectividad**

**Síntoma:** Request timeout al registrarse  
**Solución:**

- Mensaje: "Problemas de conexión. Intentando nuevamente..."
  - Retry automático (max 3 intentos)
  - Si falla: Guardar datos en localStorage
  - Permitir recuperar datos al regresar
- 

## **DEFINICIÓN DE DONE**

Esta User Story se considera TERMINADA cuando:

- Código desarrollado y revisado (peer review)
- Tests unitarios escritos y pasando (>80% cobertura)
- Tests de integración pasando
- Tests E2E del flujo completo pasando
- Diseños implementados según wireframes aprobados
- Responsive verificado en 5 dispositivos reales
- Accesibilidad validada (WCAG 2.1 AA)

- Performance: Lighthouse score > 90
  - Documentación técnica actualizada
  - API documentada en Swagger/OpenAPI
  - Variables de entorno configuradas en todos los ambientes
  - Desplegado en ambiente de QA
  - QA realizó tests manuales y aprobó
  - Product Owner revisó y aprobó funcionalidad
  - Eventos de analytics implementados y verificados
  - Logs de error configurados
  - Monitoreo configurado (uptime, errores)
  - Desplegado en producción
  - Smoke tests en producción pasando
  - Documentación de usuario creada (si aplica)
- 

## REFERENCIAS Y RECURSOS

### Documentos Relacionados:

- [PRD] Product Requirements Document - SIVD v1.0
- [DES] Design System Banco Andino v2.3
- [API] API Specification - Authentication Module
- [SEC] Security Guidelines - Customer Data Protection
- [LEG] Términos y Condiciones - Apertura de Cuenta Digital
- [LEG] Política de Privacidad - Banco Andino 2024

### Repositorios:

- Frontend: [github.com/banco-andino/sivd-frontend](https://github.com/banco-andino/sivd-frontend)
- Backend: [github.com/banco-andino/sivd-backend](https://github.com/banco-andino/sivd-backend)
- Design: Figma - "SIVD Onboarding Flows"

## TIMELINE

**Estimación:** 5 puntos de historia ≈ 3-4 días de desarrollo

- **Día 1:** Setup de proyecto, estructura base, landing page
- **Día 2:** Formulario de registro, validaciones, API de registro
- **Día 3:** Verificación de email, login, casos de error
- **Día 4:** Tests, fixes, QA, despliegue

**Sprint Planning:** Sprint 1 - Semana 1

**Review:** Sprint 1 - Final de Semana 2

**Release:** Sprint 1 - Con US-002 a US-005

---

## NOTAS ADICIONALES

### Consideraciones de Negocio:

- Esta es la puerta de entrada al sistema; cualquier fricción aquí impacta conversión significativamente
- Benchmark: Bancos digitales líderes tienen tasa de conversión de 65-75% en este paso
- Cada segundo adicional de carga reduce conversión en ~7%

### Mejoras Futuras (No en MVP):

- Login con biometría (Touch ID, Face ID)
- Recordar dispositivo para no pedir verificación cada vez
- Registro social (Google, Facebook, Apple)
- Detección de bot/fraud en registro
- A/B testing de diferentes copys en landing page

### Riesgos Identificados:

- ● ALTO: Emails en spam - Configurar SPF, DKIM, DMARC correctamente
- ● MEDIO: Abandono por complejidad de contraseña - Considerar simplificar requisitos
- ● MEDIO: Competencia captura leads - Optimizar SEO y SEM
- ● BAJO: Problemas de escalabilidad - Arquitectura cloud auto-escalable

## 3. Tickets Jira - US-001: Iniciar Proceso de Onboarding Sistema Inteligente de Validación Documental (SIVD) - Banco Andino

User Story: US-001

Epic: Registro y Formulario Digital

Sprint: Sprint 1

Story Points: 5

Total Estimado: 96 horas

## ARQUITECTURA Y CONFIGURACIÓN

### SIVD-101: Configuración inicial del proyecto Frontend

ID del Ticket: SIVD-101

Título del Ticket: Setup inicial de proyecto React para módulo de onboarding

Tipo: Technical Task

Componente: Frontend

**Descripción:** Configurar la estructura base del proyecto React para el módulo de onboarding del SIVD, incluyendo configuración de herramientas de desarrollo, estructura de carpetas, librerías esenciales y configuración de variables de entorno.

### Criterios de Aceptación:

- Proyecto React 18+ inicializado con Vite o Create React App
- Estructura de carpetas siguiendo arquitectura feature-based:  
`src/ |— features/ | |— onboarding/ | |— components/ | |— pages/ | |— hooks/ |  
 |— services/ | |— types/ | |— shared/ | |— components/ | |— utils/ | |— constants/ | |— config/ |  
 |— styles/ |`
- ESLint y Prettier configurados según guía de estilo del equipo
- TypeScript configurado con strict mode
- Librerías instaladas: React Router v6, React Hook Form, Yup, Axios, TailwindCSS
- Variables de entorno configuradas (`.env.example` creado)
- Git hooks con Husky para pre-commit (lint + format)
- README.md con instrucciones de setup
- Scripts npm configurados (dev, build, test, lint)

**Prioridad:** Highest

**Estimación de Esfuerzo:** 4 horas

**Tareas Técnicas:**

1. Inicializar proyecto con npm create vite@latest y template React + TypeScript
2. Instalar dependencias principales y dev dependencies
3. Configurar tsconfig.json con paths aliases (@features, @shared, @config)
4. Configurar TailwindCSS con Design System tokens del Banco Andino
5. Crear estructura de carpetas base
6. Configurar ESLint con reglas: react-hooks, typescript-eslint, prettier
7. Configurar Husky con lint-staged
8. Crear archivos .env.example y .env.development
9. Configurar Axios instance con interceptors base
10. Crear archivo de constantes de configuración
11. Documentar estructura y decisiones en README.md

**Notas:**

- Usar Vite por mejor performance en desarrollo
- Asegurar compatibilidad con Node.js v18+
- Variables de entorno: VITE\_API\_BASE\_URL, VITE\_ENVIRONMENT
- Considerar configurar absolute imports para evitar "./././"

**Dependencias:** Ninguna

**Bloqueadores:** Ninguno

**Assignee:** Frontend Lead

---

**SIVD-102: Configuración inicial del proyecto Backend**

**ID del Ticket:** SIVD-102

**Título del Ticket:** Setup API REST con Node.js + Express para módulo de autenticación

**Tipo:** Technical Task

**Componente:** Backend

**Descripción:** Configurar la estructura base del backend para el módulo de autenticación y onboarding, incluyendo Express server, configuración de base de datos PostgreSQL, middleware de seguridad y estructura modular.

**Criterios de Aceptación:**

- Proyecto Node.js + Express inicializado con TypeScript
- Estructura de carpetas siguiendo arquitectura limpia:  
src/ |—— modules/ |—— auth/ |—— controllers/ |—— services/ |—— repositories/ |—— dtos/ |—— validators/ |—— routes.ts |—— shared/ |—— middleware/ |—— utils/ |—— types/ |—— errors/ |—— config/ |—— database.ts |—— env.ts |—— server.ts
- PostgreSQL connection configurada con TypeORM o Prisma
- Middleware de seguridad configurado: helmet, cors, rate-limit
- Sistema de logging configurado (Winston o Pino)
- Manejo centralizado de errores implementado
- Health check endpoint: GET /api/health
- Variables de entorno configuradas
- Docker Compose para desarrollo local (PostgreSQL + Redis)
- Migraciones de BD configuradas

**Prioridad:** Highest

**Estimación de Esfuerzo:** 6 horas

**Tareas Técnicas:**

1. Inicializar proyecto: npm init -y y configurar TypeScript
2. Instalar dependencias: express, cors, helmet, express-rate-limit, dotenv, bcrypt, jsonwebtoken
3. Instalar Prisma: npm install prisma @prisma/client
4. Configurar estructura de carpetas modular
5. Crear middleware de error handling centralizado
6. Configurar Winston para logging (info, error, debug)
7. Configurar CORS permitiendo solo dominios autorizados
8. Implementar rate limiting: 100 req/15min por IP
9. Crear docker-compose.yml con PostgreSQL 15 + Redis 7
10. Inicializar Prisma: npx prisma init
11. Configurar variables de entorno (.env.example)
12. Crear endpoint de health check
13. Documentar arquitectura en README.md

**Notas:**

- Usar Prisma por mejor DX con TypeScript
- PostgreSQL en Docker para desarrollo, RDS en producción
- Rate limiting global + específico por endpoint crítico
- Variables: DATABASE\_URL, JWT\_SECRET, JWT\_EXPIRATION, PORT, NODE\_ENV
- Considerar usar pnpm en lugar de npm para mejor performance

**Dependencias:** Ninguna

**Bloqueadores:** Ninguno

**Assignee:** Backend Lead

---

**SIVD-103: Configuración de base de datos y esquema inicial**

**ID del Ticket:** SIVD-103

**Título del Ticket:** Diseñar e implementar esquema de BD para módulo de autenticación

**Tipo:** Technical Task

**Componente:** Database

**Descripción:** Crear el esquema de base de datos inicial para el módulo de autenticación, incluyendo tabla de clientes, auditoría y configuración de índices para optimización.

**Criterios de Aceptación:**

- Esquema Prisma creado con modelo Cliente
- Tabla clientes con todos los campos requeridos
- Tabla auditoria\_auth para trazabilidad
- Índices creados en campos de búsqueda frecuente
- Constraints de unicidad configurados
- Valores por defecto establecidos
- Migración inicial creada y probada
- Seed data para desarrollo creado (usuarios de prueba)
- Documentación del modelo de datos

**Prioridad:** Highest

**Estimación de Esfuerzo:** 4 horas

**Tareas Técnicas:**

1. Crear schema.prisma con modelo Cliente:
2. model Cliente { id String @id @default(uuid()) email String @unique telefono String? passwordHash String @map("password\_hash") estadoOnboarding String @default("REGISTRO\_COMPLETADO") @map("estado\_onboarding") emailVerificado Boolean @default(false) @map("email\_verificado") tokenVerificacion String? @map("token\_verificacion") tokenExpiracion DateTime?

```

@map("token_expiracion") fechaRegistro DateTime @default(now()) @map("fecha_registro") ipRegistro
String? @map("ip_registro") dispositivoRegistro String? @map("dispositivo_registro") origenRegistro String?
@map("origen_registro") ultimoAcceso DateTime? @map("ultimo_acceso") intentosFallidos Int
@default(0) @map("intentos_fallidos") bloqueadoHasta DateTime? @map("bloqueado_hasta") createdAt
DateTime @default(now()) @map("created_at") updatedAt DateTime @updatedAt @map("updated_at")
@@index([email]) @@index([estadoOnboarding]) @@index([fechaRegistro]) @@map("clientes")}
3. Crear modelo AuditoriaAuth para logs
4. Configurar enum para estadoOnboarding
5. Crear índices compuestos si necesario
6. Ejecutar npx prisma migrate dev --name init
7. Crear archivo seed.ts con 5 usuarios de prueba
8. Ejecutar seed: npx prisma db seed
9. Generar diagrama ER del esquema
10. Documentar modelo en docs/database/schema.md

```

**Notas:**

- Usar UUID v4 para IDs (mejor seguridad que auto-increment)
- passwordHash con bcrypt (nunca almacenar password plano)
- tokenVerificacion para verificación de email
- intentosFallidos + bloqueadoHasta para prevenir brute force
- Todos los timestamps en UTC
- Considerar particionamiento por fecha si volumen es alto

**Dependencias:** SIVD-102

**Bloqueadores:** Ninguno

**Assignee:** Backend Developer

---

## 👉 FRONTEND - UI COMPONENTS

### SIVD-104: Implementar componente Landing Page

**ID del Ticket:** SIVD-104

**Título del Ticket:** Desarrollar landing page de onboarding con información y CTA

**Tipo:** Development

**Componente:** Frontend

**Descripción:** Desarrollar la landing page inicial del proceso de onboarding que presente información clara sobre requisitos, beneficios y CTA principal para iniciar el proceso. Debe ser responsive y seguir el Design System del Banco Andino.

**Criterios de Aceptación:**

- Componente LandingPage.tsx creado en features/onboarding/pages/
- Responsive en breakpoints: mobile (320px), tablet (768px), desktop (1024px+)
- Sección hero con título principal y CTA destacado
- Lista de beneficios con iconos (lucide-react)
- Sección de requisitos clara y legible
- Tiempo estimado del proceso visible
- Footer con enlaces a términos y políticas
- Botón "Comenzar ahora" con estado hover y loading
- Link "¿Ya iniciaste? Ingresar" visible
- Imágenes optimizadas (WebP con fallback)
- Accesibilidad: landmarks, alt texts, contraste WCAG AA
- Lighthouse score > 90

**Prioridad:** High

**Estimación de Esfuerzo:** 8 horas

**Tareas Técnicas:**

1. Crear componente LandingPage.tsx con TypeScript
2. Crear sub-componentes reutilizables:
  - o HeroSection.tsx
  - o BenefitsSection.tsx
  - o RequirementsSection.tsx
  - o Footer.tsx
3. Implementar layout con TailwindCSS Grid/Flexbox
4. Agregar iconos de lucide-react (Check, Clock, Shield, FileText)
5. Implementar animaciones sutiles con Framer Motion (opcional)
6. Optimizar imágenes con lazy loading
7. Agregar meta tags para SEO
8. Implementar tracking de eventos: page\_view\_landing, click\_start\_onboarding
9. Crear tests con React Testing Library
10. Validar responsive en Chrome DevTools + dispositivos reales
11. Verificar accesibilidad con Axe DevTools
12. Optimizar performance (code splitting si necesario)

**Notas:**

- Copy text debe venir de archivo de constantes (i18n-ready)
- Usar lazy loading para imágenes below the fold
- Considerar A/B testing de diferentes CTAs (preparar estructura)
- Hero image: ilustración de persona con celular
- Colores: Azul #0052CC (primario), Verde #00875A (success)
- Font: Roboto 700 (headings), Open Sans 400 (body)

**Dependencias:** SIVD-101

**Bloqueadores:** Pendiente aprobación de diseños finales

**Assignee:** Frontend Developer 1

---

**SIVD-105: Implementar formulario de registro**

**ID del Ticket:** SIVD-105

**Título del Ticket:** Crear formulario de registro con validaciones en tiempo real

**Tipo:** Development

**Componente:** Frontend

**Descripción:** Desarrollar el formulario de registro de nueva cuenta con campos de email, contraseña, teléfono y aceptación de términos. Incluir validaciones en tiempo real, indicadores visuales de error y fortaleza de contraseña.

**Criterios de Aceptación:**

- Componente RegisterForm.tsx con React Hook Form
- Campos implementados: email, password, confirmPassword, telefono, termsAccepted
- Validaciones en tiempo real con Yup schema
- Indicador visual de fortaleza de contraseña (débil/media/fuerte)
- Mensajes de error específicos por campo
- Botón "Registrarse" deshabilitado si form inválido
- Loading state durante envío de formulario
- Manejo de errores de API (email duplicado, server error)
- Links a términos y políticas abren en modal
- Accesibilidad: labels asociados, error messages con aria-describedby
- Tests unitarios para validaciones

**Prioridad:** High

**Estimación de Esfuerzo:** 10 horas

#### Tareas Técnicas:

1. Crear componente RegisterForm.tsx con useForm de React Hook Form
2. Crear schema de validación con Yup:
3. const registerSchema = yup.object({ email: yup.string().email('Email inválido').required('Email requerido'), password: yup.string() .min(8, 'Mínimo 8 caracteres') .matches(/[A-Z]/, 'Requiere mayúscula') .matches(/[a-z]/, 'Requiere minúscula') .matches(/\d{0-9}/, 'Requiere número') .required('Contraseña requerida'), confirmPassword: yup.string() .oneOf([yup.ref('password')], 'Contraseñas no coinciden'), telefono: yup.string() .matches(/^\+57[0-9]{10}\$/, 'Formato: +57XXXXXXXXXX') .required('Teléfono requerido'), termsAccepted: yup.boolean() .oneOf([true], 'Debes aceptar términos y condiciones'))});
4. Crear componente PasswordStrengthIndicator.tsx
5. Implementar lógica de cálculo de fortaleza (zxcvbn library)
6. Crear componente Input.tsx reutilizable con estados error/success
7. Implementar input type="tel" con máscara para teléfono
8. Crear componente Checkbox.tsx con link a términos
9. Implementar modal de términos con TermsModal.tsx
10. Integrar con API de registro (POST /api/v1/auth/register)
11. Manejar estados: idle, loading, success, error
12. Implementar feedback visual: toast/alert de éxito o error
13. Agregar tracking: form\_started, form\_error, registration\_completed
14. Crear tests con React Testing Library + MSW para mock API

#### Notas:

- Usar debounce en validación de email para no saturar
- Password no debe ser visible por defecto (toggle eye icon)
- Validar email disponible con API antes de submit (optional)
- Guardar datos en sessionStorage si abandona para recuperar
- Considerar Google reCAPTCHA v3 para prevenir bots (futuro)

**Dependencias:** SIVD-101, SIVD-104

**Bloqueadores:** Ninguno

**Assignee:** Frontend Developer 2

---

#### SIVD-106: Implementar pantallas de verificación de email

**ID del Ticket:** SIVD-106

**Título del Ticket:** Crear pantallas de confirmación y verificación de email

**Tipo:** Development

**Componente:** Frontend

**Descripción:** Desarrollar las pantallas de flujo de verificación de email: pantalla de confirmación después del registro, página de verificación exitosa y manejo de errores (link expirado, inválido).

#### Criterios de Aceptación:

- Componente EmailSentPage.tsx con mensaje de confirmación
- Instrucciones para revisar inbox y spam
- Botón "Reenviar email" disponible después de 2 minutos
- Componente EmailVerifiedPage.tsx con mensaje de éxito
- Redirección automática después de verificación
- Componente EmailVerificationError.tsx para errores
- Manejo de link expirado con opción de reenviar
- Manejo de link inválido con mensaje apropiado
- Loader/spinner durante verificación
- Tests para cada escenario

**Prioridad:** High

**Estimación de Esfuerzo:** 6 horas

**Tareas Técnicas:**

1. Crear EmailSentPage.tsx con ilustración de sobre
2. Implementar countdown timer para botón "Reenviar"
3. Crear servicio resendVerificationEmail()
4. Crear ruta /verify-email/:token en React Router
5. Crear EmailVerifiedPage.tsx con checkmark animado
6. Implementar verificación automática al cargar página:
7. 

```
useEffect(() => { const verifyEmail = async () => { try { await authService.verifyEmail(token); setStatus('success'); setTimeout(() => navigate('/onboarding'), 3000); } catch (error) { setStatus('error'); } }; verifyEmail(); }, [token]);
```
8. Crear EmailVerificationError.tsx con manejo de diferentes errores
9. Implementar lógica para distinguir error de expiración vs inválido
10. Agregar animaciones con Framer Motion (optional)
11. Implementar tracking de eventos de verificación
12. Crear tests con diferentes escenarios de error

**Notas:**

- Countdown de 120 segundos antes de permitir reenvío
- Guardar timestamp de último envío en localStorage
- Token viene en URL como query param: /verify-email?token=xxx
- Mostrar mensaje amigable, no código de error técnico
- Considerar enviar analytics de cuántos usuarios no reciben email

**Dependencias:** SIVD-105

**Bloqueadores:** Ninguno

**Assignee:** Frontend Developer 1

---

**SIVD-107: Implementar formulario de login**

**ID del Ticket:** SIVD-107

**Título del Ticket:** Crear formulario de inicio de sesión con manejo de errores

**Tipo:** Development

**Componente:** Frontend

**Descripción:** Desarrollar formulario de login para usuarios que ya iniciaron registro, con manejo de credenciales incorrectas, cuenta bloqueada, opción de recordar sesión y recuperación de contraseña.

**Criterios de Aceptación:**

- Componente LoginForm.tsx con React Hook Form
- Campos: email y password
- Checkbox "Recordar mi sesión" funcional
- Link "¿Olvidaste tu contraseña?" visible
- Manejo de errores específicos:
  - Credenciales incorrectas
  - Cuenta no verificada
  - Cuenta bloqueada (mostrar tiempo restante)
  - Error de servidor
- Redirección inteligente según estado de onboarding
- Loading state durante autenticación
- Persistencia de sesión con JWT en localStorage/cookie
- Tests para diferentes escenarios

**Prioridad:** High

**Estimación de Esfuerzo:** 8 horas

**Tareas Técnicas:**

1. Crear componente LoginForm.tsx con validaciones básicas
2. Crear schema Yup para validación:
3. const loginSchema = yup.object({ email: yup.string().email().required(), password: yup.string().required(), rememberMe: yup.boolean()});
4. Implementar servicio authService.login(credentials)
5. Manejar response con JWT token y refresh token
6. Guardar tokens según "Recordar sesión":
  - Si true: localStorage
  - Si false: sessionStorage
7. Crear función de redirección inteligente:
8. const redirectByStatus = (status: string) => { const routes = { REGISTRO\_COMPLETADO: '/onboarding/personal-info', FORMULARIO\_COMPLETADO: '/onboarding/documents', DOCUMENTOS\_CARGADOS: '/onboarding/status', VALIDACION\_COMPLETADA: '/dashboard' }; return routes[status] || '/onboarding';};
9. Implementar manejo de diferentes errores de API
10. Crear componente ForgotPasswordModal.tsx (UI básico)
11. Implementar contador de intentos fallidos (cliente side)
12. Agregar tracking: login\_success, login\_failed, account\_locked
13. Crear tests para cada escenario de error
14. Implementar auto-focus en campo email al cargar

**Notas:**

- Máximo 5 intentos fallidos antes de bloqueo temporal (15 min)
- JWT expira en 2 horas, refresh token en 7 días
- Usar httpOnly cookies para tokens si es posible (más seguro)
- Validar token en cada request con Axios interceptor
- No mostrar si error es email o password (seguridad)
- Mensaje genérico: "Credenciales incorrectas"

**Dependencias:** SIVD-101

**Bloqueadores:** Ninguno

**Assignee:** Frontend Developer 2

---

## BACKEND - API ENDPOINTS

### SIVD-108: Implementar endpoint de registro de usuario

**ID del Ticket:** SIVD-108

**Título del Ticket:** Desarrollar API POST /auth/register con validaciones y seguridad

**Tipo:** Development

**Componente:** Backend

**Descripción:** Implementar endpoint para registro de nuevos usuarios con validaciones robustas, hash de contraseña, generación de token de verificación y envío de email de confirmación.

**Criterios de Aceptación:**

- Endpoint POST /api/v1/auth/register implementado
- Validación de datos de entrada con class-validator
- Verificación de email único (no duplicado)
- Hash de contraseña con bcrypt (salt rounds = 10)
- Generación de token de verificación único
- Registro en base de datos con transaction
- Envío de email de verificación asíncrono
- Log de auditoría de registro
- Response apropiado: 201 Created con datos (sin password)
- Manejo de errores: 400, 409 (conflict), 500
- Rate limiting: 5 registros por IP por hora
- Tests unitarios e integración

**Prioridad:** Highest

**Estimación de Esfuerzo:** 10 horas

#### Tareas Técnicas:

1. Crear DTO CreateClienteDto:
2. export class CreateClienteDto { @IsEmail() @IsNotEmpty() email: string; @IsString() @MinLength(8) @Matches(/^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)/) password: string; @IsString() @Matches(/^\+57[0-9]{10}\$/) telefono: string; @IsBoolean() termsAccepted: boolean;}
3. Crear AuthController.register() method
4. Crear AuthService.registerCliente() con lógica de negocio
5. Implementar verificación de email único en repository
6. Implementar hash de password con bcrypt:
7. const salt = await bcrypt.genSalt(10);const passwordHash = await bcrypt.hash(dto.password, salt);
8. Generar token de verificación: crypto.randomBytes(32).toString('hex')
9. Guardar token con expiración de 24 horas
10. Crear registro en tabla clientes con transaction
11. Crear registro en auditoria\_auth con IP y user agent
12. Enviar evento a queue para envío de email (implementar worker básico)
13. Implementar rate limiting con express-rate-limit
14. Crear respuesta sanitizada (sin password, sin token)
15. Manejar errores específicos con custom exceptions
16. Crear tests unitarios con Jest + mocks
17. Crear tests de integración con supertest

#### Notas:

- Usar transactions para atomicidad (rollback si falla)
- Email enviado de forma asíncrona (no bloquear response)
- Sanitizar email: lowercase + trim antes de guardar
- Extraer IP con req.ip o x-forwarded-for header
- User agent: req.headers['user-agent']
- Considerar honeypot field para detectar bots
- Rate limiting por IP + por email intentado

**Dependencias:** SIVD-102, SIVD-103

**Bloqueadores:** Ninguno

**Assignee:** Backend Developer 1

---

#### SIVD-109: Implementar endpoint de verificación de email

**ID del Ticket:** SIVD-109

**Título del Ticket:** Desarrollar API GET /auth/verify-email para validar token

**Tipo:** Development

**Componente:** Backend

**Descripción:** Implementar endpoint para verificar el token de email recibido por el usuario, marcar la cuenta como verificada y permitir el inicio de sesión.

#### Criterios de Aceptación:

- Endpoint GET /api/v1/auth/verify-email/:token implementado
- Validación de existencia de token en BD
- Verificación de no expiración (< 24 horas)
- Actualización de estado: emailVerificado = true
- Invalidation de token después de uso
- Response 200 OK con mensaje de éxito
- Manejo de errores: token inválido (404), expirado (410), ya usado (409)
- Log de auditoría de verificación
- Tests unitarios e integración

**Prioridad:** Highest

**Estimación de Esfuerzo:** 6 horas

**Tareas Técnicas:**

1. Crear ruta GET /auth/verify-email/:token en routes
2. Crear AuthController.verifyEmail() method
3. Crear AuthService.verifyEmailToken() con lógica:
4. 

```
async verifyEmailToken(token: string) { const cliente = await this.repository.findByVerificationToken(token); if (!cliente) { throw new NotFoundException('Token inválido'); } if (cliente.emailVerificado) { throw new ConflictException('Email ya verificado'); } if (new Date() > cliente.tokenExpiracion) { throw new GoneException('Token expirado'); } await this.repository.update(cliente.id, { emailVerificado: true, tokenVerificacion: null, tokenExpiracion: null }); await this.auditService.log({ action: 'EMAIL_VERIFIED', clientId: cliente.id, timestamp: new Date() }); return { success: true, message: 'Email verificado exitosamente' };}
```
5. Implementar custom exceptions: GoneException para token expirado
6. Crear respuesta con información del siguiente paso
7. Agregar logging de evento de verificación
8. Crear tests para cada escenario:
  - Token válido
  - Token inválido
  - Token expirado
  - Token ya usado
9. Test de integración end-to-end

**Notas:**

- Token debe ser URL-safe (no caracteres especiales)
- Considerar agregar CSRF protection
- Log debe incluir timestamp, IP, success/failure
- Despues de verificar, usuario puede hacer login inmediatamente
- Considerar enviar email de bienvenida despues de verificación

**Dependencias:** SIVD-108

**Bloqueadores:** Ninguno

**Assignee:** Backend Developer 1

---

**SIVD-110: Implementar endpoint de reenvío de verificación**

**ID del Ticket:** SIVD-110

**Título del Ticket:** Desarrollar API POST /auth/resend-verification para reenviar email

**Tipo:** Development

**Componente:** Backend

**Descripción:** Implementar endpoint para reenviar email de verificación cuando el usuario no lo recibe o el token expira.

**Criterios de Aceptación:**

- Endpoint POST /api/v1/auth/resend-verification implementado
- Validación de que email existe en sistema
- Validación de que email no está ya verificado
- Generación de nuevo token (invalidar anterior)
- Envío de nuevo email de verificación
- Rate limiting: máximo 3 reenvíos por hora por email
- Response 200 OK con mensaje confirmación
- Manejo de errores: email no existe, ya verificado, rate limit
- Tests unitarios e integración

**Prioridad:** High

**Estimación de Esfuerzo:** 4 horas

**Tareas Técnicas:**

1. Crear DTO ResendVerificationDto:
2. export class ResendVerificationDto { @IsEmail() @IsNotEmpty() email: string;}
3. Crear ruta POST /auth/resend-verification
4. Crear AuthController.resendVerification() method
5. Crear AuthService.resendVerificationEmail() con lógica: