Riesgo Transaccional

Documento de Diseño del Servicio de Negocio

Información de Control del Documento

Información del Documento

|  |  |
| --- | --- |
| Identificador del Documento | **DSN-2018-MRT** |
| Nombre del Documento | Documento de Diseño del Servicio de Negocio |
| Nombre del Proyecto | Riesgo Transaccional |
| Autor | David E. Fonseca Sánchez |
| Versión | 0.3 |
| Estatus | Diseño |
| Fecha de Liberación | Marzo 2018 |

Historial del Documento

| Versión | Fecha | Descripción del cambio | Responsable |
| --- | --- | --- | --- |
| 0 | 2016 | Diseño inicial | David E. Fonseca |
| 0.1 | 2016 | Re Diseño | Ejecución |
| 0.2 | 10-03-2018 | Re Diseño | David E. Fonseca |
| 0.3 | 26-03-2018 | Ajuste N Reglas | David E. Fonseca |
| 1.0 |  |  |  |

Historial de Aprobación

| Fecha | Nombre | Responsable | Comentarios |
| --- | --- | --- | --- |
|  | Jesús Héctor Mejía Álvarez | Comité de Arquitectura: Subdirector de Tecnología |  |
|  | Febe Prado Rojo | Comité de Arquitectura: Subdirector de Planeación de la Demanda |  |
|  | Rogelio Briones González | Subdirector de Aplicaciones de Producto |  |
|  | José Luis Durand Baez | Comité de Arquitectura: Gerente de Seguridad |  |
|  | Rafael Carlos Alcantara Cortes | Comité de Arquitectura: Gerente de Infraestructura |  |
|  | Wendy Massiel González Guzmán | Comité de Arquitectura: Gerente de Gobierno TI |  |
|  | David E. Fonseca Sánchez | Comité de Arquitectura: Arquitecto en Jefe |  |
|  | Carlos Alberto González Bejarano | Comité de Arquitectura: Invitado, Sistemas La Paz |  |

Índice

[1 Introducción 4](#_Toc509832761)

[1.1 Propósito 4](#_Toc509832762)

[1.2 Supuestos 4](#_Toc509832763)

[1.2.1 Diseño 4](#_Toc509832764)

[2 Arquitectura de Referencia 5](#_Toc509832765)

[2.1 Arquitectura de negocio 5](#_Toc509832766)

[2.1.1 Motivación 5](#_Toc509832767)

[2.1.2 Modelo y servicios de negocio esperados 5](#_Toc509832768)

[2.1.3 Procesos y objetos de negocio que soportan el modelo operativo actual 5](#_Toc509832769)

[2.2 Arquitectura de servicio 5](#_Toc509832770)

[2.2.1 Arquitectura de aplicaciones de referencia 5](#_Toc509832771)

[3 Diseño de Servicio 7](#_Toc509832772)

[3.1 Alcance actual 7](#_Toc509832773)

[3.2 Arquitectura general de datos 9](#_Toc509832774)

[3.3 Arquitectura general de tecnología 11](#_Toc509832775)

[3.4 Requerimiento DF01.01: Alerta 11](#_Toc509832776)

[3.4.1 Requerimientos Funcionales 11](#_Toc509832777)

[3.4.2 Requerimientos No Funcionales 11](#_Toc509832778)

[3.4.3 Descripción narrativa 11](#_Toc509832779)

[4 Estándares 16](#_Toc509832780)

[5 Requerimientos no funcionales 16](#_Toc509832781)

[5.1 Regulatorios 17](#_Toc509832782)

[6 Temas abiertos 18](#_Toc509832783)

[6.1 Operación 18](#_Toc509832784)

[7 Anexos 19](#_Toc509832785)

# Introducción

## Propósito

El documento de arquitectura presenta los requerimientos de arquitectura a cumplir por las áreas de desarrollo durante los ciclos de desarrollo y que serán revisados durante las validaciones de arquitectura para medir el cumplimiento de la arquitectura definida.

El contenido de este documento es un criterio de aceptación para los desarrolladores al final de cada ciclo de desarrollo.

## Supuestos

### Diseño

1. Se ajusta a los requerimientos bajo el entendido verbal de distintas revisiones realizadas.

# Arquitectura de Referencia

## Arquitectura de negocio

### Motivación

Este proyecto tiene el único propósito de conocer mejor al cliente y poder reducir las operaciones riesgosas.

### Modelo y servicios de negocio esperados

Hasta el momento se cuenta con una versión liberada de la plataforma. Sin embargo, la generación automática de casos no ha cumplido con la expectativa del usuario por 2 principales razones, 1) flexibilidad de la regla y selección de información, 2) capacidad de agregar nuevas reglas.

Por tal motivo la plataforma deberá ser capaz de generar esquemas flexibles de análisis que sin ser infinitos, si provean de una amplia gama de combinaciones.

Técnicamente, el componente ha degradado en general el desempeño de distintos componentes críticos, por lo que se espera que esta funcionalidad genere la menor cantidad de estrés sobre los distintos componentes, teniendo la posibilidad de crecer. Es importante considerar que la activación de un alto número de reglas por transacción puede generar un alto tiempo de respuesta o consumo de recursos.

1. Identificación de Transacción Riesgosa. Considerando el conjunto de datos establecido y las posibles combinaciones de consultas se deberá poder generar una alerta cuando se cumpla alguna regla. Se tendrán tantas alertas como reglas se cumplan.

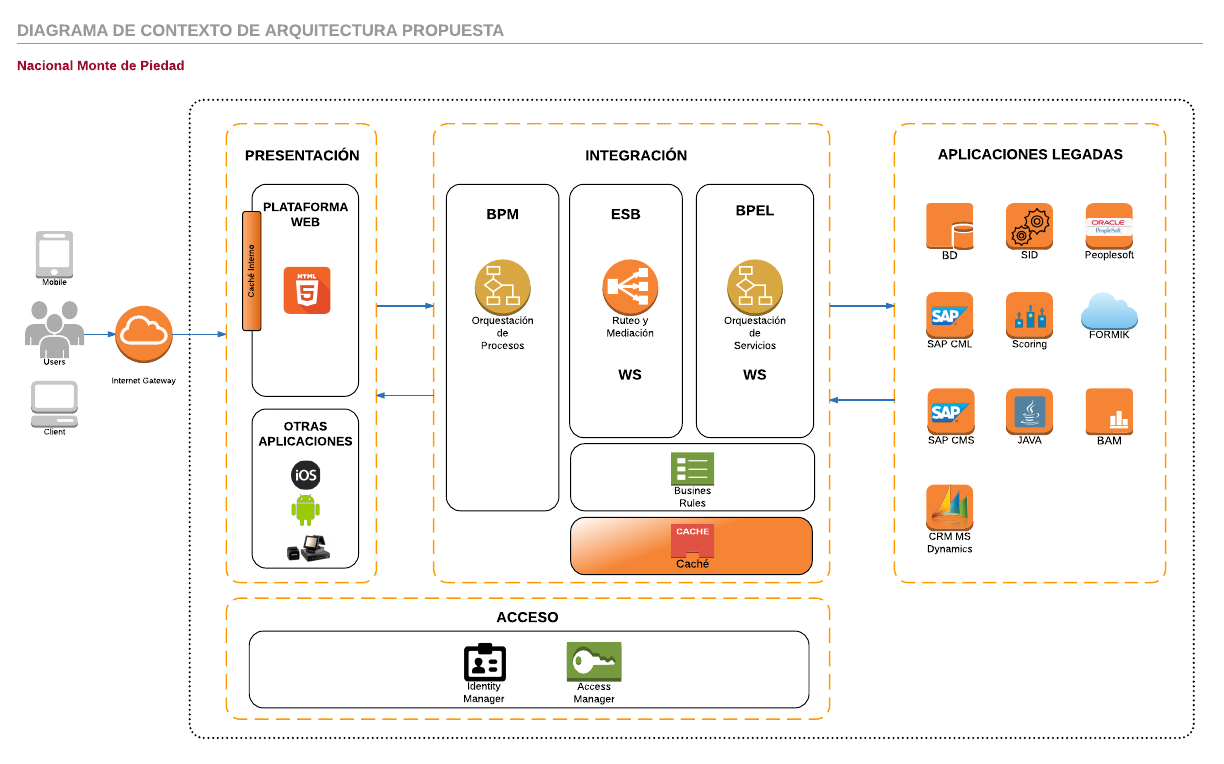
### Procesos y objetos de negocio que soportan el modelo operativo actual

Actualmente no se tiene documentado dentro del proyecto el modelo operativo. Por tal motivo, se desconocen los procesos, políticas y procedimientos asociados.

## Arquitectura de servicio

### Arquitectura de aplicaciones de referencia

A continuación, se presenta el diagrama transversal de componentes aplicativos que se encuentran dentro de la arquitectura de referencia (independiente a la plataforma y/o a la seguridad perimetral de la red).



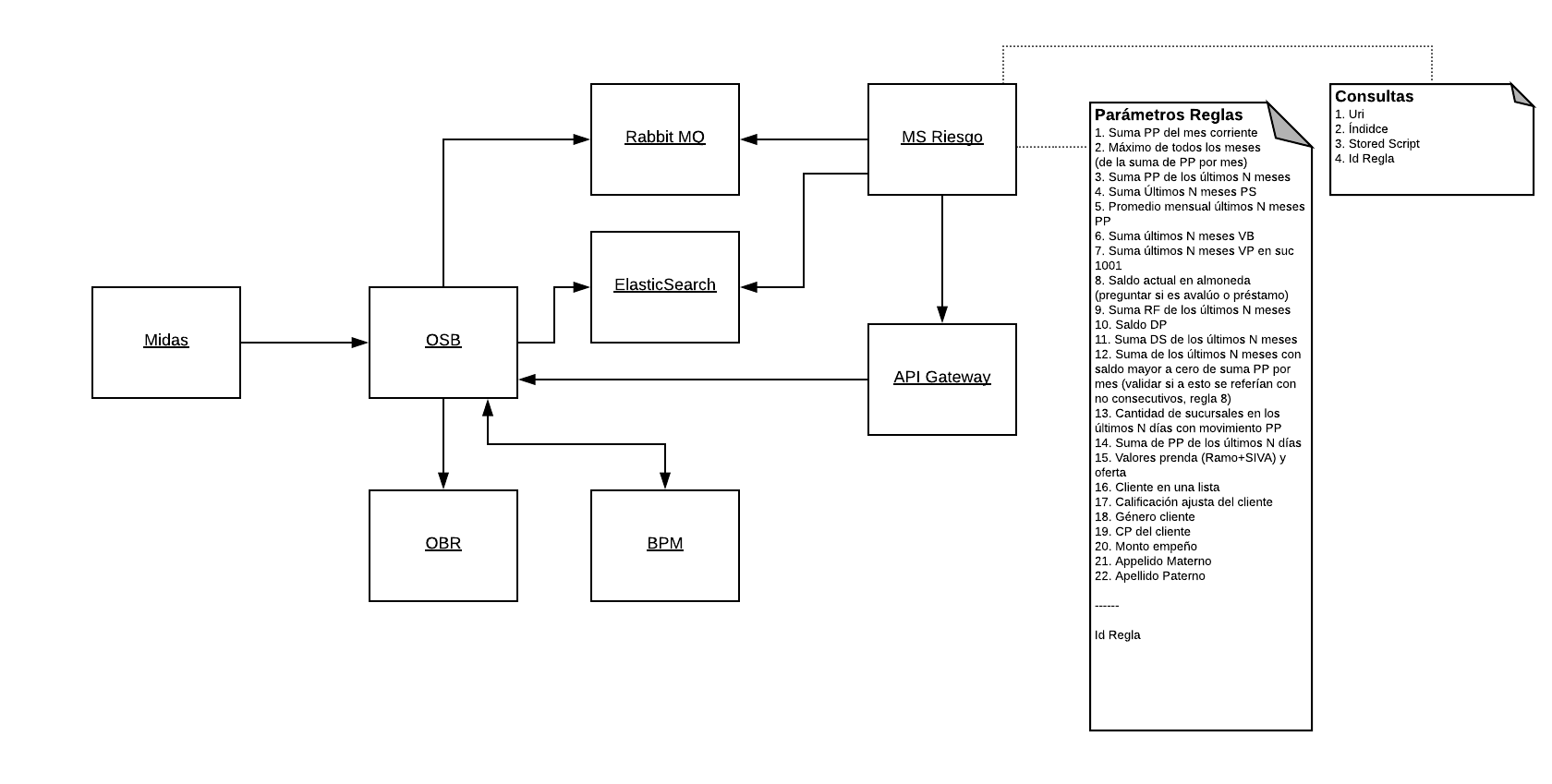
1. Diagrama de contexto de arquitectura aplicativa de referencia

# Diseño de Servicio

## Alcance actual

El Arquitectura general de aplicaciones

Los componentes aplicativos considerados para proveer los servicios de negocio requeridos en este proyecto son los siguientes:



2**. Componentes aplicativos considerados**

**MS Riesgo:** Servicio de back-end bajo el estilo de Microservicios, su propósito es desacoplar la complejidad y elasticidad de las reglas o consultas.

Debido a que la evaluación de una regla requiere variables nombradas, la flexibilidad de esta funcionalidad está dada por los contratos de servicio que proveen la información a la regla y de la mantenibilidad de las reglas en proporción a la cantidad de variables del modelo.

Para desacoplar dicha complejidad este microservicio deberá tener un modelo que, para cada una de las variables indique si es necesaria para la regla y que periodicidad es necesaria.



En este caso, para la regla “R1-AA” se considera que requiere:

* Suma del empeño del mes corriente
* Máximo mensual del empeño por mes
* **No incluye**: Suma de empeño de los últimos N meses (N=0)
* Suma de refrendos de los últimos **5** meses (N=5)
* Promedio mensual de los últimos **5** meses de empeño (N=5)
* Suma de los últimos **5** meses de Venta con Billete (N=5)
* Suma de los últimos **3** meses de Venta al Público en sucursal 1001 (N=3)

En caso de que la consulta se la misma para todas las consultas, solo se deberá consultar 1 vez. Para ello, se requiere un mapeo del tipo:

|  |  |  |
| --- | --- | --- |
| Variable | Plantilla | URL |
| Suma PP del mes corriente | {JSON @varX=###} | [https://URL](https://url/) |
| Máximo de todos los meses (de la suma de PP por mes) | {JSON @varX=###} | [https://URL](https://url/) |
| Suma PP de los últimos N meses | {JSON @varX=###} | [https://URL](https://url/) |
| Suma Últimos N meses PS | {JSON @varX=###} | [https://URL](https://url/) |
| Promedio mensual últimos N meses PP | {JSON @varX=###} | [https://URL](https://url/) |

En general la variable que se debe de enviar es el número de cliente, pero se deberá poder usar cualquier variable que el OSB envíe en la transacción.

En este componente se pretende soportar reglas que no necesariamente pasen por el OSB, para ello se define la estructura siguiente:

|  |  |  |
| --- | --- | --- |
| Regla | JSON | URL |
| RELK-A01 | {JSON @varX=##} | [https://URL](https://url/) |
| RELK-A02 | {JSON @varX=##} | [https://URL](https://url/) |

Donde, para cada regla se consumirá el JSON con las variables definidas. En caso de que devuelva algún hit, agregado o variable, deberá considerarse como válido.

Es importante considerar que deberá existir una correspondencia de los identificadores de reglas, con el objetivo de mantener consistencia.

En general, se recomienda usar las reglas por selección de elastic vs reglas intensivas de negocio por variable.

Se propone un límite de no más de 5 reglas por transacción.

Para la construcción de este MS se deberá considerar lo siguiente:

* Lenguaje: Java
* Frameworks: SpringBoot, Cloudfoundry
* DBMS: MongoDB
* Patrones: Reactive
* Seguridad: Bluemix MS

## Arquitectura general de datos

En general, las transacciones se deberán almacenar en elasticsearch, considerándose la siguiente definición (en lugar de considerar estructura padre-hijo) considerar la nueva definición de join fields (ajustar documento acorde) <https://www.elastic.co/guide/en/elasticsearch/reference/current/parent-join.html>:

Considerar que para inicios del proyecto, no necesariamente se tendrán todos los datos, pero se considera la evolución final estimada de la estructura.

{

"type":"movimiento",

"\_parent":"106269443",

"idCliente": "666666",

"idContrato": "106269443",

"origenMovimiento":"BAM",

"tipoMovimiento":"CAJA",

"fechaMovimiento": "2017-12-30T12:47:07-0600",

"fechaAplicacion":"2017-12-30T12:47:07-0600",

"fechaConciliacion": "2017-12-30T12:47:07-0600",

"estatus":"APLICADO",

"nivelRiesgoMovimiento":"01-NORMAL",

"scoreRiesgoMovimiento":80,

"claveMovimiento": "PP",

"descripcionMovimiento":"Empeño",

"nombreMovimiento":"Pago de Préstamo",

"sucursalActual": 0,

"sucursal": 0,

"geocodificacionSucursal": null,

"importe": 600.06,

"medioPago":

{

"efectivo":100.01,

"tarjetaCrédito":100.01,

"SPEI": 100.01,

"pagoEnLinea":100.01,

"OXXO": 100.01,

"Banamex":100.01

},

"empleado":"gperezg",

"RiesgoTransaccional" :

[{

"idRegla":"ROBR-01",

"nivelRiesgo":"01-NORMAL",

"scoreRiesgo":80,

"idCaso":"JKLM98"

}]

}

{

"type":"transaccion",

"idContrato":"106269443",

"idTransaccion":"456783-4444-444",

"idCliente":"666666",

"empleado":"abernald",

"montoCredito": 600,

"fechaSolicitud":"2017-12-30T12:47:07-0600",

"fechaDesembolso": "2017-12-30T12:47:07-0600",

"sucursal": 0,

"datosGarantia": {

"ramo":"AL",

"subRamo":"AL",

"idGarantia":"4567"

},

"datosCredito":{

"tasaMensual":0.04,

"tasaAnual":0.48,

"plazoMeses":5,

"frecuenciaDias":null,

"frecuenciaMeses":5,

"refrendos":0,

"idContratoAnterior":"106269443"

},

"estautusCredito":{

"estatusCaja":"PP",

"estatusGarantia":"VA",

"ultimaFechaActualizacion":"2017-12-30T12:47:07-0600",

"HistoriaEstatus":

[{

"estatusCajaAnterior":"",

"estatusGarantiaAnterior":""

}]

},

"RiesgoTransaccional" :

[{

"idRegla":"ROBR-01",

"nivelRiesgo":"01-NORMAL",

"scoreRiesgo":80,

"idCaso":"JKLM98"

}]

}

{

"type":"Caso",

"idCaso":"JKLM98",

"estatus":"ACTIVO",

"idCliente": "666666",

"idContrato": "106269443",

"idRegla":"ROBR-01",

"fechaCreacionCaso":"2017-12-30T12:47:07-0600",

"ultimaFechaActualizacion":"2017-12-30T12:47:07-0600"

}

## Arquitectura general de tecnología

A continuación se presenta los componentes tecnológicos que darán soporte a los requerimientos establecidos:

3 Componentes tecnológicos considerados

Dentro de los componentes a considerar, es necesario:

1. Generar 4 ambientes en Bluemix
   1. Cloud Foundry – Java – Liberty
   2. 1 ambiente por cada espacio: dev, test, pre, prod
   3. Memoria: 1 GB máximo
   4. Servicios Operativos
      1. Monitoreo
      2. Autoescalamiento
      3. Toolchain
   5. Servicios de aplicación
      1. Mongo DB
2. Generar 4 ambientes en RabbitMQ
   1. 1 ambiente por cada espacio: dev, test, pre, prod
   2. 1 cola para, N productores y hasta 3 consumidores con hasta 5 canales cada uno.
3. Generar 4 índices de elasticsearch
   1. Reutilizar dev, test uat
   2. Generar uno producción dedicado
4. SQL Integration Services para carga de información

## Requerimiento DF01.01: Alerta

Debido a falta de tiempo, los requerimientos se resumirán en este punto

### Requerimientos Funcionales

TBD.

### Requerimientos No Funcionales

* No se especifican

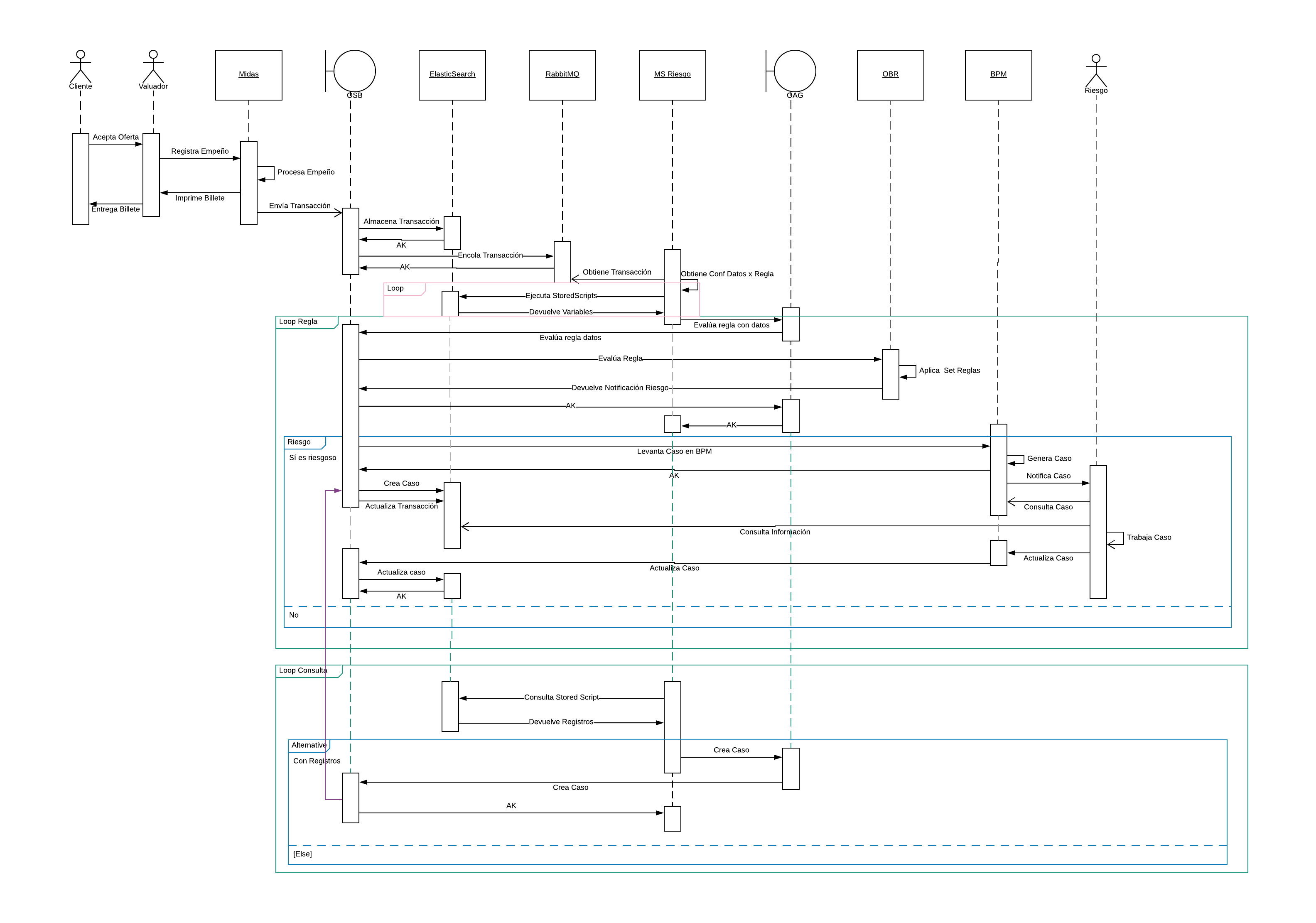
### Descripción narrativa

Para satisfacer el requerimiento se llevará a cabo la construcción o modificación de los siguientes Servicios de Negocio:

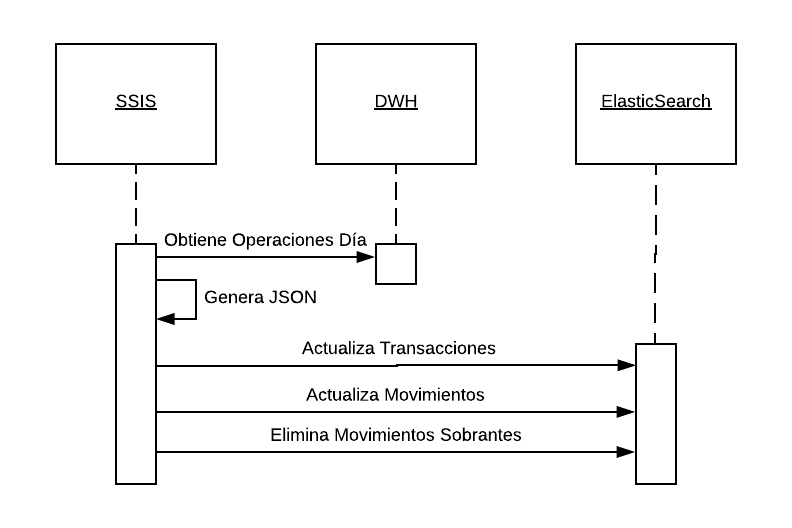
1. Gestión de alertas automáticas.

Este Servicio de Negocio será soportado mediante los componentes aplicativos antes mencionados:

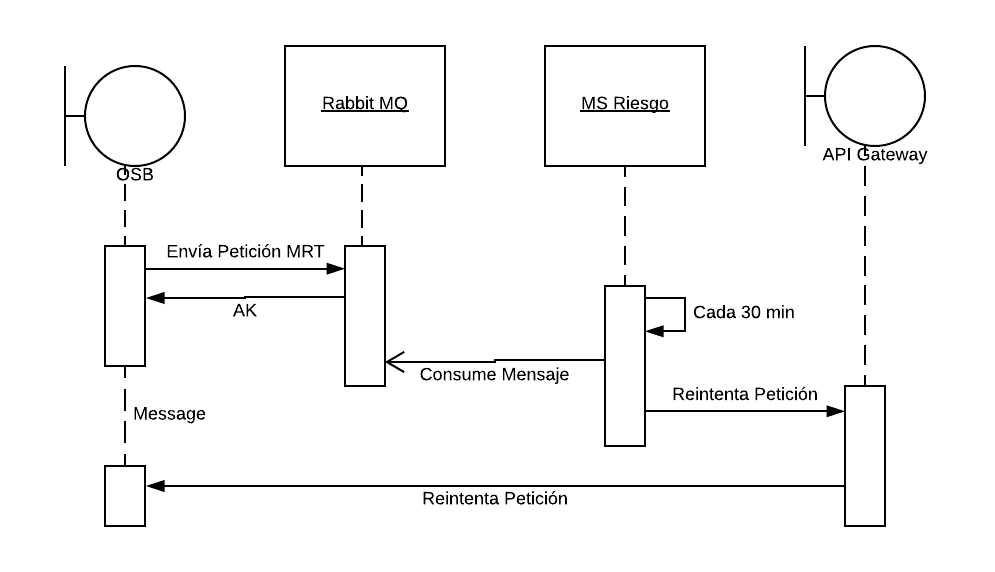
Desde una perspectiva funcional, los siguientes diagramas de secuencias ejemplifican paso a paso la interacción de cada uno de los componentes.



Todos los días, se deberá actualizar la información generada en tiempo real y se deberá conciliar en caso de discrepancia.



En caso de falla, se deberá consumir un servicio de mensajería específico. Cuando se encole el mensaje, se deberá almacenar el paso en que fallo, para que el OSB pueda retomar desde el punto adecuado.



# Estándares

| Capa aplicativa | Aplicación | Estándar Aplicable | Herramienta de Desarrollo |
| --- | --- | --- | --- |
| Tecnología | sFTP | Titan Server | Titan Server |
| Tecnología | Cliente sFTP | WebDrive | WebDrive |
| Datos / JSON | MongoDB | <https://json-ld.org/spec/latest/json-ld-api-best-practices/>  <https://www.mongodb.com/blog/post/mongodb-atlas-best-practices-part-1>  http://spring.io/guides/tutorials/bookmarks/  https://spring.io/guides/gs/accessing-mongodb-data-rest/  https://spring.io/guides/gs/rest-service/  https://spring.io/guides/gs/rest-hateoas/ |  |

# Requerimientos no funcionales

Los requerimientos no funcionales aplicables a esta solución se encuentran descritos a continuación. A pesar de que no existen definidos ni en el documento de alcance ni en los diseños funcionales actuales, se considera lo siguiente:

1. Performance de la aplicación:
   1. Mejor esfuerzo
2. Disponibilidad
   1. Garantías
      1. 24x7 con un nivel de disponibilidad del 95%
   2. Reportes
      1. Lunes a viernes 8-18 hrs en elasticsearch
3. Volumetría estimada
   1. Carga inicial
      1. 7 millones de partidas vigentes
      2. 22 millones de partidas vencidas
      3. 10 movimientos promedio por partida

Total: 300 millones de registros

* 1. Carga en casi tiempo real
     1. 200 mil empeños diarios
     2. 2 operaciones promedio por empeño
     3. 120 mil operaciones de caja

Total de actualizaciones: 520 mil operaciones por día

* 1. Reglas activas
     1. 5 reglas activas
     2. Cada regla requiere 2 consultas promedio a elasticsearch

Total de consultas por transacción:10 por movimiento

* 1. Total de consultas a elasticsearch
     1. 520 mil por actualización
     2. 2 millones por consultas
     3. Total máximo esperado: 2.5 millones de transacciones al día.
  2. Desempeño esperado elasticsearch
     1. 70 consultas por segundo
     2. Soporte estimado con 4 nodos productivos con 16 GB de RAM

1. Seguridad
   1. Basada en IAM o Directorio Activo
   2. Servicios Bluemix

## Regulatorios

No

# Temas abiertos

## Operación

TBD

# Anexos

Este documento no tiene anexos.