

Bogotá D.C. 03 de marzo de 2021

Señores,

**SERVICIOS POSTALES NACIONALES S.A**

**Jenny Téllez**

Profesional Experto Nivel II

Jenny.tellez@4-72.com.co

Dirección Nacional de IT

3178123965

4722005 Ext. 1055

Dig. 25G N° 95A - 55. Bogotá, Colombia

**SOLICITUD DE COTIZACIONES PARA EFECTUAR ESTUDIO DE MERCADO**

**1. OBJETO A CONTRATAR O BIEN A CONTRATAR**

Contratar el servicio remoto de soporte y mantenimiento preventivo y correctivo del aplicativo Sherlock para la mitigación de riesgos de los procesos de servicios Postales Nacionales S.A.

**2. ESPECIFICACIONES DEL SERVICIO O BIEN A CONTRATAR**

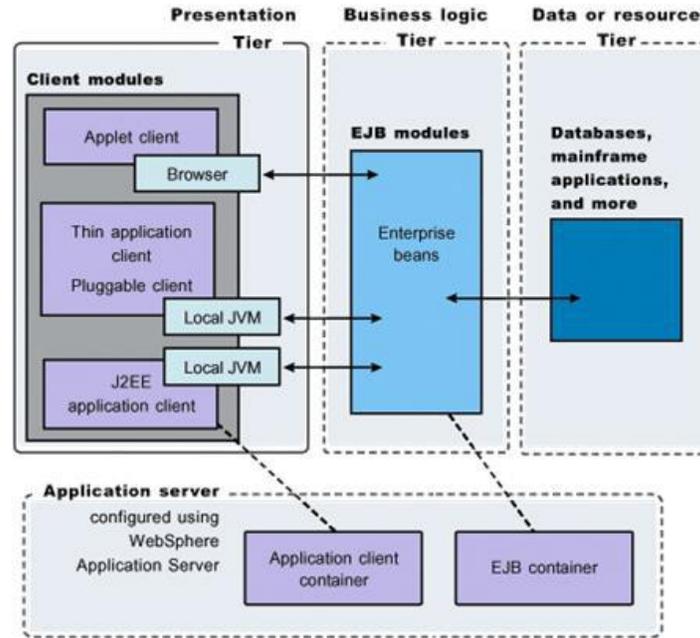
La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas software complejos como pueden ser los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad, integración.

Para solventar estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (aplicación) y por último una interfaz gráfica que facilite al usuario el uso del sistema (presentación).

Si se establece una separación entre la capa de presentación, replicada en cada uno de los entornos de usuario, y la capa modelo, que quedaría centralizada en un servidor de aplicaciones, obtenemos una potente arquitectura que nos otorga algunas ventajas:

- Centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad del modelo.
- No replicación de lógica de negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento.
- Mayor sencillez de los clientes.

Sherlock utiliza una arquitectura de tres capas, que es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de presentación.



La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de algún cambio, sólo se realiza al nivel requerido sin tener que revisar nuevamente todas las capas.

Además, permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer las interfaces que existen entre las capas: capa de presentación, capa de negocio y capa de persistencia.

### Capa de presentación

Es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser amigable, entendible y fácil de usar para el usuario.

Para este caso particular se construirá una capa de presentación totalmente Web.

### Objetivos

- Permitir al usuario a través de la interfaz (en este caso un navegador) interactuar con la aplicación modelada en la capa de negocio. Así mismo permite a la capa de negocio mostrar y controlar el flujo de la información que se presentará a través de esta capa.
- Presentará los resultados de los procesos efectuados
- Tiene la responsabilidad del manejo de validación de datos ingresados, presentación de los resultados, lo que permite mantener un aspecto visual consistente, y manejo de errores. También resuelve la transformación de los datos ingresados en datos del dominio del negocio y la transformación de resultados del negocio a datos presentables al usuario.

### Capa de negocio (aplicación)

Es donde residen los componentes que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del

negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

### Objetivos

- La capa de negocio será la encargada de modelar e implementar todos los casos de uso requeridos y las reglas de negocio definidas para los mismos.
- Se pretende construir una capa de negocio sencilla, que solamente contenga objetos del dominio buscando un mayor grado de simplicidad.
- Un aspecto importante a tener en cuenta es que, si bien la interacción con el cliente principalmente es a través de la capa web, la capa de negocio debe exponer una interfaz que sea capaz de dialogar con la capa de presentación actual y con algún otro módulo futuro, intentando alcanzar un alto grado de escalabilidad y mantenibilidad. Los cambios serán menores y más sencillos si la capa sólo maneja aspectos del negocio.

### Características

- Esta capa hace uso del servicio de persistencia para guardar la información ingresada por el usuario y procesada por el sistema. También para obtener datos para que sean informados al usuario.
- También se hace uso de la capa de presentación para mostrar la información necesaria y para que el cliente pueda interactuar con el sistema. En esta parte, la capa de negocio actúa como proveedora de datos para la capa de presentación.
- Dado que la seguridad es uno de los requerimientos importantes que debe abordar la arquitectura se aplicará una táctica que nos permita poder validar para cada pedido del usuario si el mismo posee los permisos apropiados.

### Capa de persistencia (base de datos)

Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

**Objetivo:** Proveer del soporte necesario para poder almacenar la información manejada por la aplicación y recuperarla para su procesamiento durante la vida útil de la aplicación. Para poder proveer dicho soporte esta capa debe tener una interfaz apropiada con la capa del dominio posibilitando el fácil acceso a los datos que pueden estar almacenados en distintos tipos de repositorios.

**Características:** Para poder cumplir con el objetivo de esta capa, es necesario definir cuatro (4) aspectos fundamentales:

- Medio físico donde se almacenará la información
- Forma de comunicación entre la aplicación y el medio físico de almacenamiento
- Tecnología de conversión/adaptación de los datos al ambiente de objetos
- Interfaz para el acceso a dicha capa desde el dominio de la aplicación

Enunciados los puntos anteriores pasaremos a explicar la estrategia de solución adoptada para la resolución de los mismos:

- Como medio físico, para la arquitectura descrita, se propone el uso de un motor de base de datos relacional, específicamente SQL Server 2012.

A continuación, se describe algunos de los módulos que se utilizan en la aplicación:

#### **Módulo de riesgos:**

- Construye matrices de riesgo inherente y residual
- Administra en forma independiente los sistemas de riesgo que definan los usuarios.
- Flexibilidad en la tipificación de riesgos y parametrización de matrices de riesgo.
- Diseño y monitoreo de riesgos a través de indicadores descriptivos y prospectivos.
- Generar planes de acción de riesgos como esquema de monitoreo de gestión.
- Documentación de eventos de riesgo operativo.
- Valoraciones de controles internos.

#### **Gestión de monitoreo Sarlaft:**

- Diseña y ejecuta modelos de segmentación de factores de riesgos (clientes, productos, canales de distribución y jurisdicciones).
- Genera señales de alerta automáticas.
- Gestiona la consulta a listas restrictivas nacionales e internacionales.
- Desarrolla procesos de gestión para operaciones inusuales y/o sospechosas.
- Mantiene logs de gestión con análisis de los oficiales de cumplimiento.
- Permite generar planes de acción y monitorear el cumplimiento de los mismos.
- Genera y administra la información de formularios de conocimiento al cliente.
- Genera estadísticas de inusualidades transaccionales y asociados a la información que suministran los clientes.
- Controla el inventario de formulario de conocimiento al cliente.
- Facilita los procesos de actualización de datos.
- Archiva imágenes de grabaciones asociadas a proceso de conocimiento del cliente.

#### **GESTION DE PROCESOS**

- Permite documentar procedimientos considerando las estructuras de los sistemas de gestión de calidad (ISO 9001)
- Administra control de cambio a procesos.
- Estructura de mapa de procesos. Vincula a los responsables de actividades y procedimientos en general.
- Administra en forma sistemática y metodología los manuales de procesos y procedimientos de las compañías.

A continuación la ficha técnica de la aplicación.

I. DATOS GENERALES			
Nombre de la Aplicación	Aplicación Sherlock WS		
Descripción	Sistema de información para monitoreo transaccional de los canales aliados.		
Servidor	SQH0076	Sistema Operativo	WIN SERVER 2012
URL de Acceso (Intranet/Internet)	http://172.18.166.89:8687/Formularios/Sitio/Login.aspx.		
Fecha Creación Documento	20-JUL-2016		
Fecha última Actualización	03-03_2020		
Contingencia	SI		
II. REFERENTE FUNCIONAL (Dependencia Usuaría)			
Nombre:	Luz Dary Soto		
Dependencia:	Riesgos y cumplimiento		
E-mail:	<a href="mailto:luz.soto@4-72.com.co">luz.soto@4-72.com.co</a>	Teléfono - Ext.	1089
III. REFERENTE TECNICO (Responsables)			
A. SOPORTE NIVEL 2			
Nombre:	Alan Beltrán		
E-mail:	<a href="mailto:alan.beltran@4-72.com.co">alan.beltran@4-72.com.co</a>	Teléfono - Ext.:	1089
B. ADMINISTRACION DE LA APLICACIÓN			
Nombre:	Sandra Rodriguez		
E-mail:	<a href="mailto:Sandra.rodriquez@4-72.com.co">Sandra.rodriquez@4-72.com.co</a>	Teléfono - Ext.:	1973
IV. DATOS BÁSICOS APLICACIÓN			
<<Para seleccionar la opción adecuada: Doble clic en el cuadro sombreado, seleccione la opción "Activada" >>			
Tipo de aplicación:	<input checked="" type="checkbox"/> WEB <input type="checkbox"/> Cliente/Servidor		
Servidor Web:	<input type="checkbox"/> Tomcat <input type="checkbox"/> Apache <input checked="" type="checkbox"/> IIS <input type="checkbox"/> Glassfish	Versión:	7.5.7600.16385

I. DATOS GENERALES							
Alojamiento	<input type="checkbox"/> Colocation <input checked="" type="checkbox"/> Hosting <input type="checkbox"/> En la nube						
Código ejecutado en el cliente:	<input type="checkbox"/> JAVA <input type="checkbox"/> JAVASCRIPT <input type="checkbox"/> ACTION SCRIPT <input type="checkbox"/> ACTIVE X <input type="checkbox"/> HTML5 <input type="checkbox"/> Otro    ¿Cuál? _____						
Código ejecutado en el servidor:	<input type="checkbox"/> VB .Net <input type="checkbox"/> PL / SQL <input type="checkbox"/> PHP    Cuál _____ <input type="checkbox"/> C# .Net <input type="checkbox"/> JAVA <input type="checkbox"/> XML <input type="checkbox"/> ASP <input type="checkbox"/> Otro    _____						
¿Requiere base de datos?	<input checked="" type="checkbox"/> SI <input type="checkbox"/> NO	<b>Cuál:</b> <input checked="" type="checkbox"/> SQL Server <input type="checkbox"/> PostgreSQL <input type="checkbox"/> Otro <input type="checkbox"/> Oracle <input type="checkbox"/> MySQL    Cuál _____					
¿Requiere acceso a BI?	<input type="checkbox"/> SI <input checked="" type="checkbox"/> NO	<input type="checkbox"/> Analysis Services <input type="checkbox"/> Otro <input type="checkbox"/> Integration Services    Cuál: _____					
¿Utiliza Manejador de Reportes	<input type="checkbox"/> SI <input checked="" type="checkbox"/> NO	<b>Cuál:</b> <input type="checkbox"/> Reporting Services <input type="checkbox"/> Cristal Reports					
¿Interactúa con otra aplicación?	<input type="checkbox"/> SI <input checked="" type="checkbox"/> NO Cuál? __ Multipay/Aliados _____						
Nombre de interfaces	<table border="1" style="width: 100%;"> <thead> <tr> <th>Nombre aplicación</th> <th>Nombre interfaz</th> </tr> </thead> <tbody> <tr> <td>Multipay/canales aliados</td> <td>WS_472</td> </tr> </tbody> </table>			Nombre aplicación	Nombre interfaz	Multipay/canales aliados	WS_472
Nombre aplicación	Nombre interfaz						
Multipay/canales aliados	WS_472						
Detalles: Afectación que tiene este proceso al realizarse transacciones de giros nacionales por multipay y/o aliados.							
Tipo de interface: Web Service							

Descripción	Unidad de Medida	Cantidad	RISK CONSULTING COLOMBIA S.A.S.	
			Valor Unitario	Valor Total
Servicio remoto de soporte y mantenimiento preventivo y correctivo del aplicativo Sherlock.	Mes	12	2.188.246,28	26.258.955,32
<b>SUBTOTAL</b>				26.258.955,32
<b>IVA</b>				4.989.201,51
<b>TOTAL</b>				31.248.156,83

### 3. TÉRMINO DE EJECUCIÓN

El término de ejecución será de Doce (12) meses, contados a partir del cumplimiento de los requisitos de perfeccionamiento, ejecución de la orden o hasta el agotamiento de la partida presupuestal asignada.

### 4. LUGAR DE EJECUCIÓN

El lugar de ejecución del contrato será la Ciudad de Bogotá, aunque por la naturaleza del contrato los servicios por internet cubren todo el territorio nacional.

### 5. VIGENCIA DE LA OFERTA

La vigencia de la presente oferta corresponde a 90 días calendario.

Cordialmente,




---

**LUIS CARLOS ORTEGA G.**

Gerente General

**RISK CONSULTING COLOMBIA S.A.S.**

NIT. 830.110.141-3

Bogotá, D.C., Colombia

Teléfono: 57 (1) 794 2330 – 3004887103

Email: l.ortega@riskgc.com