



UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

**“SISTEMA BASADO EN UNA ARQUITECTURA DISTRIBUIDA CON
BASE DE DATOS NOSQL PARA LA RECUPERACIÓN DE
INFORMACIÓN DE MÚLTIPLES BIBLIOTECAS CON ÓPTIMOS
NIVELES DE RENDIMIENTO”**

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

AUTOR:

Br. OSORIO PÉREZ, CARLOS ALFREDO

ASESOR

Mg. CÁRDENAS ESCALANTE, LAIN JARDIEL

LÍNEA DE INVESTIGACIÓN

SISTEMAS DE INFORMACIÓN TRANSACCIONALES

Trujillo – Perú

2016

PÁGINA DEL JURADO

El presidente y los miembros del Jurado Evaluador designado por la escuela de ingeniería de sistemas.

APRUEBAN

La tesis denominada:

“SISTEMA BASADO EN UNA ARQUITECTURA DISTRIBUIDA CON BASE DE DATOS NOSQL PARA LA RECUPERACIÓN DE INFORMACIÓN DE MÚLTIPLES BIBLIOTECAS CON ÓPTIMOS NIVELES DE RENDIMIENTO”

Presentado por:

Br. Osorio Pérez, Carlos Alfredo

Aprobado por:

Dr. Pacheco Torres, Juan Francisco

Mg. Urquiza Gómez, Yosip Vladimir

Mg. Cárdenas Escalante, Laín Jardiel

DEDICATORIA

El presente trabajo lo dedico a DIOS
porque gracias a él todo lo puedo que
me fortalece.

A mis PADRES por el esfuerzo y apoyo
que me brindan siempre.

Osorio Pérez, Carlos Alfredo

AGRADECIMIENTO

A DIOS, por darme el conocimiento y la capacidad suficiente para poder realizar este proyecto y porque siempre está delante de mí.

A mis PADRES y FAMILIARES por el apoyo y los ánimos que me siempre me brindan.

A la Universidad Cesar Vallejo, por apoyarme en el proceso de enseñanza las cuales me sirvieron para poder concluir con éxito mi formación profesional.

Al Ing. Cárdenas Escalante Laín, por el esfuerzo y paciencia que me dedico para poder concluir esta investigación y por la amistad que me brindo.

Osorio Pérez, Carlos Alfredo

DECLARACIÓN DE AUTENTICIDAD

Yo Osorio Pérez, Carlos Alfredo con DNI N° 72396372, a efecto de cumplir con las disposiciones vigentes consideradas en el reglamento de Grados y Título de la Universidad César Vallejo, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas, declaro bajo juramento que toda la documentación que acompaño es veraz y auténtica.

Así mismo, declaro también bajo juramento que todos los datos e información que se presenta en la presente tesis son auténticos y veraces.

En tal sentido asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada por lo cual me someto a lo dispuesto en las normas académicas de la Universidad César Vallejo.

Trujillo, diciembre del 2016

Osorio Pérez, Carlos Alfredo

PRESENTACIÓN

Señores Miembros del jurado:

En el cumplimiento del Reglamento de Grados y Títulos de la Universidad César Vallejo presento ante ustedes la Tesis titulada: **“SISTEMA BASADO EN UNA ARQUITECTURA DISTRIBUIDA CON BASE DE DATOS NOSQL PARA LA RECUPERACION DE INFORMACION DE MULTIPLES BIBLIOTECAS CON OPTIMOS NIVELES DE RENDIMIENTO”**. La misma que someto a vuestra consideración y espero que cumpla con los requisitos de aprobación para obtener el Título Profesional de Ingeniero de Sistemas.

Osorio Pérez, Carlos Alfredo

ÍNDICE

PÁGINA DEL JURADO	i
DEDICATORIA	ii
AGRADECIMIENTO	iii
DECLARACIÓN DE AUTENTICIDAD	iv
PRESENTACIÓN	v
ÍNDICE	6
ÍNDICE DE FIGURAS	8
RESUMEN	9
ABSTRACT	10
I. INTRODUCCIÓN	11
1.1 Realidad Problemática	11
1.2 Trabajos Previos	14
1.3 Teorías Relacionadas al Tema	17
1.3.1 FUNCIONALIDAD	17
1.3.2 EFICIENCIA	18
1.3.3 ARQUITECTURA DE SOFTWARE	18
1.3.4 SERVICIO	18
1.3.5 ARQUITECTURA ORIENTADA A SERVICIOS (SOA)	19
1.3.6 SERVICIOS WEB	23
1.3.7 BASE DE DATOS	31
1.3.8 FIREBASE	35
1.4 Formulación del Problema	38
1.5 Justificación del Estudio	38
1.6 Hipótesis	39
1.7 Objetivos	39
1.7.1 Objetivo General	39

1.7.2	Objetivos Específicos.....	39
II.	MÉTODO	40
2.1	Diseño de investigación	40
2.2	Variables y Operacionalización de variables	41
2.3	Población y muestra	43
2.4	Técnicas e instrumentos de recolección de datos, validez y confiabilidad.....	43
2.5	Métodos de análisis de datos	43
III.	RESULTADOS	47
3.1	Métrica de Funcionalidad	47
3.2	Métrica de Eficiencia.....	49
IV.	DISCUSIÓN	55
V.	CONCLUSIÓN	57
VI.	RECOMENDACIONES	58
VII.	REFERENCIAS	59
	ANEXO.....	60

ÍNDICE DE FIGURAS

Figura 1 : Inventario y Catálogo de Servicios.....	20
Figura 2: Tipos de Servicios.....	21
Figura 3: Tipo de Servicio de Utilidad	21
Figura 4: Tipo de Servicio de Utilidad	22
Figura 5: Tipo de Servicio de Tarea.....	22
Figura 6: Formato de Envío de SOAP.....	24
Figura 7: Funcionamiento de Envío de SOAP 1	25
Figura 8: Funcionamiento de Envío de SOAP 2	25
Figura 9: Estructura del Formato JSON	29
Figura 10: Estructura JSON de un Objeto.....	30
Figura 11: Estructura JSON de un Arreglo	30
Figura 12: Tipo de Datos de JSON.....	31
Figura 13: Diseño de Investigación	40
Figura 14: Métrica de la Funcionalidad - Idoneidad	44
Figura 15: Métrica de la Funcionalidad – Interoperabilidad	45
Figura 16: Métrica de la Eficiencia - Comportamiento respecto al Tiempo	46

RESUMEN

La presente investigación denominada “SISTEMA BASADO EN UNA ARQUITECTURA DISTRIBUIDA CON BASE DE DATOS NOSQL PARA LA RECUPERACION DE INFORMACION DE MULTIPLES BIBLIOTECAS CON OPTIMOS NIVELES DE RENDIMIENTO”, tiene como propósito la implementación de una arquitectura la cual estará distribuida utilizando distintos proveedores de servicio para su respectivo despliegue para así permitir la interoperabilidad entre las distintas bibliotecas y asegurar un óptimo nivel de rendimiento en la recuperación de información. Esta implementación partirá desde los documentos de Especificaciones suplementarias y el Documento de arquitectura del software los cuales se definen en un comienzo y que son pieza clave para poder realizar la implementación, cabe resaltar que se utilizara las métricas de calidad que define la (ISO 2196) para validar que dicha implementación se realizó correctamente. Finalmente, en base a las métricas de calidad se obtendrán resultados y se verificaran con el Documento de arquitectura del software para constatar que se realizó todo correctamente.

Palabras clave: servicio web, arquitectura, nosql, interoperabilidad.

ABSTRACT

The present research called "SYSTEM BASED ON A NOSQL DATABASE DISTRIBUTED ARCHITECTURE FOR THE RECOVERY OF MULTIPLE LIBRARY INFORMATION WITH OPTIMAL PERFORMANCE LEVELS", has the purpose of implementing an architecture which will be distributed using different service providers for its Respective deployment to allow interoperability between different libraries and ensure an optimal level of performance in retrieving information. This implementation will start from the Supplementary Specifications documents and the Software Architecture Document, which are defined at the outset and are the key to implementing the implementation. It should be emphasized that the quality metrics defined by ISO 2196 (ISO 2196) To validate that the implementation was successful. Finally, based on the quality metrics will be obtained and verified with the Software Architecture Document to verify that everything was done correctly.

Keywords: web service, Architecture, nosql, interoperability.

I. INTRODUCCIÓN

1.1 Realidad Problemática

Actualmente se genera mucha información en internet, como lo confirma (Akerkar, 2014) en su libro “Big Data Computing” en el cual nos muestra como el mercado internacional, las empresas, los proveedores y los clientes crean y consumen grandes cantidades de información a diario, y a su vez la empresa (Gartner, January, 210) en su informe “Technology Trends You Can’t Afford to Ignore” estima que el crecimiento de datos empresariales crecerán hasta 650% en los próximos cinco años y que los datos serán en un 80% no estructurados.

Como se menciona, la información está creciendo mucho, generando incluso que haya redundancia de información, lo cual en una institución o empresa es un peligro ya que generaría retrasos o desventajas frente a sus competidores ya que los clientes tendrían que esperar mucho más tiempo y volver a repetir procesos ya realizados anteriormente, generando así incomodidad.

Como podemos ver la tendencia en estos tiempos es el incremento de información, es por ello que ahora existen conceptos como Big data, machine learning, base de datos no estructurados, etc. Ya que con esas herramientas la gestión de las grandes cantidades de información facilita su interpretación y la forma en cómo se maneja los datos para que así la recuperación de la información sea entendible y rápida.

Desde un punto de vista más segmentado ya no hablando de toda la información generada diariamente en el mundo, sino específicamente de las revistas y libros académicos que se registran y se comparten a diario, veamos lo que según Alejandra Nardi en su artículo titulado (¿A que se debe el crecimiento exponencial del número de revistas y libros académicos con acceso libre en Internet?) nos muestra que un cierto porcentaje cada vez más significativo de científicos e investigadores alrededor del mundo participan y respaldan un nuevo movimiento llamado Iniciativa de Archivos Abiertos, el cual consiste en que el acceso a la literatura científica no debe de ser restringida, con la finalidad de que

dichos archivos estén al alcance de la comunidad internacional, esto significa, que todos los documentos deben de compartirse para que las personas puedan obtenerlos y en base a ello continuar las investigaciones o que le sirva para futuras investigaciones con la finalidad de ayudar a los nuevos investigadores.

Actualmente el Perú cuenta con La Biblioteca Nacional de Perú y según él (Estado Peruano) fue creado el 28 de agosto de 1821 en Lima por el general José de San Martín y Matorras, quien donó su propia colección de libros. Así mismo el Perú cuenta con el Sistema Nacional de Bibliotecas el cual según él (Peruano, 2013), informa en la Ley número 30034 que dicho sistema se creó como instrumento de gestión pública para el establecimiento de estándares de calidad, eficacia y eficiencia durante la presentación de los servicios brindados a la ciudadanía por las bibliotecas a cargo del estado y que tiene como finalidad la integración técnica de su gestión y el aseguramiento de la calidad en sus servicios de acceso al conocimiento, a la cultura y a la información, propiciando para ello el funcionamiento de bibliotecas organizadas en todo el territorio nacional y a la optimización del uso de sus servicios y recursos bibliográficos, dentro de la política pública de inclusión social, de construcción de la ciudadanía y de desarrollo humano. Como podemos apreciar el Perú también se preocupa de que la población pueda tener acceso a la información y poder compartirla mediante el Sistema Nacional de Bibliotecas.

Por otro parte, como hemos visto es muy importante el contar con un sistema de biblioteca ya que ayuda mucho a las personas. El estado está comprometido en brindar ese servicio, al igual como lo vienen haciendo distintas universidades de la ciudad de Trujillo porque esto genera que las personas estén informadas o puedan realizar o apoyar investigaciones. Pero por otro lado existen problemas en los sistemas de bibliotecas como, por ejemplo:

- La integración de múltiples sistemas de bibliotecas.
- La Facilidad del uso de los sistemas de bibliotecas.

- El tiempo de respuesta que tardan los sistemas de bibliotecas.

Como podemos observar existen problemas que de alguna u otra forma hacen que los sistemas de bibliotecas con los que contamos no sean los suficientemente funcionales ni tampoco que cuenten con la eficiencia para poder acceder a la recuperación de información de múltiples bibliotecas.

1.2 Trabajos Previos

1.2.1 Nivel Internacional

- **Primer Antecedente Internacional**

Título: “Biblioteca Digital Basada en Servicios Web” de (Fabela Cano, 2006) desarrollada en el Instituto Politécnico Nacional.

Resumen: En esta investigación se implementó un conjunto de Servicios Web, los cuales sirvieron como modelo para proporcionar la funcionalidad a una biblioteca digital con la finalidad de habilitar la interoperabilidad de información a través de internet y/o intranet. Con esto se logró ofrecer servicios con algunas ventajas que otras bibliotecas no cuentan tales como: búsquedas basadas en meta datos siguiendo el estándar de Dublín Core así también como la posibilidad de implementar diversas interfaces de usuario de acuerdo al uso particular, entre otras.

Aporte: Esta investigación confirmo que la implementación de servicios web para una biblioteca ayuda de manera positiva ya que le da algunas ventajas antes mencionadas que otras bibliotecas no cuentan.

- **Segundo Antecedente Internacional**

Título: “Implementar el Consumo de Servicios vía Web Services de la Información almacenada en una Base de Datos de una Biblioteca Digital perteneciente a una Institución Educativa de nivel intermedio” de (Brito Moreno, 2011) desarrollada en la Universidad de Guayaquil.

Resumen: En esta investigación se implementó el consumo de servicios web con la finalidad de que la información de las bibliotecas de los colegios de la ciudad de Guayaquil esté disponible las 24 horas del día ya que muchos de los estudiantes suelen realizar otras actividades por lo tanto no hay una hora determinada para realizar sus tareas. Además, permitirá obtener la información necesaria desde cualquier parte y a cualquier hora a través del internet.

Aporte: Esta investigación nos muestra los beneficios de los servicios web los cuales resuelven algunos problemas muy conocidos que son el tiempo y la disponibilidad, ya que la implementación de estos hace que la información esté disponible a cualquier hora y desde cualquier parte del mundo

1.2.2 Nivel Nacional

- **Primer Antecedente Nacional**

Título: “Recuperación de historias clínicas electrónicas a partir de un repositorio digital usando una arquitectura orientada a servicios” de (Sánchez Reyna, 2013) desarrollada en la Pontificia Universidad Católica del Perú.

Resumen: Esta investigación tuvo como objetivo la implementación de un servicio web el cual permitirá el registro y la recuperación de las historias clínicas electrónicas, con lo cual logro dar soluciones a problemas generados por el registro en físico tales como la ilegibilidad, deterioro, pérdida de los registros y la imposibilidad de acceso a las historias clínicas.

Aporte: Esta investigación nos muestra que la implementación de servicios web para el control de las historias clínicas electrónicas, estarán actualizadas constantemente ya que la información estará centralizada e integrada

- **Segundo Antecedente Nacional**

Título: “Implementación de un sistema de integración para las bibliotecas municipales de Lima y Callao utilizando SOA y J2ME” de (Pinedo Marín, y otros, 2010) desarrollada en la Universidad Nacional Mayor de San Marcos.

Resumen: Esta investigación tuvo como objetivo el integrar las bibliotecas municipales de lima y callo utilizando SOA en donde cada biblioteca haga disponible su información a través de servicios web y la integración mediante la plataforma ESB.

Aporte: Esta investigación nos muestra de que no solo utiliza servicios web sino también se base en un paradigma que es la Arquitectura Orientada a Servicios o también conocida como SOA la cual estará compuesta por servicio web y soportado sobre un ESB con lo que permitirá una mejor integración y disponibilidad de los servicios.

1.3 Teorías Relacionadas al Tema

1.3.1 FUNCIONALIDAD

Según la (ISO 2196), define funcionalidad como la capacidad con la que cuenta el software para proporcionar funciones las cuales puedan satisfacer las necesidades que fueron declaradas y diseñadas previamente en los requisitos funcionales. La funcionalidad cuenta con los siguientes atributos:

a) Idoneidad

Se centra básicamente en evaluar si el software cuenta con las funciones indicadas para poder realizar las tareas que fueron definidas en un comienzo.

b) Exactitud

Permite evaluar si el software presenta resultados o efectos acordes a las necesidades para las cuales fue creado.

c) Interoperabilidad

Es la capacidad con la que cuenta el software para poder interactuar y comunicarse con uno o más sistemas específicamente.

d) Cumplimiento de normas

Verifica si el software cuenta con estándares, convenciones o regulaciones las cuales están en leyes y prescripciones similares.

e) Seguridad

Es la habilidad con la que cuenta el software de poder prevenir el acceso no autorizado, ya sea de manera accidental o intencional, a los distintos programas y datos.

1.3.2 EFICIENCIA

Según la (ISO 2196), la eficiencia es la característica la cual nos permite poder medir la relación que existe entre el funcionamiento del software y la cantidad de los recursos utilizados. Por lo cual las características a evaluar son:

a) Tiempo de Respuesta

Es el tiempo en que las funciones definidas en el software tienen a procesar la información.

b) Recursos Utilizados

Es la cantidad de recursos que son utilizados por el software y la duración de los mismos en base a una determinada función.

1.3.3 ARQUITECTURA DE SOFTWARE

Según (Clements, y otros, 2001) definen que una arquitectura de software para un sistema es la estructura o estructuras del sistema, que consisten en elementos, sus propiedades visibles externamente y las relaciones entre ellos.

1.3.4 SERVICIO

Según el ingeniero (Ramirez J., Noviembre 2009) en su libro "Arquitectura Orientada a Servicios" define qué un servicio es un conjunto de actividades, las cuales probablemente están automatizadas y buscan dar respuesta a uno o más necesidades del cliente. Asimismo (Microsoft Corporation, Diciembre, 2008), define como servicio a la funcionalidad de una tarea concreta, en la cual puede ser un proceso de negocio simple como introducir un dato o como también el poder acoplarse dentro de una aplicación más compleja. Y desde un punto tecnológico según (Douglas K.) Especialista en arquitectura empresarial con énfasis en SOA define que los servicios son los que se conectan entre sí mediante servicios web. Además, un servicio esta implementado bajo algún sistema el

cual ofrece una conexión para poder soportar las invocaciones de los servicios.

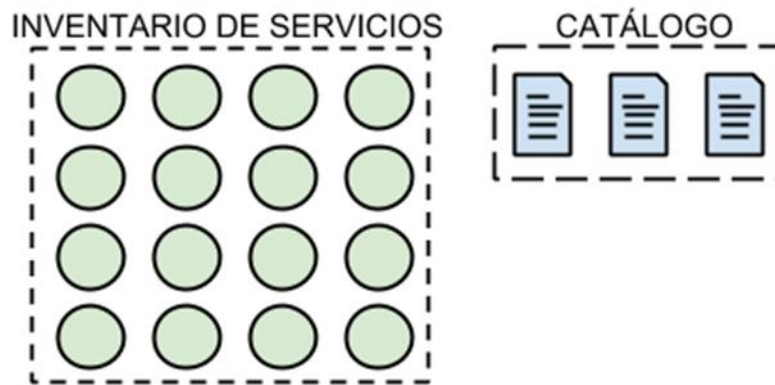
1.3.5 ARQUITECTURA ORIENTADA A SERVICIOS (SOA)

Según (Microsoft Corporation, Diciembre, 2008), define a la arquitectura orientada a servicios también conocida con su abreviatura SOA que en inglés es (Service Oriented Architecture), como un marco de diseño el cual integra aplicaciones independientes, con la finalidad que desde una red se pueda acceder a sus funcionalidades, las cuales se ofrecen como servicios.

SOA o Arquitectura Orientada a Servicios es un modelo de arquitectura tecnológica que surge de la aplicación del paradigma de orientación a servicios. La idea consiste en descomponer la lógica de una organización o partes de ella en pequeñas unidades de funcionalidad. Estas pequeñas unidades son los servicios, con esto lograremos tener una plataforma transversal formada por un inventario de servicios de forma que no resolveremos las necesidades cambiantes del negocio creando nuevas aplicaciones sino combinando diferentes servicios. De esta manera conseguimos que el área de TI y el negocio estén alineados de forma que el área de TI pueda responder de manera ágil a las exigencias del negocio. Esto no quiere decir que tenemos que crear servicios web por crear, sino que debemos tener en cuenta de que estos sean:

- Reutilizables o multipropósito como los servicios de (utilidad y entidad).
- Deben contar con un contrato estandarizado (WSDL-SOAP o convenciones-REST).
- Se deben categorizar y registrar para saber con qué servicios contamos.
- Definir nuestro catálogo de servicios correctamente categorizado es bueno para nuestra SOA.

Figura 1 : Inventario y Catálogo de Servicios



Fuente (Erl, 2008)

También tengamos en cuenta de que los servicios web basados en SOAP y REST son muy buenas opciones a la hora de crear los servicios los cuales conformaran la plataforma y también tratemos de no confundir una SOA simplemente por tener un montón de servicios web de alto nivel ya que eso sería solo integración.

a) Gobernabilidad SOA

La gobernabilidad SOA es la capacidad de gestionar y llevar a cabo los procesos del negocio en sí, y a su vez a quienes lo integran como por ejemplo las personas, teniendo en cuenta el uso adecuado de los recursos que se usaran para la implementación de SOA en la organización. A su vez se definen soluciones, políticas y prácticas los cuales estarán orientados a los servicios y al ciclo de vida para poder realizar algunas mejoras. En pocas palabras se podría decir que es un marco para la administración de los elementos de SOA con respecto a los estándares de la organización.

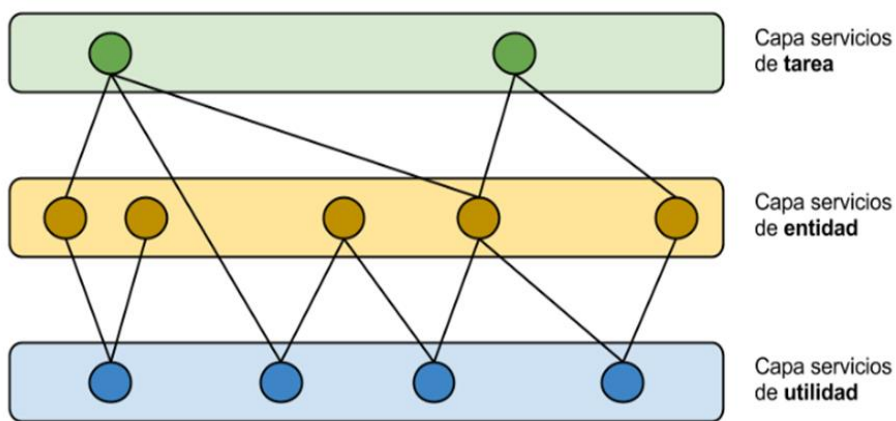
b) Tipos de Servicios

Existen diferentes clasificaciones o modelos de servicios, estas son probablemente las más sencillas y el resto de modelos suelen ser derivados de estas. El principal promotor de estos modelos de servicios es Thomas Erl, quien es actualmente el autor y fundador "Arcitura Education INC". Reconocido en el campo de la Orientación

a Servicios y Cloud Computing, sus nueve libros publicados son muy éxitos y han sido aprobados oficialmente por organizaciones de TI como IBM, Microsoft, Oracle, Intel, entre otras. Los tipos de servicios son:

- Servicios de Utilidad
- Servicios de Entidad
- Servicios de Tarea

Figura 2: Tipos de Servicios

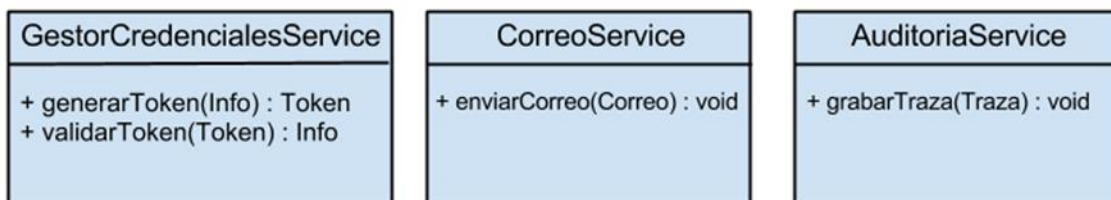


Fuente (Erl, 2008)

- Servicio de Utilidad

Son aquellos los cuales no cubren una necesidad concreta del negocio, sino que tienen una funcionalidad multipropósito en el sentido de que pueden ser utilizados múltiples veces, ósea que son reutilizables lo cual es uno de los principios básicos del diseño de servicios. Por ejemplo

Figura 3: Tipo de Servicio de Utilidad

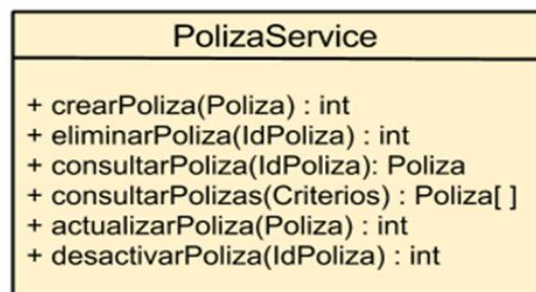


Fuente (Erl, 2008)

- Servicio de Entidad

Son aquellos que están orientados a las entidades del negocio en sí como por ejemplo Cliente, Producto, Compra, Marca, etc. Estas varían dependiendo en negocio, al igual que el servicio de utilidad estos tampoco cubren una necesidad en concreto del negocio por lo que también son reutilizables, muchas de las operaciones que este tipo de servicio tiene son las de un CRUD (Create, Read, Update, Delete).

Figura 4: Tipo de Servicio de Utilidad

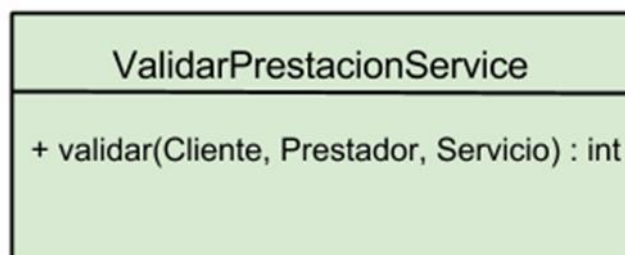


Fuente (Erl, 2008)

- Servicio de Tarea

Son aquellos que engloban un proceso de negocio apoyándose en servicios de más bajo nivel como los servicios de utilidad o entidad. Estos a diferencia del resto son aquellos que realizan una determinada tarea específica, claro está dependiendo de las necesidades del negocio. Como solo suelen realizar una tarea específica no suelen ser muy reutilizables ni muy estables ya que su funcionalidad varía según los cambios del negocio.

Figura 5: Tipo de Servicio de Tarea



Fuente (Erl, 2008)

c) Beneficios

Según (Microsoft Corporation, Diciembre, 2008), dice que los beneficios de la SOA para las organizaciones se dividen en 2 niveles: el usuario corporativo y el nivel de TI de la organización.

A nivel empresarial, SOA permite desarrollar nuevas aplicaciones dinámicas las cuales ayudan a resolver problemas de alto nivel, con la finalidad de generar crecimiento y competitividad en la organización. A su vez SOA entre otras cosas permite.

- Mejorar las decisiones en la organización.
- Mejorar la productividad de los empleados.
- Potenciar la relación entre clientes y proveedores.

A nivel TI, SOA esta orienta a simplificar la creación y el mantenimiento de los sistemas y de las aplicaciones las cuales están integradas. También permite la alineación de los recursos de TI con el modelo de negocio de la empresa y con las necesidades de la misma. A su vez SOA en TI permite:

- Aplicaciones más productivas y flexibles.
- Desarrollo de aplicaciones en un menor tiempo y costo.
- Aplicaciones más seguras y manejables

1.3.6 SERVICIOS WEB

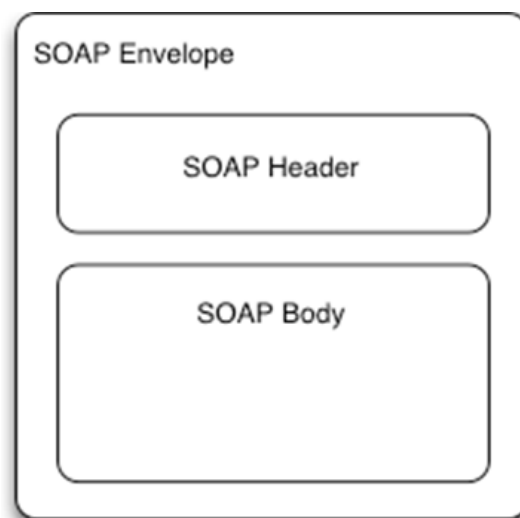
Según (Microsoft Corporation, Diciembre, 2008), menciona que los servicios web son aplicaciones los cuales utilizan estándares con la finalidad de intercambiar información, a su vez permiten la intercomunicación entre sistemas sin importar la plataforma en la cual se esté ejecutando, actualmente se utilizan en distintas empresas para interconectar y mejorar la integración de sus negocios. Estas aplicaciones están basadas en estándares de comunicación como XML y JSON.

Los servicios web fueron creados para ofrecer servicios, los cuales pueden ser consumidos desde cualquier sistema, ya sea que estén implementados en Java, C#, Visual Basic, Php, etc. ósea que no importa el lenguaje de programación con el cual estén implementados ya que el servicio web tiene como finalidad el de integrar diversos sistemas, componentes o plataformas. Y esto es posible ya que se realiza mediante estándares de comunicación generando que esta integración se convierta en interoperabilidad entre los sistemas ya que entre ellos se envían y reciben mensajes.

a) SOAP

SOAP o Simple Object Access Protocol fue el primer intento de estandarizar una interfaz de servicio web. Se basa en el envío de un mensaje XML en un formato específico y recibe una respuesta XML en otro formato específico. Este mensaje puede ser enviado mediante HTTP, FTP, SMTP, etc. La mayoría de implementaciones suelen enviar el mensaje XML mediante HTTP, este mensaje cuenta con una cabecera SOAP y un cuerpo SOAP.

Figura 6: Formato de Envío de SOAP

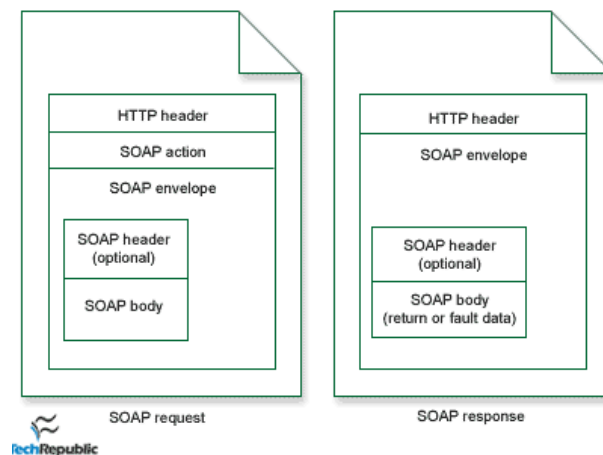


Fuente (Erl, 2008)

- **Funcionamiento de SOAP**

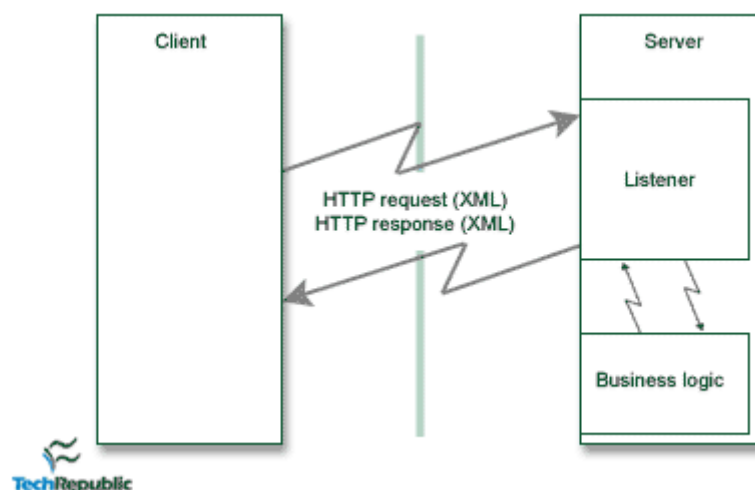
SOAP principalmente utiliza el método solicitud respuesta del protocolo HTTP, el cliente se envuelve en una llamada de método SOAP/XML. La solicitud XML se analiza para leer el nombre del método y parámetros pasados y delegado para su procesamiento, la respuesta XML se envía devuelta al cliente, que contiene el valor o la devolución de datos del error de la llamada al método. Por último, el cliente puede analizar el XML respuesta a hacer uso del valor de retorno.

Figura 7: Funcionamiento de Envío de SOAP 1



El servidor escucha el llamado al proceso SOAP

Figura 8: Funcionamiento de Envío de SOAP 2



SOAP cuenta con una estructura formal, la cual requiere de mensajes bien definidos. Cada servicio web SOAP se une a un contrato a través de la Web Service Definition Language(WSDL) que se adhiere a una definición de esquemas XML (XSD) las cuales permiten la definición de cosas como:

- ¿Qué atributos son opcionales y que atributos se requieren?
- ¿Qué atributos pueden aparecer una vez, más de una vez o nada en absoluto?
- ¿Qué tipo de datos se permite?

Los archivos WSDL y XSD previamente mencionados son de gran valor para los desarrolladores ya que estos archivos definen claramente el acuerdo de interfaz entre el cliente y el servidor. Dado un WSDL tanto el cliente como el servidor pueden generar automáticamente código que soporta el contrato, por lo que un desarrollador al construir un servicio web se centrara en la definición WSDL, esto ahorra mucho tiempo y permite una integración mucho más rápido con otros servicios.

b) REST

REpresentational State Transfer, es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP.

REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es increíblemente más simple y convencional que otras alternativas que se han usado en los últimos diez años como SOAP y XML-RPC.

REST se definió en el 2000 por Roy Fielding, coautor principal también de la especificación HTTP. Podríamos considerar REST como un framework para construir aplicaciones web respetando HTTP.

Por lo tanto, REST es el tipo de arquitectura más natural y estándar para crear APIs para servicios orientados a Internet.

Existen tres niveles de calidad a la hora de aplicar REST en el desarrollo de una aplicación web y más concretamente una API que se recogen en un modelo llamado Richardson Maturity Model en honor al tipo que lo estableció, Leonard Richardson padre de la arquitectura orientada a recursos. Estos niveles son:

- Uso correcto de URIs
- Uso correcto de HTTP.
- Implementar Hypermedia.

Además de estas tres reglas, nunca se debe guardar estado en el servidor, toda la información que se requiere para mostrar la información que se solicita debe estar en la consulta por parte del cliente.

Al no guardar estado, REST nos da mucho juego, ya que podemos escalar mejor sin tener que preocuparnos de temas como el almacenamiento de variables de sesión e incluso, podemos jugar con distintas tecnologías para servir determinadas partes o recursos de una misma API.

c) Formato de Intercambio de Datos

- XML

Es un lenguaje de marcado el cual por su sigla en inglés (Extensible Markup Language) significa Lenguaje de marcas Extensible. La manera de codificar en este tipo de lenguaje es muy simple y legible por el ser humano al igual que JSON, claro está que en XML trabaja los atributos como etiquetas algo parecido a HTML ósea que se debe de inicializar con una etiqueta de apertura y una de cierre, Así como el ejemplo.

Figura 10: Estructura del formato XML

```
<Persona>
  <codigo>1</codigo>
  <nombre>Carlos Alfredo</nombre>
  <apellido>Osorio Pérez</apellido>
  <email>hacybeyker@gmail.com</email>
  <tecnologias>
    <nombre>Java</nombre>
    <nombre>C#</nombre>
    <nombre>Php</nombre>
  </tecnologias>
  <edad>21</edad>
  <genero>M</genero>
  <estudiante>true</estudiante>
</Persona>
```

Fuente (Elaboración Propia)

XML actualmente tiene un amplio soporte ya que muchas empresas utilizan esta forma de intercambio de datos. Sin embargo, ha sido muy criticado por su verbosidad y complejidad, ya que para documentos muy grandes suele requerir mucho más memoria y procesador, además del incremento en el tamaño del archivo.

- JSON

Es un formato ligero de texto el cual nos permite intercambiar datos. Es fácil al momento de leerlo y escribirlo ya que el formato es entendible para los humanos. JSON utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C#, Java, Python, entre otros. Estas propiedades mencionadas hacen que JSON sea un lenguaje muy interesante e ideal para poder intercambiar datos.

JSON está compuesto de dos estructuras:

- **Una colección de pares de nombre/valor.** Básicamente es como un objeto el cual está compuesto por 2 propiedades que son nombre y valor, donde el nombre vendría a ser el nombre

de atributo y el valor vendría a ser el valor del atributo. Por ejemplo:

Dirección: "Av. España 125".

Donde dirección es el nombre y "Av. España 125" es el valor.

- **Una lista ordenada de valores.** Esto vendría a ser como un conjunto de objetos.

Así como se muestra en el siguiente ejemplo:

Figura 9: Estructura del Formato JSON

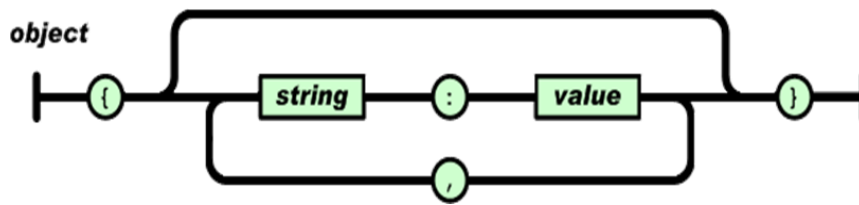
```
{
  "Persona": {
    "codigo": "1",
    "nombre": "Carlos Alfredo",
    "apellido": "Osorio Pérez",
    "email": "hacybeyker@gmail.com",
    "tecnologias": {
      "nombre": [
        "Java",
        "C#",
        "Php"
      ]
    },
    "edad": "21",
    "genero": "M",
    "estudiante": "true"
  }
}
```

Fuente (Elaboración Propia)

Estas estructuras son soportadas por todos los lenguajes de programación ya que son universales. Se presentan de las siguientes maneras:

Un objeto es un conjunto desordenado de pares nombre/valor. Al definir un objeto este empieza con una llave de apertura "{" y termina con una de cierre "}". Dentro de estas llaves van 2 valores el nombre y el valor, entre ellos se declara 2 puntos, para especificar otro par nombre/valor se separa con una coma ",". Así como lo muestra la imagen.

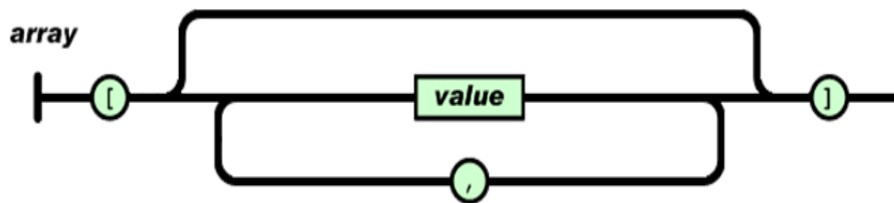
Figura 10: Estructura JSON de un Objeto



Fuente (Json)

Por otro lado, un arreglo es una colección que está compuesta de valores o se podría decir que está compuesta de objetos. Al definir un arreglo este empieza con un corchete de apertura "[" y al finalizar con un corchete de cierre "]". Y por cada objeto este es separado por una coma ",". Así como lo muestra la imagen.

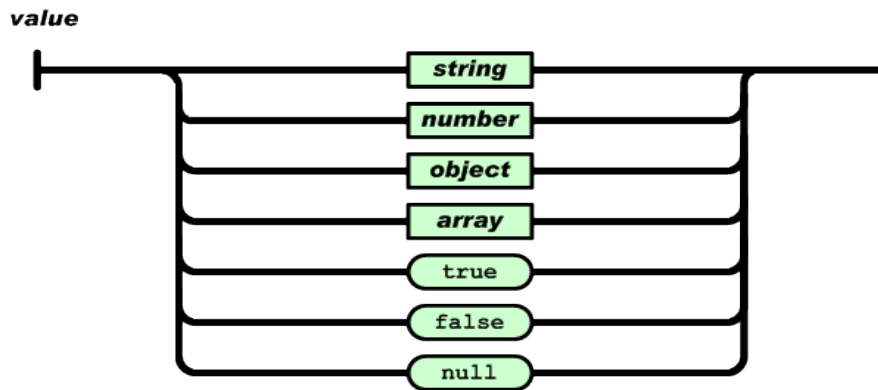
Figura 11: Estructura JSON de un Arreglo



Fuente (Json)

Dentro de las estructuras de JSON también se definen algunos tipos de datos, los cuales no se necesitan declararlos, pero si seguir con su estándar, ya que JSON verifica si los valores ingresados son válidos. Así como lo muestra la imagen.

Figura 12: Tipo de Datos de JSON



Fuente (Json)

1.3.7 BASE DE DATOS

Según (Mannino), define qué las bases de datos son una colección de datos los cuales son persistentes y que pueden compartirse e interrelacionarse. O también se podría decir que las bases de datos permiten mantener los datos almacenados de manera estable.

- **Sistema Gestor de Base de Datos**

Según (Silberschatz, 202), en su libro “Fundamentos de base de datos cuarta edición”, menciona que un Sistema Gestor de Base de Datos es una colección de datos interrelacionados y un conjunto de programas los cuales nos permitirán acceder a dichos datos. Su principal objetivo es el de proporcionar la manera de almacenar y recuperar la información de una base de datos.

Actualmente contamos con muchos softwares capaces de realizar esas tareas, algunas de pago y otras de tipo Open Source, entre ambos los más populares son:

- SQL Server.
- PostgreSQL.
- MySQL.
- Oracle
- MongoDB.

- Redis.
- Entre otras.
- **Base de Datos Relacionales**

Es un repositorio el cual contiene muchos datos y estos están compartidos, a su vez permite establecer interconexiones entre los datos, y así poder relacionarse entre ambas tablas y compartir información.

Las funcionalidades de la base de datos relacionales son:

- ACID
- Normalización
- Coherencia de la información a almacenar
- **Base de Datos No Relacionales**

Las bases de datos No relacionadas son base de datos las cuales pueden almacenar grandes cantidades de datos, ya que actualmente empresas como Facebook, Twitter, entre otras utilizan este tipo de base de datos porque a diario generan mucha información, como lo demuestra (Domo) en su infografía “Data Never Sleeps 2.0”.

Son sistemas gestores de base de datos pero que son distintos a los relacionales ya que en estos los datos no requieren de estructuras fijas como tablas, por lo general no suelen soportar los famosos Join, y tampoco garantizan la ACID, atomicidad, coherencia, aislamiento, durabilidad).

Por otro lado, la manera de almacenamiento de este tipo de base de datos es distinta, ya que el manejo de las columnas es dinámico, a diferencia de las relacionales que son estáticas. O sea que se pueden realizar cambios sin perder información, otro aspecto es que la forma de estructurar los datos no es en tablas y sus relaciones entre ellas son más flexibles, permiten almacenar en un tipo de formato como: clave – valor, mapeo de columnas, documentos. Esto varía dependiendo el sistema gestor de base de datos a utilizar.

Actualmente existen 150 sistemas de base de datos NoSQL, la elección de cual utilizar dependerá de cual se adapte a las necesidades del negocio. Entre esta lista extensa los más conocidos son:

A. Orientada a Documentos

Se encargan de gestionar los datos de manera semi estructurada ósea en documentos. Estos datos suelen ser almacenados en formatos como JSON, XML o BSON.

En esta categoría podemos mencionar a:

I. Firebase

Es una base de datos remota, la cual se encuentra alojada en la nube y es capaz de ser accedida desde navegadores y aplicaciones para Smartphone, una de sus principales características es que responde en tiempo real a los cambios que se realizan a los datos. Utiliza lo que es clave/valor para la representación de sus datos los cuales finalmente se pueden descargar en un formato json.

II. MongoDB

Es un sistema de base de datos no relacional, multiplataforma y utiliza clave/valor. Utiliza un formato BSON el cual es el JSON compilado para guardar la información. MongoDB es muy conocido y usado probablemente el más popular del momento. Actualmente las empresas que utilizan MongoDB son: Adobe, eBay, Forbes, Facebook, entre otras.

III. CouchDB

Al igual que MongoDB, CouchDB es una base de datos que utiliza JSON y está orientada a documentos, también utiliza JavaScript para índices de MapReduce y HTTP para su API. Las empresas que lo utilizan son: Credit Suisse y la BBC.

B. Orientado a Columnas

Funcionan de forma parecida a las bases de datos relacionales, pero almacenando columnas de datos en lugar de registros.

En esta categoría podemos mencionar a:

I. Cassandra

Esta orienta para proyectos en los cuales se necesita una alta disponibilidad y escalabilidad sin comprometer el rendimiento. Las empresas que lo utilizan son: CENR, GitHub, Netflix entre otros.

II. HBase

Se utiliza cuando se necesita leer y escribir en tiempo real a su Big Data, el objetivo de HBase es la organización de tablas muy grandes.

Está escrita en Java y mantenida por el Proyecto Hadoop de Apache. La utilizan Yahoo y Twitter.

C. Clave Valor

Son fáciles de entender. La forma de guardar la información es mediante tuplas, las cuales contienen una clave con su valor. Al recuperar la información, se realiza una búsqueda mediante la clave la cual retorna su valor.

En esta categoría podemos mencionar a:

I. DynamoDB

Desarrollada por Amazon, es otra opción de poder almacenar la información solo que en este caso podemos utilizar los servicios web de Amazon. Las empresas que lo utilizan son el Washington Post y Scopely.

II. Redis

Fue desarrollada en C y es de open source, las empresas que lo utilizan son: Craigslist y Stack Overflow.

D. En Grafo

Basadas en la teoría de grafos, ya que para representar los datos almacenados utilizan los nodos y las aristas. Son de mucha utilidad en los modelos con muchas relaciones como, por ejemplo: conexiones sociales. En esta categoría encontramos a:

I. Infinite Graph

Desarrollado por Objectivity, está basado en el lenguaje Java y C++. Cuenta con licencias gratuitas y de pago.

II. Neo4j

Es open source y está desarrollado en Java por la compañía Neo Technology. Las empresas que lo utilizan son: HP, Infojobs y Cisco.

1.3.8 FIREBASE

Es una plataforma móvil que te permite desarrollar rápidamente apps de alta calidad, aumentar tu base de usuarios y ganar más dinero. Firebase contiene funciones complementarias que puedes combinar y adaptar según tus necesidades.



Fuente Firebase

Al utilizar Firebase como servicio nos ayuda a despreocuparnos de algunas cosas y centrarnos a cosas más puntuales como, por ejemplo:

- **La infraestructura:**

Por lo general generar un servicio suele ser complicado desplegarlo ya que se debe de implementar un servidor hacer que este sea visible para cualquier parte del mundo, dependiendo del caso, instalar programas, etc. Cosas por las cuales nos conlleva a invertir tiempo y dinero. Pero, ahora Firebase nos ayuda y brinda el servicio para que no nos preocupes de implementar y desplegar todo lo mencionado, sino que con su APIs intuitivas contenidas en un solo SDK, es rápido y fácil, permitiéndonos poder concentrarse en resolver los problemas de los clientes y evitar perder tiempo en crear una infraestructura compleja.

- **Tomar decisiones inteligentes basadas en los datos**

Con Firebase Analytics se puede analizar de manera gratuita e ilimitada la información de nuestros usuarios, desde los clics que realizan en los anuncios hasta el uso de la app. Cabe resaltar que también puede controlar los fallos de la app.

- **Trabaja en diferentes plataformas**

Ya que actualmente está muy de moda el concepto de multiplataforma Firebase ofrece apps multiplataforma con API integradas a SDK individuales para Android, iOS y JavaScript. Lo cual nos permite que se pueda pasar a una plataforma diferente sin modificar la infraestructura.

 [Prueba el instructivo de iOS](#)

 [Prueba el instructivo de Android](#)

 [Prueba el instructivo web](#)

Fuente Firebase

- **Empresas que usan Firebase**

Firestore ya está siendo utilizado por empresas muy conocidas como se muestran en la imagen.



Fuente Firebase

- **Empresas que utilizan NoSql**

Tras el aumento acelerado de información que se genera cada minuto en el mundo las empresas han optado por utilizar este nuevo tipo de paradigma que es el NoSQL, con la finalidad de poder tener la capacidad de controlar toda la información y poder manipularla, analizarla y así obtener conocimiento para poder tomar buenas decisiones. Las empresas que actualmente utilizan estas soluciones son solo por mencionar algunas:

- Forbes.
- Bosch.
- Facebook.
- Twitter.
- Theguardian.
- Entre

otras

1.4 Formulación del Problema

¿Cómo implementar un sistema que permita la recuperación de información de múltiples bibliotecas con óptimos niveles de rendimiento?

1.5 Justificación del Estudio

Esta investigación se justifica de manera operativa de la siguiente manera, una vez que el sistema basado en una arquitectura orientada a servicios con base de datos NoSQL este diseñado, desarrollado y funcionando, tendrá como finalidad obtener la información de las todas las bases de datos creadas con la ayuda de la arquitectura orienta a servicio lo cual a su vez se conectara con una base de datos NoSQL en donde se registrara los datos más importantes de los sistemas transaccionales de las distintas bibliotecas, permitiendo así que las búsquedas a realizar puedan ser más rápidas.

Desde el punto de vista tecnológico, la arquitectura propuesta es una fusión de tecnologías existentes ya que se crearán base de datos relacionales, se implementará una arquitectura orienta a servicio, y como repositorio global se creará una base de datos no relacional basada en Firebase, esta última tecnología fue lanzada en el año 2011 pero renovada en mayo del 2016 por google, la cual trae muchas características nuevas y gracias a la documentación que se encuentra en su página oficial hace factible a la investigación.

En la parte económica de esta investigación no se tendrá que invertir comprando o adquiriendo algún tipo de producto o material que ayude a realizar esta investigación, por lo tanto, es factible.

Respecto a lo parte social esta investigación apoyara a que las personas no tengas que, literalmente “perder el tiempo ni dinero” yendo biblioteca por biblioteca a buscar un documento bibliográfico, sino que utilizaran un sistema en el que podrán buscar y obtener toda la información bibliográfica de las distintas bibliotecas y en un tiempo rápido.

1.6 Hipótesis

La implementación de un sistema basado en una arquitectura distribuida con base de datos NoSQL permite la recuperación de información de múltiples bibliotecas con óptimos niveles de rendimiento.

1.7 Objetivos

1.7.1 Objetivo General

Implementar un sistema basado en una arquitectura distribuida con base de datos NoSQL que permita la recuperación de información de múltiples bibliotecas con óptimos niveles de rendimiento.

1.7.2 Objetivos Específicos

- Implementar servicios web para la interoperabilidad con aplicaciones de múltiples bibliotecas.
- Implementar una Base de datos NoSQL para garantizar tiempos de respuesta óptimos en las búsquedas.
- Implementar un prototipo de aplicativo móvil para la búsqueda de información bibliográfica.
- Elaborar el modelo de pruebas que permita verificar el cumplimiento de los requisitos.

II. MÉTODO

2.1 Diseño de investigación

La investigación de acuerdo a la técnica de contrastación es de tipo Experimental y Pre Experimental debido a que se maneja un solo grupo. Cabe resaltar que en esta investigación solo se evaluará la parte del Post-Test ya que no habrá un Pre-Test, porque la finalidad de esta investigación no es compararla versus otra y verificar cual es la mejor, sino que en base a lo que se está proponiendo verificar si la solución está cumpliendo con los requerimientos tanto funcionales y no funcionales que se describieron en la realidad problemática.

Figura 13: Diseño de Investigación



Fuente (Elaboración Propia)

2.2 Variables y Operacionalización de variables

Variable	Definición Conceptual	Definición Operacional	Dimensión	Indicadores	Escala
<p><u>Independiente</u> Sistema basado en una arquitectura distribuida con base de datos NoSQL</p>	<p>Arquitectura orientada a servicios: Según (Microsoft Corporation, diciembre, 2008), “marco de diseño el cual integra aplicaciones independientes, con la finalidad que desde una red se pueda acceder a sus funcionalidades, las cuales se ofrecen como servicios.”</p>	<p>Arquitectura distribuida con base de datos NoSQL: permitirá la interoperabilidad entre los diversas Sistemas de las distintas bibliotecas</p>	<ol style="list-style-type: none"> 1. Componentes de servicios web. 2. Componente de base de datos NoSQL. 3. Componente de prototipo de aplicativo móvil. 	<p>Numero de servicios implementados.</p>	<p>Cuantitativa de Razón</p>

<u>Dependiente</u> Funcionalidad	Según (ISO 2196), “es la capacidad del producto software para proporcionar funciones que satisfagan las necesidades para lo que fue diseñado”.	Permitirá medir el grado de funcionalidad de software con lo cual ayudará a integración de servicios	Interoperabilidad	$X=A/B$ A= Numero de interfaces que se implementaron correctamente. B=Numero de interfaces especificados.	Cuantitativa de Razón
			Idoneidad	$X = 1 - A/B$ A=número de funciones faltantes. B número de funciones descritas en la especificación de requisitos.	
<u>Dependiente</u> Eficiencia	Según (ISO 2196), es el “grado en que optimiza los recursos de computo”.	Permitirá medir la eficiencia del software para poder realizar consultar rápidas	Tiempo de respuesta	$X = T$ T = Tiempo calculado o simulado (entre más corto mejor)	Cuantitativa de Razón

2.3 Población y muestra

En esta investigación la población no aplica debido a que la unidad de estudio es la funcionalidad del sistema sobre el cual se hará o aplicará las mediciones en base a las métricas de software.

2.4 Técnicas e instrumentos de recolección de datos, validez y confiabilidad

Este tipo de investigación no cuenta con una encuesta o entrevista como instrumento de recolección de datos ya que existen métricas de software que la (ISO 2196) define para poder realizar los cálculos respectivos al producto software y en base a ello ver la calidad con lo que el producto cuenta.

Instrumentos:

- Diseño de casos de prueba para los requerimientos funcionales y no funcionales (sirven para evaluar las funcionalidades y determinar el total de fallos).
- Herramientas de medición automática de métricas de software.

2.5 Métodos de análisis de datos

De acuerdo al trabajo de investigación, no se aplica un Pre-Test, por lo que no se hacen pruebas estadísticas. Por consiguiente, se utilizan las métricas de calidad para medir el producto software, los cuales se muestran en estos cuadros.

Figura 14: Métrica de la Funcionalidad - Idoneidad

MÉTRICAS DE FUNCIONALIDAD

- Idoneidad
- Exactitud
- Interoperabilidad
- Cumplimiento de Normas
- Seguridad

Cuadro de Métrica de Comportamiento respecto la Idoneidad	
Nombre:	Cumplimiento de Implementación Funcional
Propósito:	Que tan completa esta la implementación funcional
Método de aplicación:	Contar las funciones faltantes detectadas en la evaluación y comparar con el número de funciones descritas en la especificación de requisitos.
Medición, formula:	$X = 1 - A/B$ <p>A = número de funciones faltantes B = número de funciones descritas en la especificación de requisitos.</p>
Interpretación:	$0 \leq X \leq 1$ <p>Entre más cercano a 1, más completa.</p>
Tipo de Escala:	Absoluta
Tipo de Medida:	$X = 1 - \text{cantidad} / \text{cantidad}$ <p>A = cantidad B = cantidad</p>
Fuente medición:	<ul style="list-style-type: none"> • Especificación de requisitos • Diseño • Código fuente • Informe de revisión
ISO/IEC 12207 SLCP:	<ul style="list-style-type: none"> • Verificación. • Revisión conjunta.
Audiencia:	<ul style="list-style-type: none"> • Desarrolladores. • Requeridores.

Figura 15: Métrica de la Funcionalidad – Interoperabilidad

MÉTRICAS DE FUNCIONALIDAD

- Idoneidad
- Exactitud
- **Interoperabilidad**
- Cumplimiento de Normas
- Seguridad

Cuadro de Métrica de Comportamiento respecto a la Interoperabilidad	
Nombre:	Intercambiabilidad de Datos
Propósito:	Que tan bien han sido implementados los formatos de datos de interfaz.
Método de aplicación:	Contar el número de formatos de datos de interfaz que se han implementado correctamente como en las especificaciones y comparar con el número de formatos de intercambio de datos como en las especificaciones.
Medición, formula:	$X = A/B$ A = número de interfaces de formatos de datos que han sido implementadas correctamente como en las especificaciones. B = número de formato de datos intercambiados como en las especificaciones
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, más correcto.
Tipo de Escala:	Absoluta
Tipo de Medida:	$X = \text{cantidad} / \text{cantidad}$ A = cantidad B = cantidad
Fuente medición:	<ul style="list-style-type: none"> • Especificación de requisitos • Diseño • Código fuente • Informe de revisión
ISO/IEC 12207 SLCP:	<ul style="list-style-type: none"> • Verificación. • Revisión conjunta.
Audiencia:	<ul style="list-style-type: none"> • Desarrolladores. Requeridores.

Figura 16: Métrica de la Eficiencia - Comportamiento respecto al Tiempo

METRICAS DE EFICIENCIA

- Comportamiento respecto al tiempo
- Comportamiento respecto a recursos

Cuadro de Métrica de Comportamiento respecto al Tiempo	
Nombre:	Tiempo de respuesta.
Propósito:	Cuál es el tiempo estimado para completar una tarea.
Método de aplicación:	<p>Evaluar la eficiencia de las llamadas al SO y a la aplicación.</p> <p>Estimar el tiempo de respuesta basado en ello. Puede medirse:</p> <ul style="list-style-type: none"> • Todo o partes de las especificaciones de diseño. • Probar la ruta completa de una transacción. • Probar módulos o partes completas del producto. • Producto completo durante la fase de pruebas.
Medición, fórmula:	$X = \text{Tiempo (calculado o simulado)}$.
Interpretación:	Entre menos tiempo, mejor.
Tipo de Escala:	Proporción
Tipo de Medida:	$X = \text{Tiempo}$
Fuente medición:	<ul style="list-style-type: none"> • Sistema operativo conocido. • Tiempo estimado en llamadas al sistema.
ISO/IEC 12207 SLCP:	<ul style="list-style-type: none"> • Verificación. • Revisión conjunta.
Audiencia:	<ul style="list-style-type: none"> • Desarrolladores. • Requeridores.

III. RESULTADOS

3.1 Métrica de Funcionalidad

- **Interoperabilidad**

Para obtener la medición de la interoperabilidad entre bibliotecas la (ISO 2196) define la siguiente formula.

$$X=A/B$$

Dónde:

A: Numero de servicios implementados correctamente como en las especificaciones.

B: Numero de servicios que se definió en las especificaciones.

Tengamos en cuenta que los servicios a implementar se definieron en el Documento de arquitectura de Software - DAS.

Resultado:

A: Registrar Libros y Consultar Libros

B: Registrar Libros y Consultar Libros

Detalle del Resultado:

Servicio para registrar Libros


```

// POST api/values
[HttpPost]
0 references | Carlos Alfredo Osorio Pérez, 10 days ago | 1 author, 2 changes
public async Task<HttpStatusCode> Post([FromBody]Libro libro)
{
    if (libro.validarJson(libro))
    {
        Biblioteca biblioteca = libro.biblioteca;
        GestionarBibliotecaServicio gestionarBibliotecaServicio = new GestionarBibliotecaServicio();
        biblioteca.codigo = await gestionarBibliotecaServicio.crear(biblioteca);

        Categoria categoria = libro.categoria;
        GestionarCategoriaServicio gestionarCategoriaServicio = new GestionarCategoriaServicio();
        categoria.codigo = await gestionarCategoriaServicio.crear(categoria);

        GestionarLibroServicio gestionarLibroServicio = new GestionarLibroServicio();
        return await gestionarLibroServicio.crear(libro);
    }
    else
        return new HttpStatusCode();
}

```

Fuente: Elaboración propia – Visual Studio 2015

Servicio para consultar Libros

```

// GET: api/values
[HttpGet]//Busqueda de todos los libros
0 references | Carlos Alfredo Osorio Pérez, 11 days ago | 1 author, 1 change
public async Task<List<Libro>> Get()
{
    GestionarLibroServicio gestionarLibroServicio = new GestionarLibroServicio();
    return await gestionarLibroServicio.buscar();
}

```

Fuente: Elaboración Propia - Visual Studio 2015

Finalmente, si reemplazamos los datos quedaría de la siguiente manera:

$$X=2/2$$

$$X=1$$

- **Idoneidad**

Para obtener la medición de la idoneidad la (ISO 2196) define la siguiente formula.

$$X= 1 - A/B$$

Donde:

A: Numero de funciones faltantes.

B: Numero de funciones descritas en la especificación de requisitos.

Tengamos en cuenta que las especificaciones de requisitos del software en general se definieron en el Documento de Especificaciones Suplementarias y en el documento de arquitectura de Software - DAS.

Resultado:

Las especificaciones de requisitos fueron:

- Permitir la interoperabilidad de 1 o más bibliotecas.
- Los tiempos de respuesta deben de estar entre 3 a 4 segundos como máximo o en promedio.
- Permitir el desacoplamiento entre capas para una mayor escalabilidad.

Lo que se logro fue:

- Implementar la interoperabilidad de 1 o más bibliotecas.
- El tiempo de respuesta para acceder al servicio web en promedio fue de 2.288 segundos.
- La arquitectura del servicio se implementó mediante el patrón N-Capas para permitir el desacoplamiento y una mayor escalabilidad.

Finalmente, si reemplazamos los datos quedaría de la siguiente manera:

$$X = 1 - (A/B)$$

$$X = 1 - (0/3)$$

$$X = 1$$

3.2 Métrica de Eficiencia

- **Comportamiento respecto al tiempo**

Para obtener la medición del comportamiento respecto al tiempo la (ISO 2196) define la siguiente formula.

X

Donde:

X: tiempo (segundos)

Resultado:

- Resultado del Tiempo

Cantidad de Registros	Tiempo Transcurrido al buscar en la base de datos NoSQL
500	1.92 segundos
1000	1.71 segundos
1500	1.82 segundos
2000	1.86 segundos
5000	4.13 segundos

Detalle del resultado:

Para obtener el tiempo promedio en que tarda el servicio en devolver la información, se tomó en cuenta realizar 5 pruebas, cada una con una cantidad de registros específico y mediante la ayuda del navegador Google Chrome en la opción de “Herramientas de Desarrollador” se pudo obtener el tiempo en que tarda. E aquí los resultados de las 5 pruebas:

En un comienzo se ingresaron 500 registros, al consultar al servicio web encargado de devolverme los libros, se obtuvo mediante la herramienta de desarrollador, que la petición tardó 1.92 segundos en devolver dicha información.

The screenshot shows a browser window with the URL `serviciofindbook.azurewebsites.net/api/libro`. The developer tools are open to the Network tab, showing a request for 500 records. The response time is 1.92s. The response is a large JSON array of book records.

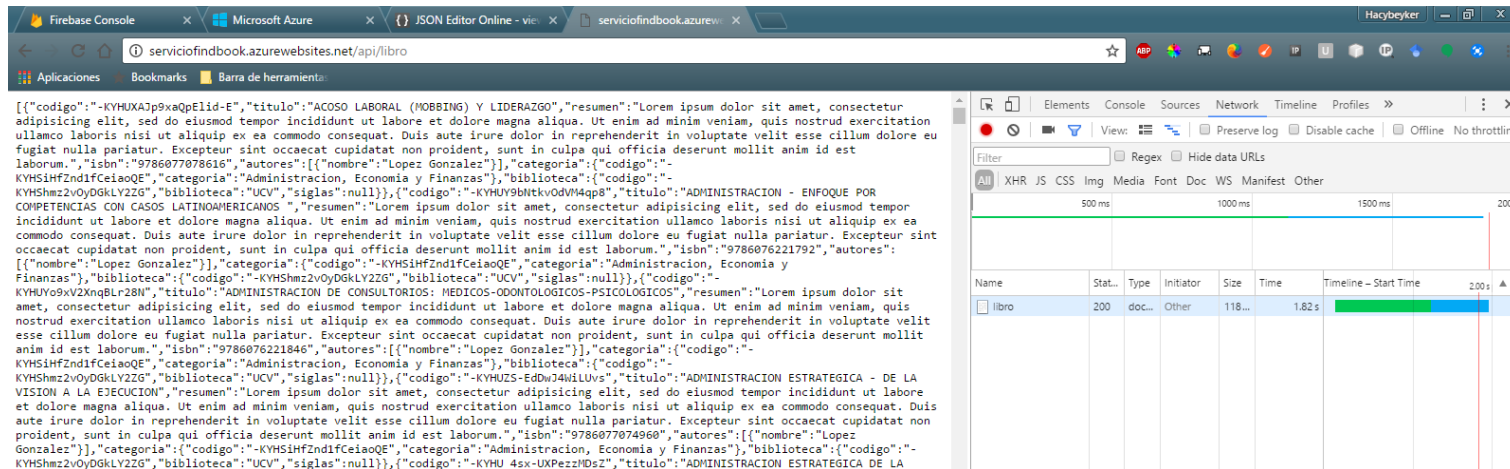
Name	Status	Type	Initiator	Size	Time	Timeline - Start Time
libro	200	doc...	Other	39.9...	1.92 s	

Como segunda prueba se ingresaron 1000 registros, al consultar al servicio web encargado de devolverme los libros, se obtuvo mediante la herramienta de desarrollador, que la petición tardó 1.71 segundos en devolver dicha información.

The screenshot shows a browser window with the URL `serviciofindbook.azurewebsites.net/api/libro`. The developer tools are open to the Network tab, showing a request for 1000 records. The response time is 1.71s. The response is a large JSON array of book records.

Name	Status	Type	Initiator	Size	Time	Timeline - Start Time
libro	200	doc...	Other	78.6...	1.71 s	

Como tercera prueba se ingresaron 1500 registros, al consultar al servicio web encargado de devolverme los libros, se obtuvo mediante la herramienta de desarrollador, que la petición tardó 1.82 segundos en devolver dicha información.



Como cuarta prueba se ingresaron 2000 registros, al consultar al servicio web encargado de devolverme los libros, se obtuvo mediante la herramienta de desarrollador, que la petición tardó 1.86 segundos en devolver dicha información.

Network Performance Data:

Name	Status	Type	Initiator	Size	Time	Timeline - Start Time
libro	200	doc...	Other	157...	1.86 s	2.00 s

Como quinta prueba se ingresaron 5000 registros, al consultar al servicio web encargado de devolverme los libros, se obtuvo mediante la herramienta de desarrollador, que la petición tardó 4.13 segundos en devolver dicha información.

Network Performance Data:

Name	Status	Type	Initiator	Size	Time	Timeline - Start Time
libro	200	docu...	Other	392 KB	4.13 s	4.00 s

- Calculando:

$$X = (1.92+1.71+1.82+1.86+4.13) / 5$$

$$X = 11.44/5$$

$$X = 2.288$$

Teniendo en cuenta que se realizaron 5 pruebas con distintas cantidades de registros y que en cada cual se obtuvo un resultado, finalmente se realizó un promedio y el resultado final fue 2.288 segundos.

IV. DISCUSIÓN

Esta investigación logro satisfacer los requerimientos definidos en el Documento de arquitectura de Software, ya que cumplió las siguientes expectativas:

Permitir la interoperabilidad entre bibliotecas, en el DAS se esperaba que se implementara un servicio web el cual fuese capaz de realizar la integración con múltiples bibliotecas, por lo que se implementaron dos métodos los cuales fueron el de registrar libros el cual permitirá desde el punto de vista de las bibliotecas poder enviar su información y almacenarla en base de datos NoSQL, y por otro lado buscar libros el cual servirá para que los usuarios que cuenten con un dispositivo móvil pueda acceder a visualizar los libros.

Estos son los dos servicios que fueron implementados satisfactoriamente, logrando cumplir la métrica de calidad denominada interoperabilidad, en la que define la siguiente formula $X=A/B$ la cual dio como resultado 1, que según la interpretación de la (ISO 2196) dice que mientras más cercano al 1 la métrica es correcta, lo cual significa que para cumplir la métrica de interoperabilidad los servicios implementados correctamente de acuerdo al DAS deben de ser los que se definieron en un comienzo en el DAS.

Por otra parte, se definió medir el comportamiento respecto al tiempo, en la tabla de factores de las especificaciones suplementarias se esperaba que el promedio de tiempo en la búsqueda de libros sea entre 3 a 4 segundos, por lo que se realizaron 5 pruebas con distintas cantidades de registros controlando el tiempo en que se demoran en devolver los libros. Finalmente se obtuvo que el promedio de las 5 pruebas dio como resultado 2.288 segundos. Según la (ISO 2196) la fórmula para esta métrica simplemente es X la cual significa tiempo, logrando así cumplir la métrica la cual fue basada en el factor de la especificación suplementaria.

Esta investigación sirve para que exista la interoperabilidad entre bibliotecas con buenos niveles de rendimiento. Utilizando dos conceptos, servicios web y base de datos NoSQL. Los Servicio Web como lo define (Microsoft Corporation, Diciembre, 2008) son aplicaciones los cuales utilizan estándares con la finalidad de intercambiar información, a su vez el uso y los beneficios de los

servicios web son confirmados por la investigación de (Fabela Cano, 2006) denominada “Biblioteca Digital en Servicios Web” en la que también implemento servicios web para poder ofrecer servicios añadiendo una ventaja para la biblioteca digital. Y también como lo dice la investigación de (Brito Moreno, 2011) denominada “Implementar el consumo de servicios vía web Services de la información almacenada en una base de datos de una biblioteca digital perteneciente a una institución educativa de nivel intermedio” el cual utilizó los servicios web para proporcionar una disponibilidad de 24 horas a los colegios de Guayaquil permitiendo que la información pueda ser accesible desde cualquier parte a través de internet. y base de datos NoSQL.

Finalmente, esta investigación logra satisfacer la problemática descrita en un comienzo, ya que una de las problemática es el de integrar múltiples sistemas de bibliotecas para lo cual la solución fue la implementación de un servicio web, como nos muestra los resultado, la implementación del servicio web se realizó correctamente, permitiendo así que múltiples bibliotecas puedan integrarse y así centralizar la información, otro problema era la facilidad del uso de los sistemas de biblioteca, lo cual se logró implementando una aplicación móvil que consumía el servicio de buscar libros y permitía que desde cualquier parte pueda acceder a visualizar los libros, y el otro problema era el tiempo de respuesta que tardan por lo general los sistemas de bibliotecas, para este problema se implementó una base de datos NoSQL Firebase y los componentes se desplegaron en distintos servidores como en Microsoft Azure y Google con la finalidad de poder obtener resultados en tiempos óptimos.

V. CONCLUSIÓN

- Se concluye que la implementación de un sistema basado en una arquitectura distribuida con base de datos NoSQL, permite la recuperación de información de múltiples Bibliotecas con óptimos niveles de rendimiento.
- La implementación del servicio web - RestFul permitió la interoperabilidad con las aplicaciones de múltiples bibliotecas.
- La implementación de una base de datos NoSQL garantizó tiempos de respuesta óptimos en las búsquedas.
- La implementación de una aplicación móvil permitió la búsqueda de información bibliográfica.
- La elaboración del modelo de pruebas permitió verificar el cumplimiento de los requisitos.

VI. RECOMENDACIONES

Luego de haber realizado esta investigación, se llegó a las siguientes recomendaciones.

- Se recomienda definir en primeras instancias los servicios a ofrecer dependiendo de la problemática, para así poder implementarlos correctamente y con las reglas necesarias.
- Se recomienda explorar y aprender más acerca del concepto de base de datos no relacionales, más que todo en como diseñar una, ya que es completamente distinta al concepto de base de datos relaciones ya que por lo general no existen relaciones entre tablas.
- Se recomienda tratar de utilizar esta investigación no solo para centralizar la información de bibliotecas, sino que esta solución se puede adaptar a otras problemáticas correctamente.

VII. REFERENCIAS

- ✓ ¿A que se debe el crecimiento exponencial del número de revistas y libros académicos con acceso libre en Internet? **Nardi, Alejandra M.** Córdoba : Facultad de Ciencias Económicas Universidad Nacional de Córdoba.
- ✓ **Akerkar, Rajendra. 2014.** Big Data Computing. Instituto de Investigación Noruega Occidental : CRC Press Taylor & Francis Group, 2014. pág. 538.
- ✓ **Brito Moreno, Jhon Edison. 2011.** Implementar el Consumo de Servicios vía Web Services de la Información almacenada en una Base de Datos de una Biblioteca Digital perteneciente a una Institución Educativa de nivel intermedio. Universidad de Guayaquil. Guayaquil : s.n., 2011. pág. 121, Tesis.
- ✓ **Domo. Domo. Domo.** [En línea] [Citado el: 30 de 06 de 2016.] <https://www.domo.com/learn/data-never-sleeps-2>.
- ✓ **Douglas K., Barry.** Web Services, Service-Oriented Architectures, and Cloud Computing, Second Edition: The Savvy Manager's Guide. 978-0123983572.
- ✓ **Erl, Thomas. 2008.** SOA: Principles of Service Design. 2008.
- ✓ **Fabela Cano, Oscar. 2006.** Biblioteca Digital Basada en Servicios Web. Instituto Politécnico Nacional. Mexico : s.n., 2006. pág. 93, Tesis.
- ✓ **Gartner. January, 210.** Technology Trends You Can't Afford to. January, 210. pág. 30.
- ✓ **Mannino, Michael V. .** Administración de Bases de Datos Diseño y Desarrollo de Aplicaciones. Vol. III.
- ✓ **Microsoft Corporation. Diciembre, 2008.** La Arquitectura Orientada a Servicios(SOA) de Microsoft aplicada al mundo real. Estado Unidos : s.n., Diciembre, 2008. pág. 21.
- ✓ **Pinedo Marín, Luis Enrique y Medina Bonilla, Luis Eduardo. 2010.** Implementación de un sistema de integración para las bibliotecas municipales de Lima y Callao utilizando SOA y J2ME. Universidad Nacional Mayor de San Marcos. Programa Cybertesis PERÚ. Lima : s.n., 2010. pág. 176, Tesis Profesional.
- ✓ **Ramirez J., Jorge. Noviembre 2009.** Arquitectura Orientada a Servicios. San Jose : Rho-Sigma S.A, Noviembre 2009. pág. 34.
- ✓ **Sánchez Reyna, Katty Sue Hellen. 2013.** Recuperación de historias clínicas electrónicas a partir de un repositorio digital usando una arquitectura orientada a servicios. PUCP. Lima : s.n., 2013. pág. 73, Tesis.
- ✓ **Silberschatz, Abraham. 202.** Fundamentos de Bases de Datos. s.l. : MC Graw Hill, 202. Vol. 4.

ANEXO

1. Cuadro de elección de metodología de desarrollo de software

Tabla 1: Selección de la metodología de desarrollo de software

CRITERIO	METODOLOGIA		
	RUP	XP	ICONIX
Información	6	4	5
Documentación	6	5	4
Tiempo y Costo	4	5	5
Aseguramiento de la Calidad	6	4	4
Total	22	18	18

De acuerdo a los criterios seleccionados y la encuesta realizada a 2 expertos en desarrollo de software, la metodología a utilizar es RUP ya que fue la que obtuvo más puntaje que el resto de metodologías, por lo que la investigación se trabajara en base a la metodología RUP.



**FACULTAD DE INGENIERÍA
ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS**

ENCUESTA PARA LA ELECCIÓN DE LA METODOLOGÍA DE DESARROLLO

Tema: "Sistema basado de una arquitectura orientada a servicios con base de datos NoSQL para la automatización funcional y eficiencia en la recuperación de la información de múltiples bibliotecas".

Finalidad: Obtener una metodología la cual me ayudara a realizar la investigación.

La información que usted brinde es muy importante para la investigación en curso. Sea muy honesto al momento de brindar la información.

Apellidos y Nombre:

Profesión:

Años de Experiencia:

Criterios de Evaluación de la Metodología: La elección de la metodología se realizará bajo los siguientes criterios de evaluación.

- **Información:** Se refiere a las guías o los ejemplos que cuenta cada metodología para su implementación.
- **Documentación:** Se refiere a que todos los procesos cuentan con una documentación clara y precisa en el proceso de desarrollo de software.
- **Tiempo y Costo:** Se refiere al tiempo y al costo que se invierten al utilizar la metodología.
- **Aseguramiento de la Calidad:** Se refiere a la capacidad que cuenta cada metodología para poder garantizar la calidad del producto.

Cuadro de Valoración: Para la elección de la metodología se utilizará la siguiente valoración.

Valoración	Escala
Deficiente	1
Regular	2
Bueno	3

Calificación de las Metodología en base a los criterios de valoración

CRITERIO	METODOLOGÍAS		
	RUP	XP	ICONIX
Información			
Documentación			
Tiempo y Costo			
Aseguramiento de la Calidad			



FACULTAD DE INGENIERÍA
 ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS

ENCUESTA PARA LA ELECCIÓN DE LA METODOLOGÍA DE DESARROLLO

Tema: "Sistema basado de una arquitectura orientada a servicios con base de datos NoSQL para la automatización funcional y eficiencia en la recuperación de la información de múltiples bibliotecas".

Finalidad: Obtener una metodología la cual me ayudara a realizar la investigación.

La información que usted brinde es muy importante para la investigación en curso. Sea muy honesto al momento de brindar la información.

Apellidos y Nombre: *Córdova Escalante, Ivan*

Profesión: *Iny. Computación y Sistemas*

Años de Experiencia: *15*

Criterios de Evaluación de la Metodología: La elección de la metodología se realizará bajo los siguientes criterios de evaluación.

- **Información:** Se refiere a las guías o los ejemplos que cuenta cada metodología para su implementación.
- **Documentación:** Se refiere a que todos los procesos cuentan con una documentación clara y precisa en el proceso de desarrollo de software.
- **Tiempo y Costo:** Se refiere al tiempo y al costo que se invierten al utilizar la metodología.
- **Aseguramiento de la Calidad:** Se refiere a la capacidad que cuenta cada metodología para poder garantizar la calidad del producto.

Cuadro de Valoración: Para la elección de la metodología se utilizará la siguiente valoración.

Valoración	Escala
Deficiente	1
Regular	2
Bueno	3

Calificación de las Metodología en base a los criterios de valoración

CRITERIO	METODOLOGÍAS		
	RUP	XP	ICONIX
Información	<i>3</i>	<i>2</i>	<i>2</i>
Documentación	<i>3</i>	<i>3</i>	<i>2</i>
Tiempo y Costo	<i>2</i>	<i>3</i>	<i>2</i>
Aseguramiento de la Calidad	<i>3</i>	<i>2</i>	<i>2</i>



FACULTAD DE INGENIERÍA
 ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS

ENCUESTA PARA LA ELECCIÓN DE LA METODOLOGÍA DE DESARROLLO

Tema: "Sistema basado de una arquitectura orientada a servicios con base de datos NoSQL para la automatización funcional y eficiencia en la recuperación de la información de múltiples bibliotecas".

Finalidad: Obtener una metodología la cual me ayudara a realizar la investigación.

La información que usted brinde es muy importante para la investigación en curso. Sea muy honesto al momento de brindar la información.

Apellidos y Nombre: Urquiza Gomez Yosip

Profesión: Ing. de Sistemas

Años de Experiencia: 14

Criterios de Evaluación de la Metodología: La elección de la metodología se realizará bajo los siguientes criterios de evaluación.

- **Información:** Se refiere a las guías o los ejemplos que cuenta cada metodología para su implementación.
- **Documentación:** Se refiere a que todos los procesos cuentan con una documentación clara y precisa en el proceso de desarrollo de software.
- **Tiempo y Costo:** Se refiere al tiempo y al costo que se invierten al utilizar la metodología.
- **Aseguramiento de la Calidad:** Se refiere a la capacidad que cuenta cada metodología para poder garantizar la calidad del producto.

Cuadro de Valoración: Para la elección de la metodología se utilizará la siguiente valoración.

Valoración	Escala
Deficiente	1
Regular	2
Bueno	3

Calificación de las Metodología en base a los criterios de valoración

CRITERIO	METODOLOGÍAS		
	RUP	XP	ICONIX
Información	3	2	3
Documentación	3	2	2
Tiempo y Costo	2	2	3
Aseguramiento de la Calidad	3	2	2

2. Cuadro de validación de la realidad problemática

Tabla 2: Cuadro de validación de la realidad problemática

CRITERIO	Sistema de biblioteca		
	UCV	UPAO	UPN
Funcionalidad integrada en el sistema de biblioteca con múltiple base de datos	1	1	1
Integridad en el sistema de biblioteca	2	2	2
Facilidad de uso en el sistema de biblioteca	2	2	2
Tiempo de respuesta en los sistemas de biblioteca	2	2	2
Total	7	7	7

De acuerdo a los criterios seleccionados y la encuesta realizada a 1 experto, el cual ha trabajado en las distintas universidades mencionadas y ha podido hacer uso del sistema de biblioteca de cada universidad, se obtuvo como resultado de que ninguna del sistema de biblioteca de las universidades cuenta con la funcionalidad de integrar múltiple base de datos y en los demás criterios tampoco tienen una valoración tan favorable, sino que tienen una valoración regular.



**FACULTAD DE INGENIERÍA
ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS
ENCUESTA PARA LA VALIDACIÓN DE LA REALIDAD PROBLEMÁTICA**

Tema: "Sistema basado de una arquitectura orientada a servicios con base de datos NoSQL para la automatización funcional y eficiencia en la recuperación de la información de múltiples bibliotecas".

Finalidad: Validar que los problemas de la realidad problemática sean ciertos.

La información que usted brinde es muy importante para la investigación en curso. Sea muy honesto al momento de brindar la información.

Apellidos y Nombre:

Profesión:

Cargo:

Cuadro de Valoración: Para la validación de la realidad problemática se utilizará la siguiente valoración.

Valoración	Escala
Bueno	3
Regular	2
Deficiente	1

Calificación de los problemas descritos en la realidad problemática en base a los criterios de valoración.

Criterio	Sistemas de Biblioteca		
	UCV	UPAO	UPN
Funcionalidad integrada en el sistema de biblioteca con múltiples base de datos			
Integridad en el sistema de biblioteca			
Facilidad de uso en el sistema de biblioteca			
Tiempo de respuesta en los sistemas de biblioteca			



FACULTAD DE INGENIERÍA
ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS
ENCUESTA PARA LA VALIDACIÓN DE LA REALIDAD PROBLEMÁTICA

Tema: "Sistema basado de una arquitectura orientada a servicios con base de datos NoSQL para la automatización funcional y eficiencia en la recuperación de la información de múltiples bibliotecas".

Finalidad: Validar que los problemas de la realidad problemática sean ciertos.

La información que usted brinde es muy importante para la investigación en curso. Sea muy honesto al momento de brindar la información.

Apellidos y Nombre: Córdova Escalante, Lam.
 Profesión: Ing. Computación y Sistemas
 Cargo: Docente UCV

Cuadro de Valoración: Para la validación de la realidad problemática se utilizará la siguiente valoración.

Valoración	Escala
Bueno	3
Regular	2
Deficiente	1

Calificación de los problemas descritos en la realidad problemática en base a los criterios de valoración.

Criterio	Sistemas de Biblioteca		
	UCV	UPAO	UPN
Funcionalidad integrada en el sistema de biblioteca con múltiples base de datos	1	1	1
Integridad en el sistema de biblioteca	2	2	2
Facilidad de uso en el sistema de biblioteca	2	2	2
Tiempo de respuesta en los sistemas de biblioteca	2	2	2

Tabla 3: Cuadro de validación de la realidad problemática

Pregunta	Alternativa	
	SI	NO
¿Por lo general suele demorar en encontrar un determinado libro o tema de interés en el sistema de biblioteca?	7	0
¿Siente que las búsquedas que realiza en los sistemas de bibliotecas son lentas?	7	0
¿Por lo general al realizar las búsquedas en el sistema de biblioteca suele encontrar lo que buscaba?	2	5
¿te gustaría encontrar información de varias bibliotecas al momento de realizar tus búsquedas en el sistema de biblioteca?	7	0

De acuerdo a las preguntas realizadas a 7 estudiantes de la Universidad Cesar Vallejo, respecto a algunas características del sistema de biblioteca se obtuvo como resultado que, por lo general se suele demorar en encontrar lo que uno busca, el tiempo que se demora el sistema de biblioteca es mucho, muchas veces no se encuentra lo que se buscaba y que les gustaría obtener información de varias bibliotecas para saber en dónde pueden encontrar lo que están buscando.

ENCUESTA PARA LA REALIDAD PROBLEMÁTICA (REQUERIMIENTOS)

Encuesta dirigida a los alumnos y profesores de la Universidad Cesar Vallejo con el fin de poder captar las necesidades que tienen acerca del uso del sistema de bibliotecas que actualmente utilizan.

La encuesta consta de 2 preguntas puntuales, los cuales tiene como opciones SI y NO, usted debe de marcar cual mejor le parezca.

1. ¿Por lo general suele demorar en encontrar un determinado libro o tema de interés en el sistema de biblioteca?

SI

NO

2. ¿Siente que las búsquedas que realiza en los sistemas de bibliotecas son lentas?

SI

NO

3. ¿Por lo general al realizar las búsquedas en el sistema de biblioteca suele encontrar lo que buscaba?

SI

NO

4. ¿Te gustaría encontrar información de varias bibliotecas al momento de realizar tus búsquedas en el sistema de biblioteca?

SI

NO

3. Evaluación de instrumentos de recolección de datos



PLANTILLAS PARA LA EVALUACIÓN DE INSTRUMENTOS DE RECOLECCIÓN DE DATOS

1. IDENTIFICACION DEL EXPERTO

NOMBRE DEL EXPERTO: Lavin Cardenas Escalante.
 DNI 18133704 PROFESION: Ing. Computación y Sistemas
 LUGAR DE TRABAJO: UCV
 CARGO QUE DESEMPEÑA: Docente.
 DIRECCION: _____
 TELEFONO FIJO: _____ MOVIL: _____
 DIRECCION ELECTRONICA: lavin.cardenas@gmail.com
 FECHA DE EVALUACIÓN: 12/07/2016
 FIRMA DEL EXPERTO: [Firma]

2. PLANILLA DE VALIDACION DEL INSTRUMENTO

CRITERIOS	APRECIACION CUALITATIVA			
	EXCELENTE (4)	BUENO (3)	REGULAR (2)	DEFICIENTE (1)
Presentación del instrumento	/			
Claridad en la redacción de los ítems	/			
Pertinencia de las variables con los indicadores	/			
Relevancia del contenido	/			
Factibilidad de la aplicación	/			

APRECIACION CUALITATIVA: _____

OBSERVACIONES: _____

3. JUICIO DE EXPERTOS:

- En líneas generales, considera Ud. que los indicadores de las variables están inmersos en su contexto teórico de forma:

SUFICIENTE ✓	MEDIANAMENTE SUFICIENTE	INSUFICIENTE
--------------	----------------------------	--------------

OBSERVACION:

- Considera que los reactivos del cuestionario miden los indicadores seleccionados para la variable de manera:

SUFICIENTE ✓	MEDIANAMENTE SUFICIENTE	INSUFICIENTE
--------------	----------------------------	--------------

OBSERVACION:

- El instrumento diseñado mide la variable de manera:

SUFICIENTE ✓	MEDIANAMENTE SUFICIENTE	INSUFICIENTE
--------------	----------------------------	--------------

OBSERVACION:

- El instrumento diseñado es:

4. Validación del instrumento

MÉTRICAS	ESCALA				OBSERVACIONES
	DEJAR	MODIFICAR	ELIMINAR	INCLUIR	
01	 				
02	 				
03	 				

DESARIA INCLUIR	COMO LO MODIFICARIA

MÉTRICAS DE FUNCIONALIDAD

- Idoneidad
- Exactitud
- **Interoperabilidad**
- Cumplimiento de Normas
- Seguridad

Cuadro de Métrica de Comportamiento respecto a la Interoperabilidad	
Nombre:	Intercambiabilidad de Datos
Propósito:	Que tan bien han sido implementados los formatos de datos de interfaz.
Método de aplicación:	Contar el número de formatos de datos de interfaz que se han implementado correctamente como en las especificaciones y comparar con el número de formatos de intercambio de datos como en las especificaciones.
Medición, formula:	$X = A/B$ A = número de interfaces de formatos de datos que han sido implementadas correctamente como en las especificaciones. B = número de formato de datos intercambiados como en las especificaciones
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, más correcto.
Tipo de Escala:	Absoluta
Tipo de Medida:	$X = \text{cantidad} / \text{cantidad}$ A = cantidad B = cantidad
Fuente medición:	<ul style="list-style-type: none"> • Especificación de requisitos • Diseño • Código fuente • Informe de revisión
ISO/IEC 12207 SLCP:	<ul style="list-style-type: none"> • Verificación. • Revisión conjunta.
Audiencia:	<ul style="list-style-type: none"> • Desarrolladores. Requeridores.

MÉTRICAS DE EFICIENCIA

- Comportamiento respecto al tiempo
- Comportamiento respecto a recursos

Cuadro de Métrica de Comportamiento respecto al Tiempo	
Nombre:	Tiempo de respuesta.
Propósito:	Cuál es el tiempo estimado para completar una tarea.
Método de aplicación:	<p>Evaluar la eficiencia de las llamadas al SO y a la aplicación.</p> <p>Estimar el tiempo de respuesta basado en ello. Puede medirse:</p> <ul style="list-style-type: none"> • Todo o partes de las especificaciones de diseño. • Probar la ruta completa de una transacción. • Probar módulos o partes completas del producto. • Producto completo durante la fase de pruebas.
Medición, fórmula:	X = Tiempo (calculado o simulado).
Interpretación:	Entre menos tiempo, mejor.
Tipo de Escala:	Proporción
Tipo de Medida:	X = Tiempo
Fuente medición:	<ul style="list-style-type: none"> • Sistema operativo conocido. • Tiempo estimado en llamadas al sistema.
ISO/IEC 12207 SLCP:	<ul style="list-style-type: none"> • Verificación. • Revisión conjunta.
Audiencia:	<ul style="list-style-type: none"> • Desarrolladores. • Requeridores.

4. Documento de Especificaciones Suplementarias

Especificaciones Suplementarias

FindBook

INTRODUCCION

Este documento muestra los requerimientos que se deberán de implementar en el proyecto para que al finalizarlo se pueda contrastar con este y verificar si se cumplieron o no.

FUNCIONALIDAD

SUP-01 Interoperabilidad entre bibliotecas. (Registrar libros, consultar libros)

USABILIDAD

No Aplica

CONFIABILIDAD

No Aplica.

DESEMPEÑO

SUP-02 Tiempo de respuestas óptimos

SOPORTABILIDAD

SUP-03 Arquitectura N-Capas

RESTRICCIONES DE DISEÑO

No Aplica.

IMPLEMENTACIÓN

No Aplica.

TABLA DE FACTORES

Factor	Escenarios de Calidad	Variabilidad	Impacto del factor	Prioridad para el éxito	Dificultad o Riesgo
Categoría de requisito					
Interoperabilidad entre bibliotecas	Permitir la interoperabilidad de 1 o más bibliotecas		El cumplimiento de esto favorece a la centralización de la información de los libros de las bibliotecas de la ciudad de Trujillo	Alto	Alto
Tiempo de respuestas óptimos	Los tiempos de respuesta deben de estar entre 3 a 4 segundos como máximo o en promedio.		El cumplimiento de este factor favorece al rendimiento al tiempo de respuestas que demandara consumir los servicios.	Alto	Alto
Arquitectura N-Capas	Permitir el desacoplamiento entre capas para una mayor escalabilidad.		El cumplimiento de este factor favorece a que la arquitectura del servicio web este desacoplado para posibles cambios en el futuro y pueda ser escalable fácilmente.	Medio	Medio

5. Documento de Arquitectura de Software

Documento de Arquitectura de Software

FINDBOOK

Historia de revisiones

Fecha	Versión	Descripción	Autor
2016-10-01	1	Descripción de la arquitectura en fase de elaboración.	Carlos Osorio Pérez
2016-10-10	2	Se mejoró la vista lógica.	Carlos Osorio Pérez
2016-11-24	3	Se agregó la vista de implementación.	Carlos Osorio Pérez
28-11-2016	4	Se modificaron los componentes y los nodos	Carlos Osorio Pérez

Introducción

Este artefacto presenta vistas arquitectónicas de la aplicación FindBook, las cuales representan los modelos más importantes del sistema guiados por los casos de uso arquitectónicamente más significativos.

La descripción de la arquitectura es importante porque nos ayudará a visualizar de una manera general como estará conformada la aplicación y así poder tener más claro que tarea o actividad realizar en cada una de ellas.

Factores y decisiones de la arquitectura

La tabla de factores se debe consultar en las Especificaciones Suplementarias.

A continuación, se muestran los memorándums técnicos donde se describe y justifica la solución propuesta a cada uno de los factores.

Memorándum Técnico 1
Asunto: Funcionalidad
Factores: <ul style="list-style-type: none">• Interoperabilidad entre bibliotecas
Solución: <p>Se debe de implementar un servicio web RestFull el cual servirá para que las bibliotecas puedan acceder y poder enviar información para registrar un determinado libro con la finalidad de poder tener la información de los libros de distintas bibliotecas centralizadas en un base de datos NOSQL.</p> <p>Este servicio web estará implementado en C# utilizando ASP NET Core y publicado en la nube de Microsoft Azure para que pueda ser accedido desde cualquier lugar, a su vez se utilizara una librería llamada FireSharp la cual permitirá la comunicación con la base de datos NoSql Firebase.</p>
Motivación: <p>Se propone un servicio web RestFull ya que a diferencia de SOAP este es más ligero en la transferencia de datos pese a que devuelve la información en un formato Json y SOAP en cambio devuelve en formato XML, por otro lado, la implementación de un servicio SOAP está orientada para soluciones empresariales las cuales tienen un catálogo de servicios y manejan</p>

estándares como los XSDL y RestFull no. Es por ello que se optó implementar un servicio RestFull ya que se propone utilizar una base de datos NoSql la cual contendrá grandes cantidades de datos.

Alternativas consideradas:

- Realizar el servicio web de tipo SOAP.

Memorándum Técnico 2

Asunto: **Desempeño**

Factores:

- **Tiempo de respuestas óptimos**

Solución:

Se debe de implementar una base de datos NOSQL para el manejo de grandes cantidades de datos, es por ello que se utilizara Firebase.

Motivación:

La base de datos FireBase se encuentra en un servidor de Google lo que facilita la conexión con el servicio web y a la vez el manejo con las grandes cantidades de datos.

Alternativas consideradas:

Existen muchas otras opciones en base de datos NoSQL como, por ejemplo:

- MongoDB, CouchDB, Hadoop, etc.

Se optó por Firebase ya que Google la ofrece como un servicio (gratis), y esto apoyaría en el sentido de que ya no habría la necesidad de levantar un propio servidor y publicarla en la nube, simplemente se configuraría para que el servicio pueda ser accedido ósea que este público y así poder consumirlo.

Memorándum Técnico 3

Asunto: **Soportabilidad**

Factores:

- **Arquitectura N-Capas**

Solución:

La implementación del servicio web RestFul estará bajo una arquitectura denominada N-Capas.

Motivación:

El motivo de implementar una arquitectura N-Capas, es porque permitirá que el código este desacoplado, entendible y contemple principios y patrones de diseño con el propósito que el servicio pueda extenderse fácilmente sin necesidad de modificar demasiadas cosas.

Alternativas consideradas:

Vista de Casos de Uso

Diagrama de paquetes de Casos de Uso

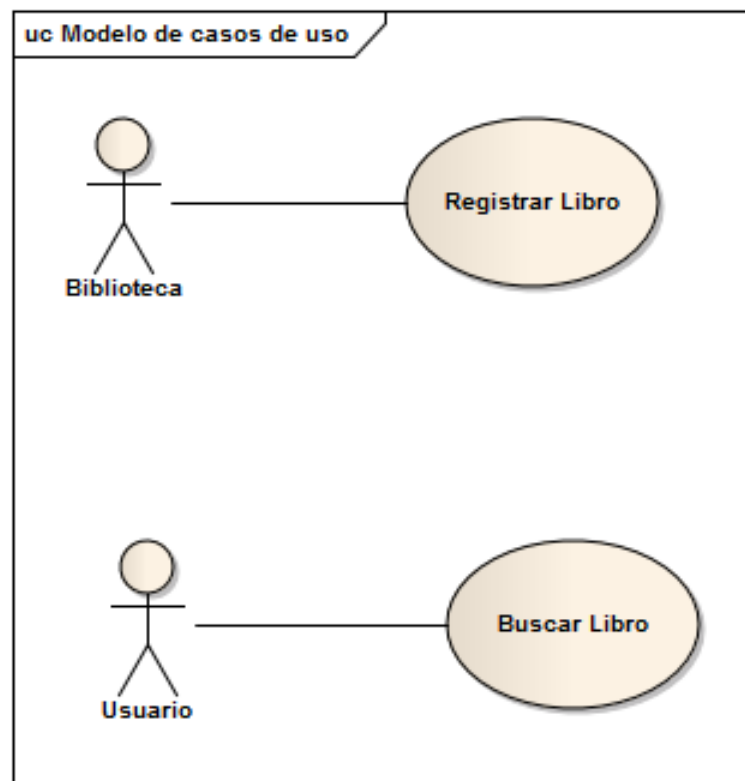


Figura 1: Diagrama de Casos de Uso

Priorización de Casos de Uso

CRITERIOS PARA VALORAR A LOS CASOS DE USO	PES O	RANG O
(RI) Riesgo tecnológico: complejo de implementar, nuevo, o puede ir variando debido a nuevas políticas o reglas de negocio.	3	0-3
(SA) Significativo para la arquitectura: debe cumplir características de calidad como fiabilidad, mantenibilidad, o rendimiento.	2	0-3
(NC) Naturaleza crítica: de valor para el negocio	1	0-3

Tabla de priorización para el sistema FindBook:

CASO DE USO	RI	SA	NC	PUNTAJE
CU - Registrar libro	3	3	3	18
CU - Buscar libro	2	3	3	15

Diagrama de Casos de Uso

Tabla de priorización agrupado por niveles de prioridad:

CASO DE USO	PRIORIDAD	COMENTARIO
CU - Registrar libro	Alta	Se debería planificar para implementar en fase de Elaboración
CU - Buscar libro		

Vista Lógica

Esquema general de la Arquitectura

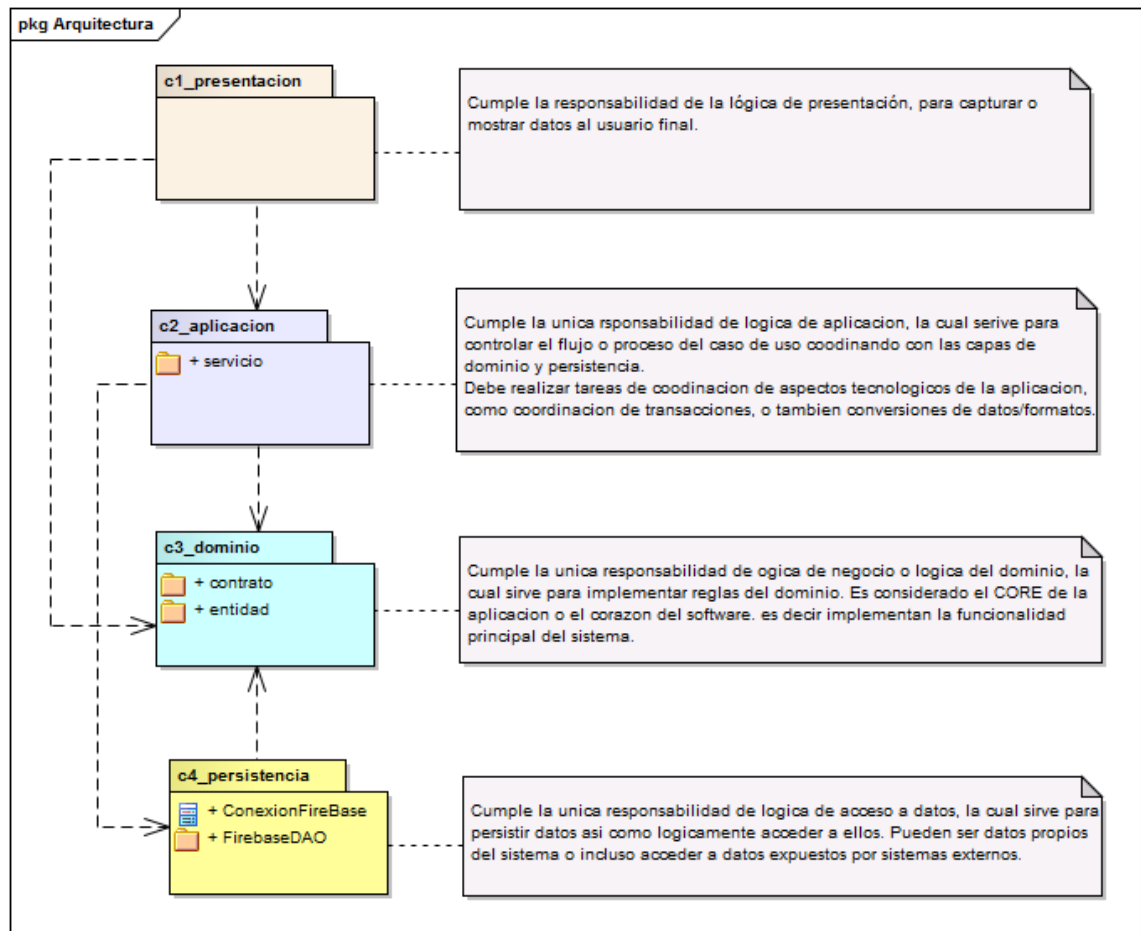


Figura 2: Esquema general de la Arquitectura de Software.

Cabe resaltar que esta arquitectura N-Capas para el Servicio Web no cuenta con la capa de Presentación ya que funcionará como un servicio el cual será expuesto tanto para las bibliotecas o para el usuario quienes lo consumirán.

Descripción de la Arquitectura

A. Capa de Aplicación: la figura 3 muestra las clases de la capa.

Todas las clases de la capa de aplicación deben ser nombradas de la siguiente forma: norma del caso de uso + “Servicio”. Por ejemplo, para el caso de uso Registrar y Buscar Libro sería, así como se muestra en la imagen:

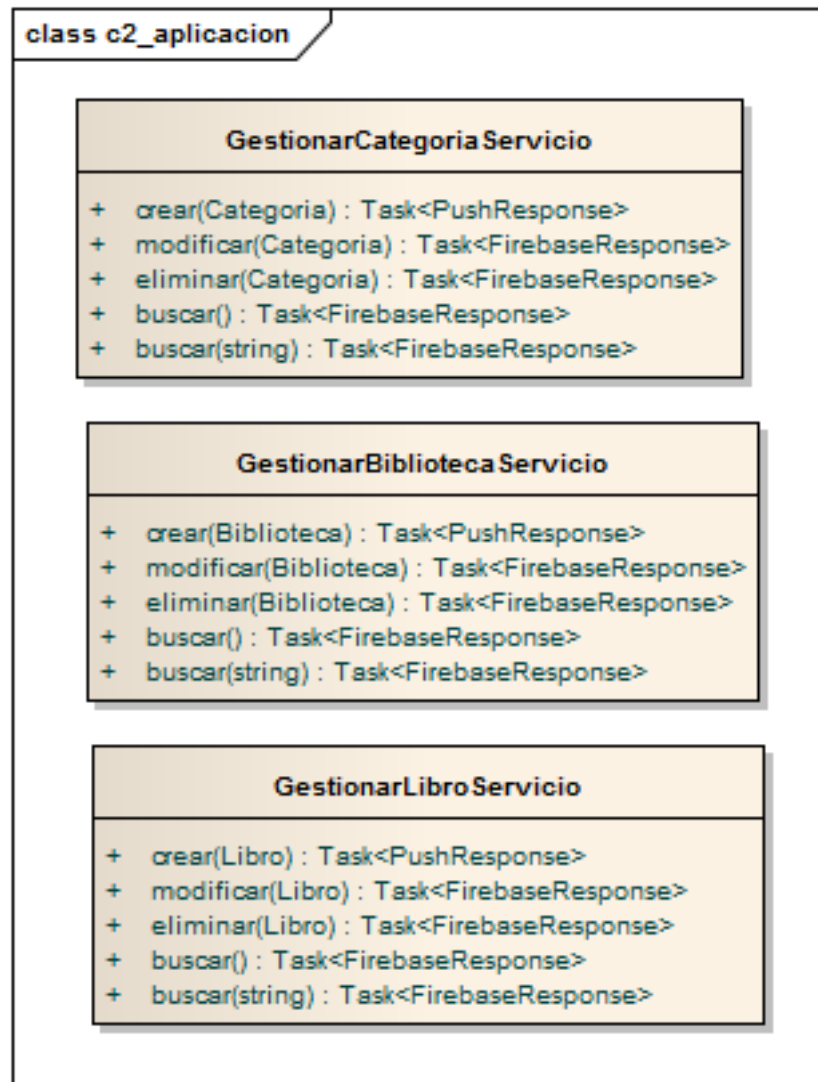


Figura 3: Diagrama de clases parcial de la capa “c2_aplicacion”

B. Capa de Dominio: la figura 4 muestra las subcapas de la capa dominio.

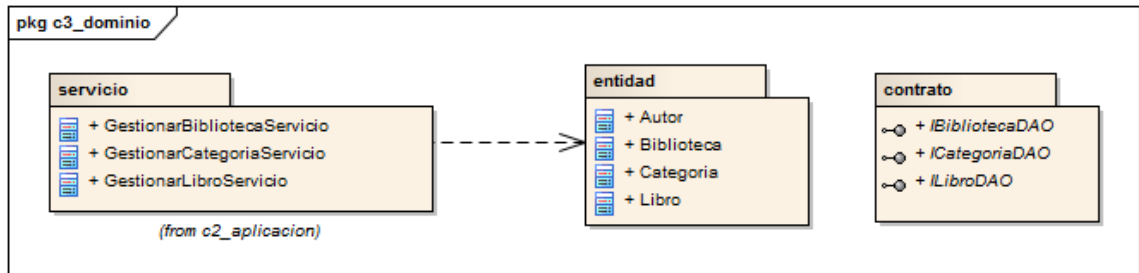


Figura 4: Diagrama de paquetes o subcapas de la capa “c3_dominio”

a. Subcapa Entidad: La figura 5 muestra las clases de la subcapa.

Esta subcapa representa la parte más significativa de la arquitectura por estar centrado en resolver o hacer cumplir las reglas de negocio. Las clases de diseño que aparecen en esta subcapa, representan entidades del dominio de problema que se está analizando, surgen de los conceptos del negocio y que son representados inicialmente en el Modelo de Dominio en una fase de análisis anterior.

Se debe colocar en las clases del diagrama sólo los atributos y operaciones más significativas, en especial las operaciones que resuelven reglas de negocio. Estas clases se nombran como los conceptos mismos del negocio.

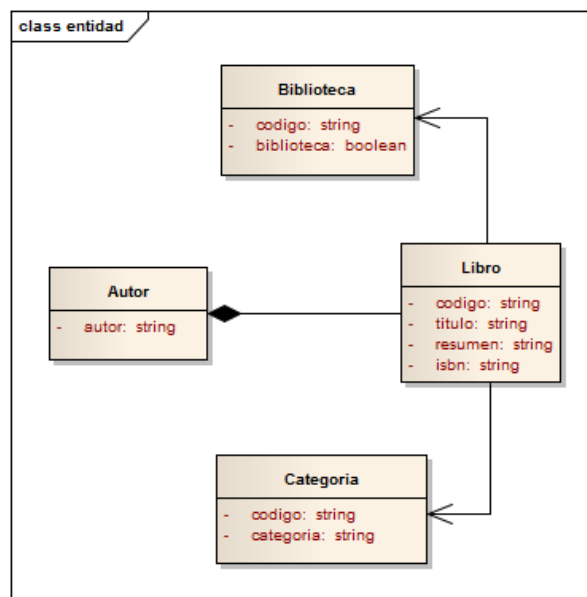


Figura 5: Diagrama de Clases parcial de la subcapa “entidades”

b. Subcapa Contrato: la figura 6 muestra las interfaces de la subcapa, estas interfaces definen las operaciones que serán realizadas por las clases de persistencia. Esta clase permitirá instanciar las clases que hacen parte de la capa de persistencia y poder lograr el desacoplamiento entre las capas “c2_aplicacion” y “c4_persistencia”

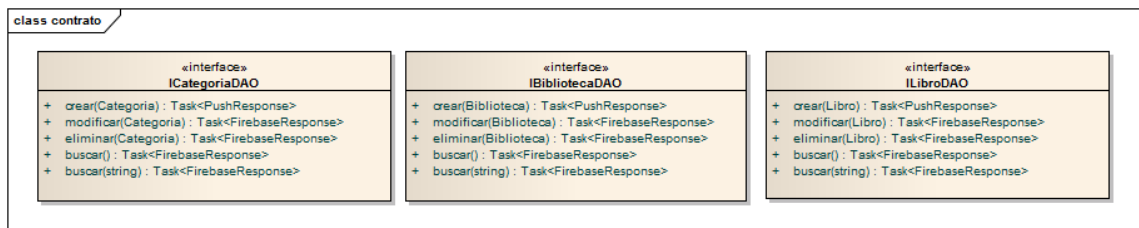


Figura 6: Diagrama de Clases e Interfaces parcial de la subcapa “contrato”

C. Capa de Persistencia: la figura 7 muestra la subcapa de la capa y a la vez la clase abstracta “GestorJDBC” encargada de gestionar el acceso a las bases de datos. Y la figura 8 muestra las clases de la subcapa “FirebaseDAO”.

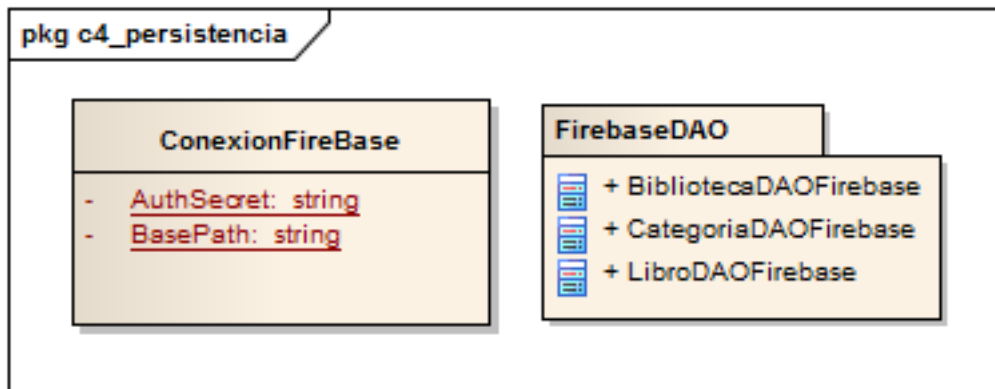


Figura 7: Diagrama de clases y paquete o subcapa de la capa “persistencia”



Figura 8: Diagrama de clases parcial de la subcapa “Firebase”

Vista de Implementación

Diagrama de Componentes

La figura 9 muestra el diagrama de componentes del servicio web FindBook. Siguiendo el patrón de arquitectura N-Capas, la aplicación será implementada en tres componentes, uno por cada capa lógica del diseño. Cada componente representa un archivo compilado con extensión .dll debido a que está desarrollado en lenguaje C#, además cada componente incluye todas las clases compiladas .class que serán generadas usando la herramienta VisualStudio 2015 Update 3.

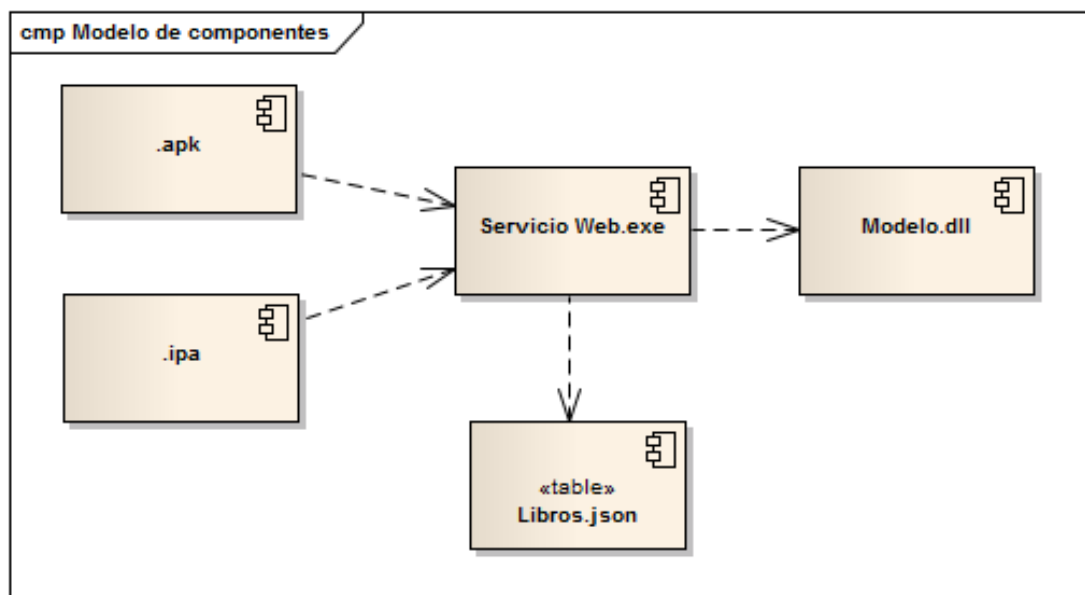


Figura 9: Diagrama de Componentes.

Modelo de Implementación

A continuación, se presenta la implementación del caso de uso registrar libro.

```
using System;
using System.Threading.Tasks;

using FireSharp;
using FireSharp.Interfaces;
using FireSharp.Response;

using Modelo.c3_dominio.contrato;
using Modelo.c3_dominio.entidad;

namespace Modelo.c4_persistencia.FireBase
{
    public class LibroDAOFirebase : ILibroDAO
    {
        private FirebaseClient firebaseClient;
        IFirebaseConfig configuracion = new ConexionFireBase().configuracion;

        public Task<PushResponse> crear(Libro libro,string
codigoUniversidad,string codigoCategoria)
        {
            firebaseClient = new FirebaseClient(configuracion);
            Task.Run(() => firebaseClient.PushAsync("libros/" + codigoUniversidad
+ "/" + codigoCategoria + "/" +codigoUniversidad+""+libro.isbn+"", libro));
            return Task.Run(() =>
firebaseClient.PushAsync("libro/" +codigoUniversidad+""+libro.isbn+"", libro));
        }
    }
}
```

Vista de Despliegue

Diagrama de Despliegue

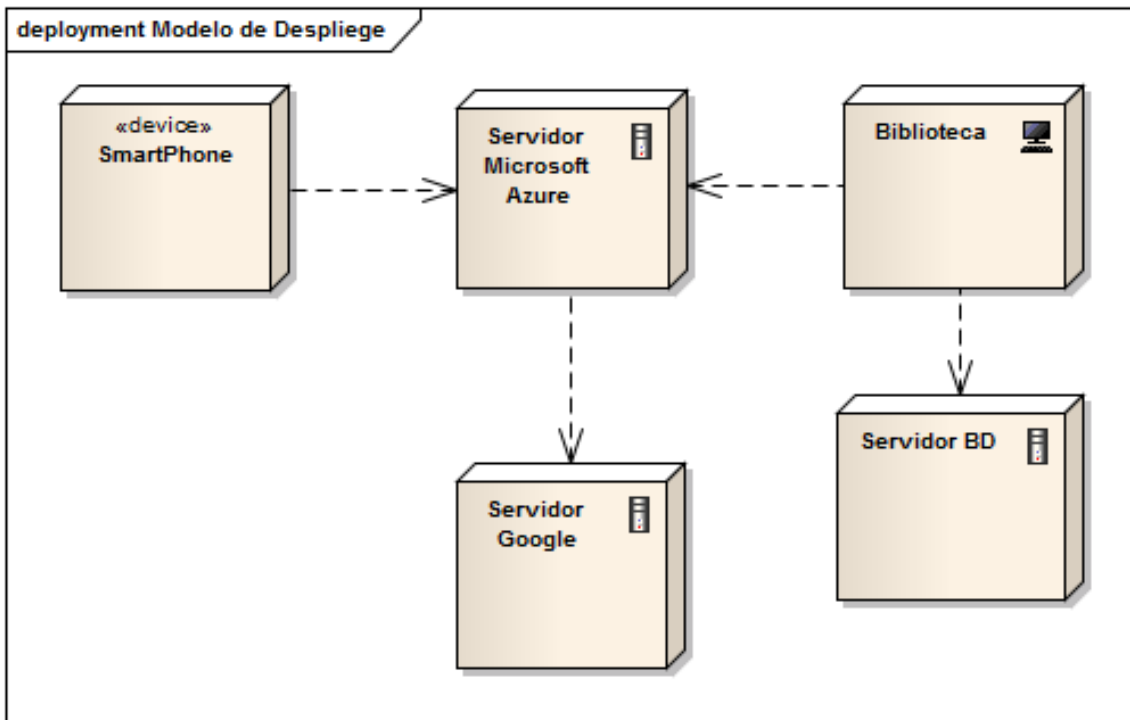


Figura 10: Diagrama de Despliegue

La siguiente arquitectura muestra el flujo de cómo se realiza la comunican del servicio web con las bibliotecas, con la base de datos y con los clientes que en este caso es para móvil.

Se podría decir que se divide en 2 procesos:

Proceso 1: Comunicación entre las bibliotecas y el Servicio Web

El Servicio Web expone un servicio el cual se encuentra en la siguiente dirección web: <http://serviciofindbook.azurewebsites.net/api/libro>, este servicio está diseñado exclusivamente para las bibliotecas, ya que permitirá a las bibliotecas poder registrar los libros, cabe señalar que para el registro de un libro se debe de utilizar el método POST.

Una vez que una determinada biblioteca envía el libro por el servicio web, el servicio realizara una serie de validaciones y finalmente se conectara con la base de datos NoSQL Firebase para realizar la persistencia de los datos.

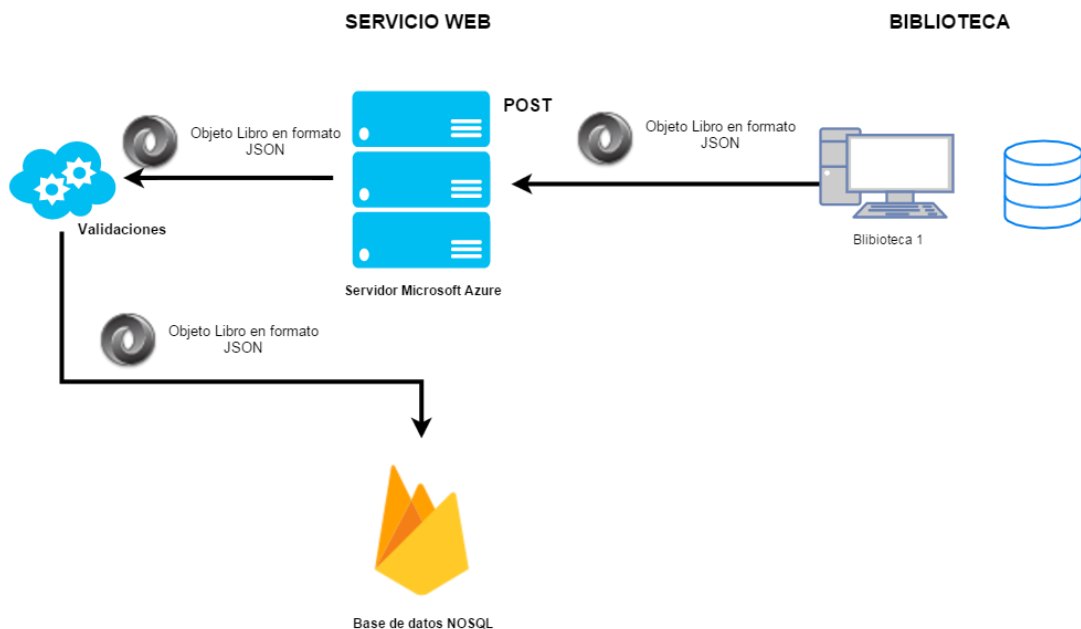


Figura 11: Diagrama de Comunicación entre las bibliotecas y el servicio web.

Proceso 2: Comunicación entre los clientes y el Servicio Web

El servicio web expone un servicio el cual se encuentra en la siguiente dirección web: <http://serviciofindbook.azurewebsites.net/api/libro>, este servicio está diseñado exclusivamente para los clientes, ya que permitirá a los clientes poder obtener los libros que busquen, cabe señalar que para la búsqueda de los libros se debe de utilizar el método GET.

Una vez que un determinado usuario abrió la aplicación y realizó la búsqueda de los libros por el servicio web, el servicio realizará la respectiva búsqueda conectándose con la base de datos NoSQL Firebase la cual le devolverá la información y el servicio realizará una transformación a los datos para la respectiva devolución de los mismos, finalmente la aplicación obtendrá la información del servicio web y lo mostrará en la aplicación móvil.

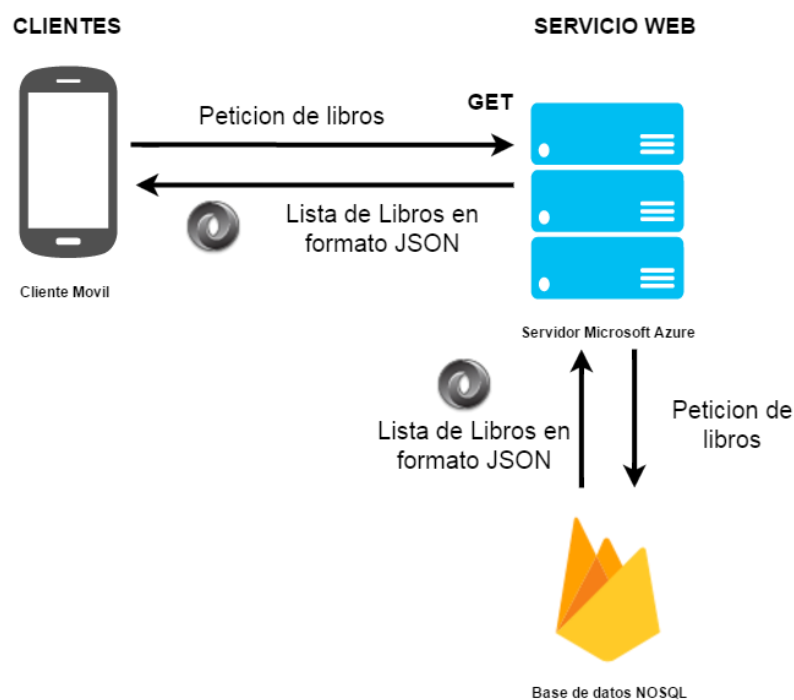


Figura 12: Diagrama de Comunicación entre las bibliotecas y el servicio web.

Nodos

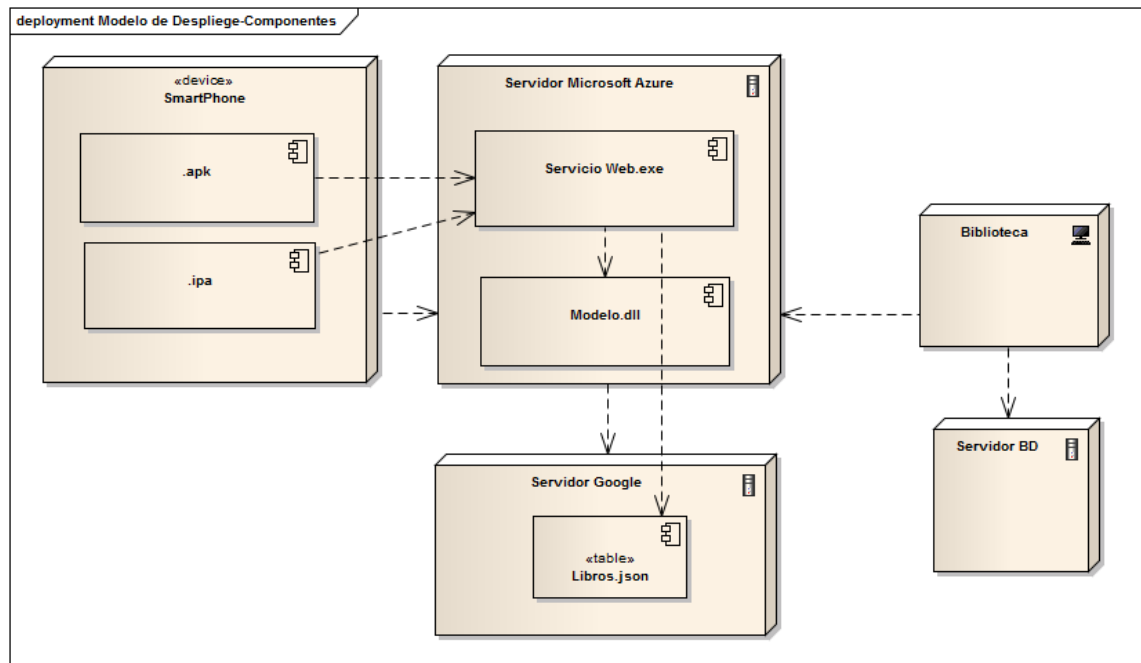


Figura 13: Diagrama de Nodos con sus Componentes.

Nodo 1: SmartPhone

Sera el encargado de consumir el servicio de buscar libros, los componentes a ejecutarse serán .apk o .ipa dependiendo del SO del SmartPhone.

Nodo 2: Servidor Microsoft Azure

Sera el encargado de exponer los servicios: registrar libros y buscar libros. Los componentes a ejecutarse en ese nodo serán Servicio Web.exe y Modelo.dll. El servicio estará desplegado en la nube de Microsoft y estará implementado en C# ASP NetCore, a su vez utilizará un plugin para poder conectarse con la base de datos Firebase.

Nodo 3: Servidor Google

Sera el encargado de exponer la base de datos NoSQL Firebase para poder realizar las acciones de registrar o buscar libros. El servicio de base de datos está desplegado en la nube de google ya que es un servicio propio de google. El componente a publicarse será la tabla que en este caso llamo Libros.

Nodo 4: Biblioteca – Servidor BD

Sera el encargado de consumir el servicio Registrar Libro, podrá implementarlo en lenguaje con el que el negocio trabaje, por ejemplo: java, c#, visual basic, etc.

6. Prueba de Caja Negra

Requerimientos Funcionales

Interoperabilidad entre bibliotecas

Registrar un determinado libro desde una biblioteca haciendo uso del servicio web, el cual estará en esta dirección: <http://serviciofindbook.azurewebsites.net/api/libro>, para esto la biblioteca al enviar la información deberá de enviar estos datos: título, resumen, isbn, autores, categoría y biblioteca. Finalmente, el envío de la información deberá ser mediante el método POST.

Condición	Clase Valida	Clase No Valida
El servicio debe estar activo.	1. el servicio este activo.	2. el servicio no este activo.
Se tiene que enviar el objeto Libro con los campos requeridos.	3. datos correctos.	4. datos incorrectos.
Enviar mediante el método post.	5. método post.	6. cualquier otro método.

Caso de Prueba:

NRO	CLASES	SERVICIO	CAMPOS	METODO	RESULTADO ESPERADO	RESULTADO OBTENIDO	PRUEBA
CP01	1,3,5	Si	Si	Si	Registro exitoso	Registro exitoso	VALIDA
CP02	2,3,5	No	Si	Si	Registro fallido	Registro fallido	VALIDA
CP03	1,4,5	Si	NO	SI	Registro fallido	Registro fallido	VALIDA
CP04	1,3,6	Si	Si	No	Registro fallido	Registro fallido	VALIDA

7. Prueba de Caja Blanca

Función para verificar que los datos del Json enviado al servicio son los necesarios para poder realizar el registro del libro:

15 references | 0/14 passing | Carlos Alfredo Osorio Pérez, 11 days ago | 1 author, 1 change

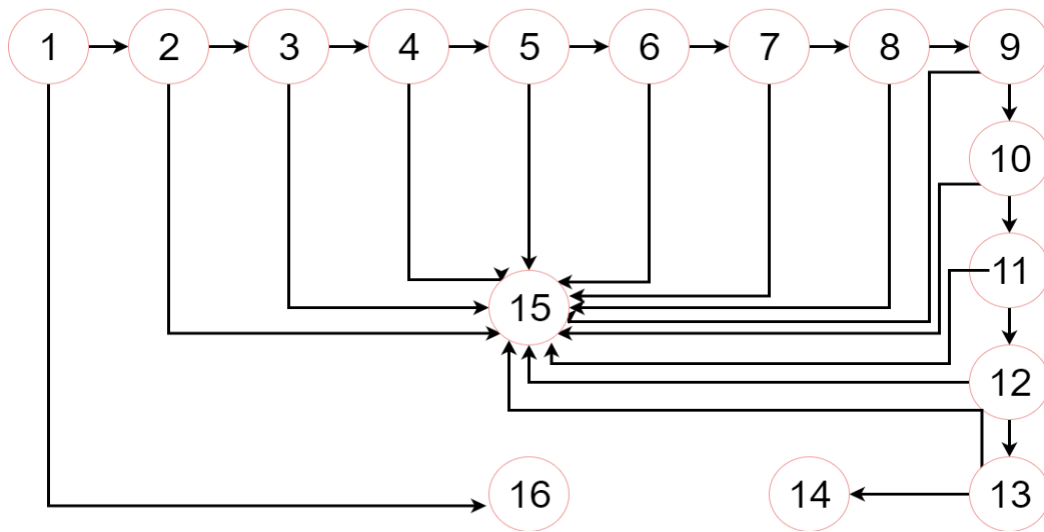
```
public bool validarJson(Libro libro)
{
    if (libro != null)
    {
        if (libro.titulo != null && libro.titulo != "" &&
            libro.resumen != null && libro.resumen != "" &&
            libro.isbn != null && libro.isbn != "" &&
            libro.autores != null && libro.autores.Count > 0 &&
            libro.categoria != null && libro.categoria.categoria != "" &&
            libro.biblioteca != null && libro.biblioteca.biblioteca != "")
            return true;
        else
            return false;
    }
    else
        return false;
}
```

Identificación de nodos en el código.

15 references | 0/14 passing | Carlos Alfredo Osorio Pérez, 11 days ago | 1 author, 1 change

```
public bool validarJson(Libro libro)
{
    if (libro != null) 1
    {
        if (libro.titulo != null && libro.titulo != "" &&
            4 libro.resumen != null && libro.resumen != "" && 5
            6 libro.isbn != null && libro.isbn != "" && 7
            8 libro.autores != null && libro.autores.Count > 0 && 9
            10 libro.categoria != null && libro.categoria.categoria != "" && 11
            12 libro.biblioteca != null && libro.biblioteca.biblioteca != "") 13
            return true; 14
        else
            return false; 15
    }
    else
        return false; 16
}
```


Diagrama del grafo de flujo.



Calcular la complejidad ciclomatica

$$V(G) = a - n + 2$$

$$V(G) = 26 - 16 + 2$$

$$V(G) = 12$$

Encontrar los caminos básicos

C1	1,16
C2	1,2,15
C3	1,2,3,15
C4	1,2,3,4,15
C5	1,2,3,4,5,15
C6	1,2,3,4,5,6,15
C7	1,2,3,4,5,6,7,15
C8	1,2,3,4,5,6,7,8,15
C9	1,2,3,4,5,6,7,8,9,15
C10	1,2,3,4,5,6,7,8,9,10,15
C11	1,2,3,4,5,6,7,8,9,10,11,15
C12	1,2,3,4,5,6,7,8,9,10,11,12,15
C13	1,2,3,4,5,6,7,8,9,10,11,12,13,15
C14	1,2,3,4,5,6,7,8,9,10,11,12,13,14

Definir caso de prueba por cada camino

CASO	CAMINO	VALORES DE ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO
C1	1,16	<code>Libro libro1 = null;</code>	False	False
C2	1,2,15	<pre> Libro libro1 = new Libro { titulo = null, resumen = "resumen del libro", isbn = "1449358063", autores = new List<Autor> { new Autor { nombre = "Mike Amundsen" }, new Autor { nombre = "Sam Ruby" } }, categoria = new Categoria { categoria = "SISTEMAS" }, biblioteca = new Biblioteca { biblioteca = "UCV" } }; </pre>	False	False

C3	1,2,3,15	<pre> Libro libro1 = new Libro { titulo = "", resumen = "resumen del libro", isbn = "1449358063", autores = new List<Autor> { new Autor { nombre = "Mike Amundsen" }, new Autor { nombre = "Sam Ruby" } }, categoria = new Categoria { categoria = "SISTEMAS" }, biblioteca = new Biblioteca { biblioteca = "UCV" } }; </pre>	False	False
C4	1,2,3,4,15	<pre> Libro libro1 = new Libro { titulo = "RESTful Web APIs", resumen = null, isbn = "1449358063", autores = new List<Autor> { new Autor { nombre = "Mike Amundsen" }, new Autor { nombre = "Sam Ruby" } }, categoria = new Categoria { categoria = "SISTEMAS" }, biblioteca = new Biblioteca { biblioteca = "UCV" } }; </pre>	False	False

C5	1,2,3,4,5,15	<pre> Libro libro1 = new Libro { titulo = "RESTful Web APIs", resumen = "", isbn = "1449358063", autores = new List<Autor> { new Autor { nombre = "Mike Amundsen" }, new Autor { nombre = "Sam Ruby" } }, categoria = new Categoria { categoria = "SISTEMAS" }, biblioteca = new Biblioteca { biblioteca = "UCV" } }; </pre>	False	False
C6	1,2,3,4,5,6,15	<pre> Libro libro1 = new Libro { titulo = "RESTful Web APIs", resumen = "resumen del libro", isbn = null, autores = new List<Autor> { new Autor { nombre = "Mike Amundsen" }, new Autor { nombre = "Sam Ruby" } }, categoria = new Categoria { categoria = "SISTEMAS" }, biblioteca = new Biblioteca { biblioteca = "UCV" } }; </pre>	False	False

C7	1,2,3,4,5,6,7,15	<pre> Libro libro1 = new Libro { titulo = "RESTful Web APIs", resumen = "resumen del libro", isbn = "", autores = new List<Autor> { new Autor { nombre = "Mike Amundsen" }, new Autor { nombre = "Sam Ruby" } }, categoria = new Categoria { categoria = "SISTEMAS" }, biblioteca = new Biblioteca { biblioteca = "UCV" } }; </pre>	False	False
C8	1,2,3,4,5,6,7,8,15	<pre> Libro libro1 = new Libro { titulo = "RESTful Web APIs", resumen = "resumen del libro", isbn = "1449358063", autores = null, categoria = new Categoria { categoria = "SISTEMAS" }, biblioteca = new Biblioteca { biblioteca = "UCV" } }; </pre>	False	False

C9	1,2,3,4,5,6,7,8,9,15	<pre> Libro libro1 = new Libro { titulo = "RESTful Web APIs", resumen = "resumen del libro", isbn = "1449358063", autores = new List<Autor> { }, categoria = new Categoria { categoria = "SISTEMAS" }, biblioteca = new Biblioteca { biblioteca = "UCV" } }; </pre>	False	False
C10	1,2,3,4,5,6,7,8,9,10,15	<pre> Libro libro1 = new Libro { titulo = "RESTful Web APIs", resumen = "resumen del libro", isbn = "1449358063", autores = new List<Autor> { new Autor { nombre = "Mike Amundsen" }, new Autor { nombre = "Sam Ruby" } }, categoria = null, biblioteca = new Biblioteca { biblioteca = "UCV" } }; </pre>	False	False

C11	1,2,3,4,5,6,7,8,9,10,11,15	<pre> Libro libro1 = new Libro { titulo = "RESTful Web APIs", resumen = "resumen del libro", isbn = "1449358063", autores = new List<Autor> { new Autor { nombre = "Mike Amundsen" }, new Autor { nombre = "Sam Ruby" } }, categoria = new Categoria { categoria = "" }, biblioteca = new Biblioteca { biblioteca = "UCV" } }; </pre>	False	False
C12	1,2,3,4,5,6,7,8,9,10,11,12,15	<pre> Libro libro1 = new Libro { titulo = "RESTful Web APIs", resumen = "resumen del libro", isbn = "1449358063", autores = new List<Autor> { new Autor { nombre = "Mike Amundsen" }, new Autor { nombre = "Sam Ruby" } }, categoria = new Categoria { categoria = "SISTEMAS" }, biblioteca = null, }; </pre>	False	False

C13	1,2,3,4,5,6,7,8,9,10,11,12,13,15	<pre> Libro libro1 = new Libro { titulo = "RESTful Web APIs", resumen = "resumen del libro", isbn = "1449358063", autores = new List<Autor> { new Autor { nombre = "Mike Amundsen" }, new Autor { nombre = "Sam Ruby" } }, categoria = new Categoria { categoria = "SISTEMAS" }, biblioteca = new Biblioteca { biblioteca = "" } }; </pre>	False	False
C14	1,2,3,4,5,6,7,8,9,10,11,12,13,14	<pre> Libro libro1 = new Libro { titulo = "RESTful Web APIs", resumen = "resumen del libro", isbn = "1449358063", autores = new List<Autor> { new Autor { nombre = "Mike Amundsen" }, new Autor { nombre = "Sam Ruby" } }, categoria = new Categoria { categoria = "SISTEMAS" }, biblioteca = new Biblioteca { biblioteca = "UCV" } }; </pre>	True	True

Resultados:

▲ Passed Tests (14)	
✓ TestMethod1	30 ms
✓ TestMethod10	< 1 ms
✓ TestMethod11	< 1 ms
✓ TestMethod12	< 1 ms
✓ TestMethod13	< 1 ms
✓ TestMethod14	< 1 ms
✓ TestMethod2	1 ms
✓ TestMethod3	< 1 ms
✓ TestMethod4	< 1 ms
✓ TestMethod5	< 1 ms
✓ TestMethod6	< 1 ms
✓ TestMethod7	< 1 ms
✓ TestMethod8	< 1 ms
✓ TestMethod9	< 1 ms