**DEFINICION DE MODELO DE ARQUITECTURA PARA APLICACIONES DESARROLLADAS EN EL ICFES BASADAS EN MICROSERVICIOS.**

**Farid Eliecer Ureche López**

**UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA**

**ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA**

**CEAD SANTA MARTA**

**BOGOTÁ, OCTUBRE 2020**

**DEFINICION DE MODELO DE ARQUITECTURA PARA APLICACIONES DESARROLLADAS EN EL ICFES BASADAS EN MICROSERVICIOS.**

**Trabajo de grado para optar por el Título de Magister en Gestión de TI**

**Farid Eliecer Ureche López**

**Tutor**

**xxxxxxxxxxxxxxx**

**xxxxxxxxxxxxxx**

**UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA**

**ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA**

**CEAD SANTA MARTA**

**BOGOTÁ, OCTUBRE 2020**

**Contenido**

[LISTADO DE IMAGENES VI](#_Toc49703981)

[LISTADO DE TABLAS VII](#_Toc49703982)

[DEDICATORIA VIII](#_Toc49703983)

[ABSTRACT IX](#_Toc49703984)

[RESUMEN X](#_Toc49703985)

[INTRODUCCIÓN 11](#_Toc49703986)

[CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA 11](#_Toc49703987)

[1.1 Descripción de la Realidad Problemática 11](#_Toc49703988)

[1.2 Identificación y Formulación del Problema 12](#_Toc49703989)

[1.3 OBJETIVOS DE LA INVESTIGACIÓN 12](#_Toc49703990)

[1.3.1 Objetivo General 12](#_Toc49703991)

[1.3.2 Objetivos específicos 12](#_Toc49703992)

[1.5 LIMITACIONES DE LA INVESTIGACIÓN 14](#_Toc49703993)

[1.5.1 Delimitación teórica 14](#_Toc49703994)

[1.5.2 Delimitación Geográfica 14](#_Toc49703995)

[CAPÍTULO II: MARCO TEÓRICO 15](#_Toc49703996)

[2.1 Bases Técnicas 15](#_Toc49703997)

[2.1.1 Modelo de sistema 15](#_Toc49703998)

[2.1.2 Arquitectura 15](#_Toc49703999)

[2.1.3 Microservio 16](#_Toc49704000)

[2.1.4 Arquitectura de microservicios 16](#_Toc49704001)

[2.1.5 Patrón 18](#_Toc49704002)

[2.1.6 Microservicio orquestación 18](#_Toc49704003)

[2.1.7 Microservicio coreográfico 18](#_Toc49704004)

[2.2 Patrones de microservicios 19](#_Toc49704005)

[2.2.1 Descubrimiento de servicio 20](#_Toc49704006)

[2.2.2 Balanceo de carga. 20](#_Toc49704007)

[2.2.3 Observabilidad y telemetría. 20](#_Toc49704008)

[2.2.4 Comunicación -API GATEWAY 22](#_Toc49704009)

[2.2.5 Seguridad 22](#_Toc49704010)

[2.2.6 Configuración 23](#_Toc49704011)

[2.3 Arquitectura de software 23](#_Toc49704012)

[2.3.1 Proceso de Arquitectura de Software 24](#_Toc49704013)

[2.3.2 Modelos o Vistas Arquitecturales 25](#_Toc49704014)

[2.3.3 Diseño basado en el dominio (DDD) 26](#_Toc49704015)

[2.3.4 Arquitectura Hexagonal 26](#_Toc49704016)

[CAPÍTULO III: METODOLOGÍA 26](#_Toc49704017)

[3.1. Investigación cualitativa. 26](#_Toc49704018)

[3.2. La Investigación de Estudio de Casos. 28](#_Toc49704019)

[3.3. Investigación / Acción. 29](#_Toc49704020)

[3.4. Tipo y Nivel de la investigación 30](#_Toc49704021)

[CAPÍTULO IV: DESARROLLO DE LA PROPUESTA 32](#_Toc49704022)

[CAPÍTULO V: PRESENTACIÓN Y ANÁLISIS DE RESULTADOS 34](#_Toc49704023)

[5.1. Presentación de resultados 34](#_Toc49704024)

[5.2.1 Validación de la arquitectura 34](#_Toc49704025)

[5.2.1.1 Resultados 34](#_Toc49704026)

[5.2.2 Análisis cualitativo 34](#_Toc49704027)

[5.2.2.1 Análisis e interpretación de resultados del instrumento 34](#_Toc49704028)

[5.3 Conclusiones 34](#_Toc49704029)

[5.4 Recomendaciones 34](#_Toc49704030)

[REFERENCIAS 34](#_Toc49704031)

# LISTADO DE IMAGENES

# LISTADO DE TABLAS

# RESUMEN

En los últimos años con las tendencias de las empresas de sistematizar y democratizar los servicios que prestan han llevado estos servicios a la internet, ofreciéndole a si a sus clientes la facilidad de atenderle sus necesidades desde la comodidad de su silla en la oficina u hogar. Este crecimiento de los servicios en línea ha obligado a las empresas a trasladar o replicar sus servicios en línea, con lo que es necesario la creación de aplicaciones o la migración o modernización de las actuales. Dada la cantidad de usuarios concurrentes que puedan tener estas aplicaciones se ha disparado la necesidad de crear aplicaciones que permitan prestar el servicio más rápido y a más usuarios concurrentes.

En este orden de idea las empresas con aplicaciones legadas han encontrado un inconveniente muy fuerte dado que muchas de estas aplicaciones no cuentan con la facilidad de escalabilidad y así se ven en la obligación de la construcción de aplicaciones nuevas con nuevas arquitecturas o mejoras de la arquitectura actual para así poder ofrecerles a sus usuarios unas buenas experiencias con sus servicios.

Por lo anterior se decide diseñar un modelo de arquitectura basado en microservicios que sirva como base y guía en la implementación o actualización de aplicaciones de alta concurrencia.

# INTRODUCCIÓN

# CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

## Descripción de la Realidad Problemática

Las aplicaciones y software de grandes empresas a través del tiempo van sufriendo transformaciones, por recorte de funcionalidades, incorporación de nuevas funcionalidades o desuso de algunas. Las aplicaciones de más de 10 años de creación en su mayoría están basadas en la arquitectura cliente – servidor, utilizando un patrón de diseño por capas, definiendo una aplicación con muchos módulos como el gran monstruo con ciertas limitaciones de crecimiento y mantenibilidad. Por lo que generan que en estas empresas crezca la filosofía de “Si funciona no lo toque”.

Actualmente en el ICFES las aplicaciones se encuentran con inconvenientes de interoperabilidad, desempeño y complicaciones para realizar modificaciones y mejoras. Al momento de escalar las aplicaciones es necesario contar con una gran cantidad de recursos dado que las aplicaciones no permiten ser escaladas por módulos, sino que toca duplicar literalmente los servidores.

En los últimos años experiencias de empresas como Spotify, Netflix, eBay, Amazon, el Servicio Digital del Gobierno del Reino Unido, Twitter, PayPal, entre otras. Han generado la inquietud de ¿Cuál es el éxito detrás de las aplicaciones basadas en arquitectura de microservicios?

## Identificación y Formulación del Problema

¿Cómo implementar aplicaciones basadas en arquitectura de microservicios en el ICFES?

## 1.3 OBJETIVOS DE LA INVESTIGACIÓN

## 1.3.1 Objetivo General

Definir un modelo de arquitectura para aplicaciones desarrolladas en el ICFES basadas en microservicios.

## 1.3.2 Objetivos específicos

* Definir a octubre de 2020, patrones de diseño y arquitectura de microservicios para las aplicaciones desarrolladas en el ICFES.
* Diseñar a octubre de 2020, modelo de desarrollo para microservicios.
* Implementar aplicación PLAPPI utilizando el estándar definido a diciembre de 2020.

**1.4 JUSTIFICACIÓN DE LA INVESTIGACIÓN**

El ICFES cuenta con una cantidad significativa de aplicaciones desarrolladas bajo la estrategia de aplicaciones monolíticas, las cuales al pasar el tiempo vienen presentado diversos inconvenientes, como son rendimiento, dificultades para agregar nuevas funcionalidades, complicaciones al momento de realizar modificaciones, dificultad para encontrar errores, falta de flexibilidad para integración con otros sistemas, entre otros.

En el momento de iniciar algunos procesos que conllevan a que un número significativo de usuarios utilicen las aplicaciones de manera concurrente, como lo son inscripción a exámenes, descarga de resultados, verificación de citación, se presentan problemas de rendimiento, por otro lado, se requiere una gran infraestructura para soportar los procesos, las PQRs por problemas en los sistemas se incrementan significativamente.

Teniendo en cuenta la situación actual mundial la que esta llevando a las empresas a realizar la mayoría de sus procesos de manera virtual, el ICFES está trabajando en la implementación de sus pruebas virtuales, lo que conlleva al mejoramiento de las aplicaciones actuales, con el fin de reducir la cantidad de recursos de infraestructura utilizados, capacidad de recurso humano, entre otros, los cuales pueden ser utilizados en los nuevos desarrollos.

Por otra parte, la implementación de metodologías agiles por parte de los equipos se dificulta por la forma como están definidas las aplicaciones. De igual manera las migraciones de base de datos, mejoras al sistema o inclusión de nuevas funcionalidades.

## LIMITACIONES DE LA INVESTIGACIÓN

## 1.5.1 Delimitación teórica

Se estudiarán metodologías, buenas prácticas, patrones de diseño y patrones de arquitectura, utilizados en la construcción de aplicaciones basadas en microservicios, con el fin de describir una arquitectura estándar para las aplicaciones desarrolladas en el ICFES.

## 1.5.2 Delimitación Geográfica

El proyecto se llevará a cabo en la sede del ICFES, ubicada en la ciudad de Bogotá.

# CAPÍTULO II: MARCO TEÓRICO

## 2.1 Bases Técnicas

### 2.1.1 Modelo de sistema

Un modelo es una especificación formal de la función, estructura y comportamiento de un sistema dentro de un determinado contexto, y desde un punto de vista específico (o punto de referencia). Un modelo suele estar representado por una combinación de dibujos y texto, normalmente con una notación formal como UML, aumentada con expresiones de lenguaje natural. Object Management Group.

según UML un modelo de sistema tiene las siguientes características:

* Representación
  + Funciones
  + Estructura
  + Comportamiento

### 2.1.2 Arquitectura

La arquitectura es la organización fundamental de un sistema incorporado en sus componentes, sus relaciones entre sí y con el medio ambiente, y el principio que guía su diseño y evolución. (ISO/IEC 42010:2007).

Por otro lado, Roy tomas define, La estructura de los componentes, sus interrelaciones y los principios y directrices que rigen su diseño y evolución en el tiempo. Una arquitectura de software es una abstracción de los elementos de tiempo de ejecución de un sistema de software durante alguna fase de su funcionamiento. Un sistema puede estar compuesto por muchos niveles de abstracción y muchas fases de operación, cada una con su propia arquitectura de software.

Una arquitectura de software se define mediante una configuración de elementos arquitectónicos (componentes, conectores y datos) restringidos en sus relaciones para lograr un conjunto deseado de propiedades arquitectónicas.

(Fielding, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. 2000)

### 2.1.3 Microservio

Los microservicios son unidades funcionales concretas e independientes, que trabajan juntas para ofrecer la funcionalidad general de una aplicación. Cada microservicio puede ser actualizado o escalado sin que esto afecte a la disponibilidad de las demás unidades y de la aplicación en su conjunto. Los microservicios son los principales componentes de una Arquitectura de Microservicios. ([Marta Benedet](https://blog.mdcloud.es/author/marta-b/), 2018).

### 2.1.4 Arquitectura de microservicios

Según Chris Richardson, es un conjunto de servicios que colaboran entre si debidamente acoplados, que cuentas con las siguientes características:

* Altamente mantenible y comprobable: permite un desarrollo e implementación rápidos y frecuentes.
* Acoplado libremente con otros servicios: permite que un equipo trabaje de forma independiente la mayor parte del tiempo en sus servicios sin verse afectado por cambios en otros servicios y sin afectar otros servicios.
* Implementable de forma independiente: permite que un equipo implemente su servicio sin tener que coordinarse con otros equipos.
* Capaz de ser desarrollado por un equipo pequeño: esencial para una alta productividad al evitar la alta comunicación del jefe de equipos grandes

Los servicios se comunican mediante protocolos sincrónicos como HTTP / REST o protocolos asincrónicos como AMQP. Los servicios se pueden desarrollar e implementar independientemente unos de otros. Cada servicio tiene su propia base de datos para desacoplarse de otros servicios.



Figura xx: Arquitectura de microservicios. Tomada de: https://docs.microsoft.com/

### 2.1.5 Patrón

Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno y luego describe el núcleo de la solución a ese problema, de tal forma que pueda usarse la solución muchas veces más, sin tener que repetir el proceso más de una vez. (Christopher Alexander).

### 2.1.6 Microservicio orquestación

El término orquestación de servicios se refiere a la coordinación de múltiples servicios a través de un mediador centralizado, como un consumidor de servicios o un centro de integración (Mule, Camel, Spring Integration, etc.). (Mark Richards).

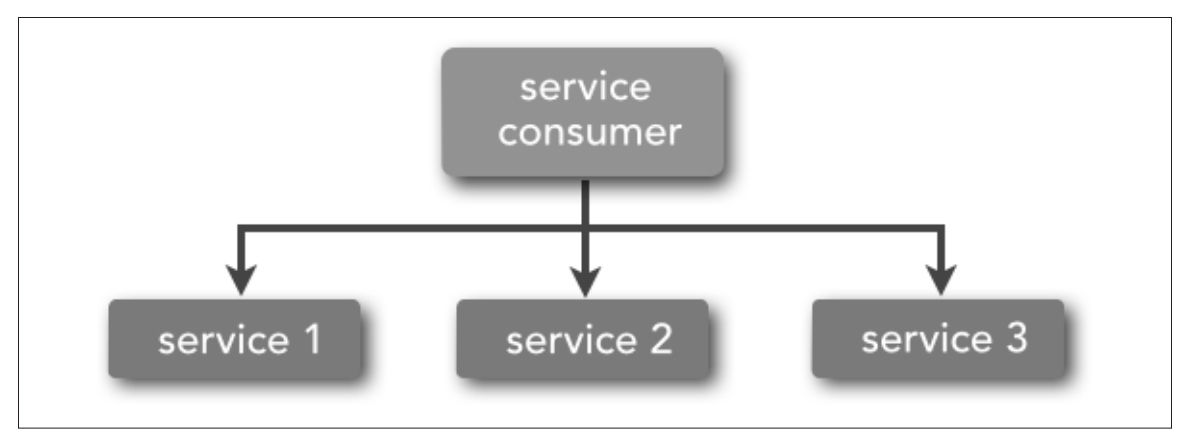


Figura xx : microservicio orquestación, tomada del libro Microservices vs. ServiceOriented Architecture.

### 2.1.7 Microservicio coreográfico

La coreografía de servicio se refiere a la coordinación de múltiples servicios. llamadas sin un mediador central. El término comunicación inter-servicios a veces se utiliza junto con la coreografía de servicio.

Con la coreografía de servicio, un servicio llama a otro servicio, que puede llamar a otro servicio y así sucesivamente, realizando lo que también se refiere como encadenamiento de servicios. (Mark Richards).

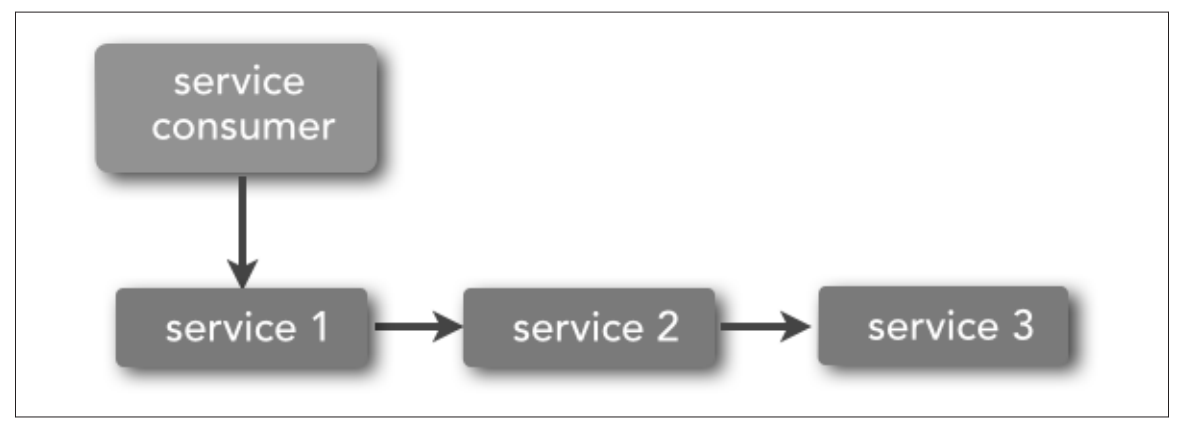


Figura xx: microservicio coreográfico Tomada del libro Microservices vs. ServiceOriented Architecture.

## 2.2 Patrones de microservicios

Patrones asociados al patrón de microservicios:



Figura xx: Patrones asociados al patrón microservicio. Tomada de https://microservices.io/.

### 2.2.1 Descubrimiento de servicio

Enrutamiento y descubrimiento es una capacidad básica para la comunicación entre microservicios atenta a la nube. Una instancia de microservicios podría ser autoescalada y tener “clusters” dinámicos, por ejemplo, y usted debería descubrir el microservicio por nombre y luego encaminarse a una instancia de ejecución. Los marcos como Zuul/Eureka proveen esto como parte de la pila Netflix OSS. Istio es una opción preferida que tiene base políglota y proporciona control de grano más fino del enrutamiento de las llamadas de servicio y control a la capa subyacente. La figura siguiente muestra un ejemplo de la arquitectura Istio.

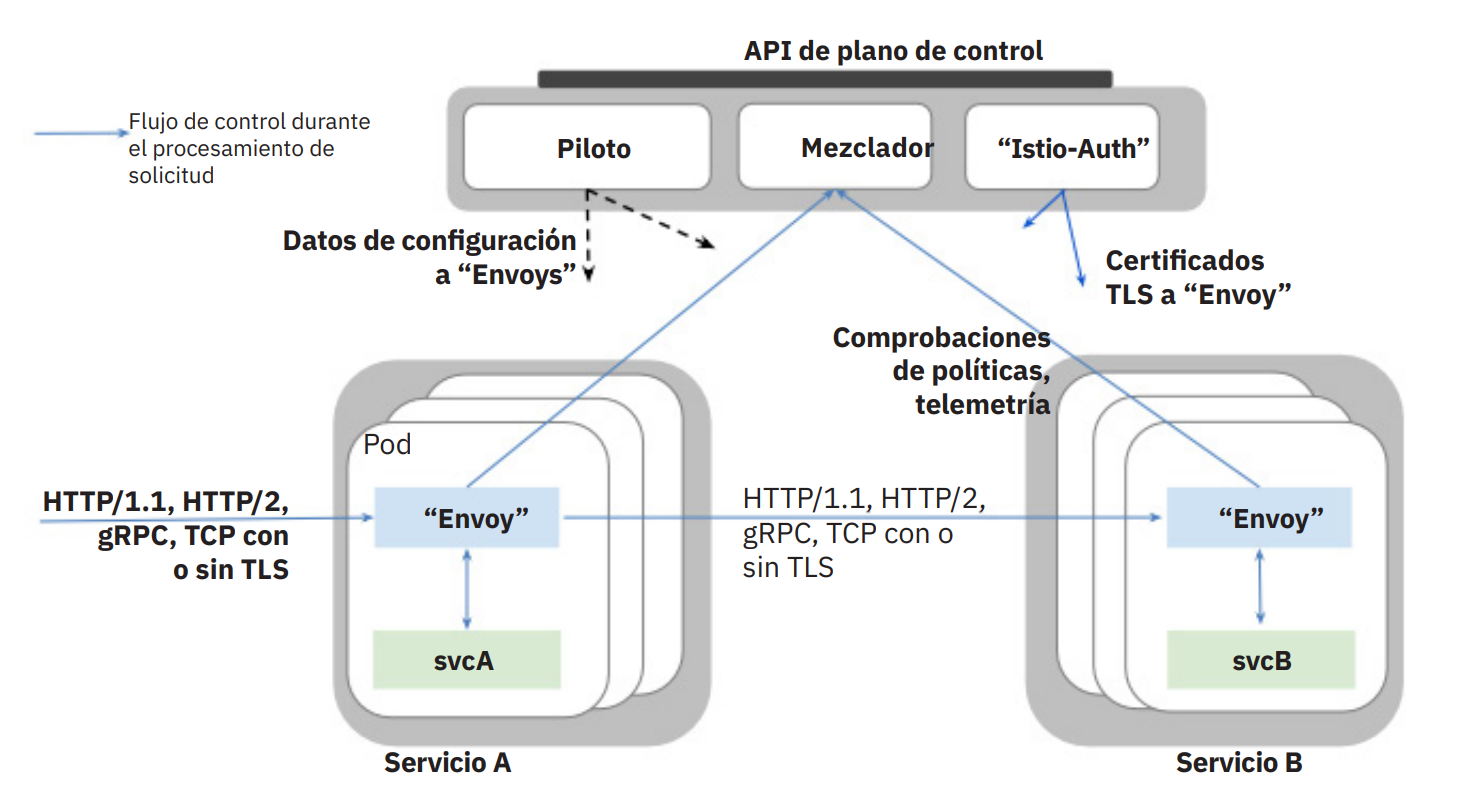


Figura xx: arquitectura Istio. tomado de Entender los microservicios.

### 2.2.2 Balanceo de carga.

### 2.2.3 Observabilidad y telemetría.

IBM Cloud, menciona que hay un cambio cultural en torno a la supervisión que se produce con un cambio a la nube nativa. Aunque se espera que las aplicaciones tanto locales como nativas de la nube tengan una alta disponibilidad y resistencia a errores, los métodos utilizados para lograr esos objetivos son diferentes. Como resultado, la finalidad de la supervisión cambia: en lugar de realizar la supervisión para evitar la anomalía, se realiza para gestionar la anomalía.

Una aplicación nativa de la nube está formada por microservicios independientes y los servicios de respaldo necesarios. Aunque una aplicación nativa en la nube como un todo debe permanecer disponible y seguir funcionando, se iniciarán o detendrán instancias de servicio individuales para que se ajusten a los requisitos de capacidad o para recuperarse de una anomalía.

#### 2.2.3.1 Observabilidad

La supervisión de este sistema fluido requiere que cada participante sea *observable*. Cada entidad debe producir los datos adecuados para dar soporte a la creación de alertas y la detección de problemas automatizados, a la depuración manual cuando sea necesario y al análisis del estado del sistema (tendencias históricas y análisis).

Los servicios deben generar los siguientes datos para poder ser observables.

* Comprobaciones de estado.
* Métricas.
* Entradas de registro.
* Rastreo distribuido, de solicitudes o de extremo a extremo.

#### 2.2.3.2 Métricas

Según IBM Cloud, Las anomalías ya no son ocurrencias disruptivas y raras. El desglose de una aplicación monolítica en microservicios empuja una parte mayor de la vía de acceso principal a la red, aumentando el impacto de la latencia y de otros problemas de red. También llegan solicitudes a procesos que no están preparados para trabajar por diversos motivos. Los servicios se reinician automáticamente si agotan los recursos, y las estrategias de tolerancia a errores permiten que el sistema en su conjunto pueda seguir en funcionamiento.

### 2.2.4 Comunicación -API GATEWAY

La puerta de enlace es el punto de entrada para los clientes. En lugar de llamar a los servicios directamente, los clientes llaman a la puerta de enlace, que reenvía la llamada a los servicios apropiados en el back-end.

Entre las ventajas de usar una puerta de enlace de API se encuentran las siguientes:

* Desacoplan los clientes de los servicios. Los servicios pueden cambiar de versión o refactorizarse sin necesidad de actualizar todos los clientes.
* Los servicios pueden utilizar los protocolos de mensajería que no son fáciles de usar para un servicio web, como AMQP.
* La puerta de enlace de API puede realizar otras funciones transversales como la autenticación, el registro, la terminación SSL y el equilibrio de carga.

Descripción docs.microsoft.com.

### 2.2.5 Seguridad

La seguridad de las API web se ocupa de la transferencia de datos a través de las API que están conectadas a Internet. OAuth (Open Authorization) es el estándar abierto para la delegación del acceso. Permite que los usuarios otorguen acceso a los recursos web a terceros, sin necesidad de compartir contraseñas. (redhat.com).

La puerta de enlace API autentica la petición y pasa un token de acceso (por ejemplo, JSON Web Token) que identifica de forma segura el solicitante en cada solicitud a los servicios. Un servicio puede incluir el token de acceso en las solicitudes que realiza a otros servicios. (Chris Richardson).

### 2.2.6 Configuración

Una aplicación suele utilizar una o más infraestructura y servicios de terceros. Ejemplos de servicios de infraestructura incluyen: un registro de servicios, un intermediario de mensajes y un servidor de bases de datos. Ejemplos de servicios de terceros incluyen: procesamiento de pagos, correo electrónico y mensajería, etc.

Externalice toda la configuración de la aplicación, incluidas las credenciales de la base de datos y la ubicación de la red. Al inicio, un servicio lee la configuración de una fuente externa, por ejemplo, variables de entorno del sistema operativo, etc.

### 2.3 Arquitectura de software

La arquitectura de un sistema intensivo en software es la estructura o estructuras del sistema, la cual comprende los elementos de software, las propiedades externas visibles de esos elementos y las relaciones entre ellos. (Bass, Clements, Kazman; AddisonWesley2003)

Vale la pena preocuparse por cómo se divide y estructura el software, en lugar de simplemente programarlo para producir un resultado correcto. Edsger Dijkstra (9168)

Dijkstra estaba escribiendo sobre un sistema operativo, y primero planteó la noción de una estructura en capas, en la que los programas se agrupaban en capas, y los programas en una capa solo podían comunicarse con los programas en capas adyacentes. Dijkstra señaló la elegante integridad conceptual exhibida por dicha organización, con las ganancias resultantes en la facilidad de desarrollo y mantenimiento.

Arquitectura del Sistema: La especificación completa y detallada de la interfaz del usuario.

No hay balas de plata: Los proyectos de software habituales pueden transformarse en un monstruo (hombre lobo) de plazos no alcanzados, presupuestos sobrepasados, y productos con errores. Entonces se escuchan gritos desesperados pidiendo una bala de plata –algo que haga disminuir los costos del software tan rápidamente como bajan los costos del hardware. Frederick Brooks (1975).

Los elementos pueden ser de procesamiento, datos o conexión; la forma se define de acuerdo con las propiedades y a las relaciones entre los elementos; la razón se contempla en términos de restricciones del sistema, que se derivan de los requerimientos del sistema. DewayneE. Perry & Alexander L. Wolf (1992).

### 2.3.1 Proceso de Arquitectura de Software

### 2.3.2 Modelos o Vistas Arquitecturales

A veces la arquitectura del software tiene secuelas de un diseño del sistema que fue muy lejos en particionar prematuramente el software, o de un énfasis excesivo de algunos de los aspectos del desarrollo del software: ingeniería de los datos, o eficiencia en tiempo de ejecución, o estrategias de desarrollo y organización de equipos. A menudo la arquitectura tampoco aborda los intereses de todos sus “clientes”.

Varios autores han notado este problema, incluyendo a David Garlan y Mary Shaw [7], Gregory Abowd y Robert Allen [1], y Paul Clements [4].

El modelo de 4+1 vistas fue desarrollado para remediar este problema. El modelo 4+1 describe la arquitectura del software usando cinco vistas concurrentes. Tal como se muestra en la Figura 1, cada vista se refiere a un conjunto de intereses de diferentes stakeholders del sistema.

* La vista lógica describe el modelo de objetos del diseño cuando se usa un método de diseño orientado a objetos. Para diseñar una aplicación muy orientada a los datos, se puede usar un enfoque alternativo para desarrollar algún otro tipo de vista lógica, tal como diagramas de entidad-relación.
* La vista de procesos describe los aspectos de concurrencia y sincronización del diseño.
* La vista física describe el mapeo del software en el hardware y refleja los aspectos de distribución.
* La vista de desarrollo describe la organización estática del software en su ambiente de desarrollo.

(Artículo publicado en *IEEE Software* 12(6), noviembre 1995).

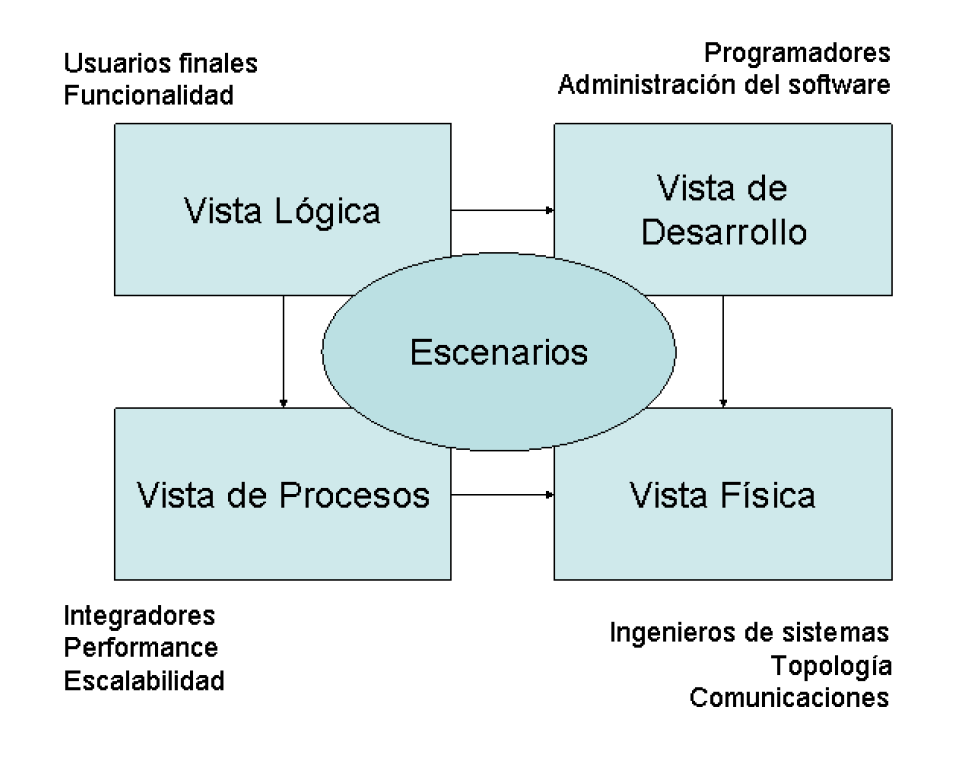


Figura xx: Vistas 4+ 1, Fuente: Artículo publicado en *IEEE Software* 12(6).

### 2.3.3 Diseño basado en el dominio (DDD)

### 2.3.4 Arquitectura Hexagonal

# CAPÍTULO III: METODOLOGÍA

### 3.1. Investigación cualitativa.

Según Jorge Martínez (2011), en su trabajo “Métodos de Investigación Cualitativa”, Jorge es docente de la Universidad Santo Tomás y San Buenaventura define que:

“La investigación cualitativa esencialmente desarrolla procesos en términos descriptivos e interpreta acciones, lenguajes, hechos funcionalmente relevantes y los sitúa en una correlación con el más amplio contexto social”

“La investigación cualitativa busca la comprensión e interpretación de la realidad humana y social, con un interés práctico, es decir con el propósito de ubicar y orientar la acción humana y su realidad subjetiva”

Martínez (2011),menciona que las características de la investigación cualitativa son las siguientes:

* La investigación cualitativa produce datos descriptivos, proporcionados por las propias palabras de las personas, y con las observaciones de su conducta.
* Una investigación cualitativa no empieza con una hipótesis, por lo que no pretende demostrar teorías existentes, más bien busca generar teoría a partir de los resultados obtenidos.
* Se basa en estudiar las personas y a los grupos en su ambiente natural y en la vida cotidiana. La interacción entre el investigador y las personas se realiza de manera natural.
* Se basa en una perspectiva histórica y dinámica.
* El empleo de la observación participante, la entrevista no estructurada, la entrevista biográfica, las historias de vida, las entrevistas grupales, las encuestas cualitativas, realiza análisis a través de esquemas y categorías abiertas.
* Su finalidad primordial es la comprensión de las experiencias individuales y/o colectivas en condiciones espacio-temporales, la aceptación de la diferencia y de la singularidad de los individuos como de sus grupos de referencia, es el fundamento de la tarea comprensiva.
* Todos los escenarios y personas son dignos de estudio.
* Los estudios cualitativos se realizan con individuos, grupos, comunidades u organizaciones.
* La forma específica de recolección de información se va definiendo y transformando durante el transcurso de la investigación, dadas las condiciones naturales en las que se realiza.

### La Investigación de Estudio de Casos.

Según Rumarriz (s.f.) y Martínez (2011), se podría definir el estudio de casos como:

“una investigación que mediante los procesos cuantitativo, cualitativo o mixto; se analiza profundamente una unidad para responder al planteamiento del problema, probar hipótesis y desarrollar teoría. También se define como una investigación sobre un individuo, grupo, organización, comunidad o sociedad; que es visto y analizado como una entidad. Otros la consideran un método para aprender de una instancia compleja, que se entiende como un todo, teniendo en cuenta su contexto

En el estudio de caso, pueden utilizarse encuestas o grupos de enfoque como herramientas para recolectar datos adicionales; esquema que resulta compatible con un proceso cuantitativo, cualitativo o mixto.”

### Investigación / Acción.

Las características principales de este método son (Rumarriz, s.f., pág. 109):

* Analiza situaciones y acciones relacionadas con problemas prácticos para intentar resolverlos.
* Considera la acción desde el punto de vista de los participantes. Es Participativa, pues todos los miembros toman parte en la investigación directa o indirectamente, y es cooperativa porque trabajan juntos para profundizar en la comprensión del problema.

Las modificaciones llevadas a la práctica son evaluadas continuamente dentro de la situación y por los propios participantes. Existe una evaluación crítica de su acción.

## 

### Tipo y Nivel de la investigación

Toda investigación requiere de un proceso sistemático y lógico, que le facilite alcanzar sus objetivos planteados, considerando para ello el apoyo de criterios del método científico, siendo a tal efecto suscrito según la naturaleza del estudio. En este sentido, como parte de la fase investigativa, se hace referencia a los aspectos de tipo y diseño de investigación; población, muestra, técnica e instrumento, validez, confiabilidad, procesamiento de los datos, y al procedimiento empleado en cada actividad de la investigación.

En este sentido, se plantea que el tipo de investigación será descriptiva, con un diseño no experimental y transversal se estudiarán los procesos de negocio de Only Systems S.A.S y se establecerán los lineamientos estratégicos para la optimación de recursos, reducción de tiempos de gestión, mejora en el servicio al cliente y satisfacción de los mismo. según Hernández y otros (2010), la investigación que se realizará se ajusta a un diseño no experimental, debido a que el estudio se efectuará sin manipulación deliberada de las variables y se hará la observación de los fenómenos en su ambiente natural, para después analizarlos.

 Así mismo, señala Kerlinger y Lee (2012), en la investigación no experimental no es posible manipular las variables o asignar aleatoriamente a los participantes o tratamientos, lo cual significa, para esta investigación, la variable de estudio no es manipulada por el investigador, siendo estudiada en su forma natural, puesto a ser hechos reales o probatorios al poner en práctica su estudio para luego ser analizados. Al respecto, Arias (2010) define el estudio no experimental como aquel donde no se manipulan ni controlan las variables, es decir, el investigador obtiene la información, pero no altera las condiciones existentes.

De tal manera, la presente investigación es de tipo aplicada ya que, se basa en el estudio de teorías para dar respuestas y soluciones a problemas específicos dentro del sector en estudio. Ello, según Chávez (2010), se lleva a cabo de acuerdo con el tipo de problema que se desea estudiar, metas que se desean alcanzar y la accesibilidad de recursos los cuales tienen como finalidad resolver un problema en un periodo corto de tiempo. Sin embargo, refieren Tamayo y Tamayo (2010), que los estudios aplicados tienen como motivación esencial enfocar su atención sobre la solución de teorías a fin de lograr la optimización de la gestión realizada por los sujetos involucrados en el estudio.

Las técnicas de recolección y herramientas que se utilizarán en el proyecto de investigación para obtener la información serán: la observación directa de los procesos, encuesta y entrevista con las personas responsables del direccionamiento de los procesos. De acuerdo con Méndez (2012), la selección de uno o varios instrumentos se hace más fácil al saber qué es lo que se desea evaluar, medir o registrar. Es necesario concretar la técnica de recogida de datos, lo que implica, una forma específica de emplear un instrumento y responde a la interrogante con qué, de tal manera que responda a los propósitos de la investigación.

# CAPÍTULO IV: DESARROLLO DE LA PROPUESTA

3.4. Desarrollo de la propuesta . . . . . . . . . . . . . . . . . . . . . . . . . . 63

3.4.1. Arquitectura y proceso de desarrollo . . . . . . . . . . . . . . . . . 63

3.4.2. Arquitectura genérica de sistemas basados en microservicios . . . 66

3.4.3. Descripción de la arquitectura genérica . . . . . . . . . . . . . . . 67

3.4.3.1. Base de datos . . . . . . . . . . . . . . . . . . . . . . . 67

3.4.3.2. Microservicios . . . . . . . . . . . . . . . . . . . . . . . 67

3.4.3.3. Comunicación . . . . . . . . . . . . . . . . . . . . . . . 68

3.4.3.4. Integración . . . . . . . . . . . . . . . . . . . . . . . . . 68

3.4.3.5. Cliente . . . . . . . . . . . . . . . . . . . . . . . . . . . 69

3.4.3.6. Seguridad . . . . . . . . . . . . . . . . . . . . . . . . . 69

3.4.4. Arquitectura de un microservicio . . . . . . . . . . . . . . . . . . 69

3.4.5. Descripción de la arquitectura de un microservicio . . . . . . . . . 71

3.4.5.1. Microservicio: . . . . . . . . . . . . . . . . . . . . . . . 71

3.4.5.2. Adaptadores y puertos . . . . . . . . . . . . . . . . . . . 71

3.4.6. Validación de la arquitectura . . . . . . . . . . . . . . . . . . . . . 71

3.4.7. Caso de estudio . . . . . . . . . . . . . . . . . . . . . . . . . . . . 73

3.4.7.1. Curul electrónica - eCurul . . . . . . . . . . . . . . . . . 73

3.4.7.2. Arquitectura actual eCurul v1.0 . . . . . . . . . . . . . . 74

3.4.8. Obtención de requerimientos . . . . . . . . . . . . . . . . . . . . . 76

3.4.8.1. Identificación de partes interesadas (Stakeholders) . . . . 76

3.4.8.2. Requerimientos funcionales . . . . . . . . . . . . . . . . 76

3.4.8.3. Requerimientos no funcionales . . . . . . . . . . . . . . 77

3.4.9. Especificación de requerimientos . . . . . . . . . . . . . . . . . . 78

3.4.10. Arquitectura propuesta . . . . . . . . . . . . . . . . . . . . . . . . 79

3.4.10.1. Contexto general del sistema . . . . . . . . . . . . . . . 79

3.4.10.2. Visión general de la arquitectura . . . . . . . . . . . . . 81

3.4.10.3. Priorización de requerimientos . . . . . . . . . . . . . . 82

3.4.10.4. Vista de casos de uso

3.4.10.5. Restricciones . . . . . . . . . . . . . . . . . . . . . . . . 85

3.4.10.6. Consideraciones arquitecturales . . . . . . . . . . . . . . 85

3.4.10.7. Vista de Módulos . . . . . . . . . . . . . . . . . . . . . 86

3.4.10.8. Vista lógica . . . . . . . . . . . . . . . . . . . . . . . . 88

3.4.10.9. Vista de procesos . . . . . . . . . . . . . . . . . . . . . 89

3.4.10.10.Vista de desarrollo . . . . . . . . . . . . . . . . . . . . . 90

3.4.10.11.Vista física . . . . . . . . . . . . . . . . . . . . . . . . 91

3.4.10.12.Diseño de microservicios . . . . . . . . . . . . . . . . . 93

3.4.10.13.Tecnologías .

# CAPÍTULO V: PRESENTACIÓN Y ANÁLISIS DE RESULTADOS

## 5.1. Presentación de resultados

## 5.2.1 Validación de la arquitectura

## 5.2.1.1 Resultados

## 5.2.2 Análisis cualitativo

## 5.2.2.1 Análisis e interpretación de resultados del instrumento

## 5.3 Conclusiones

## 5.4 Recomendaciones

# REFERENCIAS

ABI-ANTOUN,M.AND MEDVIDOVIC, N. 1999. Enabling the refinement of a software architecture into a design. In Proceedings of the Second International Conference on the Unified Modeling Language (UML'99, Fort Collins, CO). IEEE Computer Society Press, Los Alamitos, CA, 17-31.]]

<https://blog.mdcloud.es/que-son-los-microservicios-definicion-caracteristicas-y-retos/>

Microservices vs. ServiceOriented Architecture, Mark Richards, November 2015.

<https://docs.microsoft.com/es-es/azure/architecture/guide/architecture-styles/microservices>

Aroraa, G., Kale, L., y Kanwar, M. (2017). Building Microservices with .Net Core. Packt Publishing, Limited.

<https://www.redhat.com/es/topics/security/api-security>

<https://spring.io/projects/spring-cloud-netflix>

Software Architecture in Practice (2nd edition), Bass, Clements, Kazman; AddisonWesley2003

Acuña., S. (2016). La Importancia de la Tecnología Business Process Management (BPM) en la Competitividad de las Pymes en Bogotá. Tesis. Fundación Universitaria Los Libertadores Facultad Administración Empresas a Distancia. Extraido de: [http://repository.libertadores.edu.co/bitstream/11371/748/1/acu%c3%b1ape%c3%b1asandramarcela.pdf](http://repository.libertadores.edu.co/bitstream/11371/748/1/Acu%C3%B1aPe%C3%B1aSandraMarcela.pdf)

Alvarado., P. (2011). BONITA SOFT: Gestor de procesos de negocios BPM. Extraido de: <http://www.academia.edu/download/36835406/Bonita_Open_Solution.pdf>

Beltrán., J. y Rendón., C. (2012). ESTADO DEL ARTE Y TENDENCIAS DE BPM. Tesis. Universidad de San Buenaventura. Extraido de: <http://bibliotecadigital.usb.edu.co/bitstream/10819/1327/4/Estado_Arte_BPM_Rend%C3%B3n_2012.pdf>

Cadena., W. (2013). Propuesta de Mejora y Optimización a través de la Herramienta BPM para proceso de crédito del “segmento banca personas” de una Institución Financiera. Tesis.   
Pontificia Universidad Católica del Ecuador. Extraido de: [http://repositorio.puce.edu.ec/bitstream/handle/22000/9753/Disertaci%C3%B3n%20Final.pdf?sequence=1&isAllowed=y](http://www.google.com/url?q=http%3A%2F%2Frepositorio.puce.edu.ec%2Fbitstream%2Fhandle%2F22000%2F9753%2FDisertaci%25C3%25B3n%2520Final.pdf%3Fsequence%3D1%26isAllowed%3Dy&sa=D&sntz=1&usg=AFQjCNHHEqAOKg0pmrHo0SXKLfG-Mk6fwg)

Calle., X. Mayorga1., Flores., A. Lavìn., J (2014) Aplicación de la metodología BPM: RAD en una institución de educación superior. Extraído de: <http://dspace.ucuenca.edu.ec/jspui/bitstream/123456789/21407/1/TIC.EC_19_Calle%20et%20al.pdf>

Cordova mite, c. (2018). Modelamiento y simulación de la optimización del proceso de generación de facturas utilizando metodología BPM con bizagi. Ingeniera de sistemas. Unidad académica de ingeniería civil.

Club BPM. (2011). El Libro del BPM, tecnologías, conceptos, enfoques metodológicos y estándares. Madrid, España: Centro de Encuentro BPM, s.I.

Club BPM. (2009). Apuntes de Business Process Management – Gestión de Procesos de Negocio <http://www.club-bpm.com/ApuntesBPM/ApuntesBPM01.pdf>

Congacha, A.E. y García, V.J. (2017). Modelación, simulación y automatización de procesos en la gestión de servicios académicos universitarios. 3C Tecnología: glosas de innovación aplicadas a la pyme, 6(2), 32-51. DOI: . <https://dialnet.unirioja.es/servlet/articulo?codigo=6034899>

Díaz., N., (2008). Gestión de procesos de negocio BPM (Business Process Management), TIC y crecimiento empresarial ¿Qué es BPM y cómo se articula con el crecimiento empresarial?. Universidad & Empresa, [S.l.], v. 10, n. 15, p. 151-176, mayo 2008. ISSN 2145-4558. Disponible en: <>. Fecha de acceso: 03 jun. 2017. https://revistas.urosario.edu.co/index.php/empresa/article/download/1061/959

Espinosa., S. (2014). Levantamiento y Mejora de Procesos en la empresa Car Shopping S.A. y automatización en base a BPM. Tesis. Pontificia Universidad Católica del Ecuador. Extraido de: [http://repositorio.puce.edu.ec/bitstream/handle/22000/8024/Tesis%20Ricardo%20Espinoza.pdf?sequence=1&isAllowed=y](http://www.google.com/url?q=http%3A%2F%2Frepositorio.puce.edu.ec%2Fbitstream%2Fhandle%2F22000%2F8024%2FTesis%2520Ricardo%2520Espinoza.pdf%3Fsequence%3D1%26isAllowed%3Dy&sa=D&sntz=1&usg=AFQjCNHCnO9v64H0w11WYghHN476a-6-gg)

Espitia., L. (2012).  Propuesta de Modelamiento y Estandarización de los Procesos mediante soluciones BPM y BSC en proyectos de domótica para la empresa Mainframe LTDA. Tesis. Pontificia Universidad Javeriana. Extraido de: [https://repository.javeriana.edu.co/bitstream/handle/10554/13642/espitiamedinalauraviviana2011.pdf?sequence=1&isallowed=y](https://www.google.com/url?q=https%3A%2F%2Frepository.javeriana.edu.co%2Fbitstream%2Fhandle%2F10554%2F13642%2Fespitiamedinalauraviviana2011.pdf%3Fsequence%3D1%26isallowed%3Dy&sa=D&sntz=1&usg=AFQjCNGQzf1sw_fhhBXVwciQje9HgALbPw)

Funk, B., Gómez, M., Niemeyer, & Teuteberg, F. (2010). *Geschäftsprozessintegration mit SAP.* Berlín, Alemania: Springer

Garimella., K., Lees., K., Williams., B (2008). Introducción a BPM para Dummies. Publicado por Wiley Publishing. Indianápolis, Indiana. Version digital: <http://redigital.ingenieria.usac.edu.gt/bitstream/123456789/56/1/BPM.pdf>

Hernández, R., Fernández, C. y Baptista, P. (2010) Metodología de la Investigación. McGraw-Hill Interamericana Editores, SA de CV.

Huamani Gutierrez, B. y Huarcaya Tacas, N. (2018). Implementación de una solución tecnológica basada en BPM para gestionar la flota vehicular de la Municipalidad Distrital La Molina. Universidad Nacional San Luis Gozaga de Ica.

IBM. (s.f.). WebSphere Enterprise Service Bus for z/OS 7.5.0, Procesos de negocio. Recuperado el 23 de Octubre de 2017, de IBM Knowledge Center: https://www.ibm.com/support/knowledgecenter/es/SSAVUV\_7.5.0/com.ibm.wbpm.wid.bpel.doc/topics/cunder.html

Informe Observatorio BPM del ClubBPM, (2016). Extraido de: <http://www.club-bpm.com/Contenido/Estudios/est-2016-004.htm>

Kerlinger, F. y Lee, H. (2012). Investigación del comportamiento: Métodos de investigación en ciencias sociales.McGraw Hill Interamericana.

Laurentiis., R. (2017). Gestión, Automatización, Inteligencia de Procesos (BPM) y Transformación Digital Eficiencia Operacional, Innovación y Competitividad. Conferencia. Universidad Nacional Abierta y A distancia.

Martínez, J. (2011). Métodos de investigación cualitativa. *Silogismo, Revista de la Corporación Internacional para el Desarrollo Educativo*, 1-34.

Mayorga., S. Córdoba., N. (2007). Diagnóstico de la madurez de los procesos en empresas medianas colombiana. Extraido de: <http://revistas.javeriana.edu.co/index.php/iyu/article/download/944/541::pdf>

Méndez, C. (2012), Metodología: Guía para la elaboración de diseños de investigación en ciencias económicas, contables y administrativas. Mc Graw Hill. Colombia.

Restrepo., D. (2007). Las Pymes y el Crecimiento Económico en Colombia (años 1990-2000). Tesis. Pontificia Universidad Javeriana Facultad de Ciencias Económicas y Administrativas. Extraido de: <http://www.javeriana.edu.co/biblos/tesis/economia/tesis11.pdf>

Rico., C. (2011). Metodología para gestión de proyectos de administración de procesos de negocio - BPM - en empresas de servicios en Latinoamérica. Tesis de Maestría. Universidad ICESI – tulane university. Extraído de: <https://repository.icesi.edu.co/biblioteca_digital/bitstream/10906/67907/1/metodologia_gestion_proyectos.pdf>

Robledo., P.,  Laurentiis., R.(2011). Desarrollo de conocimientos y capacidades en BPM. Cómo empezar y encaminar la gestión del cambio cultural a procesos. Extraido de: [http://www.club-bpm.com/Club-BPM-Webinar-DesarrolloConocimientosyCapacidadesEnBPM.pdf](http://www.google.com/url?q=http%3A%2F%2Fwww.club-bpm.com%2FClub-BPM-Webinar-DesarrolloConocimientosyCapacidadesEnBPM.pdf&sa=D&sntz=1&usg=AFQjCNGDXcr7V3sh0Qg3B5qJsxbx1CDTiw)

Rumarriz, B. (s.f.). Técnicas y Métodos de la Investigación Cualitativa. La Coruña, España: Universidad da Coruña. Obtenido de <http://ruc.udc.es/dspace/bitstream/handle/2183/8533/CC-02art8ocr.pdf?sequence=1>

Sarmiento., N., y Franco., D. (2015). Modelo de Procesos Basado en la Metodología Business process Management (BPM) para talleres de mecánica automotriz tomando como caso de estudio la red de servicio postventa de distribuidora Nissan s.a. Tesis de Grado. Universidad distrital francisco José de caldas facultado de ingeniería. <http://repository.udistrital.edu.co/bitstream/11349/2978/1/SarmientoCastilloNataliaAndrea2015.pdf>

Sperandio., G. Andre., Eberle., L. Dorion., E. (2012). EL BPM - Business Process Management como práctica de Gestión en una Empresa Metalúrgica con estrategia de producción eto-engineer-toorder. Extraido de: <http://www.abepro.org.br/biblioteca/enegep2012_TI_ST_163_950_19442.pdf>

Tuaty.,  L. (2011). Implementación del Business Process Management (BPM) en la empresa aerovías de Integración Regional Aires S.A. Tesis. Universidad Libre.Extraido de: [http://repository.unilibre.edu.co/bitstream/handle/10901/5941/TuatyRamirezLindaBelky2011.pdf?sequence=1&isAllowed=y](http://www.google.com/url?q=http%3A%2F%2Frepository.unilibre.edu.co%2Fbitstream%2Fhandle%2F10901%2F5941%2FTuatyRamirezLindaBelky2011.pdf%3Fsequence%3D1%26isAllowed%3Dy&sa=D&sntz=1&usg=AFQjCNHqQyagIDwWlopRBhkxP_1zgP8_hA)

Weske, M., van der Aalst, W. M., & Verbeek, H. M. (2004). Advances in Business Process Management. Data & Knowledge Engineering, 50(1), 1-8