

**UNIVERSIDAD CATÓLICA SANTO TORIBIO DE MOGROVEJO**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**



**IMPLEMENTACIÓN DE UNA ARQUITECTURA ESCALABLE BASADA EN  
GOOGLE CLOUD PLATFORM PARA MEJORAR LA DISPONIBILIDAD Y  
ESCALABILIDAD DE INFORMACIÓN DE LA EMPRESA SMARTBRANDS,  
LIMA 2019**

**TESIS PARA OPTAR EL TÍTULO DE  
INGENIERO DE SISTEMAS Y COMPUTACIÓN**

**AUTOR**

**RICARDO MIGUEL LLONTOP GARCIA**

**ASESOR**

**JUAN ANTONIO TORRES BENAVIDES**

<https://orcid.org/0000-0002-0133-119X>

**Chiclayo, 2020**

**IMPLEMENTACIÓN DE UNA ARQUITECTURA ESCALABLE  
BASADA EN GOOGLE CLOUD PLATFORM PARA MEJORAR LA  
DISPONIBILIDAD Y ESCALABILIDAD DE INFORMACIÓN DE  
LA EMPRESA SMARTBRANDS, LIMA 2019**

PRESENTADA POR  
**RICARDO MIGUEL LLONTOP GARICA**

A la Facultad de Ingeniería de la  
Universidad Católica Santo Toribio de Mogrovejo  
para optar el título de

**INGENIERO DE SISTEMAS Y COMPUTACIÓN**

APROBADA POR

Maria Ysabel Aranguri Garcia  
PRESIDENTE

Jury Yesenia Aquino Trujillo  
SECRETARIO

Juan Antonio Torres Benavides  
VOCAL

## **Dedicatoria**

A mis padres por ser mi todo.

A mis hermanas por ser siempre mi motor para esforzarme y buscar siempre ser alguien mejor.

A mis sobrinas por ser la alegría de todos mis días.

## **Agradecimientos**

A mi asesor de tesis, el Ing. Juan Torres, por su incondicional en todo momento en el desarrollo del proyecto.

## Índice

<b>Resumen .....</b>	<b>5</b>
<b>Abstract .....</b>	<b>6</b>
<b>Introducción.....</b>	<b>7</b>
<b>Revisión de literatura.....</b>	<b>7</b>
<b>Materiales y métodos .....</b>	<b>10</b>
<b>Resultados y discusión .....</b>	<b>10</b>
<b>Conclusiones .....</b>	<b>15</b>
<b>Recomendaciones .....</b>	<b>16</b>
<b>Referencias.....</b>	<b>17</b>

## Resumen

En este artículo se presenta la propuesta de una arquitectura escalable para la mejora de la disponibilidad y escalabilidad de la información basada en google cloud platform para la empresa SmartBrands S.A.C., con la cual se obtuvo un incremento en su disponibilidad con respecto a su ERP de 1.5% y una mejora de 2.19% con respecto a su servidor on-premise, analizando los primeros meses del año 2020. También se demostró en este proyecto que el duplicar la capacidad del servidor central para mejorar el procesamiento de información era insuficiente ya que solo mejoró el proceso de respuesta en 6 segundos, por lo que se concluyó que para mejorar su escalabilidad este debería ser horizontal, en un segregamiento de procesos independientes, los cuales en funcionamiento y trabajando en conjunto crean una arquitectura para una mejorar adaptabilidad del sistema. Al tener esta arquitectura operando, nos permite tener un soporte para los diferentes eventos que puede acontecer en el sector retail, tales como el escalamiento de negocio por compra de marcas para su distribución, integración con api externas por contrato o por necesidad, reportes con segmentación de negocio, etc.

**Palabras clave:** Arquitectura, disponibilidad, escalabilidad, google cloud platform, información.

## Abstract

This article presents the proposal of a scalable architecture to improve the availability and scalability of information based on google cloud platform for the SmartBrands SAC company, with which an increase in its availability was obtained with respect to its ERP of 1.5 % and an improvement of 2.19% with respect to its on-premise server, analyzing the first months of the year 2020. It was also demonstrated in this project that doubling the capacity of the central server to improve information processing was insufficient since it only improved the response process in 6 seconds, so it was concluded that to improve its scalability it should be horizontal, in a segregation of independent processes, which in operation and working together create an architecture to improve the adaptability of the system. By having this architecture operating, it allows us to have support for the different events that may occur in the retail sector, such as business escalation for the purchase of brands for distribution, integration with external APIs by contract or by necessity, reports with segmentation business, etc.

**Keywords:** Architecture, disponibility, scalability, google cloud platform, information.

## Introducción

El principio, la forma sigue a la función [1], aplicada en la arquitectura y diseño del siglo 20, bien puede ser aplicada al diseño de una arquitectura de software, dado que una buena arquitectura no es creada en solitario. Una arquitectura está conformado por sistemas interconectados con un fin en específico, los cuales dada su complejidad, hace que tengan la necesidad de usar sistemas basado en componentes [2]. Este proyecto se ejecutó en la empresa Smartbrands S.A.C, el cual pertenece al fondo de inversión FARO CAPITAL SAFI S.A., cuyo rubro es la administración de tiendas por departamento retail menor o boutique especializadas. Un problema central encontrado, es que los datamart y otra data condensada la disponen externamente, la cual sufre bajas continuas, lo que ocasiona caídas de algunos sistemas e incertidumbre en la gerencia; y al no saber qué estructura está empleando el tercero y por el tipo de negocio que aplica (crecimiento acelerado) ocasiona una dependencia muy seguida a este proveedor para el desarrollo de nuevos requerimientos, ya sean estos de poca o

## Revisión de literatura

Según [3], cloud computing es un modelo para permitir un único, conveniente, acceso sobre demanda de red, para compartir un pool configurable de recursos computacionales (ej. redes, servidores, almacenamiento, aplicaciones y servicios) que pueden rápidamente ser provisionados y desplegados sin ningún mayor esfuerzo o interacción del servicio. De acuerdo [4] en la gestión de la disponibilidad, describe a la disponibilidad como la habilidad de un servicio, componente o CI (ítem de configuración) para realizar su función acordada cuando es requerido. Esta es generalmente medida y reportada en porcentajes.

gran envergadura. Otro problema es que el sistema SOLUFLEX necesita de una arquitectura cliente/servidor, por lo cual incurre en las deficiencias de este modelo tradicional. Por ultimo un problema central que ocurre en el área mayorista, en su proceso de pedidos y facturación; la generación de un pedido, aprobación y recepción de mercadería, son registrados en el ERP soluflex, pero para informar sobre pedidos nuevos, los vendedores mayoristas lo transmiten mediante el software whassap. Por tal motivo se necesita generar una arquitectura no solo para mejorar su disponibilidad de información, sino para poder escalar fácilmente según se presente las incurrencias en el negocio.

En este proyecto se busca describir el proceso de elaboración de una arquitectura escalable basada en Google Cloud y elaborar una propuesta de una arquitectura escalable, medir su nivel de disponibilidad de la información actual en su servidor on-premise y medir el nivel de disponibilidad de la información en su servicio ERP y por ultimo de qué manera se puede mejorar la escalabilidad de la información.

$$\text{Availability (\%)} = \frac{\text{Agreed service time (AST)} - \text{downtime}}{\text{AST}} \times 100$$

Fig. 1. Fórmula sobre la disponibilidad. [4]

En cloud computing se refiere a que la organización puede estar segura en que su solución va a estar disponible y corriendo más tiempo que lo que hubiera estado en localmente.

Cuando decimos que cloud computing incrementa la disponibilidad, es basado en que la mayoría de los servicios ofrecen redundancia en sus servidores físicos, obteniendo un sistema que puede expresarse en la siguiente figura:

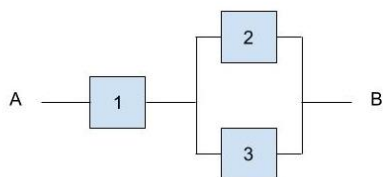


Fig. 2. Representación de un sistema en la nube con redundancia. [5]

Este gráfico se puede explicar de la siguiente manera: una salida B es esperada por un sistema dado una entrada de A. La entrada A es procesada por el software “1”. El software “1” está corriendo en el servidor “2” o en el servidor “3”. Con esta arquitectura, la salida B puede ser obtenida aun si el servidor “2” o el servidor “3” esta caído, pero no los dos.

Escalabilidad se define como cuan fácil un sistema o componente puede modificarse para arreglar el problema.

Según [6], un sistema escalable tiene 3 características:

El sistema se puede acomodar al incremento de su uso, el sistema puede acomodarse a incremento de su data y el sistema es manejable y trabaja con un razonable performance.

Escalabilidad no solo es acerca de velocidad. Performance y escalabilidad para un sistema se interrelacionan fuertemente. Performance mide cuán rápido y eficiente un sistema puede completar ciertas tareas, mientras escalabilidad mide el suceso de incremento de carga de un sistema.

Para fundamentar el cambio de un sistema en nuestros ambientes, se puede aplicar la ley de Amdahl la cual mide el impacto en el rendimiento del cambio en un subsistema, utilizada para averiguar la mejora máxima de un sistema de información cuando solo una parte de éste es mejorado.

$$F = F_a ((1-F_m) + F_m/A_m)$$

Es importante para esclarecer la metodología a utilizar en el siguiente proyecto, por lo cual listo las principales diferencias entre RUP y SCUM según su alcance, tiempo, costo y riesgos.

TABLA I  
RUP VS SCRUM, basado en [7]

RUP	SCRUM
<b>Alcance</b>	
Se definen los requerimiento de RUP Específicamente no hay mapeo de con RUP Iteraciones y fases de planeamiento	Se desarrolla una lista de elementos de la cartera de productos y se priorizan Se eligen los elementos de la cartera de productos para el próximo lanzamiento. La administración se hace a través del product backlog.
<b>Tiempo</b>	
Iteraciones y fases de planeamiento Diseño del plan por iteración Tareas de las iteraciones Problemas y manejo de excepciones Reajuste del plan del proyecto	Los escenarios se seleccionan para el próximo sprint por el equipo; las tareas se definen para lograr las funciones de lanzamiento Planificación de reuniones. La estimación de la tarea se realiza para la historia Desarrollo del cronograma general para el lanzamiento El equipo es responsable de administrar el sprint
<b>Costo</b>	
Directamente no hay mapeo de esto en RUP.	Se puede hacer estimaciones de Top-Down para las versiones y sprint usando herramientas como Project Velocity,



	Ideal Days, etc.
<b>Riesgos</b>	
Se formula un plan de manejo de riesgos	El planeamiento de los riesgos es parte de cada sprint/lanzamiento que se discute en las reuniones. Todo el equipo está involucrado en este plan y mitigación de riesgos, haciendo uso de brainstorming, checklist, etc. El monitoreo es tomado por el equipo en el planeamiento y revisión del release.

Por la capacidad intrínseca que tiene SCRUM para el rápido desarrollo de los release de un proyecto, por su involucramiento de todas las áreas donde se desarrolla el proyecto, por las constantes reuniones que se tienen que tener en cuenta para el avance de los sprint y por su maleabilidad (para la implementación de esta metodología solo se siguen guías, buenas practicas), se consideró para el proyecto esta metodología.

Según el reporte Cloud Market Share Q4 de la empresa Canals [8], empresa líder en el análisis de mercados de tecnología, genera el siguiente grafico del total mundial gastado en infraestructura cloud y crecimiento anual del año 2019.

Worldwide cloud infrastructure spending and annual growth  
Canals estimates, full-year 2019

Cloud service provider	Full-year 2019 (US\$ billion)	Full-year 2019 market share	Full-year 2018 (US\$ billion)	Full-year 2018 market share	Annual growth
AWS	34.6	32.3%	25.4	32.7%	36.0%
Microsoft Azure	18.1	16.9%	11.0	14.2%	63.9%
Google Cloud	6.2	5.8%	3.3	4.2%	87.8%
Alibaba Cloud	5.2	4.9%	3.2	4.1%	63.8%
Others	43.0	40.1%	34.9	44.8%	23.3%
<b>Total</b>	<b>107.1</b>	<b>100.0%</b>	<b>77.8</b>	<b>100.0%</b>	<b>37.6%</b>

canals

Fig. 3. Total Mundial gastado en infraestructura cloud y crecimiento anual del año 2019. [8]

En el cual se puede observar que AWS (Amazon Web Services) tiene el 34.6% del mercado global, seguido por Azure con 18.1%, luego Google Cloud con 6.2% y cerrando Alibaba Cloud con 5.2% y otras nubes con 43%.

Jeff Bezos explicaba que AWS tiene una inusual ventaja de 7 años contra cualquier competidor, por lo cual los servicios de AWS son los más evolucionados y con mayor funcionalidad. [8]

También en la figura se observa que Google Cloud es el que mayor crecimiento anual tiene con un 87.8% en comparación con los otros proveedores de servicios.

Pero elegir una tecnología solo porque es la más usada o es la que tiene mayor crecimiento carece de un análisis riguroso con respecto a que tecnologías se va a usar en un proyecto y como este proveedor se puede acoplar con las exigencias del proyecto.

Según lo analizado en este proyecto se ha detectado 4 tecnologías por las cuales se eligió Google Cloud Platform en el siguiente proyecto estas son: Firebase, G Suite, Instancias Preventivas y Kubernetes

## Materiales y métodos

Las técnicas de recolección de datos fueron la observación y el análisis documental, los instrumentos que se utilizó para esto fueron las fichas de observación y la ficha de datos que sirvieron para recoger todos los datos en las reuniones de los sprint, la población aplicada acá son las

áreas de la empresa Smartbrands en donde el proceso incidía en este caso el área de TI y el área mayorista, su propósito para esto fuera la identificación de los problemas y discusión de las medidas que se podría aplicar en estos casos por parte de las áreas en las reuniones.

## Resultados y discusión

En esta parte presentan los resultados obtenidos luego de aplicar la metodología SCRUM en base a los objetivos de la investigación, los cuales fueron:

### 1. *Describir el proceso de elaboración de una arquitectura escalable basada en Google Cloud*

Para la asociación con los indicadores críticos fue de la definición de procesos funcionales, con los cuales se comenzar a elaborar una arquitectura escalable basada en google cloud.

#### Requerimientos Funcionales

Para la identificación de los requerimientos funcionales en un ambiente de desarrollo ágil, desarrollamos historias de usuario, las cuales hemos identificado de la siguiente manera:

✓ **Mejorar en la disponibilidad de la información tanto a la empresa como para agentes externo,** se busca mejorar la disponibilidad de la información, teniendo en consideración las caídas del sistema, así como los agentes externos que conllevan en que esta información no llegue oportunamente cuando la solicitan.

✓ **Desarrollo de una arquitectura escalable que permita el soporte masivo requerimientos,** se busca desarrollar una arquitectura en cloud la cual permita el desarrollo fácil de nuevos requerimientos de la empresa, los cuales se deben en

materia del canal retail menor. Tales como reportes gerenciales, integraciones con nuevos puntos de venta, solicitudes externas de data, etc.

✓ **Mejorar el proceso de venta del canal mayorista,** se busca agilizar el proceso de venta del canal mayorista, brindándoles herramientas web para la realización de sus pedidos y que estos sean procesados oportunamente por el ERP soluflex.

✓ **Desplegar un monitoreo sobre los recursos desplegados en la nube,** Se busca que todos los recursos desplegados anteriormente, tengan un monitoreo constante, para la respuesta rápida sobre un sobreuso o un desuso de sus recursos, con el fin mejorar o ajustar el cobro sobre el uso de esto.

#### Requerimientos No Funcionales

##### **Requerimientos no funcionales de producto**

###### *Eficiencia*

La arquitectura debe ser capaz de ingestar data a la nube de más de 1'000,000 filas a un tiempo máximo por proceso de 15 minutos.

La arquitectura debe ser capaz de generar REST internos para la comunicación entre programas.

La arquitectura debe ser capaz de generar REST externos para agentes externos.

Las instancias deben poder integrarse con todos los ambientes de Google Suite (Gmail, Drive, Hangout).

Las instancias deben ser capaces de conectarse vía una red privada entre ellas, aun si su ubicación física es distinta.

Las instancias deben de tener ip públicas fijas.

#### *Seguridad lógica y de datos*

Los permisos deben de ser administrados por el arquitecto en cloud asignado, se asignan a una cuenta del área no a una persona.

Los permisos externos, para personal externo, deben de ser asignados por el administrador.

Solo debe haber un punto de comunicación con nuestra arquitectura on-premise, reduciendo el peligro de agentes externos ya que este punto debe estar protegido.

Las instancias deben de base de datos deben tener un firewall de acceso.

#### *Usabilidad*

La arquitectura debe contar con una gráfica en donde se muestre las interacciones entre estas.

La arquitectura debe de ser capaz de mandar mensajes de alerta cuando hay un sobreuso de recursos.

Los ambientes web, debe de contar con un entorno test.

En el ambiente web del canal mayorista debe de ser un ambiente responsive de acorde al tamaño de las tablets asignadas a cada vendedor, en este caso 1024x600.

#### **Requerimientos no funcionales organizacionales**

El proceso de desarrollo de la arquitectura el cual se está desarrollando en Scrum.

El entorno ETL de la arquitectura se basara en el framework Laravel.

El Gestor de Colas a desplegar será Rabbitmq.

Los Gestores de Base de Datos serán Mysql (para las páginas web), Postgresql (para tablas funcionales de la ingesta de

data) y Redis (Para las tablas NOSQL que se necesiten).

El desarrollo de los REST será basado en el framework Laravel Passport.

El desarrollo del entorno e-commerce será desarrollado en Wordpress, integrado con la herramienta e-commerce WooCommerce.

El monitoreo de recursos de las instancias se harán tanto en el ETL, como el los servicios de google alerts.

## **2. Elaborar la propuesta de una arquitectura escalable basada en Google Cloud Platform.**

Se elaboró la siguiente propuesta de arquitectura cloud, las cuales buscan mejorar la disponibilidad y escalabilidad de la información en Smartbrands.

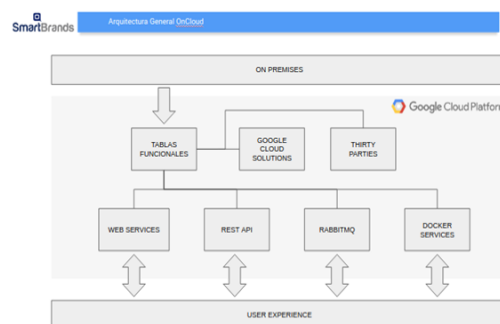


Fig. 4. Estructura General de Arquitectura.

En un primer momento se buscó hacer una ingestión de data de tablas funcionales requeridas para el funcionamiento de requerimientos globales, con esto se buscó que no todas las peticiones vayan al servidor central si no ya pasen a la nube, en este momento se buscó ingestar los Datamart de soluflex: dimproductos, dimclientes y facturacionventasfull.

En un segundo momento se buscó configurar los accesos y integración con las diferentes herramientas de google cloud, mediante key o rest apis.

Se buscó desplegar los servicios web, y rest api; los cuales servirán en un primer momento al canal mayorista, pero siempre

buscando la integración escalable con otros proyectos futuros

El despliegue de los servicios del gestor de colas, con lo cual se buscó trabajar las peticiones ya sean de los usuarios o de los procesos de manera asíncrona

Y por último se buscó hacer un servicio donde se puedan administrar contenedores, los cuales serán para proyectos internos y de Testing.

Al final se elaboró y desplego la arquitectura general de todas las entidades desplegadas por proyecto y servicios graficadas en la figura 5.

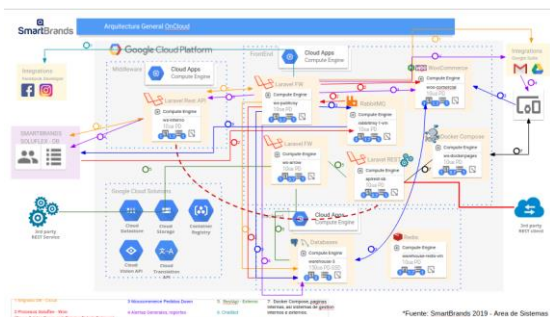


Fig. 5. Estructura General Oncloud.

Proyectos: frontend, backend y middleware.

Servicios: Migrado DB – Cloud, Procesos Soluflex – Woo (Cruce Saldos, DescuentoPrecios, EstadoOrdenes), Woocommerce Pedidos Down, Alertas Generales - reportes, RestApi – Externo, ChatBot, Docker Compose (paginas internas, así como sistemas de gestión internas y externas).

### 3. *Medir el nivel de disponibilidad y escalabilidad de información.*

Se elaboró una medición de la escalabilidad y disponibilidad de la información del año 2019, según los datos obtenidos mediante reuniones en el área de TI.

#### *Escalabilidad*

Para fundamentar el cambio de un sistema en nuestros ambientes, aplicaremos la ley

de Amdahl la cual mide el impacto en el rendimiento del cambio en un subsistema.

Según la Ley de Amdahls

$$F = F_a ((1-F_m) + F_m/A_m)$$

Se tiene que en promedio se utiliza un 80% del procesador central en todos los procesos de soluflex, ahora se va duplicar el procesador de 16 núcleos a 32. Por lo cual el  $A_m$  factor de mejora se duplica

$$F = (1/(1-0.8) + (0.8/2))$$

**$F=1.66$  (El sistema será un 60% más rápido).**

Sabiendo que el sistema tiene un promedio global de respuesta de **10 segundos**

**$F$  = Tiempo de ejecución con mejora / Tiempo de ejecución sin mejora**

**Tempo de ejecución con mejora =  $1.66/10$  (El sistema hace que la respuesta sea de 6 segundos).**

Por lo que no basta con aplicar un crecimiento en recursos, sino que se debe buscar un crecimiento horizontal, y esto se logra trabajando on-premise creando clústeres computacionales o en la nube desplegando recursos compartidos.

#### *Disponibilidad Global*

Para medir la disponibilidad global del servidor on-premise se siguió la siguiente formula:

$$a = 1 - \frac{MTTR}{MTTF + MTTR}$$

Fig. 6. Formula sobre la disponibilidad. [9]

En la tabla VI se tiene referente de todas las caídas del servidor central en los que el proceso se interrumpió por causas externas (corte de energía, corte de internet, etc.) o por causas internas (consumo de memoria RAM, cache del servidor lleno, log del servidor interrumpió procesos por falta disco duro, etc.).

TABLA VI  
CUADRO DE DISPONIBILIDAD DEL SERVIDOR  
ONPREMISE DE LA EMPRESA SMARTBRANDS  
AÑO 2019

Mes	Disponibilidad	Tiempo Promedio Fallas (MTTF)	Tiempo Medio Recuperacion (MTTR)
Enero	90.91%	2	0.2
Febrero	80.00%	2	0.5
Marzo	95.24%	4	0.2
Abril	80.00%	4	1
Mayo	83.33%	5	1
Junio	66.67%	3	1.5
Julio	90.91%	2	0.2
Agosto	90.91%	2	0.2
Setiembre	86.96%	2	0.3
Octubre	90.91%	1	0.1
Noviembre	83.33%	5	1
Diciembre	75.00%	3	1
TOTALES		2.92	0.60
	Disponibilidad	Tiempo Promedio Fallas (MTTF)	Tiempo Medio Recuperacion (MTTR)
ANUAL	81.26%	2.49	0.58

En lo que se puede observar que hay una disponibilidad de información promedio del servidor on-premise de 81.26% para el año 2019.

#### **Disponibilidad del servicio**

Para medir la disponibilidad del servicio soluflex actual aplicó la siguiente formula:

$$\text{Availability (\%)} = \frac{\text{Agreed service time (AST)} - \text{downtime}}{\text{AST}} \times 100$$

Fig. 7. Formula sobre la disponibilidad. [10]

Dada esta se tiene los siguientes resultados de la disponibilidad de la información en soluflex, dando se cuenta que los meses de Abril, Mayo, Octubre, Noviembre y Diciembre son los meses de mayores caídas y más incidencias siendo la disponibilidad global del servicio de 95.47%.

TABLA VII  
CUADRO DE DISPONIBILIDAD DEL SERVICIO  
SOLUFLEX DE LA EMPRESA SMARTBRANDS  
AÑO 2019

Mes	Disponibilidad	Tiempo Caidas(hours)
Enero	94.74%	20
Febrero	97.30%	10
Marzo	97.30%	10
Abril	91.14%	35
Mayo	75.00%	120
Junio	97.30%	10
Julio	94.74%	20
Agosto	97.30%	10
Setiembre	96.00%	15
Octubre	87.80%	50
Noviembre	85.71%	60
Diciembre	87.80%	50
		410
	Disponibilidad	Tiempo Caidas(hours)
ANUAL	95.47%	410

*Se concluye que se debe buscar una alternativa para evitar caídas en el servicio y buscar una escalabilidad horizontal que sea práctica y rentable para la empresa.*

El presente proyecto “Implementación de una Arquitectura Escalable Basada en Google Cloud Platform Para Mejorar la Disponibilidad y Escalabilidad De Información de la Empresa Smartbrands”, ha sido desarrollado para elaborar una propuesta de arquitectura escalable en la plataforma google cloud para la empresa Smartbrands apuntalando mejorar la disponibilidad y escalabilidad de su información en medio de su rublo de negocio cambiante y realidad de su arquitectura on-premise.

Esta arquitectura propuesta, se muestra en la fig.8, permite cumplir el objetivo general del proyecto el cual es la transferencia de datos del canal minorista y mayorista al ambiente oncloud, los cuales permiten desligarnos del entorno on-premise y tener la información disponible 99.95% de SLA

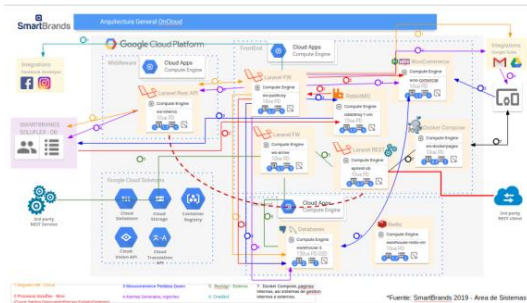


Fig. 8. Estructura General Oncloud.

Al identificar y transferir las tablas funcionales a este ambiente oncloud y almacenarlos, permite la escalabilidad de la información, ya que estas tablas tienen un formato de documento, y si hubiera una modificación en las tablas del ERP soluflex, como la agregación de nuevos campos o eliminación de estos, las tablas en la nube no sufren ningún cambio por que todos los campos y su data están almacenados en formato JSON.

Otro punto a considerar es que al adoptar esta transferencia de datos - ingesta, lo cual es muy diferente a migración, permite acoplarnos fácilmente al cambio rápido del negocio retail, un ejemplo de esto son las redes sociales, al habilitar los webhook en la instancia ws-publico, permite que este se conecte fácilmente al facebook api y por consiguiente al instagram api, con los cual permite abrir canales no solo de visibilidad de nuestro producto, sino de comercio, como son los shop de facebook e instagram o mejoras en la comunicación con el público mediante los chatbot.

Con respecto a antecedentes internacionales Pradhan [11], narraba la incertidumbre de las empresas de migrar su data sensitiva a la nube pública, además de cómo esta es de necesidad obligatoria para competir en la era de la computación hiper escalable, el presente trabajo tiene correlación con esto de que si bien es cierto esta data pasa a manos de un tercero, se debe aplicar medidas para paliar estas incertidumbres como los firewalls aplicados en las instancias como en los servicios desplegados.

Al-Dhuraibi [12], narra la elasticidad para manejar eficientemente los recursos

de provisionamiento y desprovisionamiento en cloud computing, con tecnologías como DOCKER o VIRTUALVM, este trabajo incluye en su arquitectura la instancia ws-dockerh y ws-dockerpages la cual es una instancia la cual desplegar servicios de docker-compose para en una sola instancia tener varios dominios y por ultimo Mohan [13], desplegaba en su investigación procesos de edge-computing los cuales si no están aplicados en el siguiente arquitectura fácilmente se pueden incluir en la arquitectura gracias a la escalabilidad de esta por tener servicios y data esencial migrada a la nube, con lo cual fácilmente se pueden incluir servicios de IoT (Internet de las cosas) para por ejemplo las tiendas por departamento.

Con respecto a los antecedentes nacionales Zacarias [14] y Azañedo, Bermudez y Cueva [15], proponen metodologías para la adopción de plataforma Cloud Computing en entidades públicas en el Perú, y proponer metodologías sobre la adopción de una plataforma oncloud, si bien es cierto puede servir como marco, pero hasta cierto grado, ya que los ambientes oncloud evolucionan rápidamente, y la gama de productos de cualquier proveedor aumentan exponencialmente, y dar una metodología para seguir, podría conllevar a que se pierda la oportunidad de disponer e implementar estas nuevas tecnologías, se debería dar a prioridad a definir sus procesos a escalar, luego analizar diferentes arquitecturas modelos como el propuesto en este proyecto y ver que procesos se ajustan más a su realidad (cada una es diferente de otra).

Con relación a Villanueva [16], el cual, propone prototipado que brinda soluciones de business intelligence a las área de proyecto de una empresa bancaria y utiliza soluciones SAAS en la nube, está alineado a este proyecto, ya que al brindar prototipados, esto permite tener un producto rápidamente el cual va a ser probado y modificado por el área



competente para su ejecución a posterior en el ambiente de producción.

Al elaborar esta arquitectura, se analizó el porcentaje de disponibilidad del servicio soluflex con relación de sus caídas de servicio, sus resultados son, estando funcionando el ecommerce en el canal mayorista y solo tomando la información de los meses de Enero, Febrero y Marzo se tiene que hay una disponibilidad promedio de 98.19% tal como se muestra en la figura 9, y con respecto a su disponibilidad del año anterior la cual fue de 96.44%, con lo cual se observa una mejora de 1.5% de disponibilidad de la información con respecto al ERP soluflex.

Mes	Disponibilidad	Tiempo Caidas(hours)
Enero	98.63%	5
Febrero	98.63%	5
Marzo	97.30%	10
Promedio	98.19%	

Fig. 9. Disponibilidad de información del servicio soluflex de la empresa Smartbrands Enero, Febrero y Marzo 2020.

También se analizó la disponibilidad del servidor on-premise según su disponibilidad los cuales dieron los

### Conclusiones

1. Al describir el proceso de elaboración de una arquitectura escalable basada en Google Cloud, permite dar un alcance de cómo se puede empezar a trabajar en el proceso de ingesta o migrado en información en la nube, específicamente google cloud, pero que se puede aplicar a cualquier proveedor de servicios oncloud.
2. Al elaborar la propuesta de una arquitectura escalable basada en Google Cloud Platform, según la realidad del negocio, permite ver cómo se puede diseñar una arquitectura para que escale fácilmente en este modelo de negocio de retail menor, tomando en consideración los diferentes canales donde puede escalar la información.

resultados de la figura 10, y comparándolo con los resultados del año anterior se ve una mejorar considerable de su disponibilidad en los primeros meses de 2.19%, según se muestra en la figura 11.

Mes	Disponibilidad Año 2020	Tiempo Promedio Fallas (MTTF)	Tiempo Medio Recuperacion (MTTR)
Enero	90.91%	1	0.1
Febrero	90.91%	1	0.1
Marzo	90.91%	2	0.2

Fig. 10. Disponibilidad de información de servidor on-premise de la empresa Smartbrands Enero, Febrero y Marzo 2020.

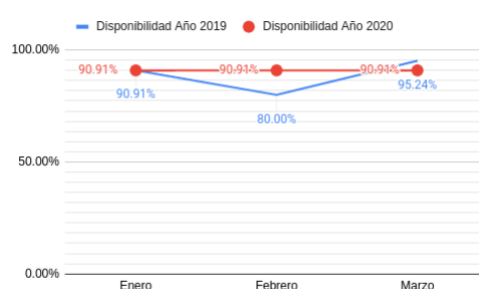


Fig. 11. Grafica de comparación de disponibilidad de información del servidor on-premise de la empresa Smartbrands en los meses de Enero, Febrero y Marzo 2020.

Además la disponibilidad de los reportes de gerencia no se caen aun si el medio on-premise esta no disponible, ya que la data a consultar esta migrada un día anterior.

3. También al elaborar la propuesta de una arquitectura escalable, se da como punto de partida las tablas funcionales, y para esto se requiere la consolidación de data en Datamart y la definición de tablas que infieren en el negocio, con los cuales se puede comenzar la ingesta o migración a la nube.
4. Al medir el nivel de disponibilidad de la información nos permite saber si los procesos ingestados o migrados al ambiente oncloud, mejoran la disponibilidad de información según el proceso medido, en este caso fue todo el servicio del ERP soluflex.
5. Midiendo el escalabilidad de la información, nos permite demostrar que no solo escalando verticalmente se mejorar

sustancialmente un proceso, sino la combinación del escalamiento vertical y horizontal, usando no solo las herramientas que los proveedores oncloud nos brindan,

sino acoplando nuestros procesos a tecnologías de escalamiento como son dockers, REST API, WebHook, Kubernetes, etc

## Recomendaciones

**1.** Cuando se ingesta o migra a la nube, no se deberían utilizar metodologías rígidas como RUP, ya que si bien es cierto permite una base documental amplia y define los procesos correctamente, la demora en la elaboración de estos en el ambiente laboral hace casi imposible su aplicabilidad a un proyecto que cambia exponencialmente, como son las arquitecturas en la nube según la realidad de negocio.

**2.** Si en las áreas de tecnología, el primer punto a tomar en consideración cuando se habla de nuevas tecnologías es migrar a la nube, algo anda mal en la definición de conceptos

**3.** No se debe caer en Recency bias, el cual es la adopción de nuevas tecnologías solo porque son nuevas, es un error muy común en estos ambiente en la nube, siempre se debe partir en tecnologías las cuales tienen tiempo de adopción

comprobadas o están definidas en especificaciones revisadas por expertos.

**4.** Se debería considerar no solo los SLA cuando se define un proveedor de servicios en la nube, sino internamente desarrollar y manejar SLI (Service Level Indicator) y SLO (Service Level Objectives), los cuales nos permiten tener una definición más clara de los que se quiere llevar a cabo con la arquitectura y como esta se puede medir según sus objetivos.

**5.** Las arquitecturas oncloud no son estáticas, por los que siempre se tienen que estar iterando con las áreas las cuales alguna modificación este cambio incida, esto puede ocasionar por lo técnico de las arquitectura, que estas no se entiendan, por lo que se recomienda tener un personal, el cual permita explicar detalladamente estos conceptos técnicos a estas áreas.



## Referencias

- [1] L. H. Sullivan, "The Tall Office Building Artistically Considered". Lippincott's Magazine, pp.403–409, 1896
- [2] R. Shoup, "Randy Shoup on eBay's Architectural Principles", 2007,[En línea]. Disponible en: <https://www.infoq.com/presentations/shoup-ebay-architectural-principles>. [Accedido: 11-12-2019]
- [3] "Cloud Computing", National Institute of Standards and Technology, Special Publication 800-145, 2011 [En línea]. Disponible en: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>. [Accedido: 22-jun-2019]
- [4] ITILV3, "Availability Management", 2018, [En línea]. Disponible en: [https://wiki.en.it-processmaps.com/index.php/Availability\\_Management](https://wiki.en.it-processmaps.com/index.php/Availability_Management). [Accedido: 27-jun-2019]
- [5] E.P.Reyes, "A Systems Thinking Approach to Business Intelligence Solutions Based on Cloud Computing", Master dissertation, Massachusetts Institute of Technology, 2010.[En línea]. Disponible en: <https://core.ac.uk/download/pdf/4423558.pdf>. [Accedido: 27-jun-2019]
- [6] C. Henderson, "What Is Scalability", in Building Scalable Web Sites: Building, Scaling, and Optimizing the Next Generation of Web Applications". USA:O'Reilly Media, 2006, pp. 203.
- [7] M. Usman, T. Rahim, M. Nawaz, "Embedding Project Management Into Xp, Scrum And Rup", European Scientific, vol.10, no.15, 2014 [En línea]. Disponible en: [https://www.researchgate.net/publication/264093672\\_EMBEDDING\\_PROJECT\\_MANAGEMENT\\_INTO\\_XP\\_SCRUM\\_AND\\_RUP](https://www.researchgate.net/publication/264093672_EMBEDDING_PROJECT_MANAGEMENT_INTO_XP_SCRUM_AND_RUP). [Accedido:15-jul-2019]
- [8] Canalys, "Cloud market share Q4 2019 and full-year 2019", 2019 [En línea]. Disponible en: [https://www.canalys.com/static/press\\_release/2020/Canalys---Cloud-market-share-](https://www.canalys.com/static/press_release/2020/Canalys---Cloud-market-share-Q4-2019-and-full-year-2019.pdf)
- [Q4-2019-and-full-year-2019.pdf](https://www.canalys.com/static/press_release/2020/Canalys---Cloud-market-share-Q4-2019-and-full-year-2019.pdf). [Accedido:01-ene-2020]
- [9] E.P.Reyes, "A Systems Thinking Approach to Business Intelligence Solutions Based on Cloud Computing", Master dissertation, Massachusetts Institute of Technology, 2010.[En línea]. Disponible en: <https://core.ac.uk/download/pdf/4423558.pdf>. [Accedido: 27-jun-2019]
- [10] ITILV3, "Availability Management", 2018, [En línea]. Disponible en: [https://wiki.en.it-processmaps.com/index.php/Availability\\_Management](https://wiki.en.it-processmaps.com/index.php/Availability_Management). [Accedido: 27-jun-2019]
- [11] R. Pradhan, "Software Defined Infrastructures - Implications for Technology and Business Strategies for competing in the era of Hyper-Scale Computing", Master dissertation, Massachusetts Institute of Technology, 2015. [En línea]. Disponible en: <http://hdl.handle.net/1721.1/100382>. [Accedido: 27-jun-2019]
- [12] Y. Al-Dhuraibi, "Flexible Framework for Elasticity in Cloud Computing", Doctoral dissertation, Univ. of Lille, 2018. [En línea].Disponible en: <https://tel.archives-ouvertes.fr/tel-02011337/document>. [Accedido: 27-jun-2019]
- [13] N. Mohan, "Edge Computing Platforms and Protocols", Doctoral dissertation, Univ. Lille, Univ. of Helsinki, 2019. [En línea].Disponible en: [https://www.nitindermohan.com/documents/thesis/phdThesis\\_PDF.pdf](https://www.nitindermohan.com/documents/thesis/phdThesis_PDF.pdf). [Accedido: 27-jun-2019]
- [14] D. Zacarías Sánchez, "Implementación de una arquitectura de servicios de TI basada en una cloud privada para la empresa virtual It-Expert", Universidad Peruana de Ciencias Aplicadas, trabajo de fin de grado, 2016 [En línea]. Disponible en: [https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/621941/ZACARIAS\\_SD.pdf?](https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/621941/ZACARIAS_SD.pdf?)

[15] H. Azañedo Silva, N. Bermudez Corzano, R. Zuñiga, Propuesta de metodología para la adopción de plataforma Cloud Computing en entidades públicas, 2016, [En línea]. Disponible en: [https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/621875/Car%c3%a1tula\\_Bermudez\\_CN.pdf?sequence=18&isAllowed=y](https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/621875/Car%c3%a1tula_Bermudez_CN.pdf?sequence=18&isAllowed=y). [Accedido: 27-Jun-2019]

[16] J.A.Villanueva Roman, "Solución de business intelligence utilizando tecnología saas.caso: área de proyectos en empresa bancaria – Perú", Univ. de Piura, trabajo de fin de grado, 2015, [En línea]. Disponible en: [https://pirhua.udep.edu.pe/bitstream/handle/11042/2790/MAS\\_DET\\_032.pdf?sequence=1&isAllowed=y](https://pirhua.udep.edu.pe/bitstream/handle/11042/2790/MAS_DET_032.pdf?sequence=1&isAllowed=y). [Accedido: 27-Jun-2019]