



Diseño de sistema asíncrono de suscripciones de productos

Juan Felipe Gallo Rendón

Informe de prácticas académicas para optar por título de Ingeniería de Sistemas

Asesor Interno Diana Margot López Herrera, Ingeniera en Sistemas UdeA

Asesor Externo Orlando Allan Rodríguez Pamplona

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería de Sistemas
Medellín, Antioquia, Colombia

2022

Cita

(Gallo Rendón, 2022)

Referencia

- [1] Gallo Rendón, J.F. (2022). Diseño de sistema asíncrono de suscripciones de productos [Trabajo de grado profesional]. Universidad de Antioquia, Medellín, Colombia

Estilo IEEE (2020)



Rector: John Jairo Arboleda Céspedes.

Decano/Director: Jesús Francisco Vargas Bonilla.

Jefe departamento: Diego José Luis Botía Valderram.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicatoria

En honor al avance de las nuevas tecnologías de la información, que nos permiten avanzar cada día más, ser más económicos y estar preparados para las adversidades.

Agradecimientos

Para todos aquellos que hicieron posible mi desarrollo profesional y graduarme de tan prestigiosa universidad, a pesar de las necesidades. Mis padres: Elizabeth Rendón y Arley Gallo. Los profesores: Diana Margot López, Gabriel Darío Uribe y Fredy Alexander Rivera. Mi jefe Jonathan Alexander Diosa y todos mis compañeros en Experimentality S.A.S.

TABLA DE CONTENIDO

RESUMEN	6
ABSTRACT	7
I. INTRODUCCIÓN	9
II. OBJETIVOS	10
A. Objetivo general	10
B. Objetivos específicos	10
III. MARCO TEÓRICO	11
IV. PLANTEAMIENTO DEL PROBLEMA	14
V. METODOLOGÍA	15
VI. DESARROLLO DEL PRODUCTO	16
A. Product Backlog	16
B. Casos de Uso	16
C. Principales retos	17
D. Diagrama de la solución planteada	17
E. Diagrama de la arquitectura	18
VII. RESULTADOS	20
VIII. CONCLUSIONES	21
REFERENCIAS	22

LISTA DE FIGURAS

Fig. 1. Ejemplo de tablero ScrumBan	14
Fig. 2. Diagrama del dominio de la solución	16
Fig. 3. Diagrama arquitectura hexagonal de la aplicación.	17
Fig. 4. Diagrama arquitectura de la infraestructura.	18

RESUMEN

Experimentality S.A.S. es una compañía de desarrollo de software que le apuesta a la innovación y desea implementar una metodología de ventas por suscripción en uno de sus desarrollos, haciendo uso de tecnologías en la nube como AWS. Las oportunidades de negocio en una compañía, permiten a los ingenieros, innovar, soñar y explotar su creatividad ideando nuevas soluciones, valiéndose de los conocimientos adquiridos durante la carrera, teniendo en cuenta todos los aspectos técnicos, tecnológicos y funcionales, tales como: rendimiento, concurrencia, tiempos de respuesta, capacidad de cómputo, etcétera. Gracias al desarrollo de las nuevas tecnologías y a la cultura devops se presenta la posibilidad de asumir retos técnicos, partiendo del conocimiento de los procesos y los negocios para presentarlo de la mejor manera. Existen modelos de negocio que permiten aprovechar mejor los recursos y monetizar la venta de manera ágil que se ha manifestado de manera disruptiva, como lo han hecho plataformas como Youtube Premium, Paramount Plus, Amazon Prime, etcétera, que han ofrecido sus productos por medio de un cobro mensual que permite al usuario ahorrar gastos de la obtención de su producto. Se observan técnicas de paradigmas de procesos asíncronos y manejo de la concurrencia que le agregan un toque interesante al proyecto.

Palabras clave — Cloud Computing, Suscripciones, AWS, Asíncrono

ABSTRACT

Experimentality S.A.S. is a software development company that is committed to innovation and wants to implement a subscription sales methodology in one of their developments, making use of cloud technologies such as AWS. Business opportunities in a company allow engineers to innovate, dream and exploit their creativity by devising new solutions, using the knowledge acquired during their career, taking into account all technical, technological and functional aspects, such as: performance, concurrency, response times, computing capacity, etc. Thanks to the development of new technologies and the devops culture, the possibility of assuming technical challenges is presented, starting from the knowledge of processes and businesses to present it in the best way. There are business models that make better use of resources and monetize sales in an agile way that has manifested itself in a disruptive way, as has been done by platforms such as YouTube Premium, Paramount Plus, Amazon Prime, etc., which have offered their products through a monthly charge that allows the user to save costs of obtaining their product. Paradigm techniques of asynchronous processes and concurrency management are observed that add an interesting touch to the project.

***Keywords* — Cloud Computing, Subscriptions, AWS, Asynchronous**

I. INTRODUCCIÓN

En este documento, se da a conocer los resultados de la práctica académica, en la modalidad semestre de industria. Donde, el estudiante dedicó el tiempo al diseño de la solución, en el cual, por términos de confidencialidad, no se revelará cuál es su nombre y el servicio que vende. El proyecto consiste en una aplicación bajo la modalidad de suscripción y se constata que se mostrarán detalles a alto nivel de cómo se llevará a cabo el proceso.

Se realizará el diseño de la arquitectura, estimación, levantamiento de requisitos, prototipos, y una versión preliminar de la solución basada en suscripciones. Debido a los tiempos límites, no se entregará una versión completamente productiva, puesto que un proyecto a esta escala, exige un equipo de desarrollo que esté trabajando tiempo completo para realizar su proyecto en su totalidad.

Los modelos de suscripción, permiten la entrega de los beneficios que trae un producto o servicio, gastando menos recursos por algo que si bien no puede pagarse debido a altos costos, puede ser alquilado y si en algún momento se desea dejar de recibir el beneficio basta con dejar de pagar, y es una consideración justa: ya que se centralizan responsabilidades de una costosa administración del producto a un tercero.

En el pasado, los modelos de suscripción a productos o servicios, se realizaba de forma lenta y manual: si un cliente quería suscribirse a un producto o servicio, debía contactarse con un asesor, el cual, normalmente iba de casa en casa a vender y convencer clientes de que su compra merecía la inversión.

Hoy en día, muchas empresas venden sus productos y servicios, debido a la masificación de los e-commerce como: Amazon, Alibaba, Aliexpress, y otras plataformas, que se dedican a desarrollar tecnologías de la información y realizar avances significativos en los frameworks de desarrollo de software y proveedores de infraestructura (Cloud Providers)[1].

Según <https://www.subscrybe.com/> una empresa danésica dedicada al mercado de suscripciones, cada vez las empresas dejan de ofrecer productos para ofrecer servicios[2] como ha sido el caso de las plataformas como Paramount+, Disney+, HBOMax, YoutubePremum, Spotify, entre otras plataformas dedicadas al streaming han invertido 10,7 billones de dólares en

distribución de contenido en lo que lleva el transcurso del año 2022 y ha obtenido cerca de 731.3 millones de suscriptores[3]. Esto no es algo nuevo, desde la década de 2010 se han explorado alternativas para venta de servicios bajo modelos de suscripciones, como automóviles, bicicletas, patines y hasta gafas para la vista.

Muchas compañías están trabajando bajo este negocio y Experimentality quiere explorar el terreno por medio de este proyecto que saldrá pronto a producción, el proyecto se realizará bajo un enfoque de arquitecturas asíncronas, y se usarán herramientas de IaaS del proveedor de nube, haciendo uso de los servicios necesarios para lograr el objetivo de presentar la versión preliminar de la aplicación, empleando la cultura DevOps, en la cuál interactúan las partes involucradas, tanto técnicas como del negocio, y se tendrá un acompañamiento de los asesores internos de la empresa Experimentality S.A.S. Se usará la metodología ScrumBan para el seguimiento de las tareas y monitorear los avances de la aplicación y por último se mostrará un calendario y un presupuesto para los costos del proyecto.

II. OBJETIVOS

A. *Objetivo general*

Realizar el desarrollo de una aplicación que permita a un e-commerce vender sus productos basada en modelos de suscripción a servicios, haciendo uso de órdenes de compra.

B. *Objetivos específicos*

- Levantar requisitos funcionales del producto con el apoyo de los implicados del negocio.
- Escoger las tecnologías de desarrollo que más se adapten a la solución.
- Estimar costos de infraestructura y desarrollo para la realización del proyecto.
- Desarrollar una fase preliminar del producto que la empresa pueda usar con fines productivos.

III. MARCO TEÓRICO

La crisis actual del COVID 19, ha traído desgracias a muchas personas alrededor del mundo, también ha impulsado el desarrollo tecnológico y la masificación de la tecnología, puesto que: “Estamos atravesando una crisis global, por lo que, independientemente del punto de vista con que desees afrontarlo, hay que destacar que los momentos de crisis a lo largo de la historia han representado un campo lleno de oportunidades que muchos han sabido aprovechar”[4]. Por ello, las tecnologías de la información, que desde mitad del siglo pasado empezaron a tener puja, así como en la segunda guerra mundial, con la computación para fines militares, y luego con la llegada del internet, que muchas empresas se enriquecieron rápidamente, mientras otras caían en bancarrota, como puede verse, en el libro “Por un Scrum popular”: “Silicon Valley era un verdadero manicomio a fines de los noventa. Como telón de fondo de semejante desquicio docenas de startups nacían y morían cada día, al ritmo alocado que marcaban las fluctuaciones en los bolsillos de dueños, inversionistas y curiosos con suerte”

[6]. El desarrollo desenfrenado, ha impulsado a los estudiosos del tema, como Robert C. Martin, Alistair Cockburn y los demás fundadores del Manifiesto Ágil[5]. Comprendieron que

estos sistemas, debían ser estabilizados, y ser sustentables, para que las empresas puedan cumplir con sus objetivos sin estar sometidos ante el azar de una posible bancarrota.

A raíz de la creciente demanda de desarrolladores de software, muchas empresas adoptaron estándares de calidad para evitar el fracaso de los proyectos relacionados con automatización. Muchos de estos estándares, pretenden ser estrictamente rigurosos con la administración del tiempo del desarrollador y del equipo conformado, manteniendo una continua presión sobre el individuo, y del cuál, muchos desarrolladores han sufrido por décadas. Poco después en el año 2008, se inventó el término cultura DevOps: “The concept of DevOps emerged out of a discussion between Andrew Clay and Patrick Debois in 2008. They were concerned about the drawbacks of Agile and wanted to come up with something better. The idea slowly began to spread and after the DevOpsDays event held in Belgium in 2009, it became quite a buzzword.”[6]. El cual fue un término disruptivo en las compañías dedicadas al desarrollo de software, ya que, en vez de medir los equipos y las personas haciendo uso de estándares de calidad que mortifican al desarrollador. Devops se encarga de medir con base a la capacidad del negocio, liberando, tareas de menor tamaño pero con mayor frecuencia (Continuous Delivery and Continuous Integration)[7]. Gracias a esta nueva metodología introducida al mundo del software ágil, muchas compañías, lograron poner pasos firmes en el desarrollo de sus proyectos que han requerido automatización, de este modo, crecieron las comunidades centradas en la tecnología en internet.

Con la llegada de la cultura DevOps, cambió el paradigma del ciclo de vida del software, puesto que antes los roles de operaciones y desarrolladores, se encontraban completamente aislados: mientras el desarrollador quería hacer cambios en el software, el encargado de las operaciones debía mantener la aplicación funcionando para los usuarios. Los constantes cambios en el desarrollo del software vulneran la disponibilidad del producto, ya que cualquier cambio, ya sea de librerías, dependencias, nuevas versiones, nuevos features pueden estropear el funcionamiento correcto del programa. Ahora, los desarrolladores y los encargados de operaciones contribuyen de manera sinérgica en la organización, y se usan otro tipo de métricas para evaluar el desempeño de los equipos[8].

Como una solución que permitió minimizar los costos de mantenimiento de las aplicaciones, surgió la computación en la nube, se pueden reservar funcionalidades específicas de hardware y software para poner en funcionamiento aplicaciones en cuestión de minutos, con alta disponibilidad y alta escalabilidad. Adicionalmente, se pueden contratar servicios administrados, para funcionalidades muy específicas con arquitecturas serverless: bases de datos, proxies, funciones como servicios, servicios de mensajería, servicios de almacenamiento, etcétera. De manera muy intuitiva permite configurar los servicios requeridos para despliegue de infraestructura, por medio de la interfaz gráfica, o puede usarse por medio del CLI (Command Line Interface).

El requerimiento principal del proyecto propuesto en la práctica se definió como una metodología para crear suscripciones de licencias a productos, aprovechando al máximo los recursos utilizados en la nube para manejar procesos con largos tiempos de respuesta.

Se realizan tareas de solicitud de producto a servicio de terceros y entrega del producto vía correo electrónico para cerca de diez mil suscriptores, teniendo en cuenta las limitantes de los servicios del tercero que sólo permite solicitar quinientas suscripciones en una sola petición y veinte peticiones cada sesenta minutos.

Se usó scala como lenguaje de programación, ya que permite aprovechar al máximo los recursos computacionales de manera eficiente con poca cantidad de memoria en una función de AWS, que dispara una petición asíncrona en una cola SQS para procesar varias peticiones en paralelo para la solicitud del producto y el delivery que se demostró duraba diez minutos por cada quinientos suscriptores solicitados. La aplicación, cuenta con un módulo de seguridad de roles y permisos gestionados por un directorio activo y una tabla de permisos en la base de datos, una interfaz gráfica donde se especifican los datos de la suscripción y un csv donde se carga la información de los suscriptores. Los tiempos de respuesta son en promedio de veinte segundos para realizar la petición y luego de finalizar la petición, el servicio de colas se encarga de completar la creación de la suscripción.

IV. PLANTEAMIENTO DEL PROBLEMA

Los microservicios[3], ofrecen múltiples soluciones de desacoplamiento y alta cohesión entre aplicaciones, puesto que, cuanto más crezca una aplicación, más difícil será mantenerla. Cuanto más crece el negocio, más robustas deben ser sus soluciones, o por lo menos, es lo que se puede apreciar en la cultura DevOps: “También es necesario contar con capacidad de ajuste y tener flexibilidad a la hora de preparar los sistemas”[9]. Puesto que, el producto que pretende desarrollarse, es un apoyo a algo que ya se encuentra en ambiente de producción, y debido a la creciente demanda, se ha optado por adoptar decisiones de diseño que permitan ampliar la capacidad del negocio.

Como se puede observar, cada vez las empresas pretenden tercerizar sus operaciones por medio de integraciones con otros servicios: como es el caso de Servientrega, Coordinadora, Logysto, que usan muchos e-commerce, para la entrega de sus productos.

A nivel nacional, a nivel de infraestructura se tercerizan servicios en la nube como AWS, Azure, Google Cloud, etcétera, donde los usuarios de estas plataformas pueden alojar y mantener sus aplicaciones, ahorrando costos excesivos en infraestructura.

El reto principal, reside en desarrollar y verificar que el producto a desarrollar sea realmente rentable y permita dar abasto a las crecientes necesidades del negocio, como es: designar más y mejores recursos a sus operarios, dar mejores incentivos a sus colaboradores, ampliar la capacidad de producción, generar más empleo, y destacar en la innovación como producto nacional.

Para lograr este objetivo, desde esta perspectiva, se desarrolló una aplicación que permite la gestión de órdenes de suscripción mediante un sistema basado en colas alojado en la nube, ya que los servicios son fáciles de integrar con los otros módulos ya existentes de la aplicación.

V. METODOLOGÍA

La metodología Scrumban, permite planificar los proyectos según se ejecuta. Se marcan los hitos, al final de los denominados sprints y, cuando se llega a esa fecha, se revisa el trabajo hecho y se planifica el siguiente sprint o iteración, según los nuevos requerimientos o cambios producidos. En esta metodología se tienen varias columnas, las cuales suelen ser: hecho, demorado, en proceso, por hacer. Cada columna está formada por actividades que todo el equipo comparte. De esta manera, todo el equipo revisa sus tareas, su prioridad y fecha de entrega.

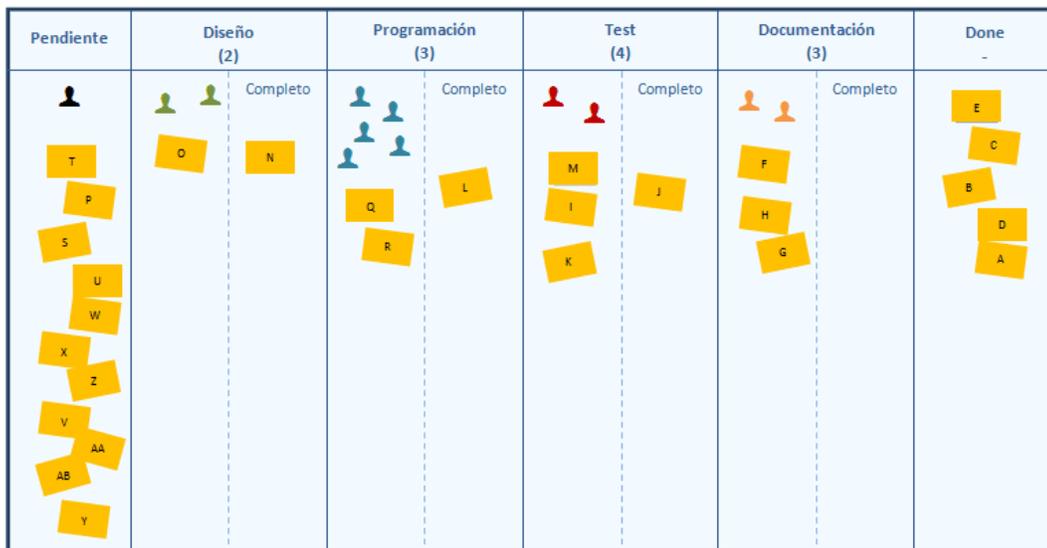


Fig. 1. Ejemplo de tablero ScrumBan

VI. DESARROLLO DEL PRODUCTO

A. *Product Backlog:*

A continuación, se listan los requisitos funcionales para el desarrollo de la primera etapa del proyecto:

Proveer un canal seguro donde se pueda solicitar la suscripción a los productos disponibles.

Realizar integración con servicio web de terceros teniendo en cuenta las consideraciones de seguridad requeridas para la conexión: certificados, banco de contraseñas, etcétera.

Guardar información retornada por el servicio web

Entregar información sobre códigos de activación al cliente solicitante.

Tener un módulo de trazabilidad donde se pueda visualizar el cambio de estados de los suscriptores con respecto al tiempo.

B. *Casos de Uso*

Debido a las limitaciones entregadas por el servicio externo, las suscripciones no pueden activarse en una sola petición, por lo que será un proceso asíncrono y masivo.

- *Crear suscripciones pendientes en el sistema:* El usuario con acceso al sistema, debe poder crear un lote de suscripciones con los datos necesarios para la suscripción.
- *Completar suscripciones:* Al crear el lote de suscripciones pendientes en el sistema, se dispara un trabajo asíncrono que se encarga de la completación de la suscripción, enviando la petición al servicio externo y realizando un delivery de códigos de activación vía correo electrónico.
- *Listar suscripciones:* Tener una interfaz gráfica donde pueda visualizarse las suscripciones creadas por los usuarios y donde pueda verse el detalle de la creación, junto con sus suscriptores y fecha de finalización.

- *Cancelar suscripciones:* El sistema debe tener una interfaz donde se pueda dar de baja los suscriptores correspondientes a suscripciones creadas en el sistema y tener registro de la cancelación.
- *Pausar suscripciones:* El sistema debe tener una interfaz donde se pueda dar de baja los suscriptores con un tiempo para volver a activar la suscripción antes de cancelarla completamente y tener registro del sistema.

C. Principales retos

A nivel tecnológico, se trabajó con un servicio web SOAP sin WSDL definido y todas las acciones debían realizarse bajo la misma petición POST con cambios en el BODY de la petición, para cada comando a realizar, es decir si se querían generar suscripciones para mil suscriptores, debían enviarse mil comandos en el cuerpo de la petición SOAP. A nivel de requerimiento, el negocio no tuvo completamente claro cuál era la necesidad puntual, hasta que surgió la negociación con un cliente y se llegaron a acuerdos que podían desfasar el alcance.

D. Diagrama de la solución planteada

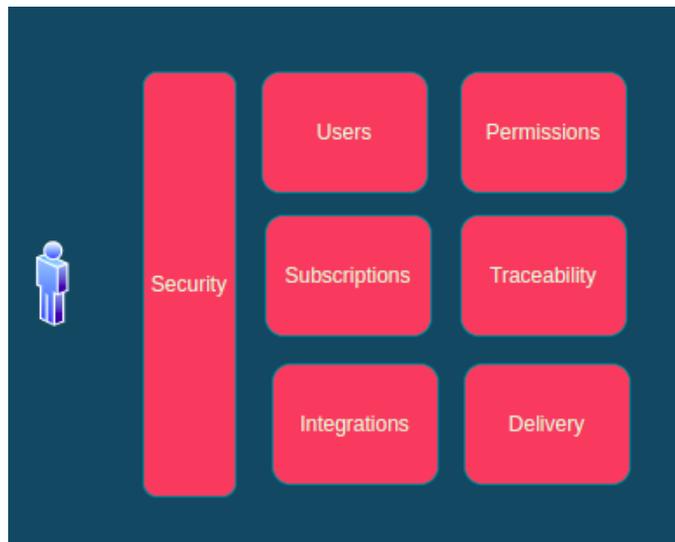


Fig. 2. Diagrama del dominio de la solución

El sistema se encuentra resguardado por un módulo de seguridad que se encarga de validar roles y permisos de los usuarios que ingresan al sistema, el sistema de suscripciones permite la creación y gestión de las mismas. El sistema de trazabilidad se encarga de mostrar los movimientos de las suscripciones creadas por el usuario, las

integraciones de las comunicaciones con el servicio externo y la entrega de códigos de activación vía correo electrónico.

E. Diagrama de la arquitectura.

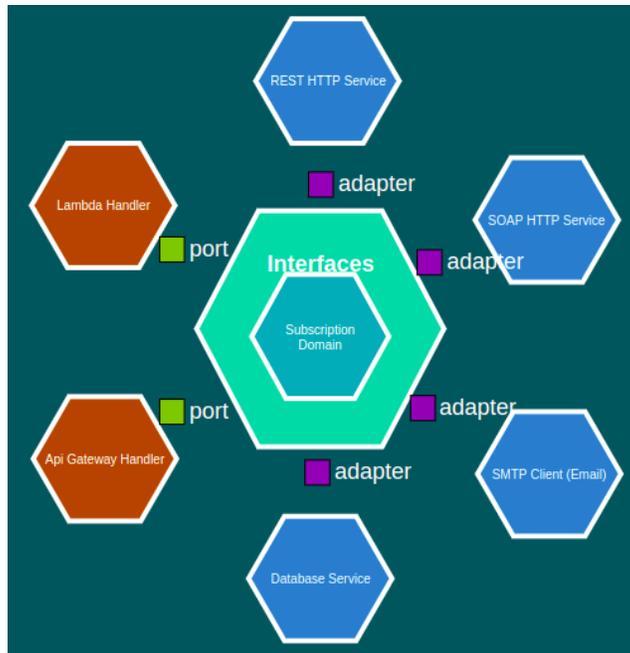


Fig. 3. Diagrama arquitectura hexagonal de la aplicación.

Se trabajó con base a una arquitectura hexagonal donde los puertos son los handlers que disparan los eventos de AWS y los adaptadores son la conexión con los diferentes sistemas con los que se integra la aplicación.

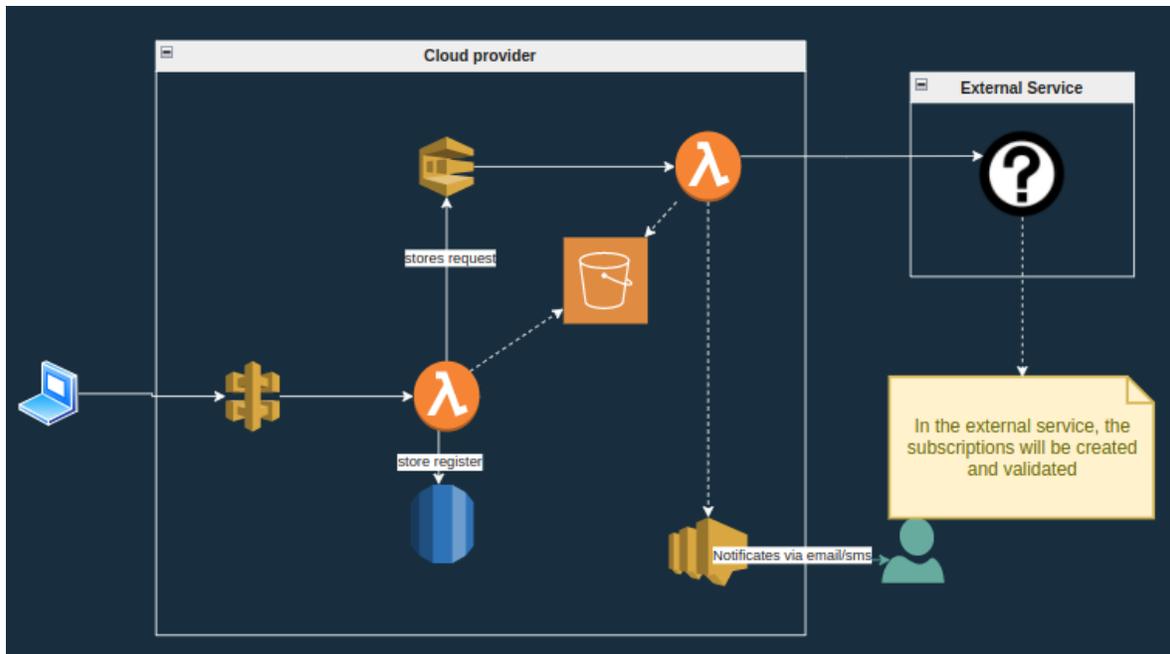


Fig. 4. Diagrama arquitectura de la infraestructura.

Se trabajó bajo una arquitectura serverless, donde los clientes se conectan a la aplicación por medio del servicio API Gateway (Proxy inverso) de AWS, el cual se conecta con una lambda (servicio function as a service de AWS) la cual envía una petición asíncrona a otra lambda desde el servicio de mensajería por colas SQS y finalmente, después de realizar la petición al servicio externo se envía un email con los códigos de activación al usuario.

VII. RESULTADOS

Se desarrolló un aplicación con arquitectura hexagonal, asíncrona y serverless, donde se puso a prueba el nivel técnico del equipo de ingeniería encargado del proyecto, se recibió aprobación de los mentores y arquitectos de la empresa Experimentality S.A.S. en compañía con la asesora Diana Margot, con algunas consideraciones técnicas a nivel de implementación y documentación.

La aplicación se puede crear suscripciones partir de un archivo con la información de los suscriptores que se almacena en primera instancia en la base de datos, en un servicio aparte se completa la creación de la suscripción que se conecta con la API del proveedor de las licencias y se encarga de hacer el delivery a los correos electrónicos de los suscriptores.

Se realizaron pruebas unitarias sobre los trozos más pequeños de código, algunas pruebas de integración para probar el ambiente de pruebas de la API externa, y algunas consultas complejas de la base de datos, además, se realizaron pruebas *to-end* para verificar el funcionamiento de la aplicación antes de desplegar sobre el proveedor de nube. Una vez verificados los escenarios, se realizó el despliegue sobre la infraestructura de la nube en AWS y se realizaron pruebas desde postman para verificar su correcto funcionamiento y después se integró al sistema front-end.

Productos próximos:

El producto mínimo viable será liberado cuando el equipo comercial logre las primeras negociaciones con los clientes, se deben definir precios, principales flujos críticos y ajustes en caso de ser necesario.

Aspectos de mantenimiento, código, secuencias, repositorios, actualizaciones:

El producto está sujeto a múltiples validaciones y ajustes que ayudan al equipo comercial a facilitar la venta y la distribución de suscripciones. En caso de que el sistema falle, el equipo de desarrollo debe estar atento a las notificaciones de los usuarios para brindar el soporte necesario.

El código de la aplicación se encuentra en el repositorio de github empresarial, donde los colaboradores pueden agregar y mantener los cambios que sean requeridos.

El proyecto se encuentra configurado con un pipeline de CICD que se dispara con cada push a las ramas de cada ambiente (master, QA, develop), el artefacto de despliegue se encuentra en un bucket de S3 de AWS del cual se sirven las lambdas, cada vez que se haga un merge a la rama principal se hará una actualización en la versión productiva de esta.

Aspectos preponderantes de la aplicación:

Alta disponibilidad del servicio, arquitectura serverless y cloud native, arquitectura asíncrona, seguridad incorporada, arquitectura hexagonal, alta escalabilidad, microservicios, CICD, excelente trabajo de equipo.

VIII. CONCLUSIONES

Experimentality S.A.S. Al ser una empresa relativamente joven (cinco años en el mercado) aún tiene muchas oportunidades de crecimiento a nivel de estandarización de procesos, como documentación formal, formatos de estimación y reportes de resultados.

El proyecto que se trabajó fue una apuesta a la innovación, ya que en él se implementaron estrategias de desarrollo nuevas que no se habían trabajado anteriormente, y puede ayudar a impulsar mayor cantidad de desarrollos con su contenido, sea en el mismo proyecto u otros, debido a su adaptabilidad, bajo costo y alta disponibilidad.

Las AWS Lambdas son una alternativa de muy bajo costo y eficientes a nivel de manejo de memoria, para ejecutar procesos con menos de quince minutos, pero existen limitantes a nivel de respuesta que el cliente puede percibir como lentas, y es que, cada que se ejecuta un proceso la Lambda debe activarse y este tiempo toma de 3 a 5 segundos. Por otra parte, API Gateway tiene un tiempo límite de respuesta de 30 segundos, por lo que si una petición demora en ejecutarse 25 segundos, el cliente obtendrá un error de timeout, por lo que hay que pensar mucho en la optimización de procesos cuando se quiere desarrollar sobre estas plataformas, y si toman mucho tiempo en ejecutarse procesos es necesario optar por sistemas de colas y procesamiento por lotes.

Trabajar en este proyecto permite a los colaboradores del equipo de desarrollo explotar su creatividad y estar en constante aprendizaje, el conocimiento del proyecto de los participantes logró de esta, ser una experiencia amena para todos, ya que se cumplieron los objetivos planteados y se mostraron resultados calidosos con prontitud.

REFERENCIAS

- [1]. SOLVED. 2021. The history of cloud computing - SOLVED. [online] Available at: <https://www.scality.com/solved/the-history-of-cloud-computing/> [Accessed 14 November 2021].
- [2]. The biggest subscription trends of 2022. Subscribe. (2022). Retrieved 27 February 2022, from <https://www.subscribe.com/the-subscription-movement-keeps-rolling-here-are-the-biggest-trends-of-2022/>.
- [3]. microservices.io. 2021. What are microservices?. [online] Available at: <https://microservices.io/> [Accessed 14 November 2021].
- [4]. CALLEJAS-CUERVO, Mauro; ALARCÓN-ALDANA, Andrea Catherine; ÁLVAREZ-CARREÑO, Ana María. Modelos de calidad del software, un estado del arte. En: Entramado. Enero - Junio, 2017. vol. 13, no. 1, p. 236-250, <http://dx.doi.org/10.18041/entramado.2017v13n1.25125> [Accessed 13 September 2020]
- [5]. Caf.com. 2021. [online] Available at: <https://www.caf.com/es/conocimiento/visiones/2020/05/covid19-una-oportunidad-para-la-innovacion-tecnologica/> [Accessed 14 November 2021].
- [6]. Mayer, T. and Cymment, A., 2014. Por un scrum popular. Sausalito, CA: Dymaxicon.
- [7]. Agilemanifesto.org. 2021. Manifesto for Agile Software Development. [online] Available at: <https://agilemanifesto.org/> [Accessed 14 November 2021].
- [8]. Agarwal, H., 2021. Roadway to IT Revolution: The History of DevOps. [online] Appknox.com. Available at: <https://www.appknox.com/blog/history-of-devops> [Accessed 14 November 2021].
- [9]. AppDynamics. 2021. DevOps Metrics and KPIs: How To Measure DevOps?. [online] Available at: <https://www.appdynamics.com/topics/devops-metrics-and-kpis#~7-customer-tickets> [Accessed 15 November 2021].