



Diseño, implementación y despliegue de Arquitectura de microservicios Cloud Native para un Ecommerce

Andrés Felipe Arango Ortiz

Trabajo de grado presentado para optar al título de Ingeniería de Sistemas

Asesor

Joan Andres Hasper Tabares, Magister en Administración de negocios

Universidad de Antioquia

Facultad de Ingeniería

Pregrado

Medellín

2022

Cita	(Arango Ortiz, 2022)
Referencia	Arango Ortiz, A. F. (2022). Diseño, implementación y despliegue de Arquitectura de microservicios Cloud Native para un Ecommerce [Trabajo de grado profesional]. Universidad de Antioquia, Medellín.
Estilo APA 7 (2020)	



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: : John Jairo Arboleda Céspedes

Decano/Director: Jesús Francisco Vargas Bonilla

Jefe departamento: Diego José Luis Botía Valderrama.

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.

Agradecimientos

A Dios, a todas esas personas que a lo largo de los años me apoyaron y estuvieron en momentos difíciles, a todos los docentes que he tenido a lo largo de mis años, a la Universidad de Antioquia por el espacio para adquirir conocimientos y por último a Accenture Ltda. por la oportunidad y la confianza en mis capacidades.

Tabla de contenido

Resumen	9
Abstract	10
Introducción	11
1 Objetivos	13
1.1 Objetivo general	13
1.2 Objetivos específicos	13
2 Marco teórico	14
3 Metodología	16
4 Resultados	18
Sprint 1:	18
Sprint 2:	20
Sprint 3:	22
Sprint 4:	25
5 Análisis	28
6 Conclusiones	29
Referencias	30

Lista de figuras

Fig 1. Vista de los repositorios	19
Fig 2. Vista del backend monolito para productos	20
Fig 3. Vista del backen para productos en microservicios (GET, PUT, DELETE Y POST)	21
Fig 4. Vista al frontend final para productos	21
Fig 5. Vista del backend monolito para usuarios	22
Fig 6. Arquitectura final del proyecto	23
Fig 7. Vista del frontend final de clientes	23
Fig 8. Vista frontend final para órdenes	24
Fig 9. Microservicios finales para clientes y ordenes	24
Fig 10. Vista de los layers usados en el proyecto	25
Fig 11. Actualización de IaC a CDK	26
Fig 12. Vistazo de pipelines finales para el proyecto	26

Siglas, acrónimos y abreviaturas

DB	Database
AWS	Amazon Web Services
SQL	Structured Query Language
CI	Continuous Integration
MP	Continuous Distribution
CI/CD	Continuous Integration /Continuous Distribution
IaC	Infrastructure as Code
CDK	Cloud Development Kit
HU	Historia de Usuario
DevOps	Development and Operations
CRUD	Create, Read, Update, Delete

Resumen

Con el paso de los años el internet ha tomado mucha relevancia en nuestra vida diaria, siendo actualmente un elemento indispensable para el desarrollo y la evolución de esta. Por ello, uno de los aspectos que más se han visto afectado y/o beneficiado por el internet ha sido el comercio, el cual ha encontrado un espacio donde circular 24/7 sin descanso y al alcance de cualquier persona. Por ello, la finalidad de este trabajo fue realizar el diseño, implementación y despliegue de una arquitectura de microservicios en la nube para un Ecommerce (comercio electrónico), haciendo uso de la metodología SCRUM y utilizando las herramientas más usadas actualmente en la industria. Como resultado, se mostrará un sitio web con una alta disponibilidad para los usuarios y con el uso de las mejores prácticas con el fin de generar un producto escalable y que aporte un gran beneficio al usuario que lo requirió.

Palabras clave: ecommerce, cloud, devops, aws, scrum.

Abstract

Throughout the years, the internet has become very relevant in our daily lives, being currently an indispensable element for the development and evolution of it. Therefore, one of the aspects that have been most affected and benefited by the Internet has been commerce, which has found a space where to circulate 24/7 without a break and within the reach of anyone. In addition, the purpose of this project was to design, implement and deploy a microservices architecture in the cloud for an E-commerce, making use of the SCRUM methodology and using the most widely used tools in the industry. As a result, it will show a website with high availability for users and with the use of best practices in order to generate a scalable product that brings a great benefit to the user who requires it.

Keywords: ecommerce, cloud, devops, aws, scrum.

Introducción

Aún hace unos años atrás pocas personas creían y/o pensaban en el Internet como un nuevo canal de distribución y comunicación entre empresas y consumidores (Gonzales, 2006), pero ahora la situación es diferente ya que el año pasado aproximadamente el 21,55% de la población mundial (*más de 1.500 millones de personas*) hacia sus compras por internet (Campos-Sánchez, Cedeño-Palacios, Palma-García, Campos-Sánchez, 2021).

Por consiguiente, tener presencia en internet es fundamental para los negocios, ya que esto significa una mayor cantidad de público que puede acceder a los productos o servicios. También, como se pudo evidenciar en época de pandemia, tener presencia en internet ayuda a que un negocio pueda seguir funcionando e incluso generar un crecimiento.

Por tal motivo, en el proyecto a realizar se planea desarrollar un sitio web para una tienda en la cual los usuarios podrán ingresar y realizar sus compras, esto con el fin de brindar un crecimiento al negocio y darle más presencia en lugares alejados o como en la pandemia, poder seguir ofreciendo los servicios aun cuando hay confinamiento.

Para lograr tener esta presencia en internet, altamente disponible y segura, se utilizaran los beneficios de la Nube de AWS, la plataforma líder por 10 años consecutivos de acuerdo a Gartner. Habilitando el uso de recursos de infraestructura bajo demanda y escalables dinámicamente de acuerdo a la necesidad real del negocio y/o concurrencia, permitiendo el ahorro de costos al pagar únicamente por los recursos reales que se utilizan y beneficiandonos con los servicios administrados del proveedor, los cuales facilitan la configuración y mantenimiento de los mismos.

A lo largo del proceso se irá adquiriendo conocimientos en DevOps (haciendo uso de Azure DevOps para este fin), de servicios cloud de AWS, buenas prácticas a la hora de trabajar en proyectos y una base de cómo funciona la industria actualmente. Cabe resaltar que este

proyecto se hará de manera remota y teniendo en cuenta aspectos estudiados a lo largo de la carrera.

1 Objetivos

1.1 Objetivo general

Diseñar, implementar y desplegar una Arquitectura de microservicios Cloud Native de 3 capas para un e-commerce altamente disponible, escalable, seguro y con alto desempeño, aplicando metodologías ágiles.

1.2 Objetivos específicos

- Diseño de arquitectura en la nube.
- Realizar desglose de tareas y actividades con el fin de construir las historias de usuario.
- Utilizar la estrategia de versionamiento Trunk-based basado en microcambios
- Crear base de datos (DB) para clientes y productos, usando un motor SQL (MySQL)
- Desarrollar backend en Node.js (con el lenguajes JavaScript) para clientes y productos
- Desarrollar frontend con el framework Angular (con el lenguajes JavaScript) para clientes y productos
- Crear una base de datos para el manejo de productos y de clientes.
- Hacer uso de los servicios de AWS para aspectos como: Bases de datos, despliegue de Back y Front, seguridad y monitoreo.
- Cifrar los datos guardados en reposo y tránsito.
- Implementar integración y despliegue continuo (CI/CD) por medio de Azure DevOps haciendo uso de pipelines.
- Implementar IaC para los recursos de AWS (con Cloudformation y CDK)
- Desplegar y testear el funcionamiento del sitio web.

2 Marco teórico

Al momento de la creación de internet, pocas personas creyeron que iba a influir demasiado en la sociedad, pero luego del año 1999, se pasó a un crecimiento activo y acelerado hasta la fecha (Gonzales, 2006). Además, el crecimiento de la infraestructura y la cantidad de personas que acceden a internet ha venido aumentando a lo largo de los años.

Debido a este crecimiento acelerado, los negocios convencionales han estado adaptándose y para ello es necesario aplicar soluciones apropiadas de comercialización electrónica que garanticen el éxito sostenido del negocio (Gerrikagoitia, et al., 2014). Con la implementación de estas prácticas nacen los e-business o negocio digital los cuales pueden concebirse como un método de gestión empresarial que utiliza las TIC (Brzozowska & Bubel, 2015).

En los últimos años los negocios digitales han tenido grandes beneficios: sólo en el 2020 cerca del 21,55% de la población mundial (más de 1.500 millones de personas) hacia sus compras por internet (Campos-Sánchez, Cedeño-Palacios, Palma-García, Campos-Sánchez, 2021). Por este motivo, la seguridad de los negocios electrónicos es uno de los componentes visibles que más afectan al usuario final durante sus interacciones económicas en el mercado digital (Niranjanamurthy, 2013).

Por consiguiente, una de las mejores opciones en temas de flexibilidad rigurosa, aplicaciones rápidas y sencillas, escalabilidad de la solución, mantenimiento de datos seguros y alcance masivo de las operaciones (Brañes, 2019) es AWS, ya que cuenta con más de 200 servicios integrales de centros de datos a nivel global ("¿Qué es AWS?", 2021), entre los cuales uno de ellos puede detectar clases enteras de configuraciones erróneas que podrían exponer potencialmente datos vulnerables (Rungta, N. 2019) ayudando en temas de seguridad.

Asimismo, para mantener una alta disponibilidad en todo momento se recomienda usar DevOps el cual permite que se unan los equipos de Desarrollo (Dev) y los equipos de

Operaciones (Ops), permitiendo que el proceso de desarrollo y lanzamiento del software sea más eficiente, obteniendo así resultados más satisfactorios (Diaz-Crespo, 2018). Para ello, una buena opción es Azure DevOps, el cual es un servicio en la nube que proporciona servicios y herramientas para apoyar el desarrollo y entrega continua de software (Castro).

3 Metodología

Para la realización del proyecto se trabajó con la metodología SCRUM, en la cual se contó con 4 sprints y cada sprint con una duración de 2 semanas. Además, se tuvieron las siguientes actividades principales por sprint:

- Sprint 1:
 - Definición y corrección de arquitectura en la nube.
 - Creación de historias de usuario (HU) y repositorios (para backend y frontend) en Azure DevOps.
 - Desarrollo de backend y frontend para productos, teniendo en cuenta la metodología trunk-based development para los repositorios en Azure DevOps.
- Sprint 2:
 - Correcciones a backend y frontend de productos.
 - Implementación de IaC con Cloudformation para productos.
 - Despliegue del backend y frontend de productos con pipelines usando CI/CD en Azure DevOps.
- Sprint 3:
 - Actualización de la arquitectura.
 - Correcciones a backend y frontend de clientes.
 - Implementación de IaC con Cloudformation para clientes.
 - Despliegue del backend y frontend de clientes con pipelines usando CI/CD en Azure DevOps.
- Sprint 4:
 - Entrega del sitio web, el cual tendrá IaC con AWS CDK y pipelines de despliegue con CI/CD en Azure DevOps.

También, hay 2 actividades las cuales se realizan antes o durante el primer sprint y son:

- Realización del cronograma de actividades: se realiza con el fin de estimar la duración de cada actividad planteada en el sprint y tratar en la medida de lo posible de cumplir con los plazos de entrega.
- Estimación de costos para el proyecto: elaborado para conocer el coste total del proyecto y cerciorarse de que el presupuesto destinado para el proyecto sea suficiente para finalizarlo.

4 Resultados

En el transcurso del proyecto se recibió una serie de capacitaciones por parte de personal capacitado de la empresa con el fin de mejorar y adquirir nuevos conocimientos que ayudaran en el desarrollo del proyecto. También, se confirió acceso a una serie de cursos para explorar alternativas a la hora de realizar las diferentes actividades del proyecto.

El proyecto se dividió en 4 partes, las cuales fueron: funcionalidades para productos, funcionalidades para clientes, pipelines de despliegue e IaC, esto con el fin de enfocarse en asuntos específicos en cada sprint y poder tener un control en los entregables.

Al final del proyecto, se logró cumplir con el objetivo general y se creó un Ecommerce funcional con los requerimientos solicitados y/o planteados, además, de que esto se llevó a cabo en el tiempo estipulado y sin sobrepasar el presupuesto asignado para la ejecución del proyecto.

En cuanto a los resultados puntuales obtenidos en cada sprint, se detallaran a continuación:

- **Sprint 1:**

- *Definición de arquitectura:* Se realiza la definición de la arquitectura y los diferentes elementos que se implementaran en ella. Esta arquitectura fue totalmente implementada en AWS. La arquitectura consta de:
 - Una capa para seguridad para proteger el sitio web de ataques malintencionados
 - Dos zonas de disponibilidad (para evitar caídas del sitio en caso de un problema con AWS).
 - DB para guardar los diferentes productos así como la información de los usuarios y los registros de compras.

- *Creación de repositorios:* Se crean los repositorios en los cuales se aloja el código fuente del aplicativo. Estos repositorios serán los que se conecten a los pipelines para un CI/CD.

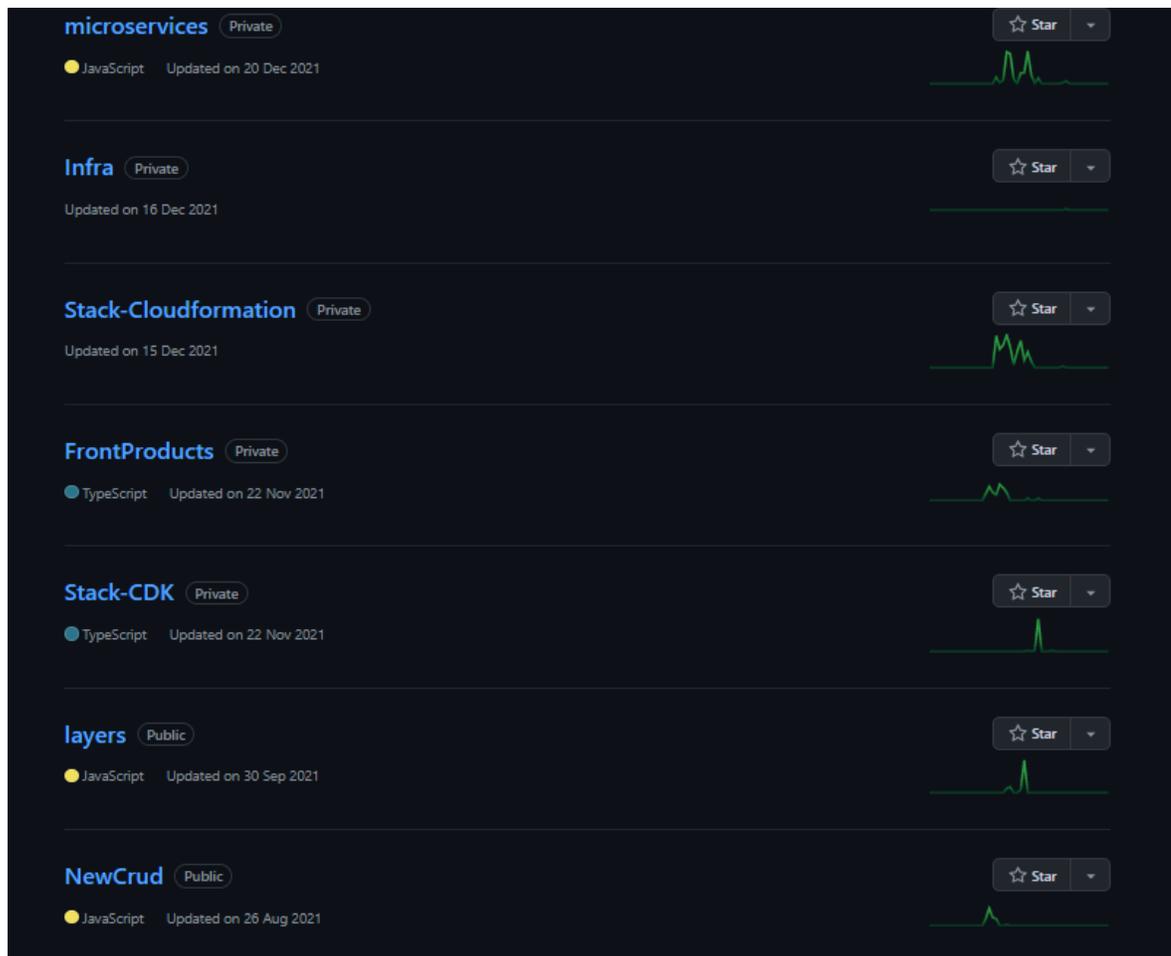


Fig 1. Vista de los repositorios

- *Desarrollo de backend de productos:* Se desarrolla el backend que consta de un CRUD para la creación, actualización, listado y eliminación de productos.

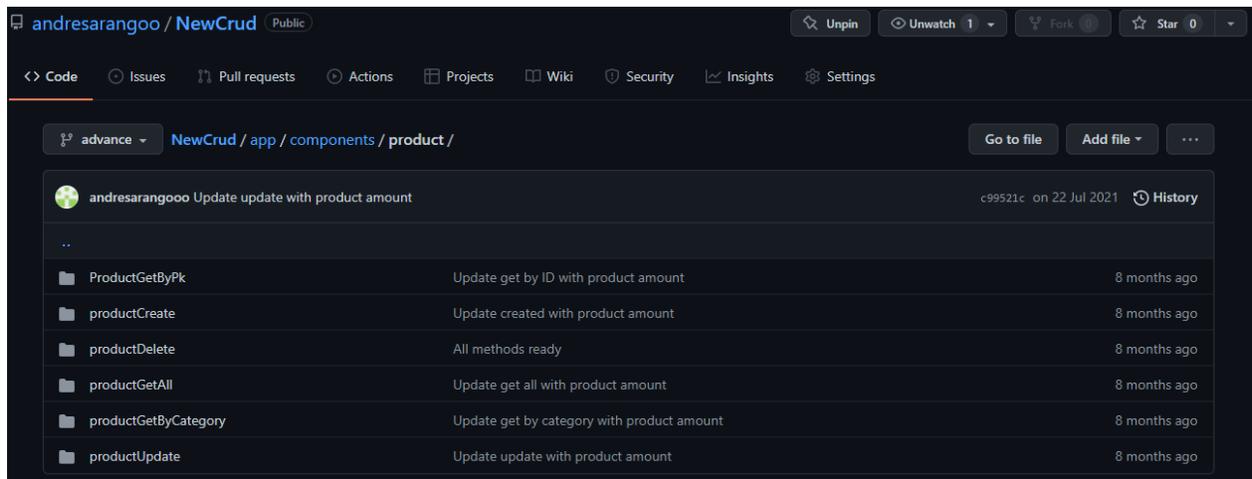


Fig 2. Vista del backend monolito para productos

- *Desarrollo de frontend de productos:* Se desarrolla el frontend que consta de los componentes visuales necesarios para conectarse y consumir el backend (usar los servicios expuestos del backend).
- **Sprint 2:**
 - *Correcciones backend de productos:* Se realizan las correcciones necesarias al backend y se cambia de ser un monolito a microservicios.

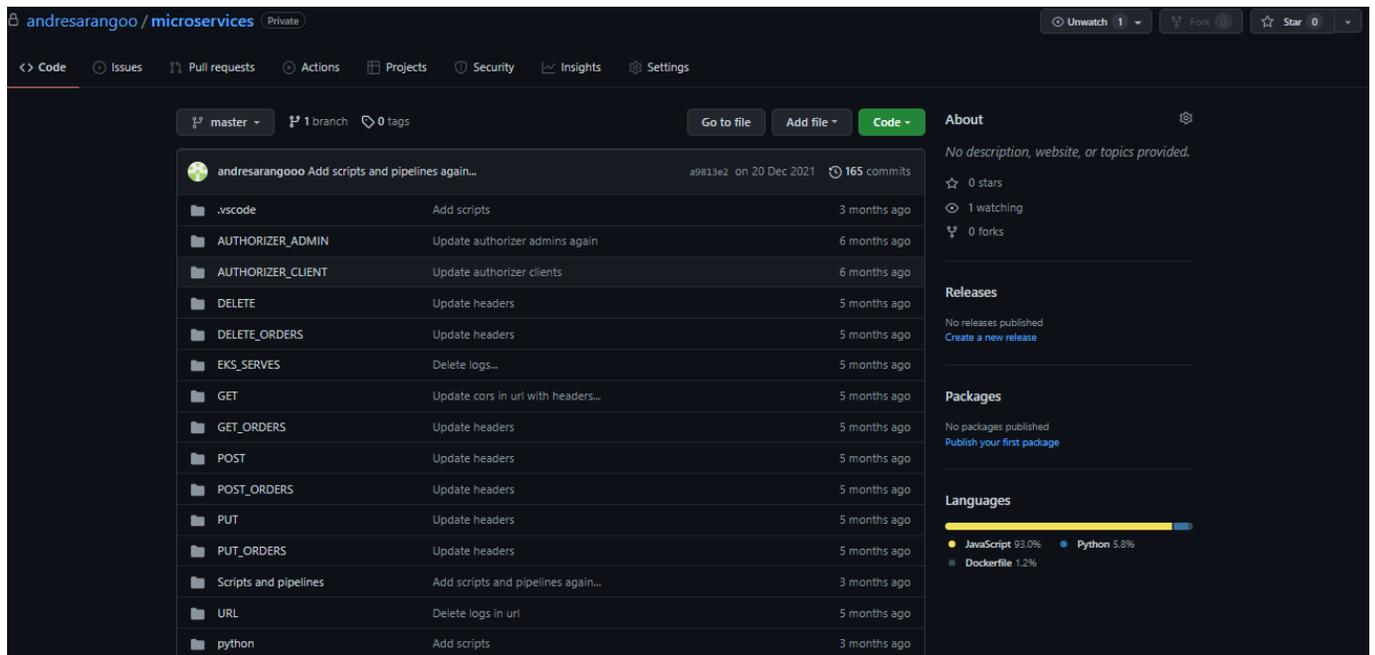


Fig 3. Vista del backen para productos en microservicios (GET, PUT, DELETE Y POST)

- *Correcciones frontends de productos:* Se actualiza el frontend para así poder poder consumir el nuevo backend y realizar los cambios visuales sugeridos por el asesor externo.

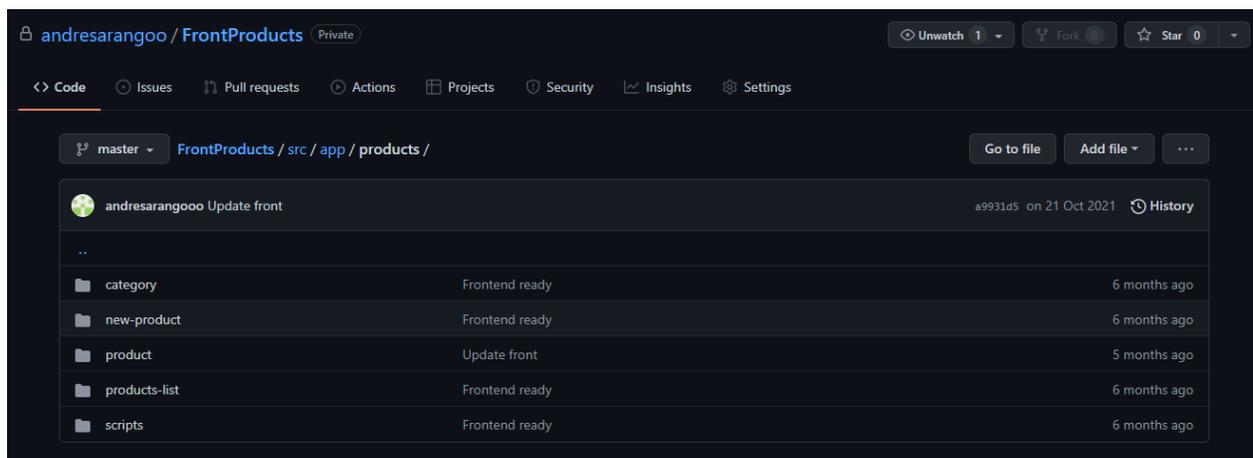


Fig 4. Vista al frontend final para productos

- *Implementación de IaC para productos:* Se construye la infraestructura necesaria en AWS para alojar el backend y frontend de los productos.

- *Despliegue con pipelines:* Se procede a crear los pipelines, conectarlos con los repositorios (de backend, frontend e IaC) y al final conectarlos a AWS, para así, tener un CI/CD en lo que queda del proyecto para la parte de productos.
- *Desarrollo de backend de clientes:* Se desarrolla el backend que consta de todo el flujo y métodos de autenticación, así como los roles para los clientes. Además, se agrega las funcionalidades para realizar órdenes y compras

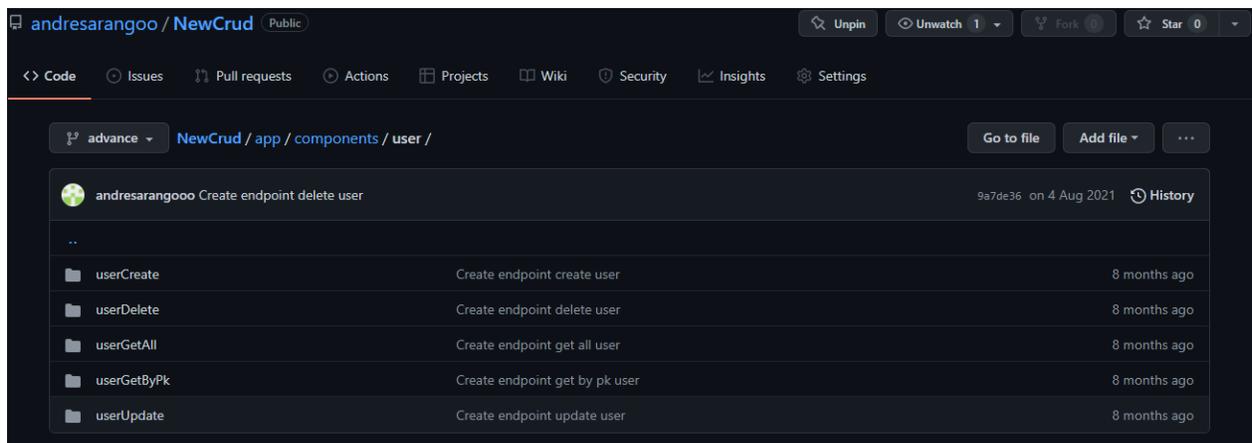


Fig 5. Vista del backend monolito para usuarios

- *Desarrollo de frontend de clientes:* Se desarrolla el frontend que consta de los componentes visuales necesarios para conectarse y consumir el backend (usar los servicios expuestos del backend). Adicionalmente, se crean los componentes visuales para realizar manejo de usuarios (creación, actualización y eliminación de usuarios) y manejo de órdenes y compras
- **Sprint 3:**
 - *Actualización de arquitectura:* Se realiza la actualización de la arquitectura en base a recomendaciones realizadas por el asesor externo y consideraciones hechas por el desarrollador (en este caso, mi persona).

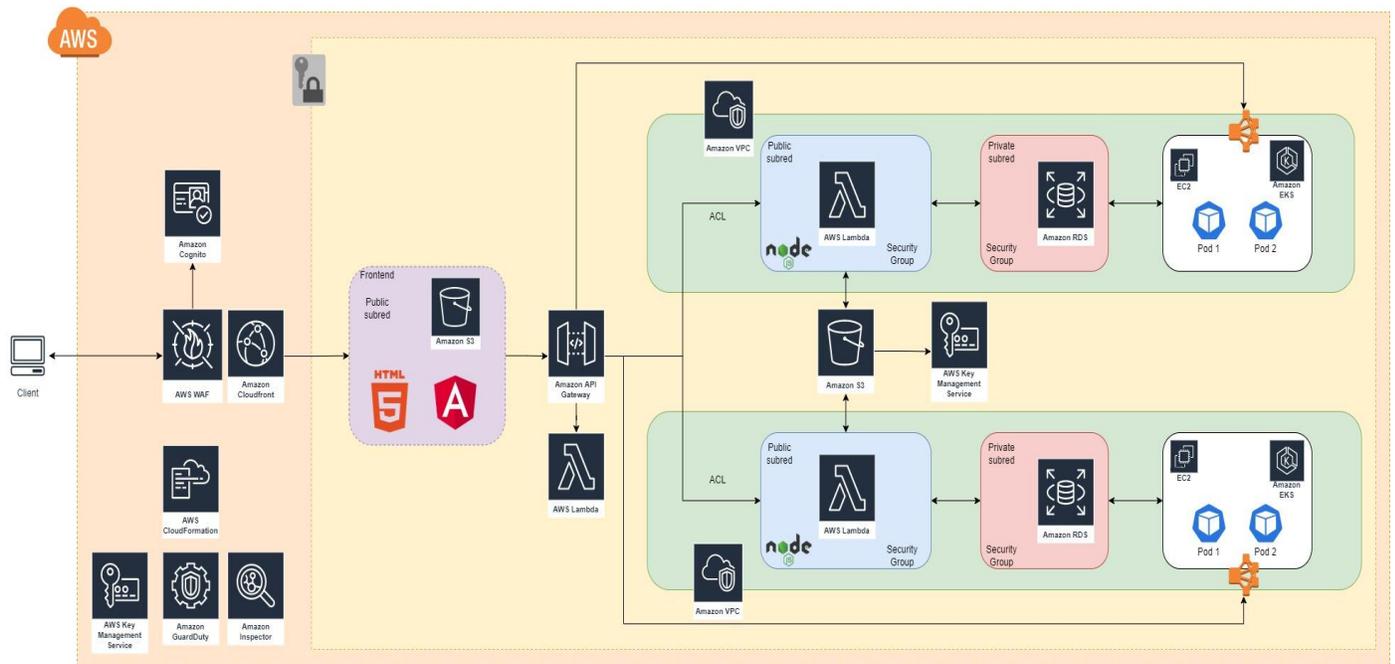


Fig 6. Arquitectura final del proyecto

- *Correcciones frontend de clientes:* Se actualiza el frontend para así poder consumir el nuevo backend y realizar los cambios visuales sugeridos por el asesor externo.

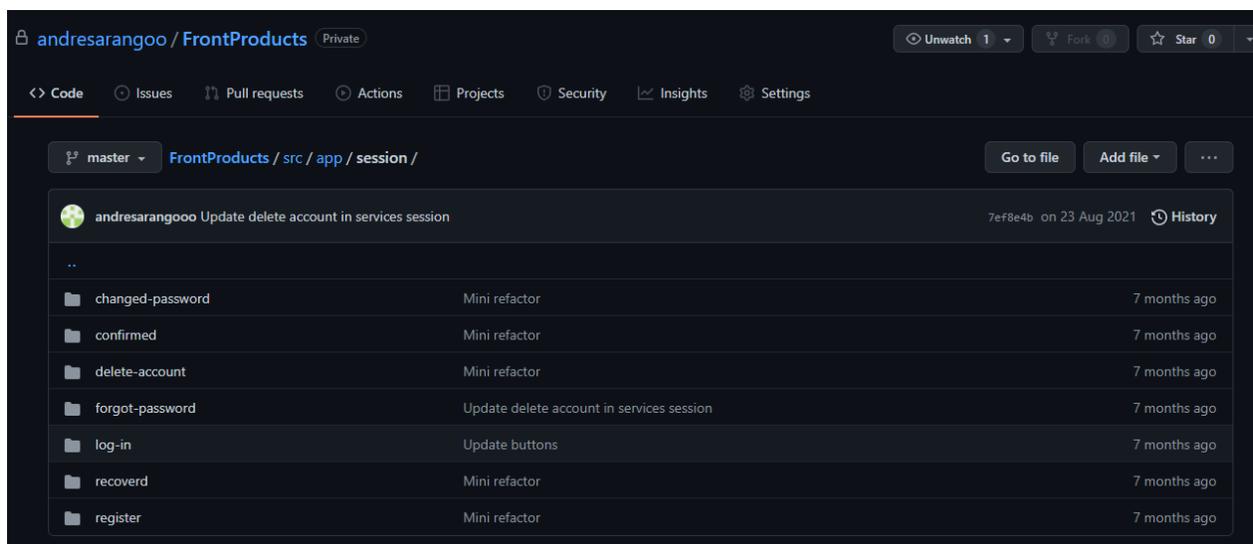


Fig 7. Vista del frontend final de clientes

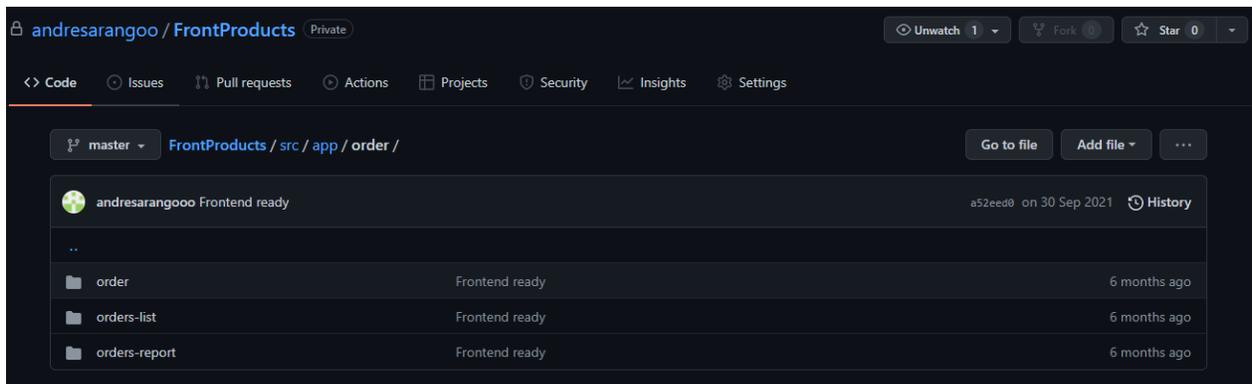


Fig 8. Vista frontend final para órdenes

- *Correcciones backend de clientes:* Se realizan las correcciones necesarias al backend y se cambia de ser un monolito a microservicios.

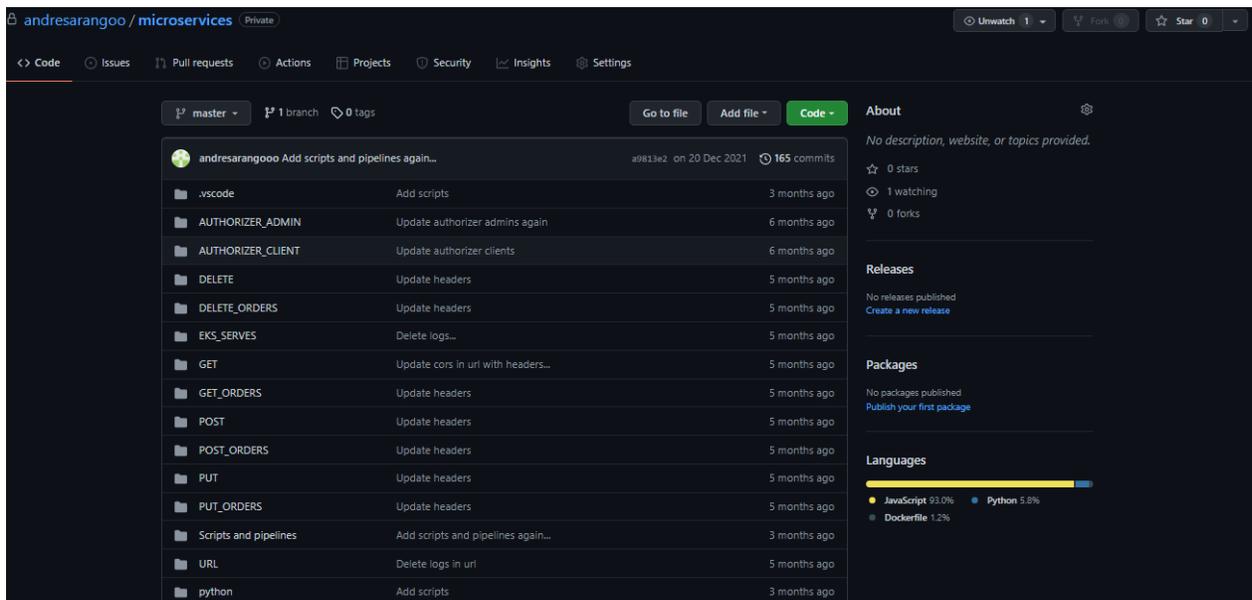


Fig 9. Microservicios finales para clientes y ordenes

- *Implementación de layers:* Por el uso de microservicios (lambdas en AWS) se realiza la creación de layers (capas para código común compartido por las lambdas), además, de sus respectivos pipelines.

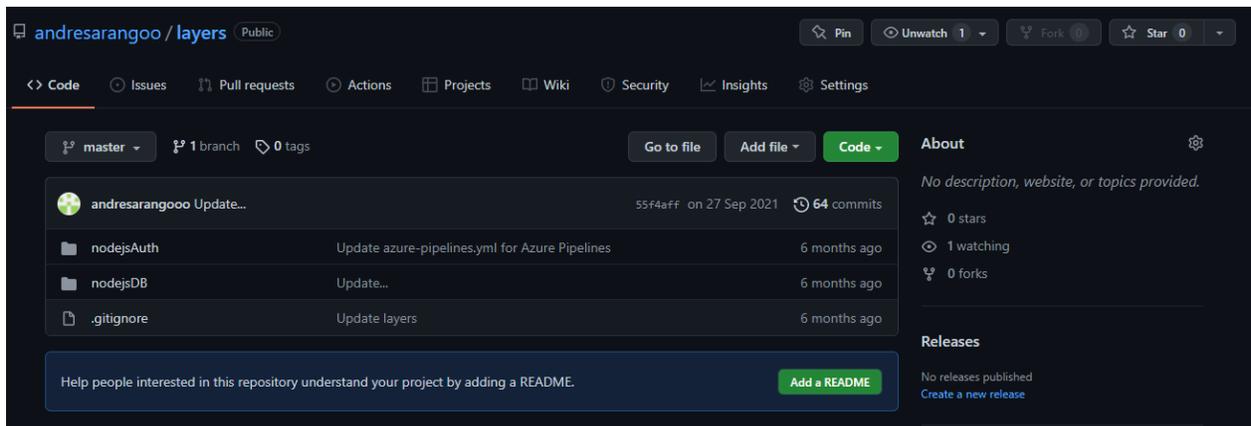


Fig 10. Vista de los layers usados en el proyecto

- *Implementación de IaC para clientes:* Se construye la infraestructura necesaria en AWS para alojar el backend y frontend de los clientes. También, se consolida con la infraestructura de los productos.
- *Despliegue con pipelines:* Se procede a crear los pipelines, conectarlos con los repositorios (de backend, frontend e IaC) y al final conectarlos a AWS, para así, tener un CI/CD en lo que queda del proyecto para la parte de productos.
- **Sprint 4:**
 - *Actualización de IaC:* Se cambia la herramienta usada para IaC (de CloudFormation a CDK) y se realizan los ajustes necesarios para que la infraestructura no se altere y no se deba realizar mejoras a lo ya obtenido en sprints anteriores.

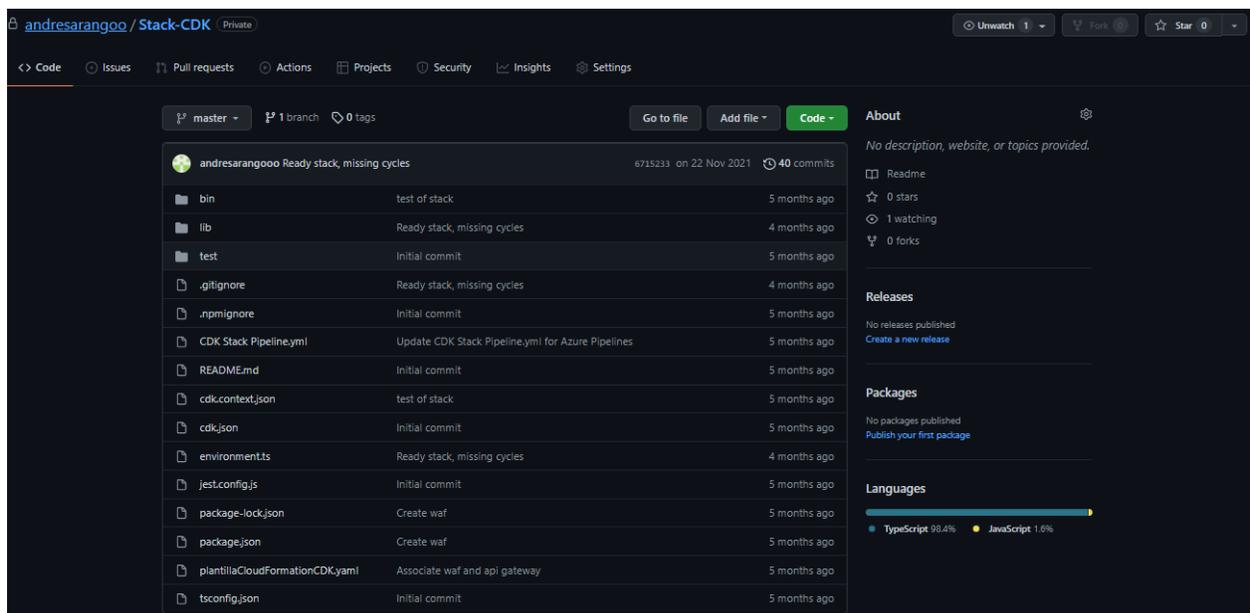


Fig 11. Actualización de IaC a CDK

- *Actualización de pipelines:* Se actualizan los pipelines para IaC y se corrigen los ya existentes en caso de ser necesarios.

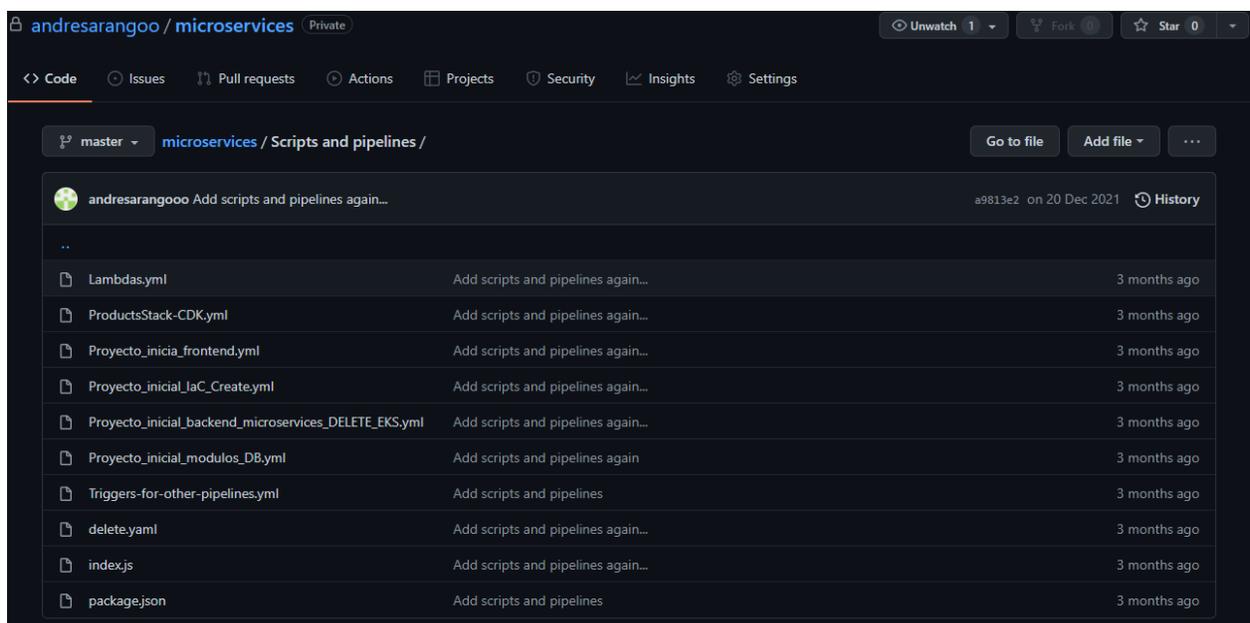


Fig 12. Vistazo de pipelines finales para el proyecto

- *Sitio web funcional*: Se hace entrega del sitio web para Ecommerce contando con los módulos necesarios para el manejo de productos y clientes (creación, listado, actualización y eliminación), el manejo de pedidos y compras y lista el registro de pedidos y compras.

5 Análisis

El desarrollo del sitio web para Ecommerce tiene un alto impacto en la manera de consumir un producto, esto debido a que permite a los consumidores tener un acceso 24/7 en caso de que lo necesiten y se traduce se puede traducir en un aumento en las ganancias de los dueños del sitio. También, permite llegar a posibles clientes que no se encuentren en la misma ciudad, región o incluso país y/o continente, lo cual da un posible margen de crecimiento al negocio.

Cabe resaltar que aunque el sitio puede generar un coste adicional al dueño del mismo, los beneficios pueden sobrepasar por mucho dichos costos, además, puede ayudar a generar más empleos ya sea en el área que maneja el Ecommerce o en otras no relacionadas, como lo pueden ser a desarrolladores para mantener el sitio web en óptimas condiciones o personas encargadas de hacer los envíos en caso de ser necesario.

En lo que se refiere a la parte de desarrollo y mantenimiento, el sitio al estar en la nube (haciendo uso de servicios de AWS), tiene una alta posibilidad de mejora continua además de crecer, sin invertir mucho esfuerzo y/o dinero para ello.

Como conclusión, el desarrollo del Ecommerce da la oportunidad de crecer no solo al negocio para el cual se construye, sino también a las personas involucradas en el desarrollo de este mismo. Adicionalmente, ayuda al aumento de las oportunidades del negocio como de las demás partes involucradas.

6 Conclusiones

- El hacer el despliegue de un proyecto en la nube agiliza en gran medida el desarrollo de este debido a que todas las partes involucradas tienen acceso rápido a lo que necesitan y ayuda a no depender de una máquina o entorno específico.
- El hacer usos del marco de trabajo DevOps ayuda en gran medida a la puesta en marcha de las diferentes partes del proyecto, además, disminuye en gran medida la intervención directa por parte de los miembros del equipo de desarrollo al momento de realizar los diferentes despliegues.
- Hacer uso de modelo Serverless ayuda en gran medida a mantener el sitio funcional debido a que los diferentes módulos están poco acoplados, también, ayuda a reducir costos ya que en este modelo se realiza cobro por consumo de recursos sin necesidad de apartar de antemano una cantidad fija de recursos.
- La necesidad de seguir aprendiendo nuevas metodologías, modelos y/o marcos de trabajo y tecnologías hacen que continuamente nos estemos retando a nosotros mismos con el fin de estar a la vanguardia en nuestro campo laboral.
- El acompañamiento de la Universidad, la empresa y los asesores (interno y externo) ayudan al crecimiento en la primera experiencia laboral, además, ayuda a pulir nuestras competencias y enfocarnos en aquello que más nos llama la atención.
- La realización de un proyecto se puede influenciada por factores externos a este, además, puede que tenga cambios significativos a lo largo de su ejecución. Por tal motivo es necesario tener una comunicación asertiva entre todos los agentes involucrados y estar abiertos a los cambios que se puedan presentar.

Referencias

- ¿Qué es AWS?. Amazon Web Services, Inc. (2021). Retrieved 2 August 2021, from <https://aws.amazon.com/es/what-is-aws/>.
- Álvaro Díaz-Crespo, M. (2018). Estudio y desarrollo de funcionalidades de la herramienta “Chef” e integración con el entorno Azure.
- Brañes Vilchez, R. E. (2019). Arquitectura de back end con amazon web services (AWS) para sistemas escolares.
- Brzozowska, A., & Bubel, D. (2015). E-business as a new trend in the economy. *Procedia Computer Science*, 65, 1095-1104.
- Campos-Sánchez, S. T., Cedeño-Palacios, C. A., Palma-García, Y. V., & Campos-Sánchez, V. S. (2021). Los negocios digitales y su importancia en la era actual. *Dominio de las Ciencias*, 7(1), 1029-1043.
- Coronado Castro, P. (2019). Gestionando el ciclo de vida de aplicaciones MEAN desplegadas en contenedores Docker con Azure DevOps.
- Gerrikagoitia, J. K., Castander, I., Rebón, F., & Alzua-Sorzabal, A. (2015). New trends of Intelligent E-Marketing based on Web Mining for e-shops. *Procedia-Social and Behavioral Sciences*, 175, 75-83.
- González, J. M. C. (2006). Internet: Oportunidades de negocios y otras perspectivas relacionadas. *Posgrado y Sociedad*, 6(1), 58-90.
- Niranjanamurthy, M., Kavyashree, N., Jagannath, S., & Chahar, D. (2013). Analysis of e-commerce and m-commerce: advantages, limitations and security issues. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(6), 2360-2370.
- Rungta, N. (2019). Provable Security at {AWS}. In *Enigma 2019* (Enigma 2019).