

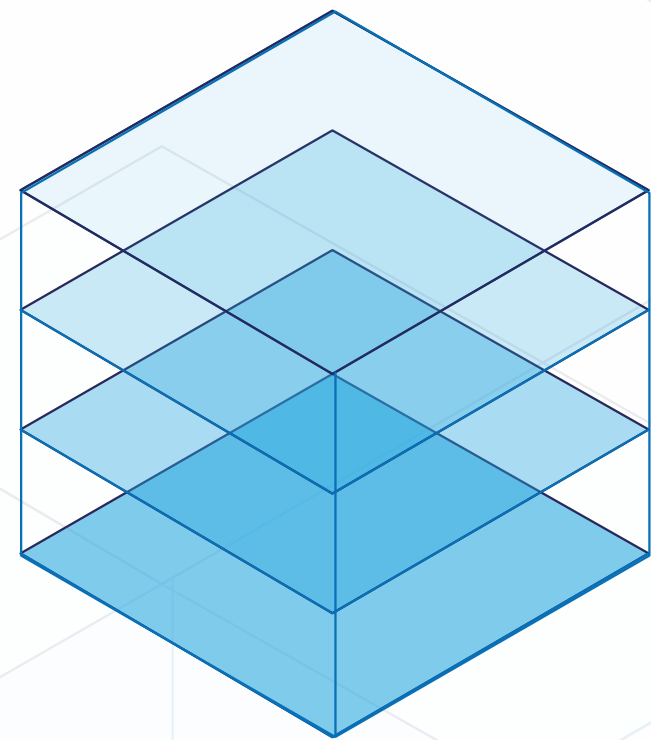
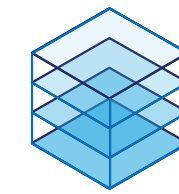
MISO

Maestría en Ingeniería de Software

Sustentación

Proyecto de grado 1

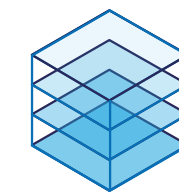
Grupo 7



MISO

Maestría en Ingeniería de Software

Modelamiento



Contexto

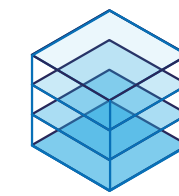
CCP es una empresa de comercialización de productos con operaciones en diferentes países del mundo. Su núcleo de negocio es la distribución y venta de productos como un tercero entre fabricantes y consumidores finales.

CCP manifestó unas metas de crecimiento del 3% anual por los próximos 3 años. Para ello, desean poder tener más recursos al minimizar los sobrecostos generados por reclamaciones e inconformidades que sus clientes generan debido a errores en la operación logística y por la pérdida de producto perecedero. Esta suma de dinero alcanza hasta los \$1'250.000 USD anuales

Entre los retos principales se encuentran:

- Adquisición de productos con baja rotación que no son atractivos para los clientes
- Desconocimiento del mercado de fabricantes y las tarifas
- Falta de herramientas para la gestión del inventario y promesa comerciales por parte del equipo de ventas
- No existe un método para la optimización de las rutas de entrega, aumentando el costo de los despachos y provocando despachos erróneos
- Pérdida de producto en las bodegas al no tener un sistema de inventario por posiciones

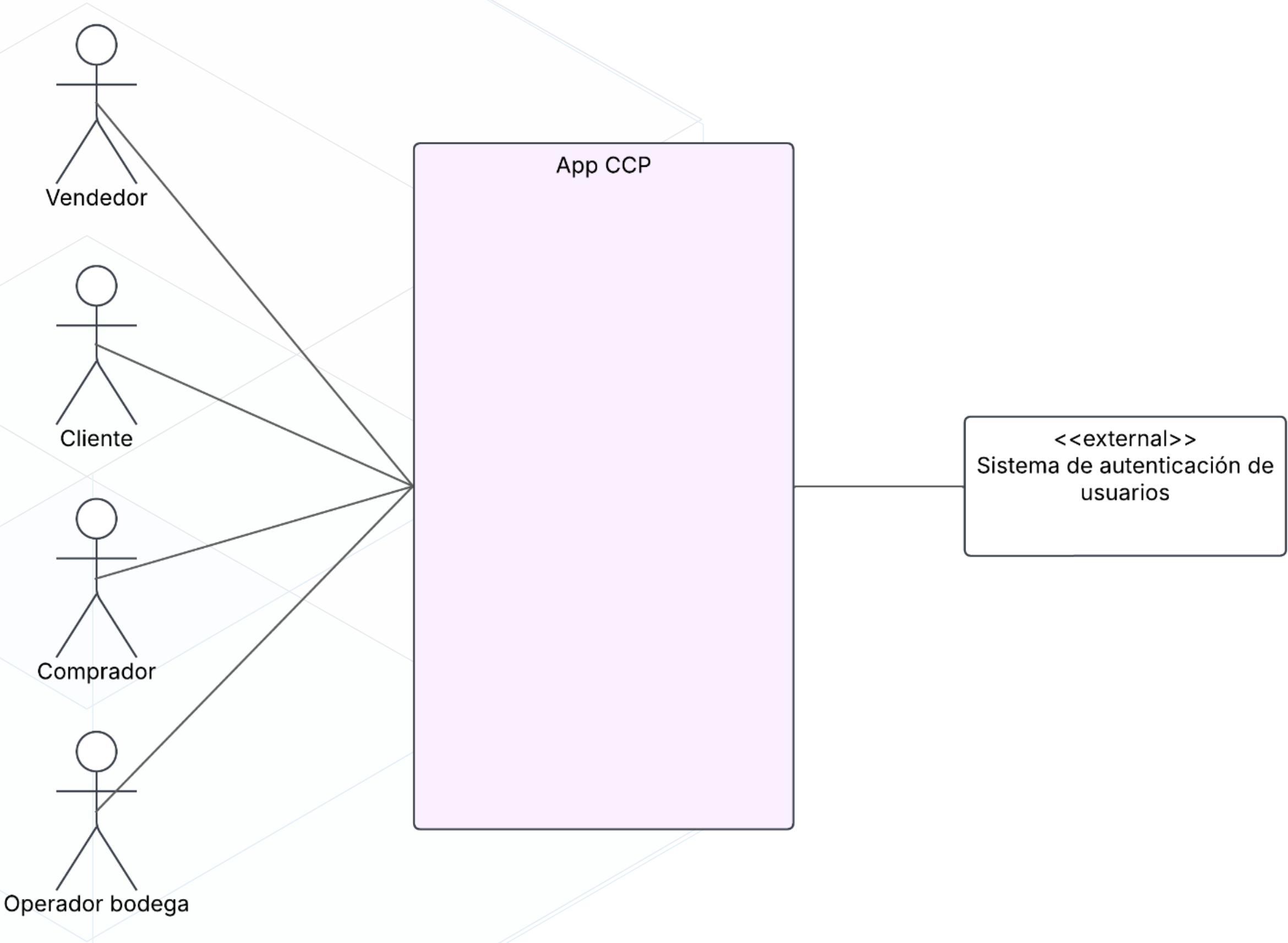
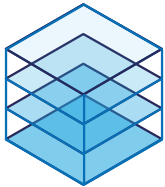
A partir de lo anterior, el equipo fue solicitado para desarrollar una propuesta de aplicación para suplir estas necesidades y permitir a CCP cumplir sus metas de crecimiento



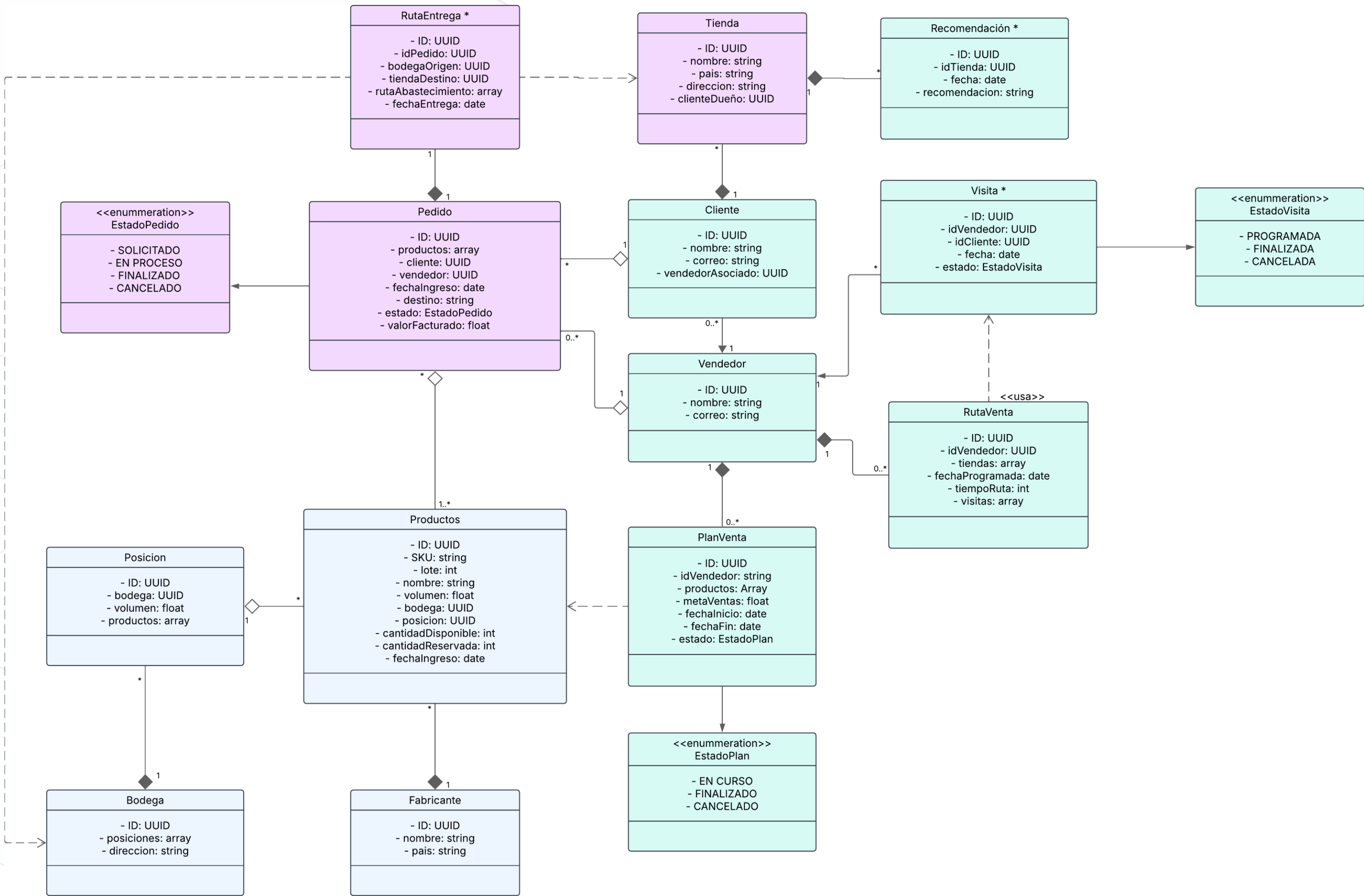
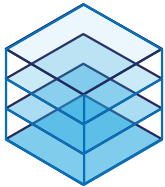
Requisitos y restricciones

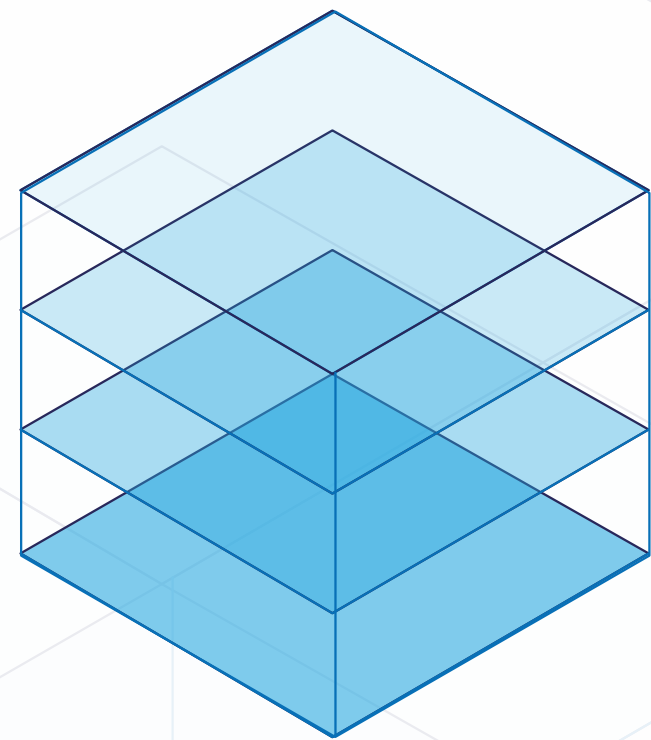
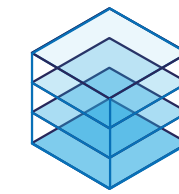
- **Tiempo de ejecución del proyecto:** 16 semanas a partir de la fecha de inicialización
- **Hitos de entrega:**
 - 8 semanas para el modelamiento de la arquitectura
 - 8 semanas para el desarrollo de la aplicación
 - Presupuesto: 4 desarrolladores senior
- **Entregables:**
 - **Aplicación web:** Orientada a la gestión del inventario y su stock disponible, creación de nuevos productos y fabricantes, al igual que para gestionar las rutas de entrega y los despachos generados. También permite la administración de la fuerza de ventas y sus planes de trabajo
 - **Aplicación móvil para la fuerza de ventas:** Contiene las funcionalidades para que la fuerza de ventas pueda crear pedidos a clientes, registrar visitas a clientes, gestionar cronogramas y administrar planes de venta. Finalmente, permite generar recomendaciones inteligentes a los clientes de CCP para mejorar la distribución de sus tiendas
 - **Aplicación móvil para clientes de CCP:** Permite el ingreso de pedidos de forma autónoma

Vista de contexto



Vista de dominio

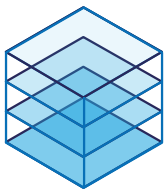




MISO





Maestría en Ingeniería de Software

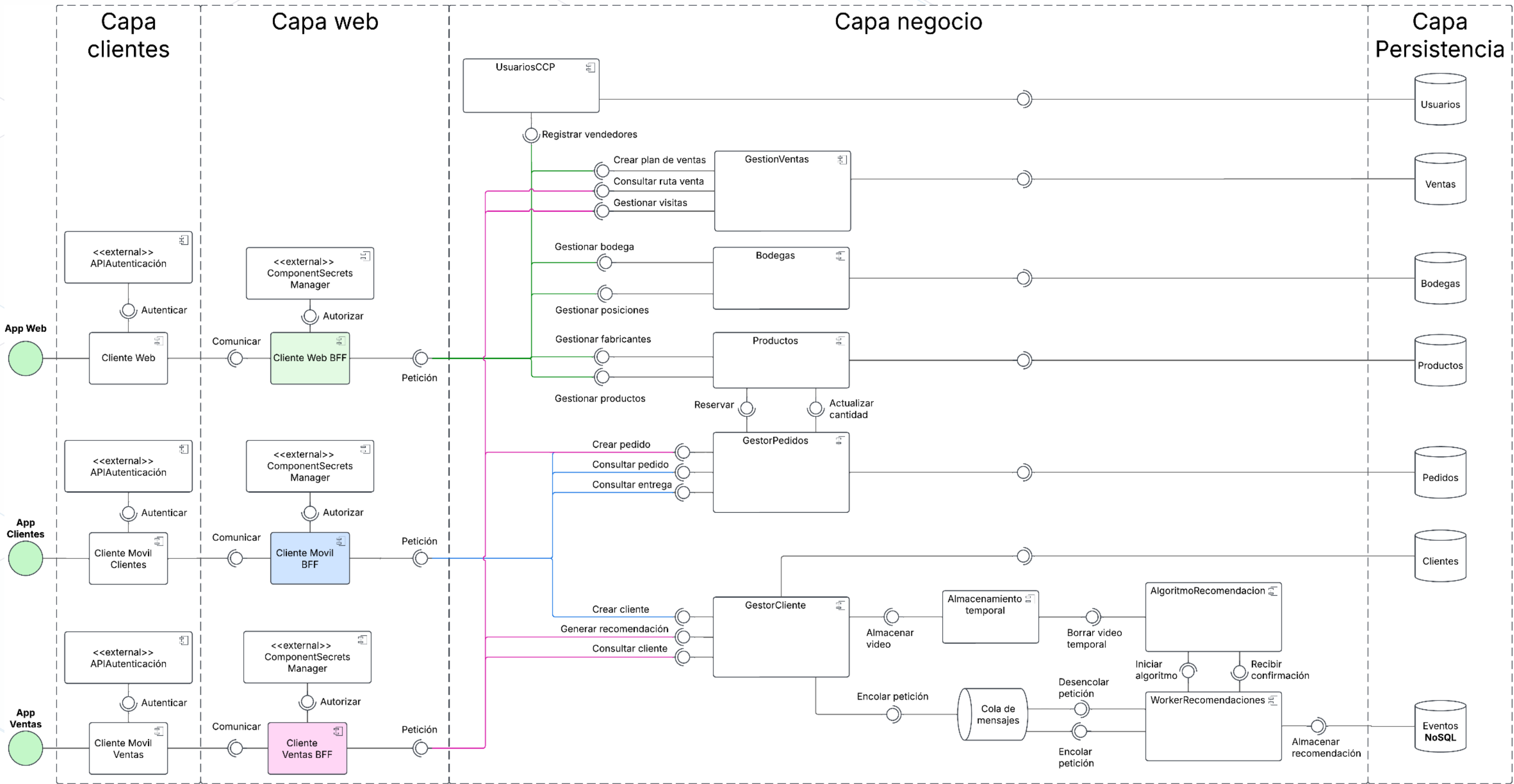
Propuesta de arquitectura

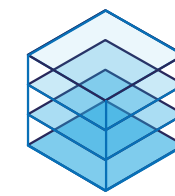


Vista Funcional – (Estilo Microservicios)

Proyecto	CCP Supply	ID	VC-001	Elaboración	Grupo 7	Versión	1.0
Vista	Funcional		Modelo	Componentes			

Convención
Estilo microservicios
 Interfaz
 Estímulo
 Componente
 Base de datos





Vista Funcional – (Estilo Microservicios)

Proyecto	CCP Supply	ID	VC-001	Elaboración	Grupo 7	Versión	1.0
Vista	Funcional		Modelo	Componentes			

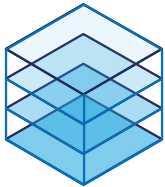
Patrones y tácticas de arquitectura utilizadas

- Microservicios
- Backend For Frontend (BFF)
- Autenticación
- Autorización
- Colas de mensajería

Razonamiento sobre las principales decisiones de arquitectura tomadas en este modelo

- Por medio del estilo de microservicios se espera incrementar la disponibilidad del sistema y tener lógicas de negocio más focalizadas en los componentes asociados, mejorando la facilidad de modificación
- El patrón BFF se escogió en contraposición al patrón de API Gateway para reducir el tamaño de los diferentes aplicativos a realizar. En este caso particular, reducirá la dependencia de un único componente, aumentará la disponibilidad y permitirá redistribuir adecuadamente las cargas
- Se introdujo un servicio externo de autenticación. Este permitirá mantener, por medio del estándar OAuth2, mantener la identidad de los usuarios, tener trazabilidad de sus actividades y delegar roles y permisos. Así mismo, este servicio reducirá la carga de código y es más seguro que un servicio in-house
- Se utilizará un servicio cloud para el almacenamiento y uso de secrets, el cual permitirá autorizar a los componentes para realizar sus debidas operaciones
- Para el manejo del procesamiento de videos para la generación de recomendaciones se implementará una cola de mensajería que permitirá activar un worker que realizará el proceso de forma asíncrona. De esta forma se reduce el tiempo de respuesta y al implementar la lógica “exactly once”, se garantizará el procesamiento de todos los videos

Vista de despliegue

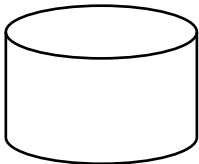


Proyecto	CCP Supply	ID	VC-001	Elaboración	Grupo 7	Versión	1.0
Vista	Funcional		Modelo	Despliegue			

Convención



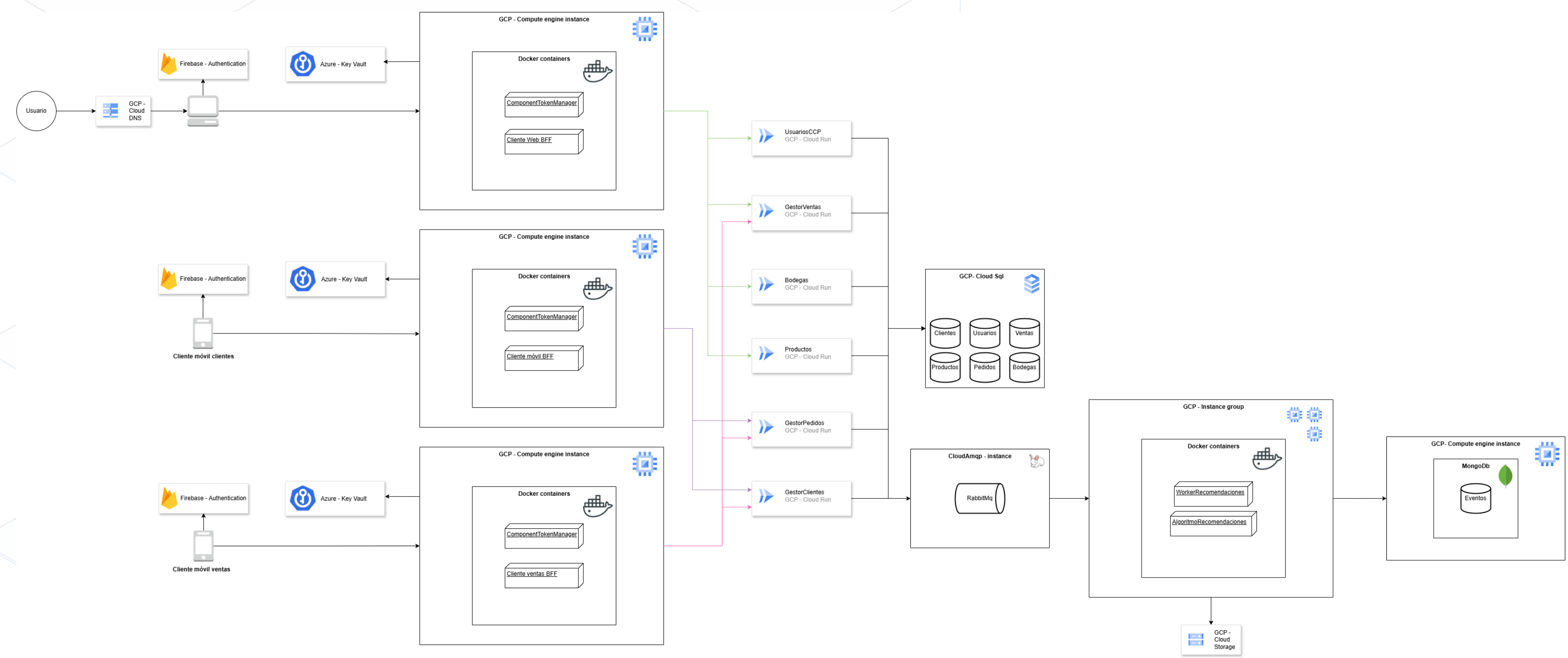
Nodos de ejecución

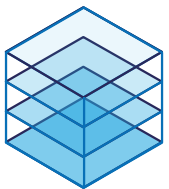


Base de datos



Cola de mensajes.





Vista de despliegue

Proyecto	CCP Supply	ID	VC-001	Elaboración	Grupo 7	Versión	1.0
Vista	Funcional		Modelo	Despliegue			

Patrones y tácticas de arquitectura utilizadas

- Microservicios.
- Cliente-servidor

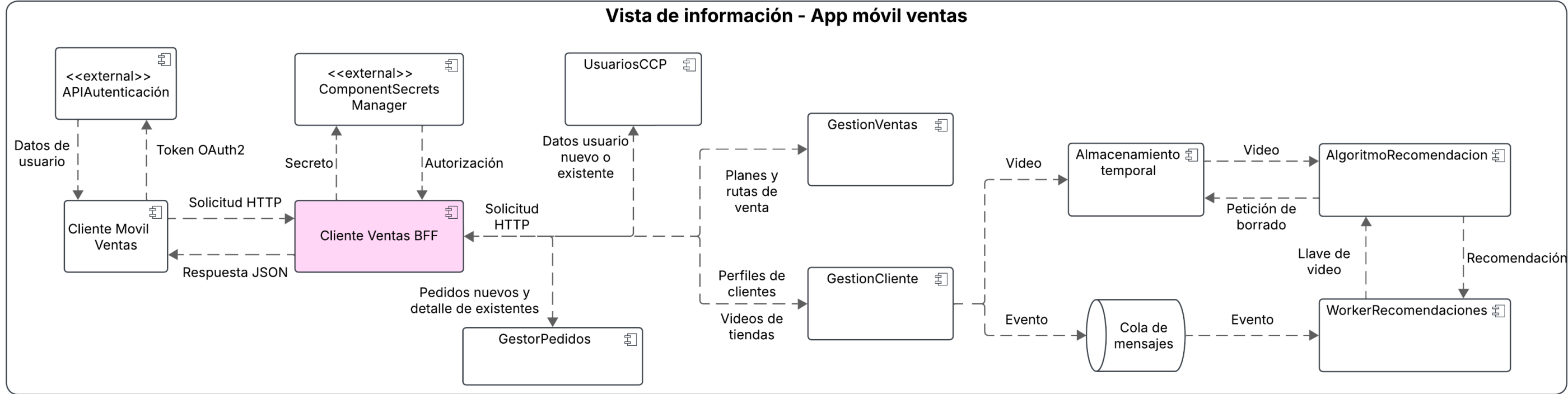
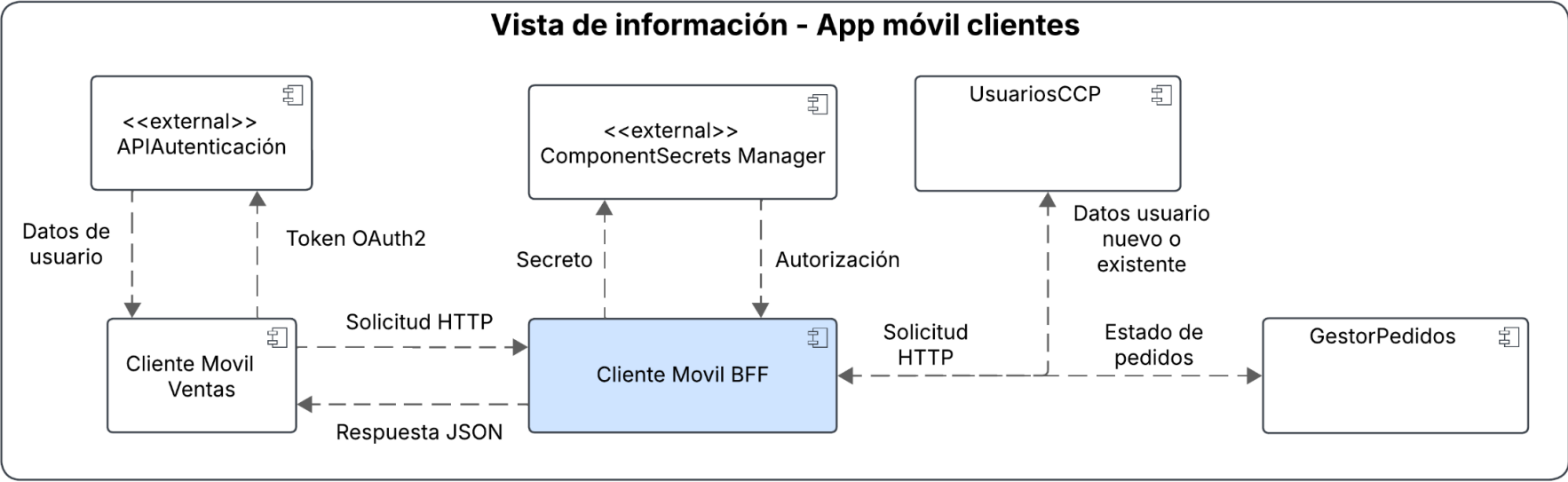
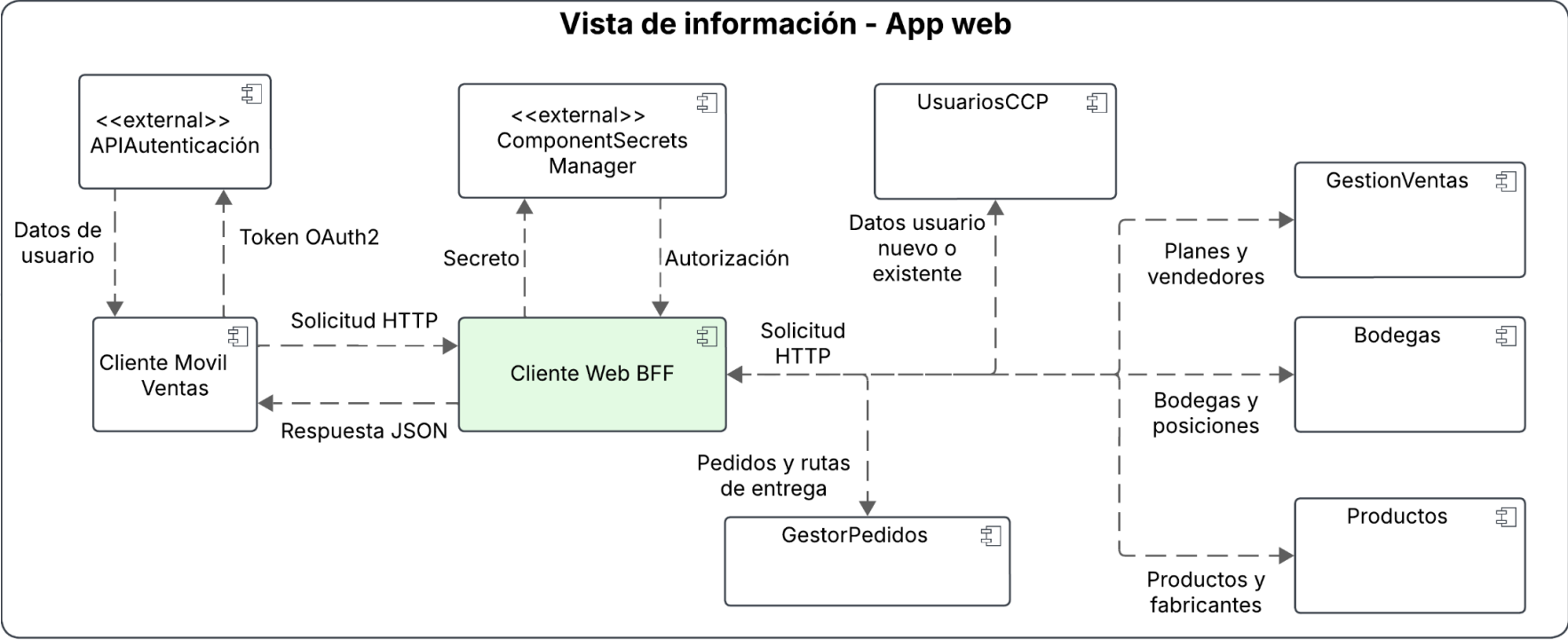
Razonamiento sobre las principales decisiones de arquitectura tomadas en este modelo

- El despliegue de la solución se hará principalmente utilizando la nube y los servicios de GCP.
- Para el proceso de autenticación de cada área del proyecto se utilizarán los servicios de Firebase authenticator, esto nos ayudara principalmente a acelerar el proceso de desarrollo y gestionar una misma herramienta para la parte web y móvil.
- Los diferentes servicios serán desplegados en un compute engine y cada uno en su propio container. Con esto en mente, se debe tener en cuenta la configuración de la VPC para la interconexión de todos los servicios (compute engines, bases de datos, sistema de encolamiento, entre otros.).
- Debido a la complejidad del sistema de recomendaciones para los clientes (basándonos en videos tomados por los vendedores), se utilizará una base de datos No relacional con el fin de tener mayor flexibilidad en el proceso de desarrollo del modelo y los objetos a persistir en una base de datos.



Vista de Información

Proyecto	CCP Supply	ID	VC-001	Elaboración	Grupo 7	Versión	1.0	Convención
Vista	Información		Modelo	Información				



Componente



Vista de Información

Proyecto	SportApp	ID	VC-004	Elaboración		Versión	1.1
Vista	Información		Modelo	Información			

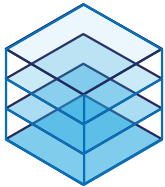
Patrones y tácticas de arquitectura utilizadas

- Backend For Frontend (BFF)
- Colas de eventos

Razonamiento sobre las principales decisiones de arquitectura tomadas en este modelo

- Para el manejo del procesamiento de videos para la generación de recomendaciones se implementará una cola de mensajería que permitirá activar un worker que realizará el proceso de forma asíncrona. Esto, en conjunto con una lógica exactly once, muestra que el sistema será resiliente a fallos ya que podrá procesar más adelante los videos cargados cuándo el sistema esté disponible, y tener un espacio eficiente al eliminarlos del almacenamiento temporal enlazado a él
- El patrón BFF muestra en este caso como el sistema distribuido se parte en partes más pequeñas. Por tal razón, cada uno deberá manejar menos tipos de datos, aumentando significativamente la facilidad de modificación

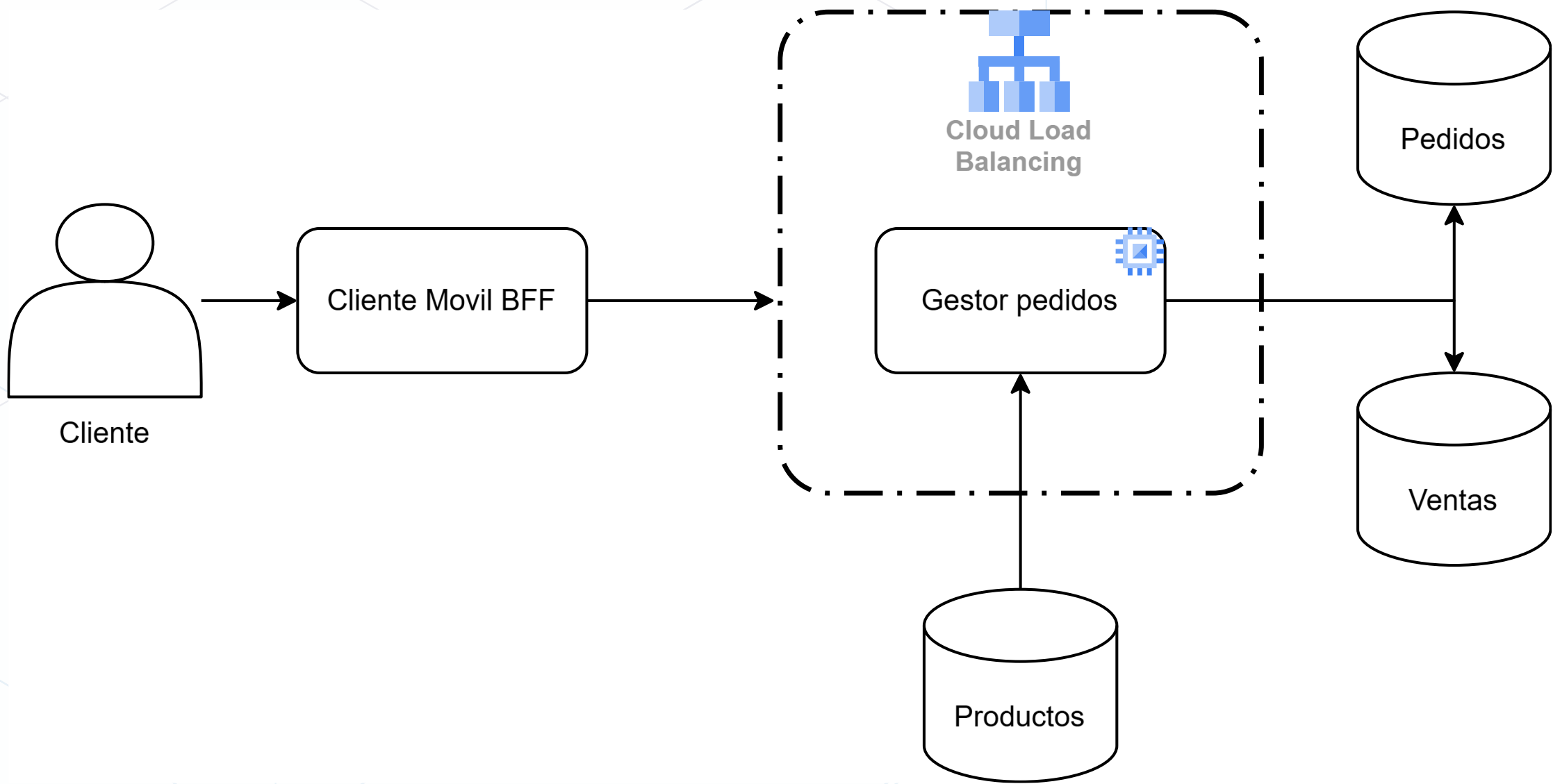
Vista de Concurrency

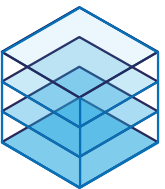


Proyecto	CCP Supply	ID	VC-001	Elaboración	Grupo 7	Versión	1.0
Vista	Concurrency		Modelo	Concurrency			

Convención
<div><div></div><div>Componente</div></div> <div><div></div><div>Base de datos</div></div>

Vista de concurrencia de la operación de escalamiento para la creación de pedidos





Vista de Información

Proyecto	SportApp	ID	VC-004	Elaboración		Versión	1.1
Vista	Concurrencia		Modelo				

Patrones y tácticas de arquitectura utilizadas

- Backend For Frontend (BFF)
- Load balancer

Razonamiento sobre las principales decisiones de arquitectura tomadas en este modelo

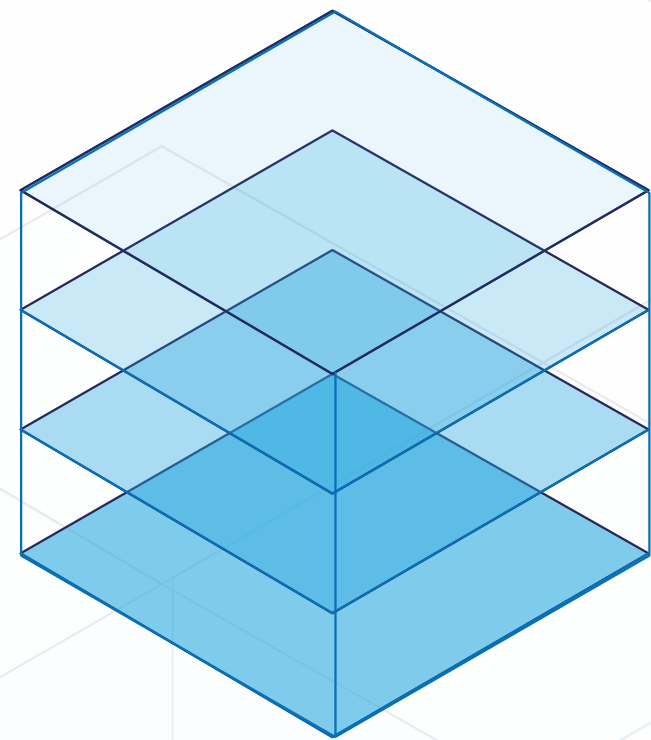
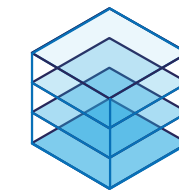
- Como uno de los requisitos de calidad principales en nuestra arquitectura es la escalabilidad, y dado que vamos a hacer un experimento relacionado a ello. Desarrollamos a un alto nivel el segmento del diagrama completo de arquitectura para el componente de realización de pedidos. Esto con el fin de adelantar los experimentos de arquitectura, documentar la hipótesis y probarla para este ciclo.
- El principal componente que debe escalar y al cual se evaluara su escalamiento es el Gestor de pedidos.
- Se busca probar el patrón BFF en un ambiente de alta concurrencia, evaluar cómo se comporta el balanceador de cargar y el BFF y evaluar los límites del BFF en este tipo de condiciones.
- El backend de la aplicación independientemente de ser móvil o web estará alojada en la nube. En este caso las peticiones serán hechas directamente a la nube.

Estrategia de pruebas

Objetivos:

1. Garantizar la estabilidad de los componentes individuales a través de pruebas unitarias, alcanzando una cobertura mínima de pruebas del 75%.
2. Identificar defectos y problemas de usabilidad mediante pruebas exploratorias, evaluando la aplicación desde una perspectiva mixta (caja blanca y caja negra).
3. Asegurar la integración efectiva de los módulos y flujos de trabajo por medio de pruebas E2E manuales y automatizadas, verificando la coherencia y comunicación entre los diferentes componentes del sistema.
4. Detectar errores en la traducción de palabras y segmentos de texto por medio de pruebas unitarias y pruebas semiautomatizadas de GUI Ripping
5. Garantizar que la aplicación sea accesible para todos los usuarios mediante la realización de pruebas de accesibilidad. Esto incluye evaluar la conformidad con las pautas WCAG (Web Content Accessibility Guidelines) y verificar la funcionalidad con tecnologías de asistencia.
6. Validar la respuesta entre componentes que requieren una comunicación síncrona por medio de pruebas de integración, con el fin de detectar respuestas o interacciones no esperadas

Nivel	Tipo	Técnica	Objetivo
Unitarias	Caja blanca	Unit testing	1, 5
Integración	Caja gris	Integration testing	6
Sistema	Caja negra	Pruebas E2E (APIs)	3
Sistema	Mixtas	Pruebas manuales exploratorias	2, 5
Sistema	Caja negra	GUI Ripping	4

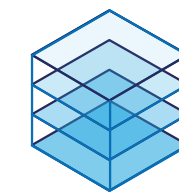


MISO

Maestría en Ingeniería de Software

Experimentación

Titulo del experimento	#1 – Efectos de las tácticas de seguridad en los requisitos de latencia de la plataforma
Propósito del experimento	Determinar cómo afecta la latencia la implementación de las tácticas de seguridad propuestas durante la creación de fabricantes en caso de alta demanda.
Resultados esperados	La creación de un fabricante ocurre: <ul style="list-style-type: none"> - Escenario sin táctica de seguridad en tiempo menor a 2 segundos. - Escenario con táctica de seguridad en tiempo mayor a 3 segundos. - Escenario con táctica de seguridad y manejo de latencia en tiempo menor a 3 segundos.
Recursos requeridos	Python, Javascript, Flask, 1 instancia de GCP App Service, Locust, 1 app en Firebase
Elementos de arquitectura involucrados	ASR #EC001 Componentes: Productos Vistas: Funcional, información y concurrencia Puntos de sensibilidad: <ul style="list-style-type: none"> • Las capas de seguridad agregadas afectan de forma importante la latencia del sistema. • Combinar tácticas de seguridad y de latencia permiten llevar a cabo la creación en menos de 3 segundos.
Esfuerzo estimado	Se estiman 22 horas hombre para la configuración de la librería, el ambiente de pruebas y la ejecución de las pruebas



Experimento #1

Pruebas con mecanismos de seguridad y reducción de latencia

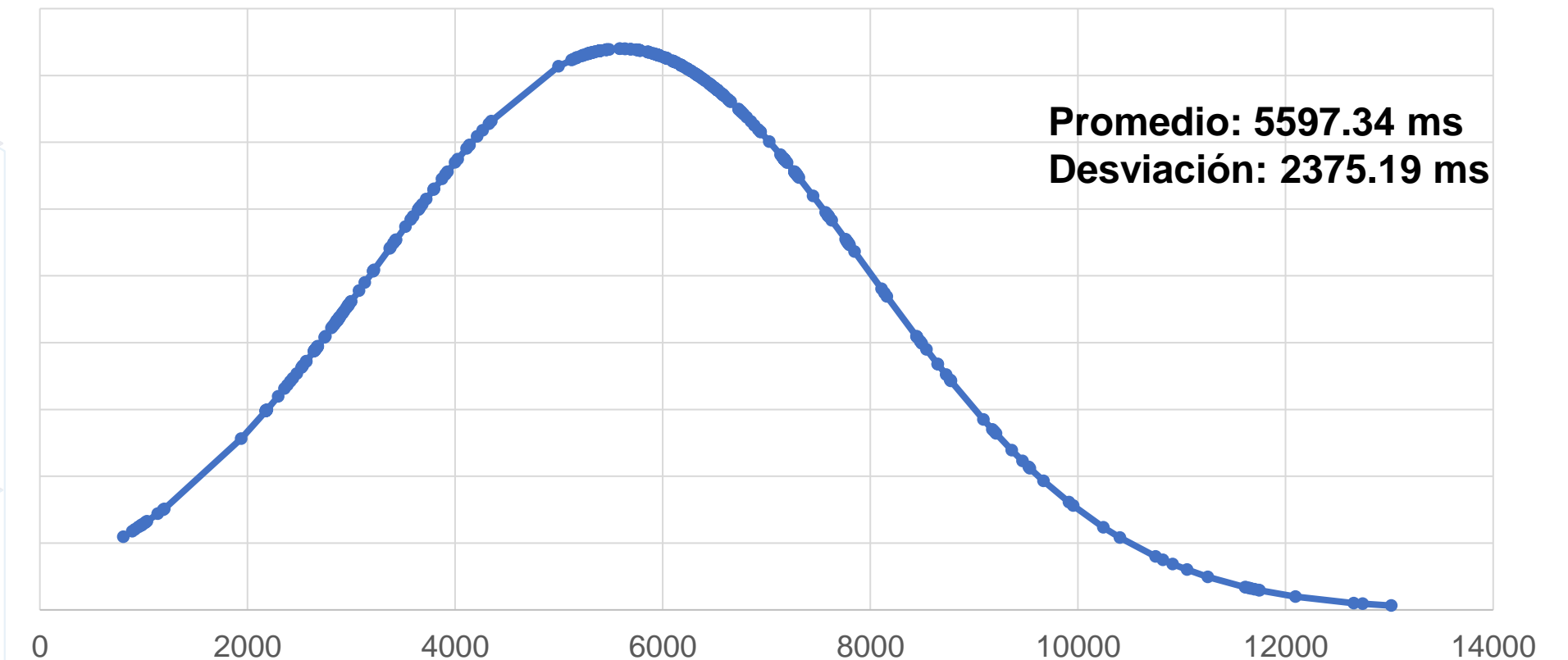
Para la prueba con mecanismos de seguridad se implementó un frontend construido en Next.JS y desplegado en cloud para simular el ambiente real en el que se ejecutará el servicio. En este caso se utilizó una estrategia para almacenar en caché el token de autenticación y así evitar los llamados a Firebase.

Las pruebas se ejecutaron usando un script construido en Playwright para poder ejecutar un script en la página web que se encarga de hacer el llamado al API Gateway

- Los resultados muestran que el sistema tiene un tiempo de carga promedio de 5597ms. Teniendo en cuenta el experimento con Firebase, **podemos estimar el tiempo con autenticación en 6577ms**
- No se observaron fallos durante la creación, y la base de datos contiene todos los objetos, sin embargo, el tiempo de creación es, en promedio, 2597ms más a lo especificado en los requisitos de calidad

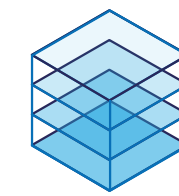
Estas prueba demuestra que la hipótesis de diseño no se cumple, y que es necesario realizar un reajuste en este proceso, y todos aquellos que tengan una alta concurrencia.

Distribución de tiempo de ejecución - Seguridad con caché



Total Users: 400
Successful Requests: 400
Failed Requests: 0
Avg Load Time: 5597.34ms
Min Load Time: 803ms
Max Load Time: 13019ms

Titulo del experimento	#2 - Tiempo de escalamiento y estabilización del sistema en una situación de alta demanda de solicitud de pedidos
Propósito del experimento	Identificar, bajo los patrones de seguridad utilizados, si es posible conseguir un tiempo de escalamiento inferior a los 3 minutos requeridos, cuando el sistema supera un límite de solicitudes concurrentes.
Resultados esperados	Durante el escalamiento: <ul style="list-style-type: none">• El tiempo completo de replicación del componente de pedidos es menor a 3 minutos• Durante la puesta a punto, no se pierden solicitudes de creación de pedido
Recursos requeridos	Python, Javascript, Flask, 1 instancia de GCP App Service, Locust, 1 app en Firebase, servicio de base de datos
Elementos de arquitectura involucrados	ASR #EC003 Componentes: Pedidos Vistas: Funcional, información y concurrencia Puntos de sensibilidad: <ul style="list-style-type: none">• Las tácticas de seguridad reducen la capacidad del sistema para procesar solicitudes significativamente en una situación de alta demanda• Un límite máximo de 80% de uso de recursos (CPU y/o RAM) es adecuado para replicar el componente antes de perder solicitudes
Esfuerzo estimado	Se estiman 15 horas hombre para la configuración de la librería, el ambiente de pruebas y la ejecución de las pruebas



Experimento #2

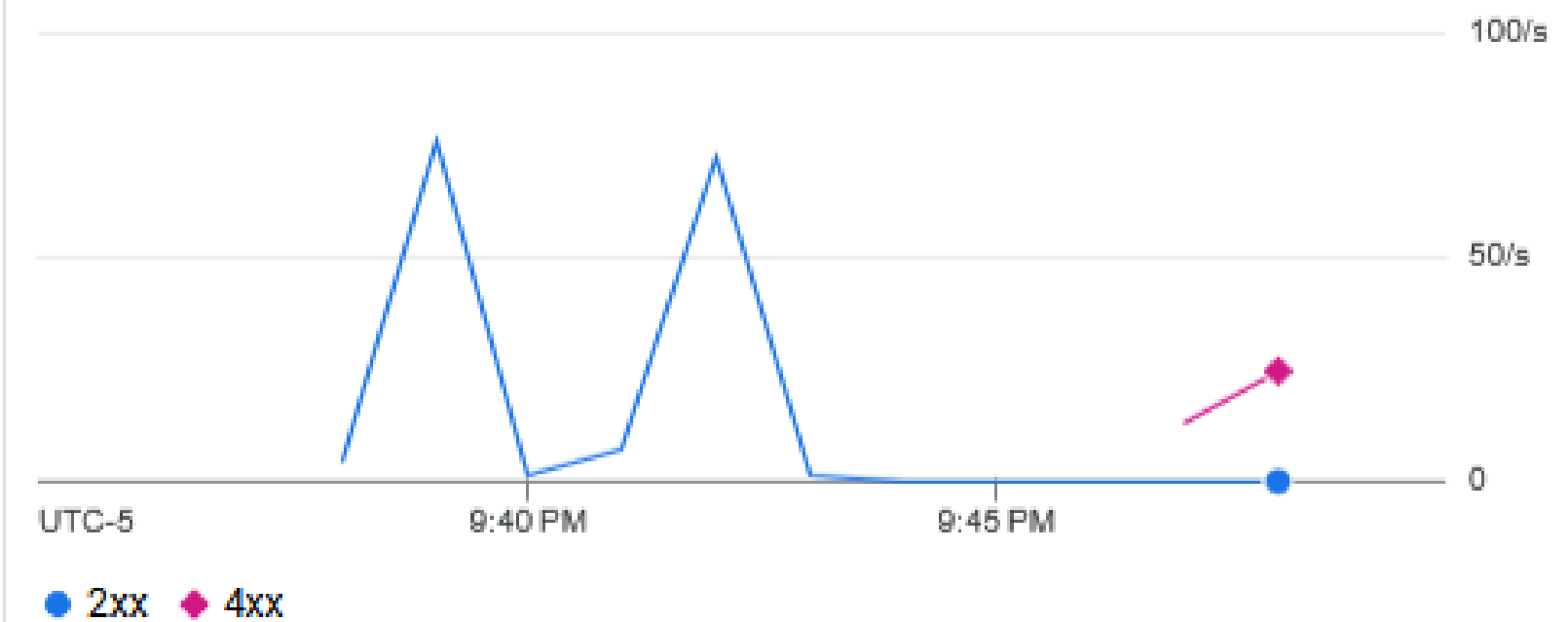
Pruebas de tiempo de escalamiento

- Observando el escalamiento del componente web, se observa un valle entre las 9:44 PM y las 9:46 PM. Este espacio muestra el tiempo de inactividad de los contenedores de Cloud Run.
- Por otra parte, entre las 9:46 PM y las 9:47 PM, el sistema escala hasta 2 instancias del contenedor, demostrando que el sistema puede escalar en un tiempo de alrededor de 1 minuto
- Entre 9:39 PM y 9:43 PM se observa la última ejecución del experimento 1, demostrando la capacidad de escalado del sistema

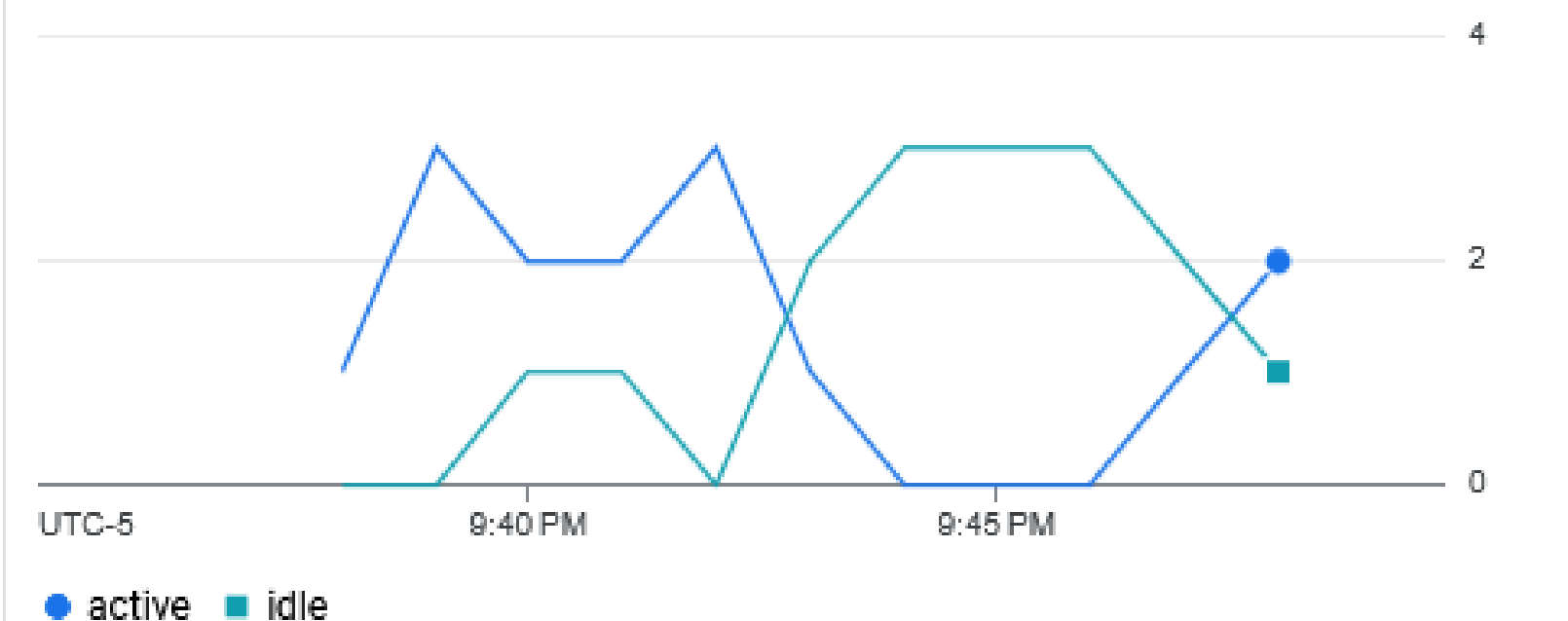
En el caso del experimento, se demostró que el tiempo de escalamiento, incluso con un backend complejo, se pudo hacer en un periodo de tiempo corto, en especial con herramientas como Cloud Run de GCP. Esto permite demostrar que se cumplió la hipótesis de diseño.

Sin embargo, **el tiempo elevado de respuesta demuestra que es necesario implementar un sistema para la gestión de las peticiones.** El tiempo actual de respuesta en un ambiente síncrono es completamente inaceptable para un usuario

Request count ?



Container instance count ?



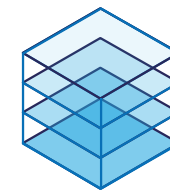
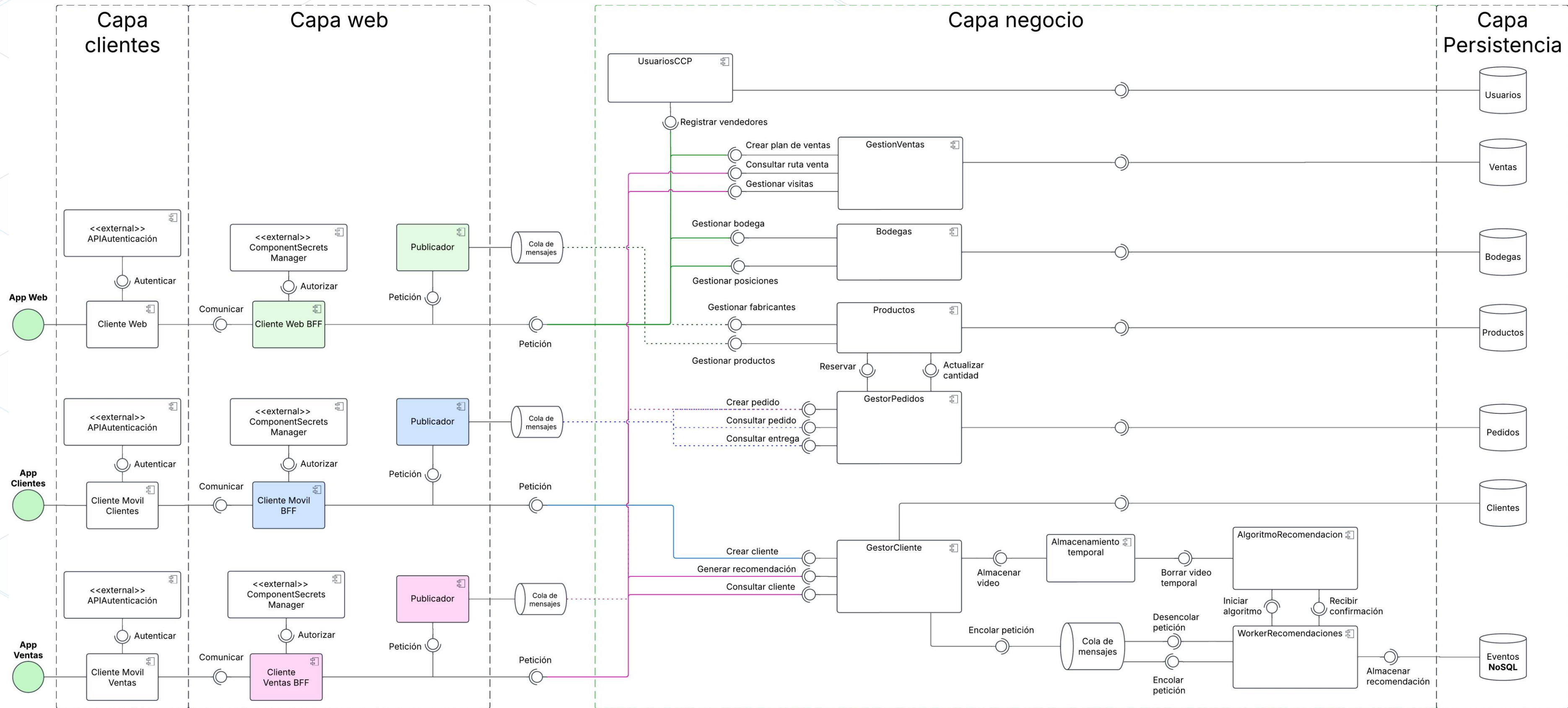


Diagrama refinado - Componentes

A partir de los resultados obtenidos se plantea el siguiente diagrama de componentes como una respuesta a la baja latencia del sistema, se agrega manejo de eventos para las peticiones que llegan a los componentes de Productos y GestorPedidos, considerados como cruciales en la arquitectura



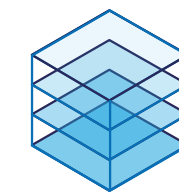
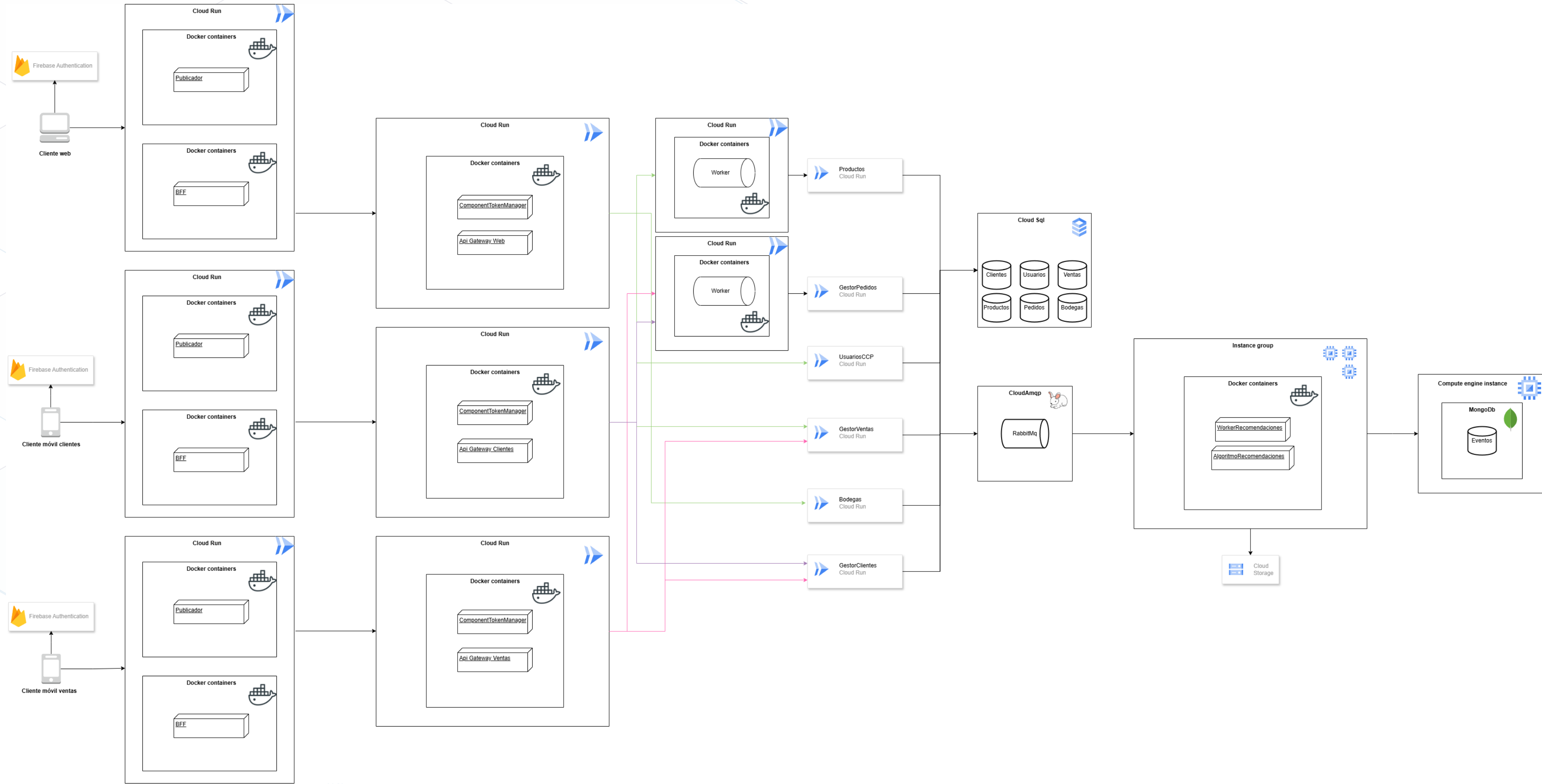
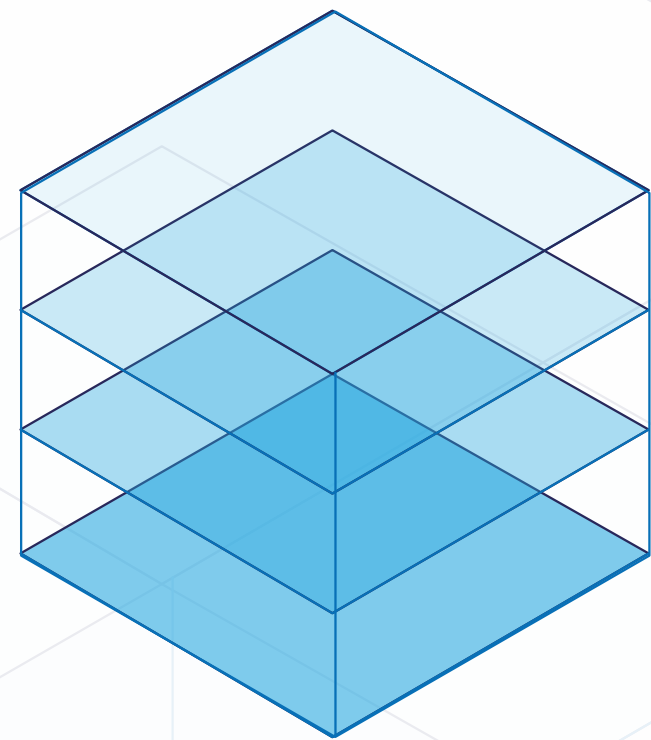
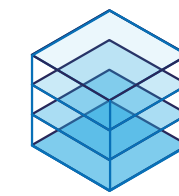


Diagrama refinado - Despliegue

A partir de los resultados obtenidos se plantea el siguiente diagrama despliegue como una respuesta a la baja latencia del sistema. Diagrama con mejor resolución en este [link](#). El refinamiento de este diagrama incluye el cambio en el uso de máquinas virtuales a instancias de Cloud Run, un publicador para cada cliente que se ejecuta en un docker container junto al BBF y dos brokers de eventos de los cuales consumirán productos y gestor pedidos.

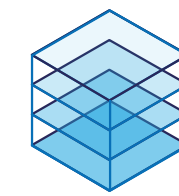




MISO

Maestría en Ingeniería de Software

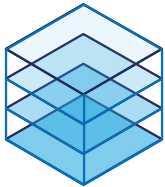
Diseño UI



Enlaces a prototipos interactivos UI

- [Prototipo web](#)
- [Prototipo móvil ventas](#)
- [Prototipo móvil](#)

Accesibilidad diseño UI



CPP

Perfil

Pedidos

Mis Planes

Buscar por ID

Filtrar

+ Agregar Plan de Venta

Generar Reporte

ID Plan de Ventas	Estado	Fecha Inicio	Fecha Fin	Meta de Ventas	Productos
123456	Finalizado	01/01/2025	01/02/2025	\$ 500.000	Producto 1, Producto 2, Producto 3
123456	Finalizado	01/01/2025	01/02/2025	\$ 500.000	Producto 1, Producto 2, Producto 3
123456	Finalizado	01/01/2025	01/02/2025	\$ 500.000	Producto 1, Producto 2, Producto 3
123456	Finalizado	01/01/2025	01/02/2025	\$ 500.000	Producto 1, Producto 2, Producto 3
123456	Finalizado	01/01/2025	01/02/2025	\$ 500.000	Producto 1, Producto 2, Producto 3
123456	Finalizado	01/01/2025	01/02/2025	\$ 500.000	Producto 1, Producto 2, Producto 3
123456	Finalizado	01/01/2025	01/02/2025	\$ 500.000	Producto 1, Producto 2, Producto 3
123456	Finalizado	01/01/2025	01/02/2025	\$ 500.000	Producto 1, Producto 2, Producto 3
123456	Finalizado	01/01/2025	01/02/2025	\$ 500.000	Producto 1, Producto 2, Producto 3

1 - 10 de 13 páginas

Esta página 1

Copyright © 2025. All rights reserved.

Español (Latinoamérica)

MISO
Maestría en Ingeniería de Software

CPP

Productos

Pedidos

Productos

Busca por SKU, lote, nombre, etc.

Q

Filtrar

+ Ingresar Lotes

Nombre Producto	SKU	Lote	Volumen	Bodega	Posición	Cantidad Disponible	Cantidad Reservada	Fecha Ingreso
ID 12345 Producto 1	ABC-12345	1	1000	550e8400-e29b-41d4	a716-446655440000	500	500	01/01/01
ID 12345 Producto 1	ABC-12345	1	1000	550e8400-e29b-41d4	a716-446655440000	500	500	01/01/01
ID 12345 Producto 1	ABC-12345	1	1000	550e8400-e29b-41d4	a716-446655440000	500	500	01/01/01
ID 12345 Producto 1	ABC-12345	1	1000	550e8400-e29b-41d4	a716-446655440000	500	500	01/01/01
ID 12345 Producto 1	ABC-12345	1	1000	550e8400-e29b-41d4	a716-446655440000	500	500	01/01/01
ID 12345 Producto 1	ABC-12345	1	1000	550e8400-e29b-41d4	a716-446655440000	500	500	01/01/01
ID 12345 Producto 1	ABC-12345	1	1000	550e8400-e29b-41d4	a716-446655440000	500	500	01/01/01
ID 12345 Producto 1	ABC-12345	1	1000	550e8400-e29b-41d4	a716-446655440000	500	500	01/01/01
ID 12345 Producto 1	ABC-12345	1	1000	550e8400-e29b-41d4	a716-446655440000	500	500	01/01/01

1 - 10 de 13 páginas

Esta página

1

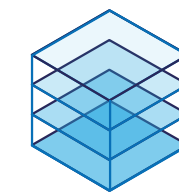
<

>

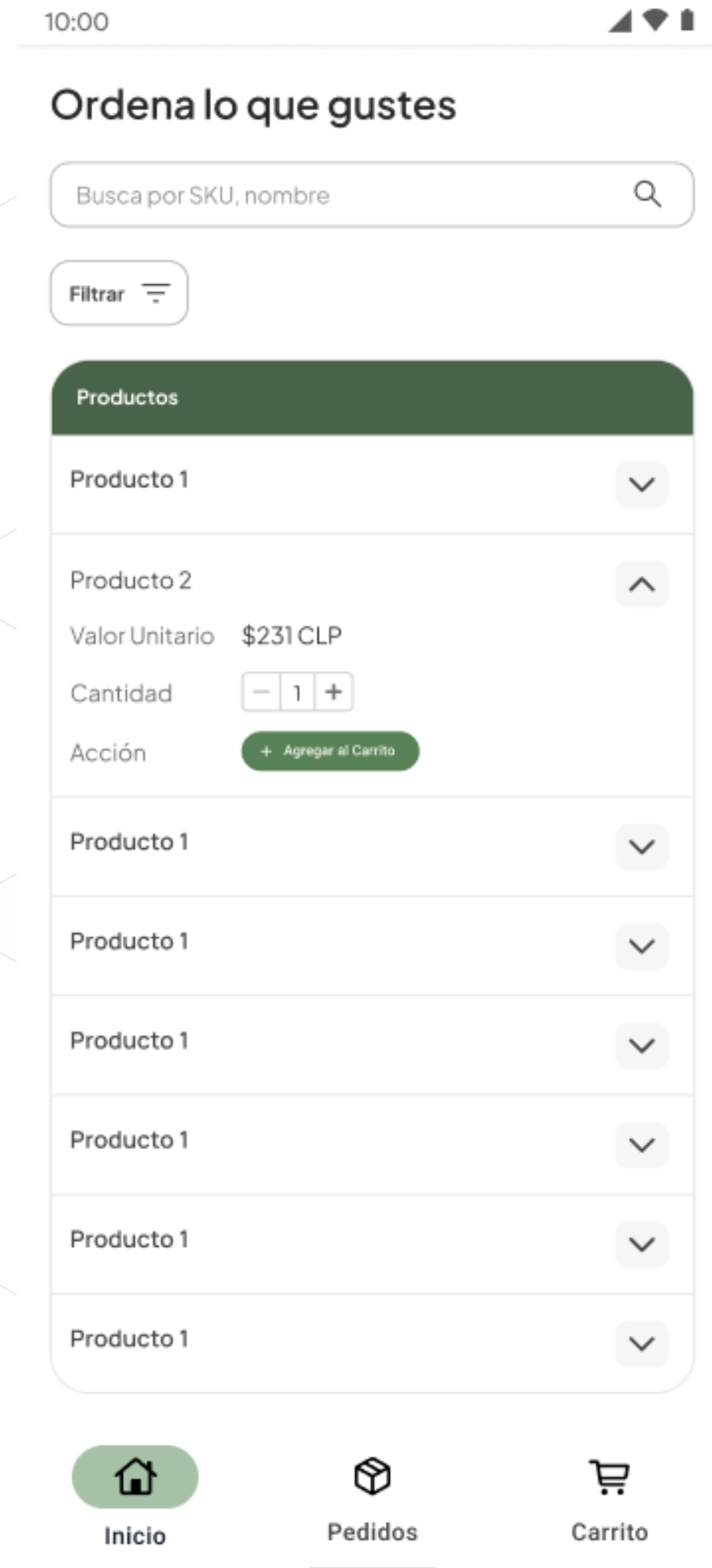
Copyright © 2025. All rights reserved.

Español (Latinoamérica)

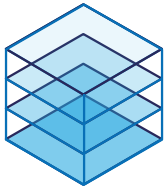
Localización diseño UI



MISO
Maestría en Ingeniería de Software



Internacionalización diseño UI



CPP

Sellers

Sellers

Filter

+ Add Seller

Name Seller	E-mail
ID 54321 Seller 1	seller_1@gmail.com
ID 54321 Seller 1	seller_1@gmail.com
ID 54321 Seller 1	seller_1@gmail.com
ID 54321 Seller 1	seller_1@gmail.com
ID 54321 Seller 1	seller_1@gmail.com
ID 54321 Seller 1	seller_1@gmail.com
ID 54321 Seller 1	seller_1@gmail.com
ID 54321 Seller 1	seller_1@gmail.com
ID 54321 Seller 1	seller_1@gmail.com

1 - 10 of 13 pages

This page

1

<

>

Copyright © 2025. All rights reserved.

English

Internacionalización diseño UI

