

Diseño y construcción de soluciones no monolíticas

Diseño y arquitectura de dominio

Nombre del equipo: Grupo

Integrantes.

1. Harold Hernando Funeme Molano (h.funeme@uniandes.edu.co)
2. Óscar Fernando Álvarez (o.alvareze@uniandes.edu.co)
3. Felipe Leyva (d.leyvad@uniandes.edu.co)
4. Juan Pablo Feliciano Báez (jp.feliciano10@uniandes.edu.co)

Entrega 1:

1. Documentación de dominios y sub-dominios

El taller que se toma para realizar esta práctica considera que se identificaron las personas con mayor conocimiento del negocio y de la tecnología para realizar la primera reunión de reconocimiento.

Etapas 1: Generar diagrama de contexto general y entendimiento de funcionamiento:

Se pretende entender el funcionamiento de la empresa a nivel general, sus integraciones y los flujos de cada una de las acciones realizadas en cada tipo de entrega.



Diseño y construcción de soluciones no monolíticas

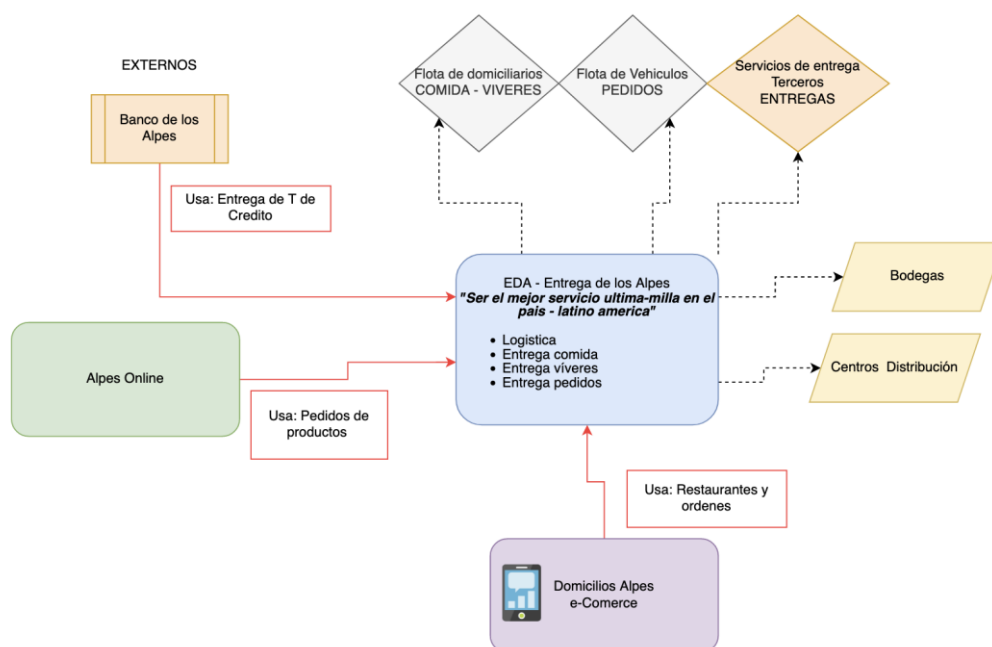
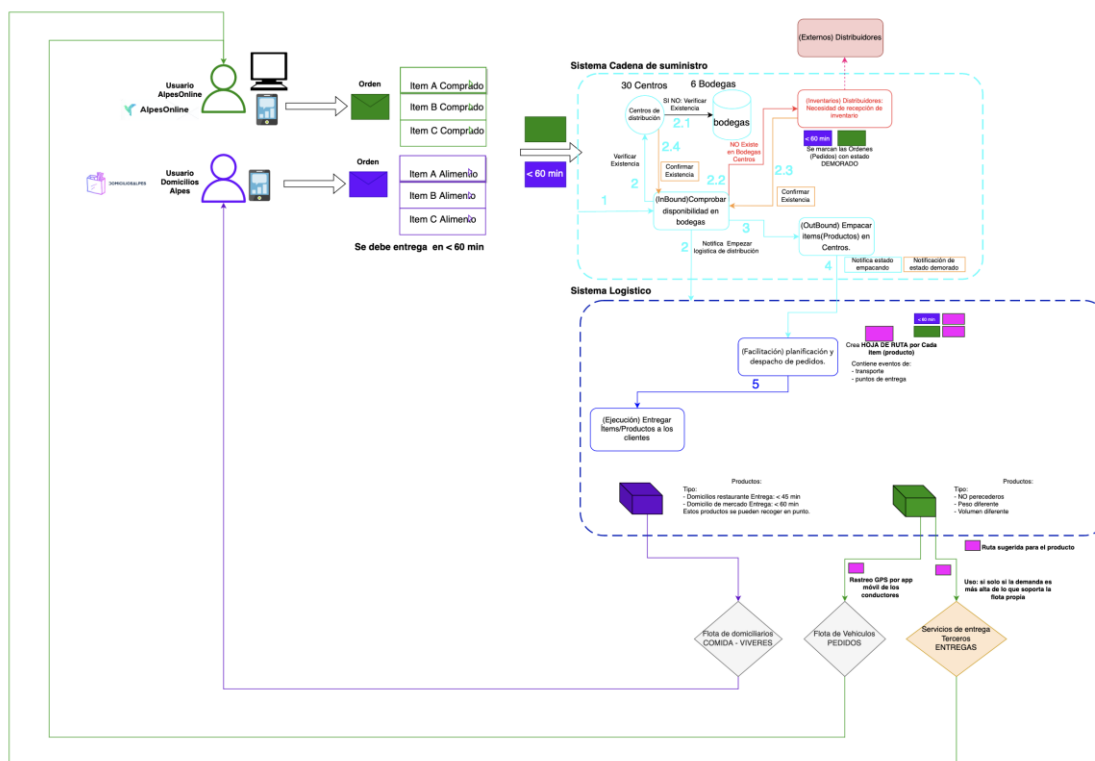


Imagen 1: Diagrama de contexto base generado por el equipo para el entendimiento del negocio

Como segundo recurso a generar se muestra el entendimiento extraído por las partes (Negocio y técnico) para los flujos generado por el negocio.



Diseño y construcción de soluciones no monolíticas



Con este entendimiento sincronizado en el equipo experto (Técnico - Negocio), se puede continuar.

Concluimos el Dominio: Realizar la logística y entrega de bienes (productos de mercado, perecederos y no perecederos) alrededor del país.

Etaa 2: Listado de conceptos

Concepto	Sinónimo	Definición
Orden	Pedido	solicitud de un usuario de comprar uno o más productos
Producto	ítem	objeto de venta. Hay dos tipos: Perecederos (Mercado y Restaurante) y No perecederos.
Centros de disponibilidad		30 centros localizados en todo el país con el fin de enviarlos al recorrido final
Bodegas		6 bodegas que almacenan inventario para ser enviados a los centros de distribución
Empacar producto para envío a Bodega o centro distribución		Se realiza en los centros de distribución y bodegas
Empacar producto para envío final		Se realiza en los proveedores, restaurantes y mercados
Hoja de ruta		Documento en que constan las mercancías que contienen los bultos cargados en un tren, camión u otro medio de transporte,



Diseño y construcción de soluciones no monolíticas

		sus puntos de destino, los nombres de los consignatarios, etc.
proveedor productos		Proveedores de productos inexistentes
Flota tractores y vehículos		grupo de vehículos que sirve para la entrega de pedidos
Flota de domiciliarios	servicio de entrega tercero	grupo de personas con vehículo pequeño que entregan pedidos de mercados o restaurantes
proveedor transporte		Servicios de entrega de terceros para suplir sobre demanda
Distribuidores suministros		Servicio de transporte que realiza la distribución de los productos hacia bodega o centros de distribución
Distribuidores finales		Servicio de transporte que realiza la distribución de los productos hacia el usuario final
pickup point		punto de entrega de productos en oficina de AlpesOnline
Supermercado - Restaurante	Vendedor	Tiendas que ofrecen productos perecederos

Finalmente generamos nuestros dominios y subdominios que se identificaron en la aplicación actual:

```
Domain RecepcionDomain {
    domainVisionStatement = "Recepción de órdenes y pedidos"
    Subdomain OrdenDomain {
        type = SUPPORTING_DOMAIN
        domainVisionStatement = "Subdominio para recepción de órdenes."
        Entity Orden {
            String idOrden
            Date fecha
            - List<Producto> productos
        }
        Entity TipoProducto {
            String tipoProducto
        }
        Entity Cliente {
            String id
            - TipoIdentificacion tipoIdentificacion
            String identificacion
            - Nombre nombres
        }
        Entity TipoIdentificacion {
            String tipoIdentificacion
        }
        Entity Nombre {
            String nombres
            String apellidos
        }
    }
}
```



Diseño y construcción de soluciones no monolíticas

```
Domain CadenaSuministroDomain {
    domainVisionStatement = "Cumplimiento de la orden"

    Subdomain InboundDomain {
        type = SUPPORTING_DOMAIN
        domainVisionStatement = "Subdominio en el que se gestina todo lo relacionado con las actividades de entrada."
    }

    Subdomain GestionInventarioDomain {
        type = SUPPORTING_DOMAIN
        domainVisionStatement = "Subdominio en el que se gestina todo lo relacionado con actividades de gestión del inventario."
        Entity Producto {
            String idProducto
            String descripcion
            - TipoProducto tipo
        }
        Entity Bodega {
            String idBodega
            String descripcion
            - Ciudad ciudad
            - TipoBodega tipoBodega
            - Coordinada coordenadas
            String direccion
            - List<Producto> existencias
        }
        Entity TipoBodega {
            String tipoBodega
        }
        Entity CentroDistribucion {
            String idCentroDistribucion
            - Ciudad ciudad
            - Coordinada coordenada
            String direccion
        }
    }
}
```

```
Domain LogisticaDomain {
    domainVisionStatement = "Cumplimiento de la orden."
    Subdomain FacilitacionDomain {
        type = CORE_DOMAIN
        domainVisionStatement = "Subdominio en el que se gestina todo lo relacionado con planificación, despacho de pedidos."
        Entity Transporte {
            - Orden orden
        }
        Entity HojaRuta {
            String idHojaRuta
            - List<Pedido> pedidos
            - List<PuntoEntrega> puntosEntrega
        }
        Entity PuntoEntrega {
            String idPuntoEntrega
            String nombre
            - Coordinada coordenada
            String direccion
        }
        Entity Coordinada {
            String latitud
            String longitud
        }
    }

    Subdomain EjecucionDomain {
        type = CORE_DOMAIN
        domainVisionStatement = "Subdominio en el que se gestina todo lo relacionado con gestión de última milla e entrega."
        Entity Entrega {
            String idEntrega
            - Orden orden
            - TipoEntrega tipoEntrega
        }
    }
}
```

2. Documentación del lenguaje ubicuo

Para realizar los dos modelos utilizamos las siguientes convenciones:



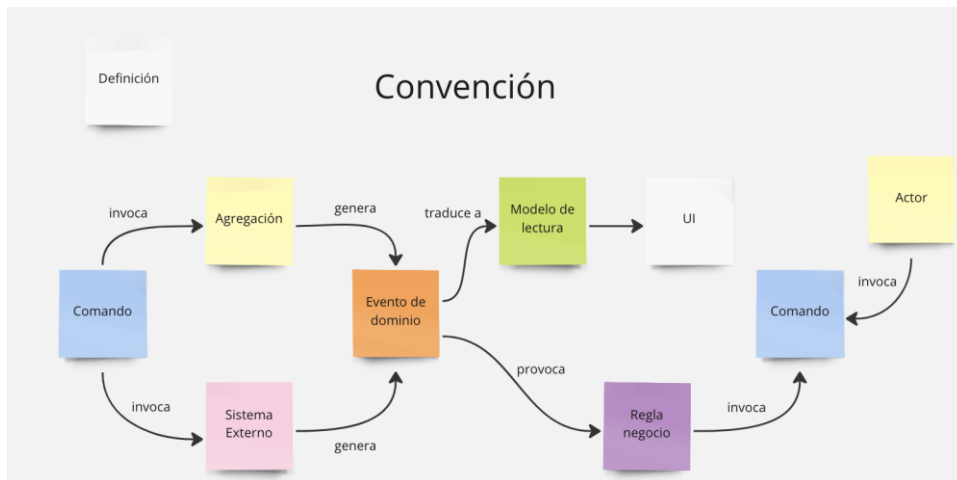
Diseño y construcción de soluciones no monolíticas

Convención usada

Tipo	Descripción	Color
Evento de dominio	Son ocurrencias de algo que pasó en el dominio. Este tópico se explicará con mayor detalle en el lecciones posteriores del curso. Por ahora, tenga presente que debe escribirlos con verbos en pretérito perfecto. i.e <code>Orden creada</code> , <code>Orden cancelada</code> , etc.	Evento de dominio
Comando	Representan intenciones/acciones/decisiones de un actor. Similar a los eventos, debe usarse verbos para representar las acciones, solo que éstos en tiempo presente. i.e <code>Crear nueva orden</code> , <code>Cancelar orden</code> , etc.	Comando
Actor	El quién o que inicia una acción	Actor
Definición	Definen términos de negocio. En cualquier caso donde se encuentre con un término específico del negocio, acrónimo o incluso del argot común, use este post-it	Definición
UI	Bosquejos de la UI para hacernos una idea del sistema. Aunque la estamos describiendo en esta sección, la verdad es que este post-it tiene más sentido en el formato nivel de diseño	UI
Reglas de negocio	Documentan condiciones y reglas de negocio. Normalmente se localizan entre eventos y siguen la sintaxis Gherkin: <code>Dado <evento en presente> cuando <condición> entonces <evento en presente></code>	Regla negocio
Modelo de lectura	Los datos que debemos proveer para un paso en nuestro workflow	Modelo de lectura
Sistema externo	Organizaciones, servicios o aplicaciones externas o de terceros	Sistema Externo
Agregación	Son agrupamiento de objetos asociados que trataremos como unidad para el propósito de cambios. Por ahora, no vamos a entrar en detalle acerca de ellas.	Agregación

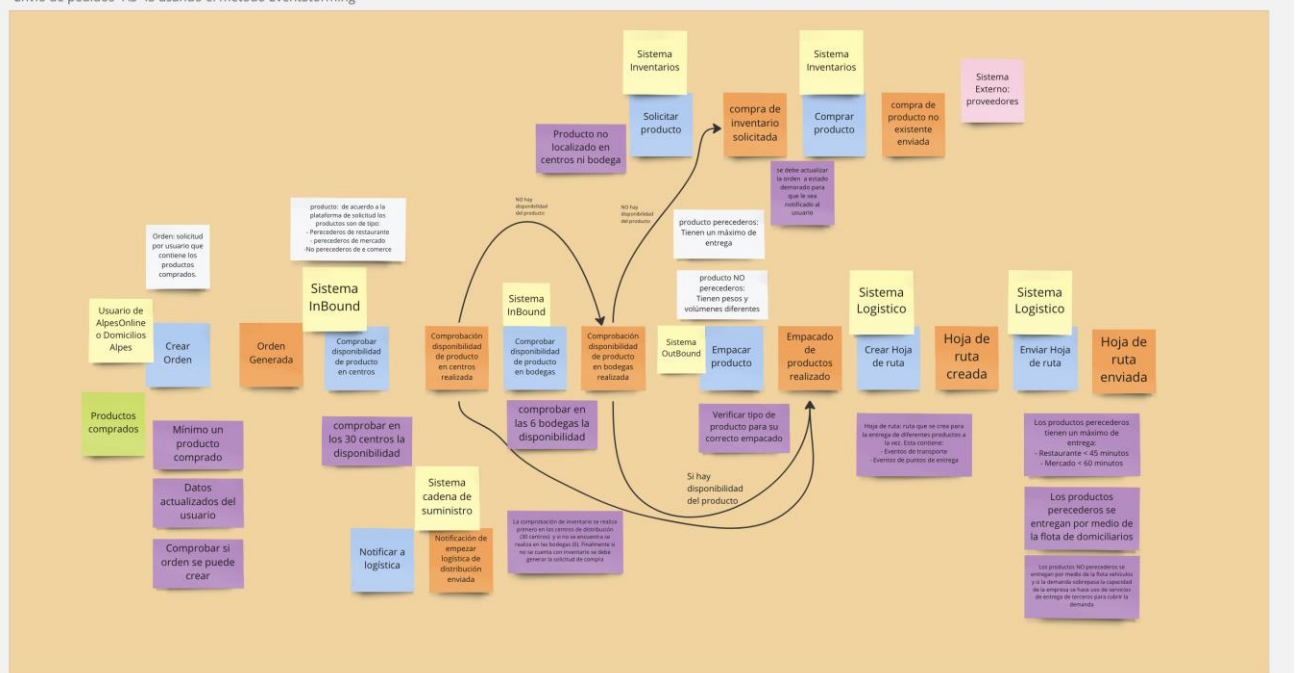


Diseño y construcción de soluciones no monolíticas



Modelo AS-IS

"Envío de pedidos" AS- IS usando el método EventStorming

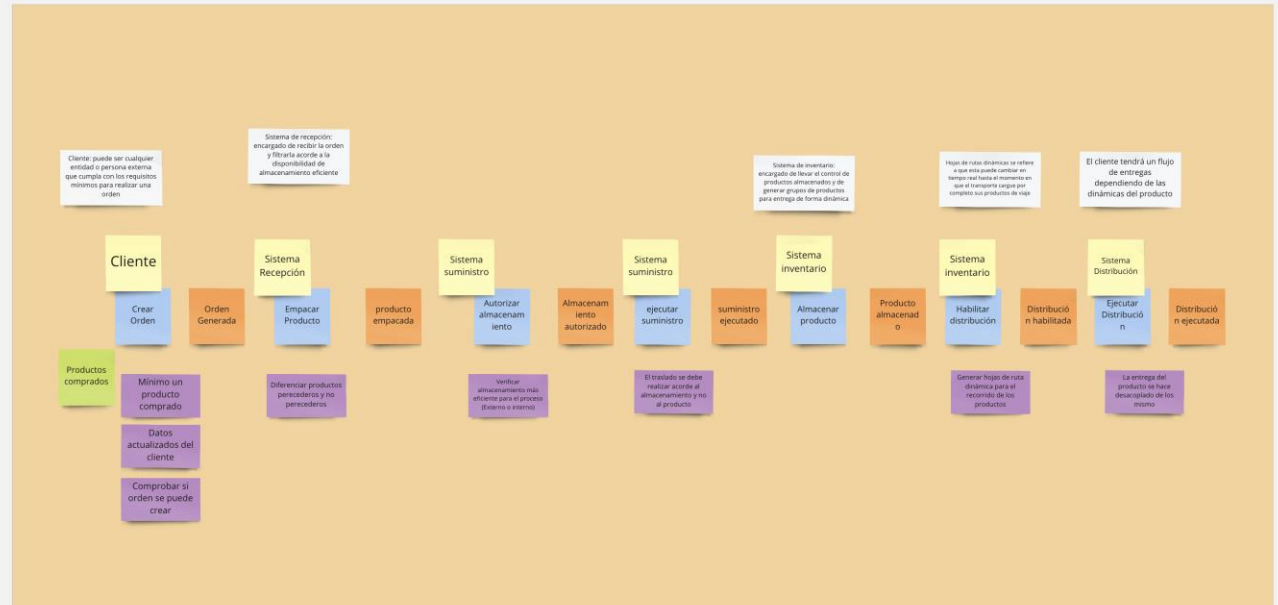


Modelo TO-BE



Diseño y construcción de soluciones no monolíticas

"Cadena de suministro como Servicio CSaaS" TO-BE usando el método EventStorming



Los modelos que se muestran anteriormente se pueden encontrar en el siguiente enlace:

[EventStorming EDA](#)

3. Documentación de contextos acotados

Con el listado y el modelo de TO-BE, hemos agrupado los conceptos que creemos tienen una relación y le hemos puesto un nombre para identificarlos, el resultado final es el siguiente:



Diseño y construcción de soluciones no monolíticas

Dominio:
Entrega de bienes
alrededor del país
(Servicio Última
milla)

Sub dominios

Recepción

- Orden
- Producto
- Supermercado o restaurante
- usuario
- proveedor productos

Soporte

Suministro

- Empacar Producto para envío a CD o Bodega
- distribuidores suministro

Generico

Inventario

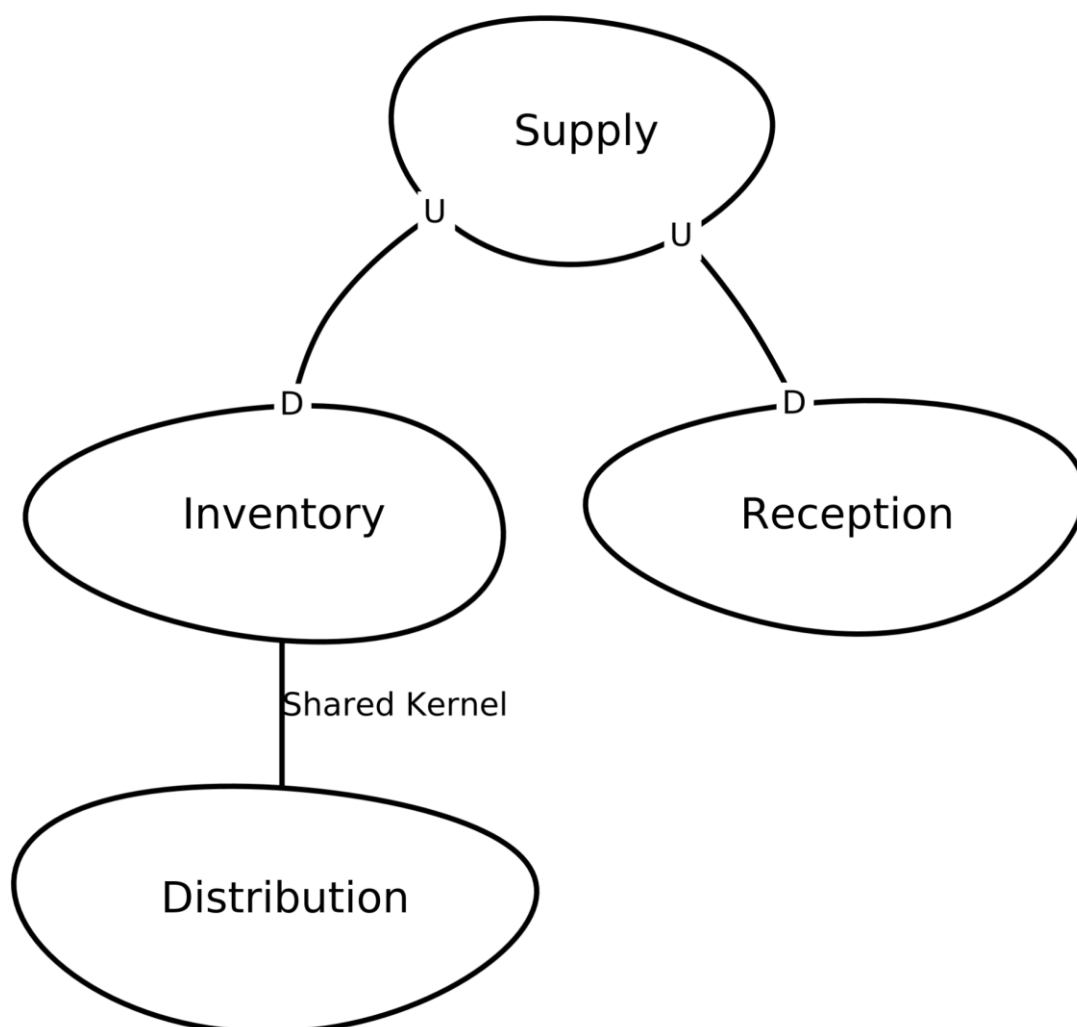
- Centros de disponibilidad
- Bodegas
- Proveedor de productos
- pickup point

soporte

Distribución

- Empacar Producto para envío a usuario final
- Hoja de ruta
- Flota Tractores
- Flota domiciliarios
- proveedor transporte
- distribuidores final

core





Diseño y construcción de soluciones no monolíticas



Diseño y construcción de soluciones no monolíticas