



MAESTRÍA EN INGENIERÍA DE SOFTWARE-ONLINE
CURSO DE DISEÑO DE APLICACIONES NO
MONOLÍTICAS

INTEGRANTES:
PATRICK MYKODA

ALBA LIGIA CARDONA
MONTOYA

CARLOS EDUARDO VILLAMIL
CASTAÑEDA

CHRISTIAN BORRÁS TORRES

ENTREGA SEMANA DOS

2024

Entrega 1: Diseño y arquitectura de dominio

Grupo ANM

Repositorio del proyecto: <https://github.com/PatrickMykoda/proyecto-propiedades-alpes>

=> Los tres entregables *documentación de dominios*, *mapa de contexto as-is* y *mapa de context to-be* se encuentran en la carpeta cml dentro de src/main:

proyecto-propiedades-alpes / src / main / cml /

PatrickMykoda Delete src/main/cml/ContextMap-to-be_patrick.cml 8aa4720 · yesterday History

This branch is 12 commits ahead of ContextMapper/web-ide-demo:master . Contribute Sync fork

Name	Last commit message	Last commit date
..		
ContextMap-as-is.cml	Avance ContextMap to-be	yesterday
ContextMap-to-be.cml	Avance ContextMap to-be	yesterday
dominios-subdominios.cml	Add files via upload	2 days ago

=> El entregable lenguaje ubicuo se encuentra en la carpeta lenguaje-ubico dentro de src/main. En el archivo *Event Storming - Grupo ANM.jpg* tiene una vista completa del lenguaje ubicuo mientras los otros archivos jpg enfocan cada uno una parte específica del flujo.

Name	Last commit message	Last commit date
..		
Event Storming - Grupo ANM - PULL.jpg	Se agregó documentación del lenguaje ubicuo	now
Event Storming - Grupo ANM - PUSH.jpg	Se agregó documentación del lenguaje ubicuo	now
Event Storming - Grupo ANM - Procesamiento de datos.jpg	Se agregó documentación del lenguaje ubicuo	now
Event Storming - Grupo ANM - Validación y Enriquecimiento de dato...	Se agregó documentación del lenguaje ubicuo	now
Event Storming - Grupo ANM.jpg	Se agregó documentación del lenguaje ubicuo	now

En las siguientes páginas explicamos las decisiones que tomamos al diseñar los dominios, el lenguaje ubicuo y los mapas de contexto.

1. Documentación de dominios y subdominios

```

Domain InformacionBienesRaicesDomain {
    domainVisionStatement = "Se encarga de consolidar, procesar, analizar y entregar información de bienes raíces comerciales"

    Subdomain CapturaDeDatosManual {
        type = CORE_DOMAIN
        domainVisionStatement = "Este subdominio es el encargado de obtener la información de forma manual"
    }

    Subdomain IngestionDeDatosFuente {
        type = CORE_DOMAIN
        domainVisionStatement = "Este subdominio es el encargado de obtener la información de fuentes externas de manera automatizada"
    }

    Subdomain ProcesamientoDeDatos {
        type = CORE_DOMAIN
        domainVisionStatement = "Este subdominio es el encargado de procesar y limpiar los datos obtenidos por los fuentes manuales y automáticas"
    }

    Subdomain ValidacionDeDatos {
        type = CORE_DOMAIN
        domainVisionStatement = "Este subdominio es el encargado de validar y enriquecer los datos"
    }

    Subdomain AnalisisYDisponibilizacionDeDatos {
        type = CORE_DOMAIN
        domainVisionStatement = "Este subdominio es el encargado de implementar las herramientas de analítica y consulta de información usadas por los clientes"
    }

    Subdomain Infraestructura {
        type = SUPPORTING_DOMAIN
        domainVisionStatement = "Este subdominio es el encargado de proveer la infraestructura necesaria para ingesta, transformación y analítica de datos"
    }
}

```

La imagen previamente mostrada, es el modelo que describe los dominios y subdominios para un proyecto denominado "**Propiedades de los Alpes**", enfocado en proveer información sobre bienes raíces comerciales. A continuación, se detalla cada parte del código y su significado en el contexto del proyecto:

Dominio "**InformacionBienesRaicesDomain**" es el dominio raíz del modelo, que encapsula toda la lógica y los subdominios relacionados con la gestión y procesamiento de la información sobre bienes raíces comerciales. La declaración `domainVisionStatement` proporciona una visión general o misión del dominio, indicando que su responsabilidad principal es consolidar, procesar, analizar y entregar información sobre bienes raíces comerciales.

Entre los sub-dominios encontrados está el **CapturaDeDatosManual**, este subdominio es esencial para el proceso de recolección de datos, encargándose de la obtención manual de información. Es clasificado como un `CORE_DOMAIN`, lo que indica que es un dominio central para la operación del negocio, fundamental para la propuesta de valor de la empresa.

IngestionDeDatosFuente, se encarga de la adquisición automatizada de datos de fuentes externas. Al igual que el subdominio anterior, es un `CORE_DOMAIN`, reflejando su importancia crítica en la arquitectura del sistema.

ProcesamientoDeDatos, su finalidad es la procesa y limpia los datos recopilados, tanto de fuentes manuales como automáticas, asegurando que la información sea precisa y útil para su posterior análisis.

ValidacionDeDatos, es el responsable de la validación y enriquecimiento de los datos, este subdominio juega un papel crucial en garantizar la calidad y fiabilidad de la información gestionada.

AnalisisYDisponibilizacionDeDatos, implementa herramientas analíticas y de consulta para los clientes, facilitando el acceso y análisis de la información sobre bienes raíces.

Infraestructura, esta, a diferencia de los anteriores, es clasificado como `SUPPORTING_DOMAIN`, indicando que su propósito es apoyar a los dominios principales proporcionando la infraestructura necesaria para la ingesta, transformación y análisis de datos.

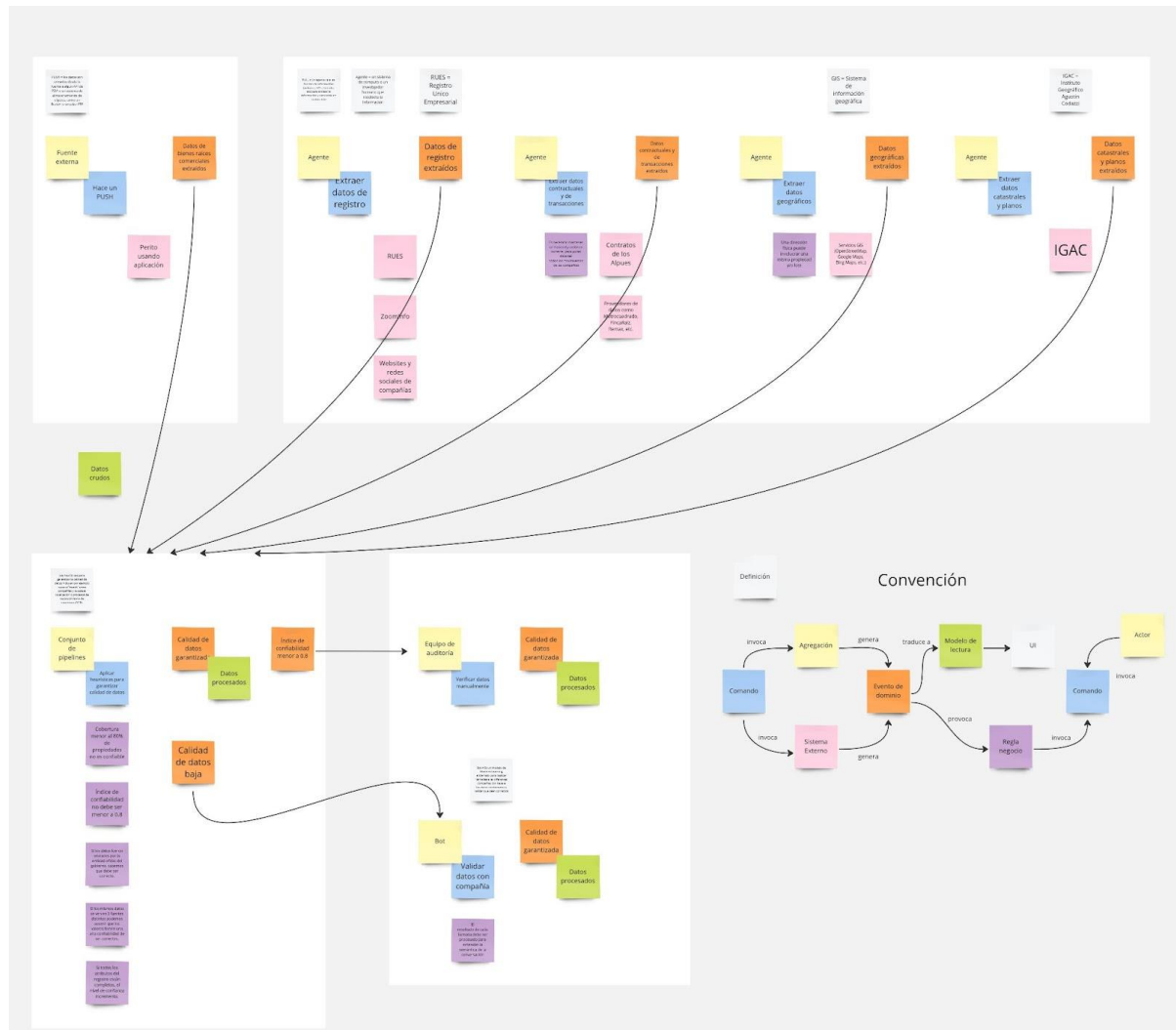
2. Event Storming => Lenguaje Ubicuo

En la imagen se puede ver el tablero del Event Storming que realizó el grupo para encontrar el Lenguaje Ubicuo para el proyecto. Para observarlo con más detalle puede dar click en [este enlace](#) que le lleva al

tablero

en

Miró.



Ahora nos enfocamos en las diferentes etapas del flujo que identificamos en el Event Storming.

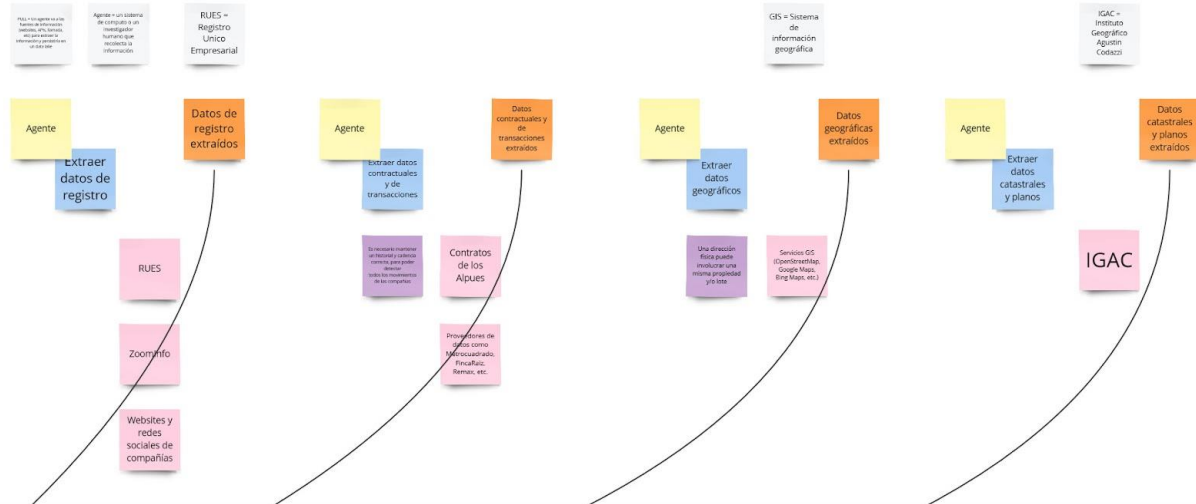
PUSH

Como es mencionado en el enunciado del proyecto, la captura y ingestión de datos se realiza a través de dos modos diferentes, PUSH y PULL. Decidimos que cada una de estas acciones debería ser representada en el tablero, dado los comandos que llevan a los eventos son realizados por diferentes actores. En el caso del PUSH el actor es una fuente externa que envía datos de todo tipo comunicándose con una API de PDA.



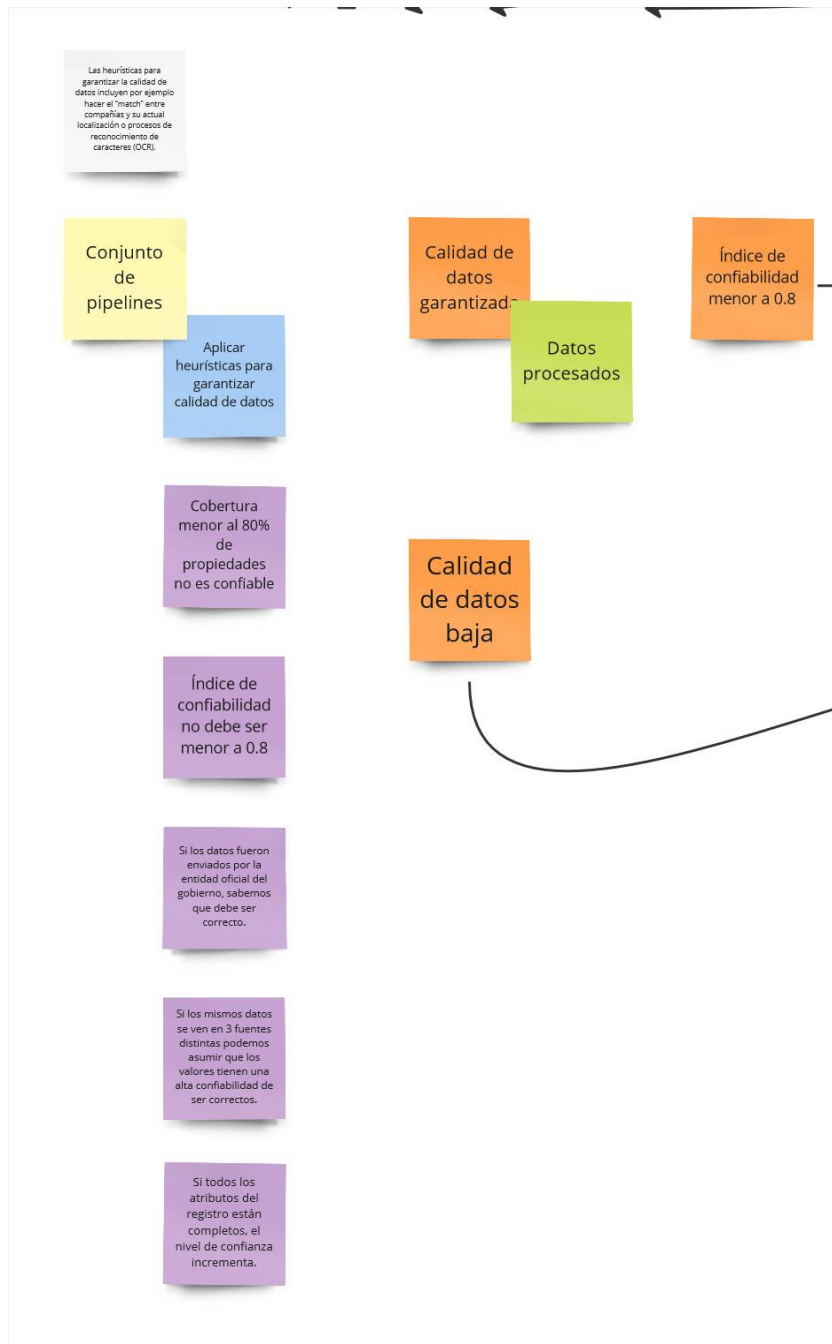
PULL

En el caso del PULL los eventos son causados por actores internos como programas de software o investigadores humanos. Repartimos los eventos entre los diferentes tipos de datos ya que los agentes extraen la información de diferentes fuentes y, por lo menos en el caso de los programas software, los agentes se distribuyen entre diferentes partes del sistema de PDA.



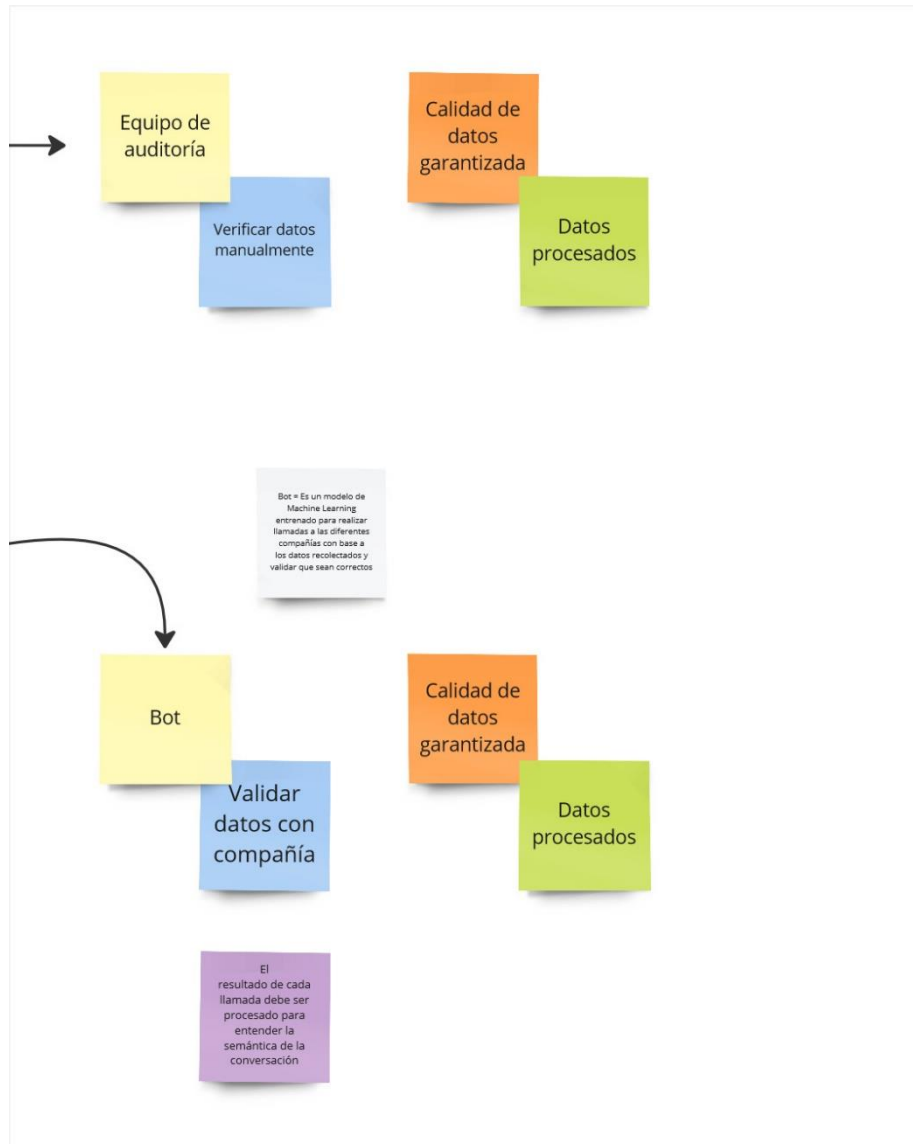
Procesamiento de Datos

Todos los eventos mencionados anteriormente tienen que ver con la extracción de datos y se realizan simultáneamente. La siguiente etapa del flujo es el Procesamiento de Datos donde un conjunto de piplines aplica heurísticas para asegurar la confiabilidad de los datos. Puede haber tres posibles eventos como resultado: La calidad de los datos es garantizada, el índice de confiabilidad es menor a 0.8 o la calidad de los datos es tan baja que un bot trata de enriquecerlos a través de una llamada en el siguiente paso.



Validación y Enriquecimiento de datos

Finalmente, si los datos no cumplen con la calidad esperada, llegan a la etapa de validación y enriquecimiento. Dependiendo del caso, es el equipo de auditoría o un bot programado en base de machine learning que tratan de aumentar la calidad de los datos hasta que puedan ser puestos a disponibilidad de los clientes.



3. Mapa de contexto As-Is

```

/* The DDD Cargo sample application modeled in CML. Note that we split the application into multiple bounded contexts. */
ContextMap DDDMapPropiedadesAlpesAsIs {
  contains CapturaDeDatosManualContext
  contains IngestionDeDatosFuenteContext
  contains ProcesamientoDeDatosContext
  contains ValidacionDeDatosContext
  contains DisponibilizacionDeDatosContext
  contains AnalisisDeDatosContext

  CapturaDeDatosManualContext [U, OHS]->[D, ACL] ProcesamientoDeDatosContext

  IngestionDeDatosFuenteContext [U, OHS]->[D, ACL] ProcesamientoDeDatosContext

  ProcesamientoDeDatosContext [U]->[D] ValidacionDeDatosContext

  ValidacionDeDatosContext [U]->[D] DisponibilizacionDeDatosContext

  ValidacionDeDatosContext [U]->[D] AnalisisDeDatosContext
}

/* The original booking application context */
BoundedContext CapturaDeDatosManualContext {
  domainVisionStatement = "Este contexto es el encargado de obtener la información de forma manual"
}

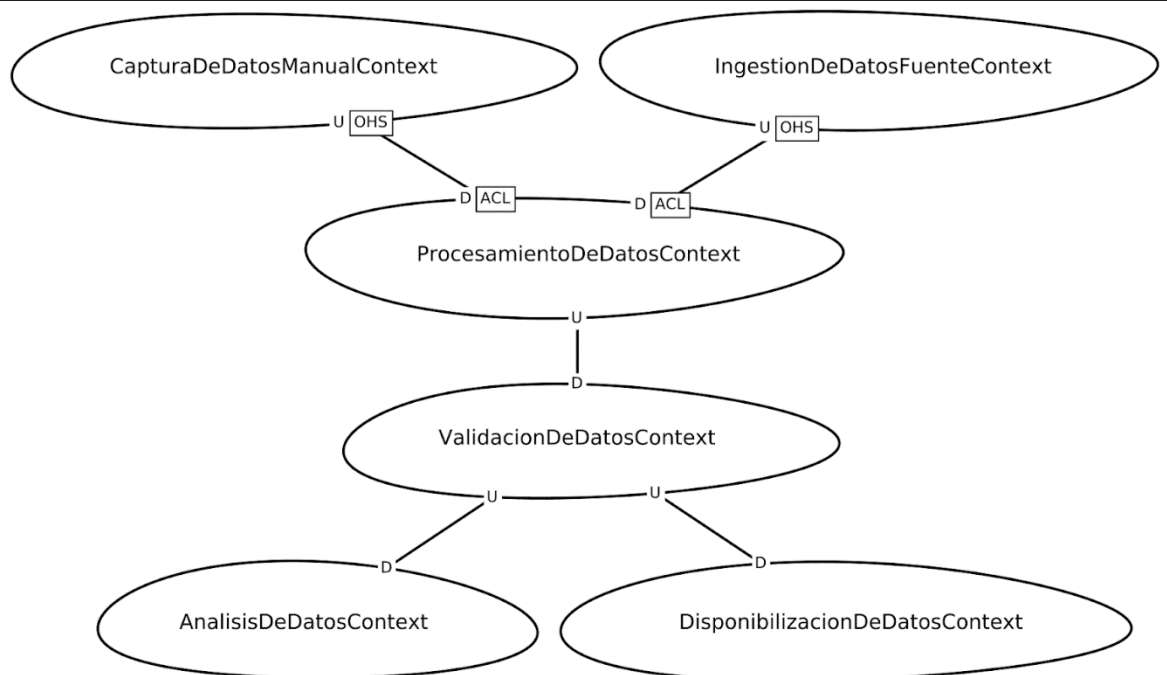
BoundedContext IngestionDeDatosFuenteContext {
  domainVisionStatement = "Este contexto es el encargado de obtener la información de fuentes externas de manera automatizada"
}

BoundedContext ProcesamientoDeDatosContext {
  domainVisionStatement = "Este contexto es el encargado de procesar y limpiar los datos obtenidos por los fuentes manuales y aut
BoundedContext ValidacionDeDatosContext {
  domainVisionStatement = "Este contexto es el encargado de validar y enriquecer los datos"
}

BoundedContext DisponibilizacionDeDatosContext {
  domainVisionStatement = "Este contexto es el encargado de implementar la consulta de información usadas por los clientes"
}

BoundedContext AnalisisDeDatosContext {
  domainVisionStatement = "Este contexto es el encargado de implementar las herramientas de analítica usadas por los clientes"
}

```



Conforme a la imagen anterior, que describe la interacción de diferentes contextos acotados en el sistema de “Propiedad de los Alpes”, en el AS-IS, los cuales se describirán a continuación: **CapturaDeDatosManualContext**, este contexto se encarga de la recopilación manual de datos, esto involucra a investigadores de campo y otros agentes que manualmente ingresan datos en el sistema, como características de las propiedades y transacciones realizadas. Para la **IngestionDeDatosFuenteContext**, como la obtención automática de datos de las fuentes externas, el **ProcesamientoDeDatosContext**, para la limpieza y procesamiento de los datos obtenidos manual y automáticamente, aplicando las reglas de negocio, por otro lado la **ValidacionDeDatosContext**, para el enriquecimiento de los datos, verificación

la precision y la complementacion de la informacion, la **DisponibilidadDeDatosContext**, implementa las consultas a la informacion, hace que los datos validados esten disponibles para los clientes. Finalmente, el **AnalisisDeDatosContext**, encargado de las funcionalidades de analisis de tendencia de mercado, la compracion de precios etc...

Asi mismo, frente a las relaciones y tipo de integracion de los datos, cada una de ellas es unidireccional, toda vez que los datos recolectados fluyen a una sola direccion, **CapturaDeDatosManualContext** -> **ProcesamientoDeDatosContext** [U, OHS], como contexto es Upstream, lo que significa que los datos fluyen desde la captura manual hacia el procesamiento de datos. Utiliza el estilo de integración Open Host Service (OHS), indicando que ofrece una API estandarizada que otros contextos pueden utilizar para interactuar con él. [D, ACL]: Desde la perspectiva de **ProcesamientoDeDatosContext**, la relación es Downstream, lo que implica que depende de los datos del contexto de **CapturaDeDatosManual**. Utiliza una Anti-Corruption Layer (ACL) para asegurarse de que los datos recibidos se traducen correctamente a su propio modelo de dominio, protegiéndolo de posibles cambios o diferencias en el modelo de datos del contexto upstream.

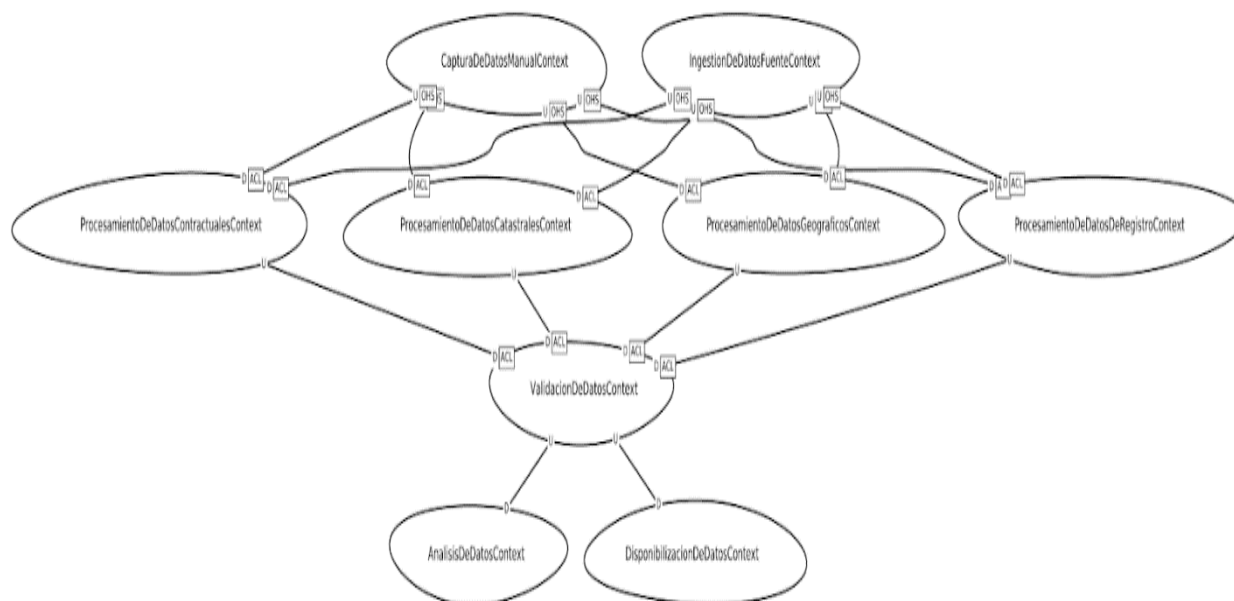
IngestionDeDatosFuenteContext -> **ProcesamientoDeDatosContext** [U, OHS] que es similar a la primera relación, **IngestionDeDatosFuenteContext** es Upstream y proporciona datos a **ProcesamientoDeDatosContext** a través de un Open Host Service. [D, ACL], para **ProcesamientoDeDatosContext**, esta es una relación Downstream con una Anti-Corruption Layer para manejar los datos que entran.

ProcesamientoDeDatosContext -> **ValidacionDeDatosContext** [U]->[D] **ProcesamientoDeDatosContext** es Upstream y envía datos procesados y limpios al contexto de **ValidacionDeDatosContext**, que actúa como Downstream. Aquí no se especifican mecanismos de integración como OHS o ACL, lo que podría implicar una integración directa o interna entre estos contextos.

ValidacionDeDatosContext -> **DisponibilizacionDeDatosContext** [U]->[D] **ValidacionDeDatosContext** es Upstream para **DisponibilizacionDeDatosContext**, proporcionando datos validados y posiblemente enriquecidos para su uso en consultas y análisis.

ValidacionDeDatosContext -> **AnalisisDeDatosContext** [U]->[D]: De igual forma, **ValidacionDeDatosContext** envía datos al contexto de **AnalisisDeDatos**, que los utiliza para realizar análisis y generar insights.

4. Mapa de contexto ToBe



Como se puede apreciar en la imagen superior, para la arquitectura propuesta decidimos conservar los contextos de captura de datos manual y de ingestión de datos de fuentes externas separados, teniendo en cuenta que cada uno de ellos puede involucrar bastantes funcionalidades y lógica de negocio diferentes. Esto debido a que para la captura de datos manual, por ejemplo, necesitamos implementar herramientas de interfaz de usuario para que las personas encargadas de recolectar la información puedan ingresar la información en sus diferentes tipos (geográfica, contractual, evidencia fotográfica, información de avalúos...) de manera práctica y efectiva (además de toda la lógica de negocio y funcionalidades del backend). Evidentemente para esto se necesita un buen equipo de diseño de interfaz de usuario, desarrollo de frontend, lo cual sería innecesario en el contexto de la ingestión automática desde fuentes externas. En cambio, en dicho contexto necesitaremos un equipo especialista en la implementación de buenas herramientas para el consumo de esa información (apis, protocolos síncronos y asíncronos, seguridad...).

Dentro del contexto de captura de datos manual se implementa una api que permite que las diferentes interfaces de usuario entreguen los datos al sistema. En contraparte, el contexto de ingestión de datos desde fuentes externas consume apis de servicios externos (IGAC, Google Maps, Bing Maps, Contratos de Los Alpes, ZoomInfo, RUES...) para traer los datos.

El contexto de procesamiento de datos se dividió en 4 contextos diferentes teniendo en mente servicios especializados en el procesamiento de cada tipo de información.

Conservamos también el contexto de validación de datos y este persiste como una sola unidad teniendo en cuenta que debe validar la coherencia de los paquetes de datos ya consolidados para los diferentes predios, propietarios y demás entidades.

Finalmente, conservamos también los contextos de análisis de datos y disponibilización de datos como dos unidades independientes, teniendo en cuenta que se debe presentar al usuario la posibilidad de ver información consolidada pero también probablemente quiera visualizar análisis a partir de ella.

En los contextos de procesamiento de datos y validación de datos se propone la implementación de capas anticorrupción, con la finalidad de que sea dentro de estos contextos donde se defina el formato en que debe llegar la información desde los contextos que la entregan.