

Entrega 2 - ECSDI

Amazon2



Alex Huang Feng

Nico Ernesto Francisquelo Tacca

Eloi Roca Corcelles

1. Cambios arquitectónicos del sistema

A continuación vamos a explicar los diferentes cambios arquitectónicos que hemos aplicado a nuestra especificación del sistema.

Hemos creado un nuevo agente llamado *Ag Mostrar Productos* a fin de separar tareas del *Ag Venta Productos*. De esta manera, conseguimos separar las peticiones de mostrar productos de las peticiones de compra de productos. Eso puede ser clave ya que las peticiones de productos pueden ser bastante más numerosas que las peticiones de compra.

Por lo tanto, ahora tendremos un agente *Ag Mostrar Productos* dedicado exclusivamente a las peticiones de *Ver Productos disponibles* y el otro agente *Ag Venta Productos* que se dedicará a gestionar todo el proceso de una compra de uno o más productos.

La distribución de las demás tareas en los diferentes agentes se mantiene como en la primera entrega.

2. Diseño detallado

Agente Recomendador

El agente Recomendador está compuesto por un plan independiente y una capacidad. El plan *guardar reseñas* es disparado por la acción *Usuario quiere escribir una reseña* y pretende almacenar la reseña creada por el usuario comprador para su uso posterior. Por eso mismo, accede a los datos de *Feedback de productos* para escribir la reseña.

La capacidad de *Enviar Recomendación* es más compleja. Contiene dos planes: por una parte, obtener los intereses de un usuario y, por otra, enviar la recomendación. El primer plan es disparado por un evento de *Toca enviar recomendación* que será periódica definida por nosotros. Este plan accederá al *Historial de compras* y *Feedback de Productos* para saber qué productos pueden ser de interés para el usuario. Una vez haya calculado los intereses, se transmitirá el interés al segundo plan, quien se encargará de enviar la recomendación anteriormente calculada al usuario comprador.

Agente Venta productos

Este agente se dedica a la gestión íntegra del proceso de compra de un producto. Está formado por un plan de *Crear pedido* y una capacidad de *Cobrar pedido*.

El primer plan se dedica a la interacción con el usuario y al procesamiento del pedido que ha hecho el usuario. Será disparado por una petición externa al sistema de *Petición de compra de productos* y como respuesta a esta petición, lanzará una respuesta al usuario con la confirmación del pedido. Asimismo, enviará un mensaje al agente *Ag Logístico* para que éste pueda procesar el envío de este pedido. Este plan accederá a la información de productos, a la información de clientes y al historial de compras/pedidos.

Por otra parte, la capacidad *Cp Cobrar pedido* se dedicará a ejecutar el proceso íntegro del pago del pedido. Contiene tres planes diferentes. El *Plan Cobrar pedido* es disparado por un mensaje del agente *Ag Almacén* con la confirmación del envío del producto. Una vez recibe esta confirmación, de procederá a cobrar el pedido del comprador donde en función del tipo de producto comprado se ejecutará el *Plan emitir factura* (si el producto es propio) o el *Plan pagar vendedor externo* (si el producto es de terceros). El *Plan emitir factura* tan solo enviará la factura al cliente una vez pagado y el *Plan pagar vendedor externo* ingresará el dinero al vendedor externo.

Agente Almacén

Este agente estará formado por una capacidad y un plan. Por una parte tendrá que negociar con los transportistas, y por otra, comunicar al transportista que ha sido elegido del nuevo envío. La capacidad *Negociar con transportistas* es disparado por un mensaje recibido del agente Logístico en el que se informa que hay un nuevo envío que realizar. Dentro de esta capacidad, se separa en dos planes, uno es la creación del lote y otro el negociar el lote con los transportistas. El *Plan Crear Lote* pretende agrupar los diferentes productos para que el envío sea en menor cantidad. El segundo plan *Negociar Lote con transportistas* se dedica a negociar el precio del envío del lote creado anteriormente con los diferentes transportistas. Enviará peticiones de precios a los diferentes transportistas externos y éstos responderán con el precio con el que trabajan. Deberá guardar en memoria estos precios para poder compararlos posteriormente.

Cuando finaliza, elegirá a un transportista según unos criterios y enviará un mensaje al segundo plan *Comunicar transportista*. Este plan avisará al transportista de que hay un nuevo envío y después comunicará al *Agente Venta Productos* para que actualice su stock.

Agente Logístico

El agente logístico contiene dos planes. El primero es disparado por un mensaje del agente *Ag Venta Productos* en el que se informa de que ha habido una nueva compra efectuada. Se accede a los *datos del almacén*, se informa al agente *Ag Almacen* de que hay un nuevo envío que realizar y en caso de que sea un producto externo y es el vendedor quien lo tiene que realizar, se lanza el segundo plan. Éste comunica al vendedor externo del producto pedido y comunica un mensaje al agente *Ag Venta productos* para actualizar el *stock* y proceder al cobro del pedido.

Agente Gestor de productos externos

Este agente encargado de gestionar los productos externos sólo contiene un plan sencillo. Es disparado por la petición de un vendedor externo que quiere vender un producto a partir de nuestra plataforma. El plan debe guardar en memoria información del producto que va a poner a la venta y información del vendedor. Cuando el proceso ha finalizado, se envía una confirmación al vendedor de que el producto ha sido publicado.

Agente Devoluciones

Este agente contiene una capacidad formada por tres planes de trabajo. El primer plan *Decidir resultado devolución* se dedica a procesar la petición externa de nueva devolución y a decidir si la devolución será aceptada según los motivos del cliente. Una vez definida la decisión final, se disparará el segundo plan *Comunicar datos de la devolución*. Éste informará al comprador del resultado de la petición y información asociada a cómo realizar la devolución.

El segundo plan se disparará cuando haya un producto devuelto en nuestros almacenes. Se devolverá el dinero al comprador.

Agente Mostrar productos

Este agente contiene un solo plan ya que se dedicará a procesar todas las peticiones de usuarios que quieren acceder a la plataforma para ver los productos puestos a la venta. El plan *Mostrar Productos* se disparará por una petición de *Petición Productos disponibles*. Como respuesta, enviará la información de los productos disponibles del sistema.

3. Descripción de ontologías

Para la realización de nuestra versión de Amazon utilizaremos parte de una ontología externa ya existente que contiene conceptos que nos resultan útiles junto con la ontología que hemos diseñado para poder ampliar y representar correctamente el dominio de nuestro proyecto.

3.1. Conceptos ontología externa: <http://schema.org>

- **Person:** Este concepto representa una persona cliente en nuestro sistema. Nos interesan concretamente las propiedades: *address, email, name* y *telephone*.
- **GeoCoordinates:** Este concepto representa una ubicación o localización real. Nos interesan concretamente las propiedades: *address, addressCountry, latitude, longitude, postalCode*.
- **Product:** Este concepto representa un producto de nuestro sistema. Nos interesan concretamente las propiedades: *productID, brand, color, aggregateRating* (valoración), *manufacturer* (para saber si es producto propio o externo) y *propertyValue* (precio).
- **Rating:** Este concepto representa aquello que en nuestro sistema llamamos valoración o feedback del usuario. Nos interesan concretamente las propiedades: *author, ratingValue, description*.
- **Organization:** Este concepto representa tanto a los vendedores externos como a los transportistas de nuestro sistema. Nos interesan concretamente las propiedades: *address, name, telephone, email* y *disambiguatingDescription* (que utilizaremos para indicar si se trata de un vendedor o de un transportista)

3.2. Ontología diseñada

Percepciones:

Conceptos que representan las percepciones que recibe el sistema del entorno exterior. Serán recibidas y procesadas por los agentes del sistema. Hemos definido las siguientes percepciones:

Añadir_producto_externo: Representa la percepción del sistema que recibe cuando un vendedor tercero quiere vender un producto a partir de nuestro sistema. Contiene una propiedad *productoExternoContiene* que la relaciona con el concepto Vendedor y el Producto_externo asociado.

Nueva_compra_producto: Representa la percepción en la que un comprador quiere comprar uno o más productos del sistema. Contiene la propiedad *nuevaCompraContiene* que la relaciona con un pedido del usuario. Éste pedido contiene los productos que quiere comprar y la localización a la que debe ser enviado el pedido.

Nueva_reseña: Representa la percepción externa de la escritura de una nueva reseña de un producto del sistema. Contiene la propiedad *nuevaReseñaContiene* que relaciona esta percepción con el concepto Feedback que representa una reseña.

Petición_devolución: Representa la percepción de una petición de devolución de un producto por un usuario que ha comprado a partir de nuestra plataforma. Contiene una propiedad *peticionDevolucionContiene* que la relaciona con el concepto Devolución. Este está relacionado con el producto asociado.

Petición_productos_disponibles: Representa la percepción de un cliente que quiere ver los productos disponibles de nuestro sistema. Contiene una propiedad *peticionProductosDisponibles* que la relaciona con el concepto Restricciones_cliente. Éste representa las restricciones de compra que tiene el cliente como la marca o el rango del precio que busca.

Precios_envio: Representa la percepción que envía el agente Almacén para conocer los precios de envío que proporcionan los diferentes transportistas. Tiene una propiedad *preciosEnvioPercep* que contiene la relación con el pedido asociado.

Producto_devuelto: Representa la percepción de un producto que ha sido físicamente devuelto al almacén. Contiene una propiedad *devolucionProductoConfirmado* que la relaciona con el producto que ha sido devuelto.

Toca_enviar_recomendación: Representa el evento periódico que recibe el sistema para enviar las recomendaciones al cliente.

Acciones:

Conceptos que representan las acciones o respuestas que deben realizar los agentes después de recibir ciertas percepciones o mensajes. En nuestro sistema tenemos:

Comunicación_resultado_devolución: Representa la acción de comunicar al cliente que ha solicitado una devolución del resultado de dicha solicitud. Tiene una propiedad *resultadoDevolución* que la relaciona con el concepto Devolución.

Comunicar_pedido_enviar: Representa la acción de comunicar a un vendedor externo que debe encargarse del envío de un pedido. Tiene una propiedad *pedidoExterno* que la relaciona con el concepto Pedido.

Avisar_transportista_envio: Representa la acción de avisar al transportista elegido, que debe encargarse del envío de un lote. Tiene una propiedad *enviaLote* que lo relaciona con Lote.

Cobrar_pedido: Representa la acción de cobrar el pedido al cliente una vez se ha gestionado el envío de éste. Tiene una propiedad *cobrarCliente* que lo relaciona con el concepto Person(schema.org)

Confirmacion_producto_externo_añadido: Representa la confirmación de que un producto de un vendedor externo ha sido añadido al stock del sistema.

Confirmacion_cesta: Representa la acción de confirmar los productos que formarán el pedido del cliente.

Devolver_dinero_cliente: Representa la acción de devolver el dinero al cliente cuando se ha aceptado la devolución de un producto y se ha recibido dicho producto.

Emitir_factura: Representa la acción de generar una factura con los detalles del pedido (desglose) y los datos de la entrega. Tiene una propiedad *tieneFactura* que lo relaciona con el concepto Factura_de_compra.

Enviar_recomendacion: Representa la acción de enviar una recomendación al cliente cuando el sistema decide que toca enviarla. Tiene una propiedad *enviaRecomendacion* que la relaciona con el concepto Recomendación.

Informacion_productos_disponibles: Representa la acción de informar al cliente sobre los productos disponibles que cumplen su restricción. Tiene una propiedad *productosFiltrados* que la relaciona con el concepto Products(schema.org).

Ingresar_dinero_vendedor_externo: Representa la acción de devolución de dinero al vendedor externo cuando se ha devuelto un producto.

Pedir_precio_envio: Representa la acción de pedir precios a los transportistas con el fin de seleccionar el más adecuado para enviar el lote.

Mensajes Internos:

Conceptos que representan los mensajes internos entre agentes de nuestro sistema que forman un protocolo de comunicación.

Solicitud_envio: mensaje que avisa al centro logístico que debe procesar el envío de un pedido.

Realiza_envio: mensaje que avisa al almacén que debe encargarse de enviar un pedido. Tiene una propiedad *realizaEnvioPedido* que lo relaciona con Pedido.

Confirmacion_envio: mensaje que confirma que el almacén ya ha avisado al transportista que debe realizar el envío del lote que contiene ese pedido.

Confirmacion_envio_externo: mensaje que confirma que el pedidos ya se está procesando por el vendedor externo, por lo tanto ya se puede proceder al cobro y pago al vendedor.

Pedido

Concepto que representa un pedido de un cliente; es decir, los productos que ha confirmado que va a comprar. Tiene las propiedades: *pedidoContiene* que lo relaciona con Productos, *pedidoDestinadoA* que lo relaciona con GeoCoordinates (schema.org), *pedidoPerteneceA* que lo relaciona con Person (schema.org), *fechaPedido* y *precioTotal*.

Recomendación

Concepto que representa una recomendación del sistema a un cliente sobre los productos en los que puede estar interesado.

Almacén

Concepto que representa un almacén físico. Tiene como propiedades *almacena* que lo relaciona con Products(schema.org), *almacenAgrupa* que lo relaciona con Lote, *almacenTransportistas* que lo relaciona con los Organization(schema.org) que sean transportistas de ese almacén y *esLocalizadoEn* que lo relaciona con GeoCoordinates(schema.org).

Razón_devolución

Concepto que representa el motivo por el cual el cliente decide devolver un producto previamente comprado en la tienda. Hay 3 tipos de razones de devolución en nuestro sistema:

- Defectuoso: El producto no ha llegado en las condiciones esperadas y presenta defectos.
- Equivocado: El cliente ha recibido un producto que no había pedido.
- No satisfactorio: El producto no ha superado las expectativas del cliente.

Tiene como propiedad *tiene_razon*, que lo relaciona con Devolución y indica que esa devolución tiene esta instancia como razón de devolución.

Restricciones_cliente

Concepto que representa las restricciones que introduce el cliente en el momento de realizar una búsqueda de productos y encontrar solamente productos que cumplan con todas sus restricciones. Tiene como propiedades *cumpleRestriccion*, que lo relaciona con los elementos de Producto que cumplen con la restricción que representa el concepto, *peticionProductosDisponibles*, que lo relaciona con la percepción *Peticion_productos_disponibles*, y *tieneRestriccion*, que lo relaciona con Cliente y representa al cliente que genera la restricción.

Categorías

Concepto que representa a las distintas categorías de productos de nuestro sistema. Tiene una propiedad llamada *perteneceA* que lo relaciona con el concepto Products (schema.org) y representa a los productos que pertenecen a esa categoría.

Centro_logístico

Concepto que representa al centro logístico de nuestro sistema. Tiene como propiedades *tienePedidos*, que lo relaciona con Pedido y representa el pedido que tiene que ser enviado, y *creaEnvios*, que lo relaciona con Envío y representa el envío que genera el centro logístico.

Devolución

Concepto que representa a una devolución de un producto en nuestro sistema. Tiene las siguientes propiedades:

- *devolucionAsociada*: Lo relaciona con Product (schema.org), representa el producto que tiene que ser devuelto.

- *devolucionContiene*: Lo relaciona con el concepto Factura_de_compra y representa la factura de compra asociada a esta devolución.
- *peticionDevoluciónContiene*: Lo relaciona con la percepción Petición_devolución y representa la percepción que activa ese proceso de devolución.
- *resultadoRevolución*: Lo relaciona con la acción Comunicacion_resultado_devolucion.
- *tieneRazon*: Lo relaciona con el concepto Razon_Devolución y representa el motivo de la devolución.

Envío

Concepto que representa el envío de un pedido en nuestro sistema. Tiene como propiedades *creaEnvios*, que lo relaciona con Centro_logístico y representa el centro logístico que ha generado el envío concreto, y *estáFormadoPor*, que lo relaciona con Product (schema.org) y representa los productos que contiene el envío y que serán enviados al cliente.

Factura_de_compra

Concepto que representa la factura de un pedido en nuestro sistema. Tiene como propiedades *devolucionContiene*, que lo relaciona con Devolución y representa la factura de la compra del pedido que contenía el producto a devolver, y *tieneFactura*, que lo relaciona con la acción Emitir_factura.

Lote

Concepto que representa a un lote, que es un conjunto de productos a enviar, en nuestro sistema. Tiene las siguientes propiedades:

- *almacenAgrupa*: Lo relaciona con el concepto Almacen y representa al almacén que crea ese lote.
- *enviaLote*: Lo relaciona con la acción Avisar_transportista_envio.
- *esEnviadoPor*: Lo relaciona con Organization (schema.org) y representa al transportista encargado de enviar ese lote.
- *loteTiene*: Lo relaciona con Product (schema.org) y representa a los productos que forman parte de ese lote.

Oferta_transportista

Concepto que representa la oferta que hace un transportista para enviar un lote. Tiene la propiedad *recibeOferta* que lo relaciona con Almacén y representa al almacén que recibe esa oferta para enviar un lote.

Recomendación

Concepto que representa las recomendaciones que reciben los usuarios sobre productos que podrían interesarles. Tiene como propiedades *recomienda*, que lo relaciona con Product (schema.org) y representa los productos que se recomiendan, y *enviaRecomendación*, que lo relaciona con la acción Enviar_recomendación.

3.3 Relación de agentes con la ontología

En este apartado explicaremos los diferentes agentes que tenemos en mente implementar con Python. Cada agente utilizará ciertos conceptos definidos anteriormente con tal de que el sistema tenga unas ontologías que puedan entender todos.

Ag Recomendador

Como hemos explicado en el diseño detallado, este agente se dedica a enviar recomendaciones al usuario de manera periódica. Conocerá los conceptos de Recomendación, Producto y Rating para poder construir esta recomendación.

El proceso lo iniciará el evento *Toca_enviar_recomendación*. A partir de este evento calculará a partir de los diferentes usuarios, Productos, Rating y Recomendaciones de otros usuarios la publicidad para el usuario. Una vez haya construido esta publicidad, enviará la acción *Enviar_Recomendación*.

Ag Venta productos

Este agente se dedica a gestionar todo el proceso íntegro de compra de un producto. Conocerá los conceptos de Producto, Pedido, Vendedor y Factura_de_compra.

El proceso se lanzará a partir de una petición externa de *Nueva_Compra_producto*. El sistema procesará el pedido y en caso de que esté bien definido enviará una acción de *Confirmación_Cesta*. Al mismo tiempo, enviará un mensaje interno al agente logístico de *Solicitud_envio* con tal de procesar la compra efectuada.

Entonces, cuando el agente reciba la *Confirmación_envio* o *Confirmación_envio_externo*, enviará una acción de *Cobrar_pedido*. Si el producto enviado es del sistema, emitirá una factura con la acción *Emitir_factura*. En caso contrario, hará la acción *Ingresar_dinero_vendedor_externo*.

Ag Almacén

Este agente se dedica a negociar con los transportistas el mejor precio posible de envío.

Conoce los conceptos de Almacén, Lote, Pedidos, Productos y Transportista. El concepto lote le permitirá agrupar diferentes pedidos geolocalizados a fin de realizar menos envíos.

Su plan de trabajo se lanzará cuando reciba un mensaje interno de *Realiza_envio* con su información asociada. Entonces comenzará el proceso de negocio de precios con los transportistas externos enviando acciones de *Pedir_precios_envio* y recibiendo en respuesta *Precios_envio*. Esta

respuesta de Precios_envio tendrá asociada una Oferta_transportista. Se decidirá entonces, qué transportista tiene mejor precio y se lanzará el segundo plan gracias al Avisa_transportista_envío.

Al final del proceso se enviará un Confirmación_envío al agente Venta Productos para actualizar el stock.

Ag Logístico

Este agente se dedica a gestionar la logística del sistema.

Conocerá los conceptos de Pedido, Producto, Vendedor, Centro_Logístico y GeoCoordinates (schema.org)

Se pondrá en marcha gracias a un mensaje interno de Solicitud_envio que llegará del agente Venta Productos. Entonces en función del tipo de producto realizará dos acciones diferentes. Si el producto es de un vendedor externo, se ejecutará el segundo plan de Comunicar envío externo. Sino, se enviará la acción un mensaje Realiza_envio para que el agente Almacén pueda realizar su tarea.

En el plan de comunicar envío externo, por una parte enviara un mensaje interno de Confirmacion_envio_externo al Agente Venta productos para actualizar el stock y por otra, realizará la acción de Comunicar_pedido_enviar.

Ag Gestor de productos externos

Este agente se dedica a guardar en memoria los productos que los vendedores externos quieren poner en nuestro sistema.

Conocerá los conceptos de Producto_Externo y Vendedor.

El agente se pondrá en marcha cuando reciba una nueva percepción de Añadir_producto_externo. Entonces confirmará los campos requeridos y actualizará el stock del sistema. Finalmente enviará una acción al vendedor de Confirmación_producto_externo_añadido para confirmar que su producto ha sido puesto en la tienda con éxito.

Ag Devoluciones

Este agente gestiona todas las devoluciones del sistema.

El Agente debe conocer los conceptos de Devolución, Pedido, Razón_devolución y Producto.

Se lanzará a partir de una percepción de Peticion_devolución. Entonces analizará la razón que lleva asociada y en función del resultado de ésta se efectuará la acción de

Comunicación_resultado_devolucion. Cuando el usuario envíe el paquete y el sistema la reciba, obtendrá la acción de Producto_devuelto. Entonces, desarrollará la acción de Devolver_dinero_cliente.

Ag Mostrar Productos

Este agente se limita a hacer queries de productos que puedan interesar a los clientes.

Debe conocer los conceptos de Producto, Categoría, Persona (que representa el cliente) y las Restricciones_cliente, que le servirán al sistema para filtrar los productos.

Recibirá la percepción de Peticion_productos_disponibles con las Restricciones_cliente asociadas y realizará un filtrado de los productos según estas restricciones. Entonces, enviará la percepción de Informacion_productos_disponibles.

4. Implementación del sistema

Por ahora, el proceso de implementación se ha centrado en entender en profundidad los ejemplos de Python y la estructuración del ejemplo proporcionado por el profesorado, incluyendo asimismo el análisis y el desarrollo de una ontología que se verá posteriormente aplicada en el futuro sistema con Python. Se ha llevado a cabo un estudio y comprensión de los conceptos necesarios para el sistema.