Informe sprint 2

Proyectos IV: Equipo BI

Proyecto: ERP

Juan Carlos Ávila* Javier Coque† Alejandro García‡ Chantal López§

9, marzo 2024

1 Resumen Sprint 2

Este Sprint 2 constaba de tres **historias** principales:

- **Historia 1**: Como comprador quiero poder realizar una compra mediante un botón de compra en Odoo.
- **Historia 2**:
  - a) Como **vendedor** me gustaría poder cancelar un pedido.
  - b) Como **comprador** me gustaría poder cancelar un pedido.
- **Historia 3**: Como usuario me gustaría poder visualizar las ventas de mis compras y pedidos.

*juan.avila@live.u-tad.com

†javier.coque@live-u.tad.com

‡alejandro.gallego@live.u-tad.com

§chantal.lopez@live.u-tad.com

1.1 Historia 1: Yo como comprador quiero poder realizar una compra mediante un botón de compra en Odoo

Esta historia se ha dividido en 8 tares de las cuales se han hecho 4 y no ha dado tiempo a hacer 2.

1.1.1 Hechas

Las dos tareas que se han realizado durante este Sprint 2 son las de:

- **Generar un botón de compra**: El objetivo de esta tarea era el de poder crear un botón de compra en el módulo de `Compras` para permitir al usuario poder realizar esta acción en dicho módulo.
- **Crear fecha de entrega**: Otra de las cosas que se ha conseguido hacer este Sprint 2 es la tarea de, una vez realizada una compra. crear un campo **Fecha de entrega** en la que se indique al cliente que acaba de realizar la compra, la fecha esperada de entrega.

> **Nota**
>
> Se espera para el próximo Sprint implementar alguna lógica a esta fecha de entrega. Algunas ideas que se barajan por el momento son las de basar esta fecha de entrega en base al precio de la compra o del los productos en stock. No está decidido aún.

- **Desencolar una petición de producto.**
- **Encolar una petición de confirmación de pedido.**
- **Desencolar la confirmar del pedido.**
- **Encolar una petición de producto.**

Como se comentó en la auditoría del Sprint 1, se ha decidido utilizar como sistema de mensajería Kafka (en vez de RabbitMQ, que era la otra opción).

El objetivo de estas tareas es que, al pulsar un botón en Odoo—realizar un pedido por parte del cliente o confirmar el pedido por parte del vendedor respectivamente—, esta petición se encolase en un **Topic** de Kafka por parte del *Producer*, para que posteriormente pudiera ser desencolado por el *Consumer*.

Un problema que se ha encontrado a la hora de realizar estas tareas, es que, en un primer momento, no se conseguía establecer una comunicación entre los contenedores de Odoo y de Kafka dentro de la misma *network* en Docker.

Finalmente, después de investigar, se encontró el problema y se pudo solucionar. Esta solución pasó por añadir una regla de *Docker-Compose* para que habilitara el puerto local.

Se deja una gráfica que muestra visualmente la arquitectura diseñada de Docker que hay por detrás para habilitar encolar y desencolar en Kafka. Véase figura 1.

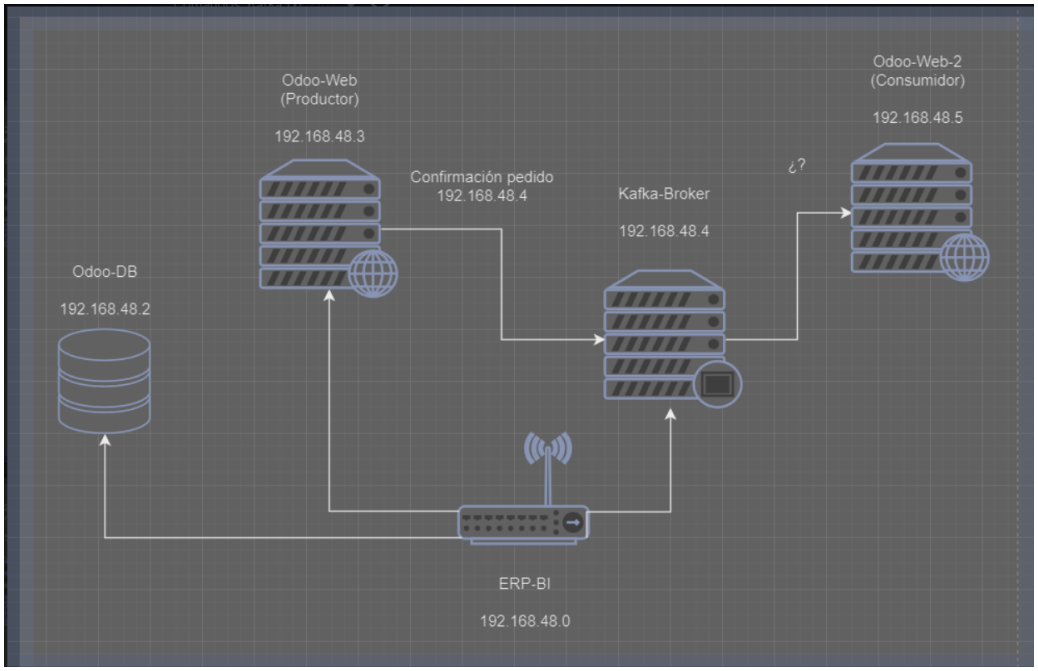

Figure 1: Subred Docker.

> **Nota**
>
> Se deja al final de este documento cómo hay que modificar esta red en los siguientes Sprints (figura 2).

1.1.2 No ha dado tiempo

De esta historia, han quedado dos tareas por hacer. Estas tareas son:

- **Actualizar el estado del pedido**: Esta tarea tenía como objetivo poder mostrar al cliente el estado de su pedido. En principio, se había pensado que este estado podía ser:
  1. Pedido pendiente de confirmar.
  2. Pedido confirmado.
  3. Pedido Enviado.
  4. En espera de recogida.

5. Entregado

- **Enviar un mensaje de confirmación al vendedor**

1.2 Historia 2_a: Como comprador me gustaría cancelar un pedido

Esta historia se ha dividido en 4 tareas: 3 que he han hecho y 1 que no ha dado tiempo.

1.2.1 Hechas

Entre las tareas que se han hecho, se encuentran:

- **Botón desde el módulo de Compras a Pedidos**: Crear un botón para desde la sección de `Compras` poder ser redirigidos—nosotros como clientes— a la sección de `Ventas`. Esta es una tarea previa y necesaria para la siguiente.
- **Botón para cancelar un pedido**: Crear un botón desde la sección de `Ventas` para poder cancelar un pedido realizado.

1.2.2 No ha dado tiempo

De esta historia, ha habido únicamente 1 tarea que no ha dado tiempo a completar. Esta tarea se la de:

- **Cancelación de pedido en un rango de tiempo**: La idea que habíamos tenido es la de que el comprado pudiera cancelar un pedido en un rango de tiempo determinado, e.g. en las primeras 24h. Para el próximo Sprint se baraja la idea de cambiar la condición de tiempo por la del estado de producto, e.g. si el producto tiene un estado de *enviado* o *en espera de recogida*, entonces que no se pueda cancelar el producto.

1.3 Historia 2_b: Como vendedor me gustaría cancelar un pedido

Las tareas de la historia anterior son las mismas que esta con una pequeña modificación, pues ahora es desde el punto de vista del vendedor: la cancelación de pedido la realiza el vendedor con un criterio no establecido. En palabras más coloquiales, el vendedor puede cancelar un pedido *si le da la gana*.

1.3.1 No ha dado tiempo

- **Cancelación de pedido por parte del vendedor**

1.4 Historia 3: Como vendedor me gustaría poder visualizar las ventas de mis pedidos

Para esta historia, nuestro equipo se había comprometido a crear un módulo para poder mostrar un Dashboard BI. El objetivo era el de mostrar, al menos, un par de gráficas muy sencillas. En el próximo Sprint se investigará más a fondo cómo añadir más gráficas y perfeccionar las ya existentes.

1.4.1 Hechas

- **Dashboard de compras**: El objetivo de esta tarea era que el usuario pudiera ver distintos gráficas en las que se pudiera visualizar información acerca de sus **compras**.
- **Dashboard de ventas**: El objetivo de esta tarea era que el usuario pudiera ver distintos gráficas en las que se pudiera visualizar información acerca de sus **ventas**.

Algunos de los problemas que se han encontrado en ese apartado son los siguientes:

- Crear una vista que pueda contener más de una gráfica.
- Generar una gráfica específica. Es decir, en vez de mostrar una única gráfica de un tipo (barras, líneas, pieChart), se muestra la de barras por defecto, dejando al usuario libertad para elegir cualquier otro tipo de gráfica

Para el primer problema, lo que se ha

Para el primer problema, la solución que se ha pensado—y llevado a cabo— es la de generar dos submenús (una para compra y otro para venta) y dentro de estos submenús crear una página distinta para cada gráfica. Indicando al usuario, por supuesto, qué página (vista) muestra qué gráfica.

El segundo problema, en realidad, se puede ver como una ventaja, pues el usuario tiene más libertad acerca de en qué grafica(s) quiere visualizar unos datos determinados.

2 Ideas para siguiente Sprint

- Añadir una confirmación al botón de confirmar pedido.
- IP estática—i.e. la misma para todos— en el *Docker-Compose* para la red de Docker. De esta manera, la conexión entre los contenedor dentro de la red está homogeneizada.

- Computar el campo de Fecha de entrega en base a una lógica.
- Encriptar la cola de mensajería de Kafka.
- Terminar de desencolar los pedidos en Kafka. En otras palabras, se tiene que ver cómo convertir el Odoo_web2 en un *Kafka Consumer*.
- Establecer una autenticación única para que el Productor y Consumidor de Kafka sean los únicos habilitados para leer los mensajes de sus respectivos Tópicos asignados.
- Añadir gráficas más representativas al módulo de Gráficas.

Como se comentó con anterioridad, se deja una imagen de la subred de Docker que se espera tener en el próximo Sprint.

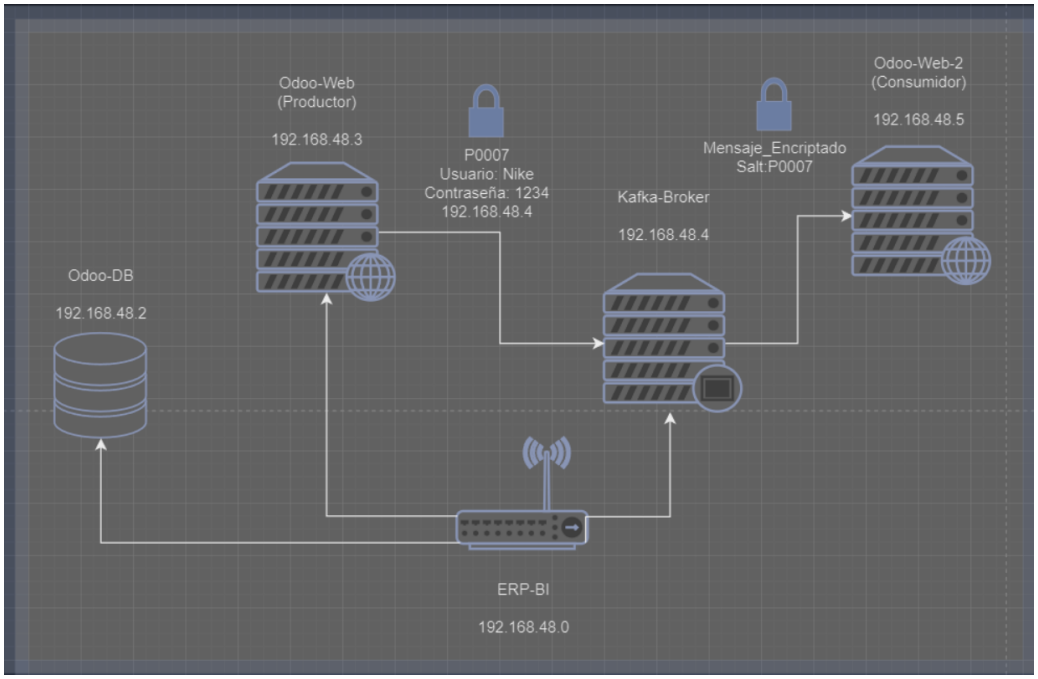

Figure 2: Subred Docker