UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

DOCUMENTO DE REQUERIMIENTOS

Fecha: 14/8

Versión: 1.0

Integrantes:

Legajo	Nombre	E-mail
163.784-8	Fiamma Coedo	coedo.fiamma@gmail.com
163.237-1	Karen Manrique	kmanrique.utn@gmail.com
157.159-0	Julian Schiffer	schifferjulian@gmail.com

ÍNDICE

Documento de Requerimientos

Este documento tiene como objetivo determinar los requerimientos funcionales y no funcionales del sistema, también las restricciones del mismo.

Requerimientos Funcionales

A continuación se listan los requerimientos funcionales del sistema:

- Desarrollar un website de ventas propio
- Integración con brokers de pago
- Integración con AFIP
- Integración con los sistemas de empresa compradora
- Manejo de stock digital
- Realizar Informe diario

Desarrollar un website de ventas propio

Se deberá desarrollar un website de ventas propio desde cero. El mismo se integrará con Mercado Libre, así como también se incorporará una sección para la administración de las ventas en el local. El website va a ser desarrollado utilizando React TypeScript.

Integrar con brokers de pago

Esta integración consta de habilitar múltiples medios de pago, no solamente ventas online; aumentando el alcance de nuestra PyME.
El sistema se integrará con los siguientes brokers de pago:

- Mercado Pago
- PayPal
- Visa
- MasterCard
- American Express

Integrar con AFIP

Dicha integración tendrá la finalidad de realizar múltiples operaciones, el sistema se integrará con los módulos de:

- facturación electrónica,
- constatación de comprobantes.

Integrar con sistemas de empresas compradoras

Una finalidad de este requerimiento es proveer a nuestra empresa compradora herramientas de auditoría, ya sea para el control de stock, de producción, de compras y de ventas. Estos datos van a estar disponibles a través de una API que la empresa compradora va a poder utilizar para consumir los datos.

Manejar stock digital

Se administrará todo el stock que posee la empresa, permitiendo realizar cambios al inventario actual de forma manual y a su vez administrando el stock disponible para ventas online y ventas en los portales externos integrados.

Realizar informe diario

Se emitirá un informe diario con toda la información de la producción, de las ventas y de las compras realizadas. Este informe se va a enviar diariamente por correo electrónico a las personas que corresponda. A su vez, esta información va a estar disponible a través de una API para consumos futuros y realización de auditorías.

Requerimientos No Funcionales

A continuación se listan los requerimientos no funcionales del sistema:

- Utilizar SUSE linux (deseable)
- Operar en múltiples plataformas digitales
- Soportar 20.000 transacciones por hora
- Alta disponibilidad de la solución
- Asegurar un RTO maximo de 1 hs
- Asegurar un RPO igual a 0

Utilizar SUSE Linux *(deseable)*

Se mantendrá como sistema operativo SUSE Linux, ya que la solución es soportada en cualquier versión de Linux dado que se utilizará Docker para la misma. De esta forma, si se desea cambiar de sistema operativo, no va a requerir ningún cambio en la solución propuesta.

Operar en múltiples plataformas digitales

Se garantizará al cliente accesibilidad, obteniendo desde cualquier dispositivo la misma experiencia. Para esto se va a utilizar un equipo de UI/UX que garantice la misma experiencia de usuario para todos los dispositivos móviles a continuación: Teléfonos/tablets Android, teléfonos/tablets iOS.

Soportar 20.000 transacciones por hora

Se espera un pico de 10.000 ventas por hora, conllevando la necesidad de manejar un promedio de 20.000 transacciones por hora en total para que el sistema no colapse.

Alta disponibilidad de la solución

El sistema deberá garantizar una alta disponibilidad, superior al 99%. Para alcanzar dicha disponibilidad, se contará con cuatro servidores para toda la carga del sistema, en caso de que uno esté muy saturado o sufra una caída, un balanceador de carga va a redirigir todas las peticiones al otro servidor. Dos de los servidores estarán en otro data center para evitar caídas en caso de que el edificio que actualmente tiene los servidores sufra un siniestro.

Asegurar un RTO máximo de 1hs

El sistema deberá garantizar un tiempo objetivo de recuperación de máximo 1 hora ante una posible caída del mismo. En caso de que el servidor sufra un daño físico, el balanceador va a repartir la carga en los servidores que estén en otro data center.

Asegurar un RPO igual a 0

El sistema deberá garantizar que, en caso de fallar el sistema, no se perderán datos. Para evitar pérdida de datos producto de una caída, se va a contar con 2 bases de datos espejadas (formando un RAID 1), donde una de las bases de datos va a estar en el data center actual y la otra va a estar en la nube Microsoft Azure.

Restricciones

A continuación se listan las restricciones del sistema:

- Utilizar stack tecnológico con amplia comunidad de uso
- Mantener los datos del pasado
- Utilizar estándares de mercado para la seguridad

Utilizar stack tecnológico con amplia comunidad de uso

Se utilizará un stack tecnológico que cuente con una amplia comunidad que lo utilice, tanto desarrolladores como consumidores finales. El stack tecnológico elegido es TypeScript dado que va a permitir mantener un único lenguaje tanto para frontend como para backend.

Mantener los datos del pasado

La nueva solución mantendrá intactos los datos del pasado, permitiendo visualizar cualquier transacción anterior.

Utilizar estándares de mercado para la seguridad

Para todas las solicitudes que se hagan a las API's, se va a exigir tokens para la autenticación y se van a enmascarar dichos tokens utilizando servidores internos para no exponer la información. Además, para poder ingresar al backoffice, se deberá autenticar utilizando la VPN con la que cuentan hoy en día.