

Universidad Modelo



Escuela de Ingeniería

Ingeniería en desarrollo de tecnología y software

8vo Semestre

Proyectos VIII

Implementación de MVP

Enero - Junio 2024

Alumnos:

José Roberto Gamboa Rivero
Andrés Enrique Ortega Alcalde
Juan Ángel Franco Heredia

Maestra:

Ivonne Emire Cervera Cetina

Descripción del MVP (Minimum Viable Product)

El MVP de nuestro proyecto de e-commerce para la venta de utensilios de cocina se centrará en ofrecer una experiencia básica pero funcional tanto para los clientes como para la gestión interna del negocio. Incluirá las siguientes características principales:

Catálogo de productos

Una interfaz de usuario que permita a los clientes explorar y buscar los utensilios de cocina disponibles para la compra. Esta sección incluirá imágenes de alta calidad, descripciones detalladas de los productos y opciones de filtrado para facilitar la navegación.

Carrito de compras

Funcionalidad básica para que los clientes agreguen productos al carrito, ajusten las cantidades y eliminen elementos si es necesario. Esta característica permitirá a los usuarios revisar su selección antes de proceder al pago.

Proceso de pago

Integración con una pasarela de pago segura que admita diferentes métodos de pago, como tarjetas de crédito, débito y transferencias bancarias. Este proceso deberá ser rápido y sencillo para garantizar una experiencia de compra fluida.

Gestión de pedidos

Una interfaz de administración que permita al equipo interno del negocio ver y gestionar los pedidos recibidos. Esto incluirá la capacidad de actualizar el estado del pedido, generar facturas y enviar confirmaciones de envío a los clientes.

Integración con Odoo y PrestaShop

Utilización de las funcionalidades proporcionadas por Odoo y PrestaShop para la gestión de inventario, seguimiento de ventas, administración de clientes y otras tareas administrativas clave. Esta integración garantizará una gestión eficiente de los procesos internos del negocio.

Objetivos del MVP

Validación de la viabilidad del producto

- **Tasa de conversión:** Alcanzar una tasa de conversión del 2-3% dentro del primer mes de lanzamiento del MVP.
- **Valor promedio del pedido:** Lograr un valor promedio del pedido de al menos \$50.
- **Ingresos generados:** Generar ingresos mensuales de al menos \$5,000.

Recopilación de retroalimentación temprana de los usuarios

- ***Tasa de respuesta en encuestas:*** Obtener una tasa de respuesta del 10% en las encuestas de satisfacción del cliente.
- ***Comentarios en el sitio web:*** Recibir al menos 20 comentarios directos en la sección de comentarios del sitio web durante el primer mes de lanzamiento.
- ***Ánalysis de la experiencia del usuario:*** Actuar sobre el 80% de los puntos destacados en el análisis de la experiencia del usuario dentro de las primeras dos semanas de lanzamiento.

Reducción del tiempo y costo de desarrollo

- ***Lanzamiento oportuno:*** Lanzar el MVP dentro del plazo establecido inicialmente, sin retrasos significativos.
- ***Utilización de funcionalidades existentes:*** Utilizar al menos el 80% de las funcionalidades proporcionadas por Odoo y PrestaShop.
- ***Iteraciones del producto:*** Implementar al menos dos iteraciones significativas del producto dentro de los primeros tres meses de lanzamiento.

Diseño final del proyecto

Visión general del diseño

Descripción del diseño final del proyecto

Componentes Principales

API en Node.js con TypeScript

- Desarrollamos una API utilizando Node.js con TypeScript. Esta API actúa como el punto de entrada para la comunicación y la lógica del negocio.
- Utilizamos TypeScript para agregar tipado estático a nuestro código JavaScript, lo que mejora la mantenibilidad y la escalabilidad del proyecto.

Instancia Docker con Odoo

- Odoo está alojado en una instancia Docker. Odoo es un sistema de gestión empresarial de código abierto que proporciona un conjunto de aplicaciones empresariales. Al estar en Docker, simplifica la gestión y la escalabilidad del entorno de Odoo.

Conexión con Base de Datos Local(MySQL)

- Hemos establecido una base de datos local para la aplicación. Esta base de datos se utiliza para almacenar datos específicos del proyecto y se conecta tanto con Odoo como con PrestaShop.

Comunicación entre Odoo y PrestaShop

- La instancia de Odoo se comunica con PrestaShop, que también está en una instancia Docker separada.
- Esta comunicación puede ser gestionada a través de la API que hemos desarrollado en Node.js. La API actúa como un intermediario para intercambiar datos entre Odoo y PrestaShop.

Arquitectura del software

Cliente

- El cliente interactúa con nuestra API desarrollada en Node.js para realizar diversas acciones, como consultas, actualizaciones o inserciones de datos.

API (Node.js)

- La API actúa como una capa intermedia entre el cliente y otras aplicaciones como Odoo y PrestaShop.
- Se encarga de manejar las solicitudes del cliente, procesar la lógica del negocio y realizar llamadas a Odoo y PrestaShop según sea necesario.

Odoo (Docker)

- Odoo proporciona funcionalidades de gestión empresarial y es utilizado para ciertas operaciones dentro de tu sistema.
- La API en Node.js puede interactuar con Odoo para leer datos, escribir en la base de datos de Odoo o realizar otras operaciones relacionadas con la gestión empresarial.

PrestaShop (Docker)

- PrestaShop es una plataforma de comercio electrónico que puede integrarse con Odoo.
- La API en Node.js puede facilitar la sincronización de datos entre PrestaShop y Odoo, lo que permite mantener actualizada la información entre ambos sistemas.

Base de Datos Local(MySQL)

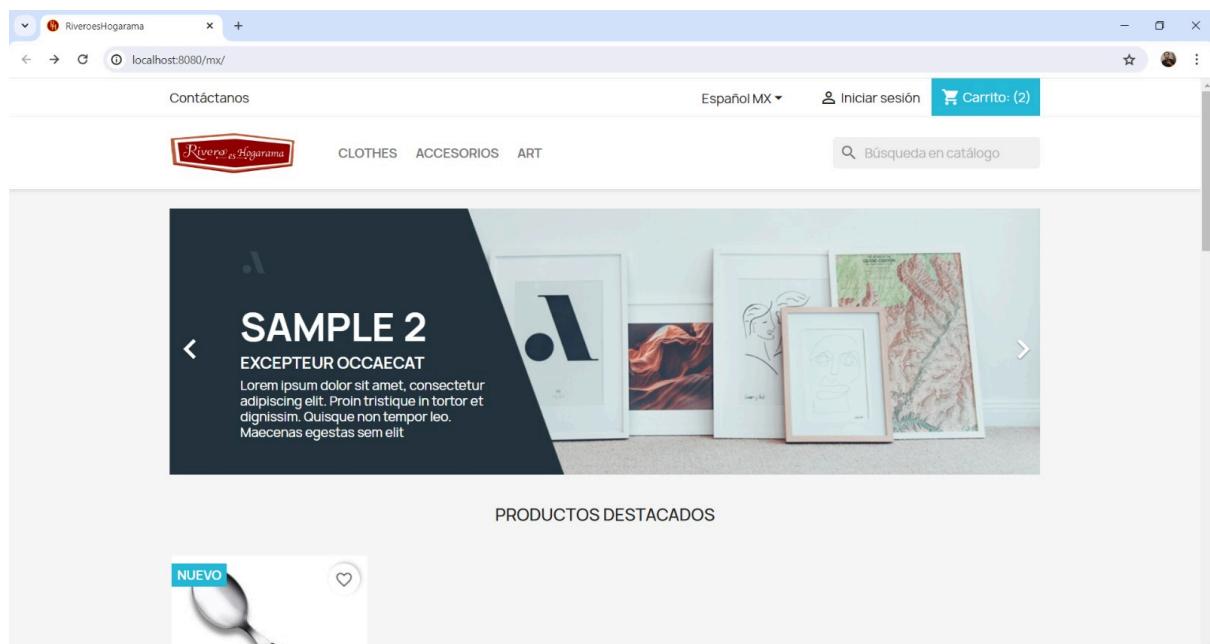
- Se utiliza una base de datos local para almacenar datos específicos del proyecto y para actuar como un punto central de comunicación entre Odoo, PrestaShop y la API en Node.js.
- Esta base de datos local puede contener información relevante para la operación del sistema, como registros de transacciones, configuraciones específicas del proyecto, etc.

Flujo de Datos

1. El cliente realiza una solicitud a la API en Node.js.
2. La API procesa la solicitud y puede interactuar con la base de datos local para recuperar o almacenar datos.
3. Si es necesario, la API realiza llamadas a Odoo para realizar operaciones empresariales específicas.
4. La API también puede comunicarse con PrestaShop para sincronizar datos relacionados con el comercio electrónico.
5. La API responde al cliente con la información solicitada o con el resultado de la operación realizada.

Interfaz de usuario

Como diseño final tenemos todo el prototipo del apartado ECommerce de usuario de manera estética y estructurada de la siguiente manera.



Pantalla de inicio

Esta pantalla es donde el usuario encontrará los productos más vendidos o que estén en promoción en la tienda en línea. Además de tener un buscador y divididos de manera categórica.

Inicio / Cuchara de Acero con Mango Negro

NUEVO

CUCHARA DE ACERO CON MANGO NEGRO

\$11.60

IVA incluido

Cantidad

1 **AÑADIR AL CARRITO**

Compartir

Security policy (edit with the Customer Reassurance module)

Delivery policy (edit with the Customer Reassurance module)

Return policy (edit with the Customer Reassurance module)

Detalles del producto

En existencia 11 Artículos

Vista del producto

En esta vista el usuario va a poder visualizar el producto que desee con el nombre del producto, su precio, una imagen y el stock disponible. Y podrá añadirlo al carrito.

Carrito: [localhost:8080/mx/carrito?action=show](#)

Contáctanos [Iniciar sesión](#) Carrito: (2)

Español MX

Rivero & Hamama CLOTHES ACCESORIOS ART Búsqueda en catálogo

CARRITO

	Cuchara de Acero con Mango Negro \$11.60	2 <input type="button" value="^"/> <input type="button" value="▼"/>	\$23.20
--	---	---	---------

Seguir comprando

2 artículos \$23.20
Envío \$8.12
Total (IVA Incluido) \$31.32

Security policy (edit with the Customer Reassurance module)
 Delivery policy (edit with the Customer Reassurance module)
 Return policy (edit with the Customer Reassurance module)

IR A LA CAJA

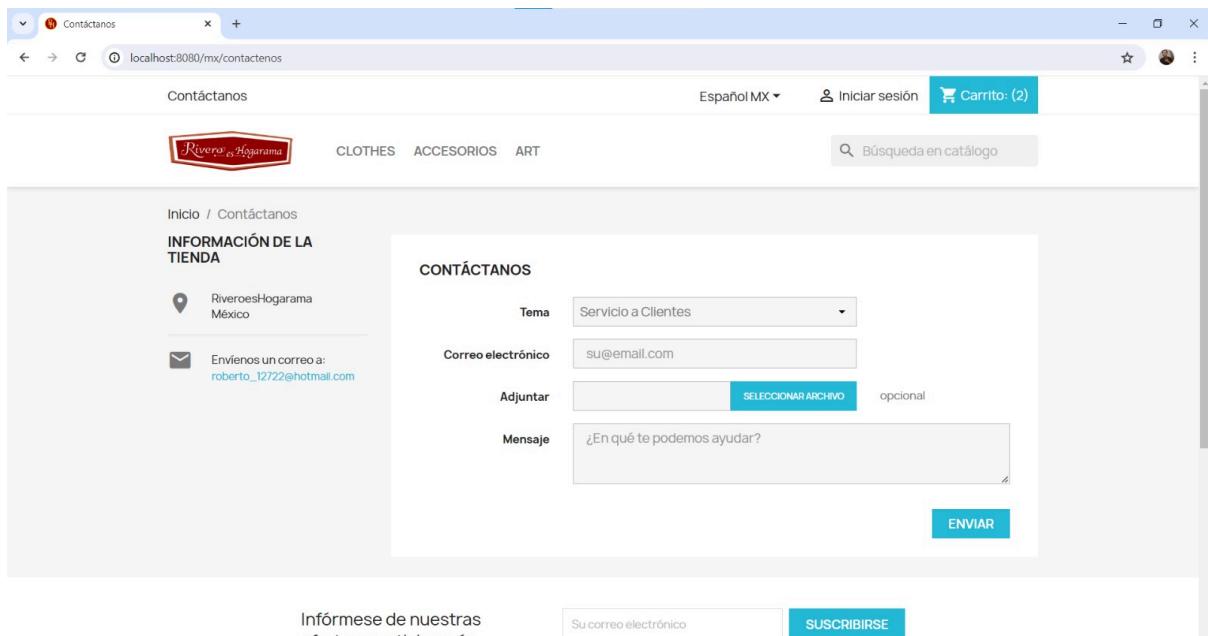
Infórmese de nuestras

[Suscribirse](#)

[SUSCRIBIRSE](#)

Vista del carrito

En esta vista el usuario va a ver su carrito de compras el cual le va a mostrar cuales son los productos que agarró su precio, el envío, que producto es el que agarro y va a poder seguir las siguientes acciones para poder hacer la compra.



Vista para contactarnos

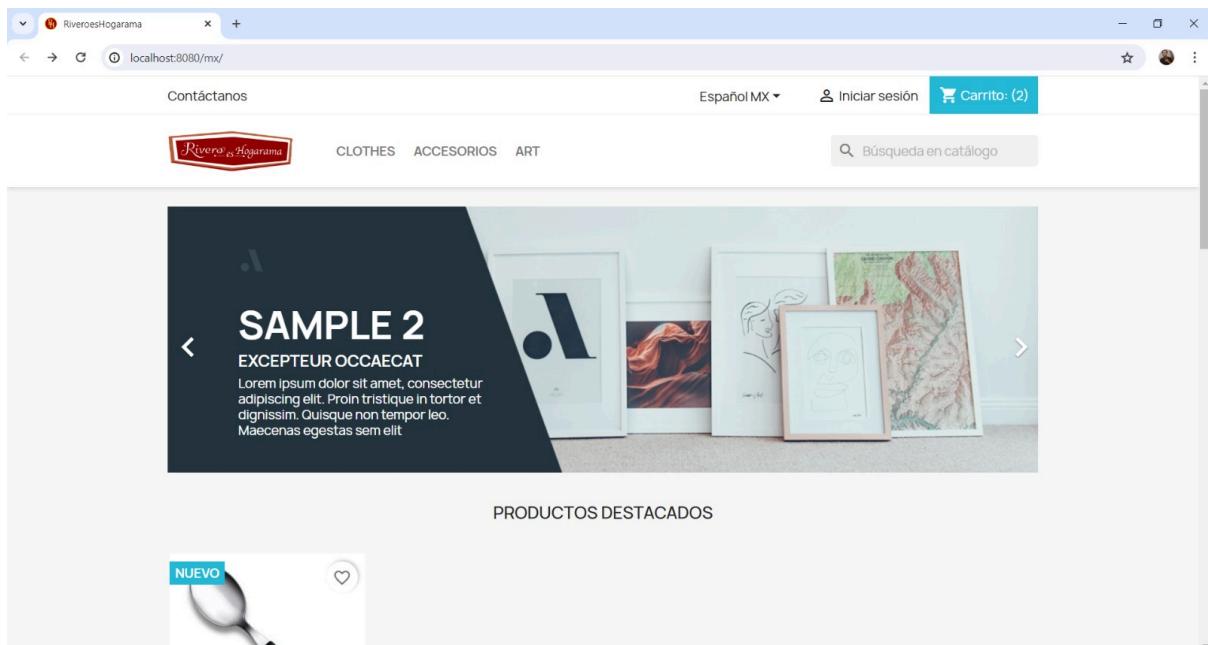
Esta vista será para que los clientes puedan contactarnos y enviarnos peticiones directamente a nuestro correo electrónico.

Iteraciones y mejoras

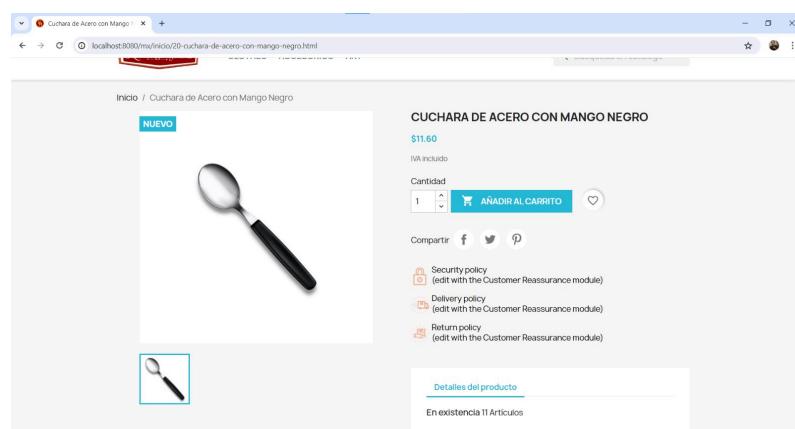
Hasta ahorita los únicos cambios relevantes que hemos hecho durante todo el prototipado del Ecommerce es la implementación de un tipo CRUD para el área administrativa que es la creación de sus productos, la actualización de sus productos y la eliminación de sus productos que no es necesariamente eliminar sino más bien inhabilitarlos para que no se hagan mas compras, esto viene hacienda más de la api y en un futuro lo haremos más un interfaz para el admin por el momento se hace mediante una llamada al postman sobre nuestra api.

Con base a la interfaz de usuario se hicieron modificaciones en los logos, en el banner, el nombre de la empresa que es Riveroeshogarama, la agregación de productos del establecimiento al Ecommerce entre otras cosas que enseñamos anteriormente. Pero lo agregaremos aquí para que sean visualizadas.

En la primera iteración lo que trabajamos fueron en los logos y sus diseños que algunos fueron proporcionados por la empresa y otros los hacemos y enseñamos al cliente para darnos el visto bueno o malo. Además de poder ver la agregación de un producto que era la primera iteración la agregación de productos a prestashop y que se guardaran en Odoo y Prestashop además de guardarla en la base de datos local en MySQL.



Podemos apreciar que crea el producto dentro de nuestra tienda en línea todo mediante postman haciendo llamados a la api.



Nuestra segunda iteración fue nuestra actualización de productos mediante llamadas a nuestra api mediante el postman. Lo que hará es actualizarlo en la base de datos local, odoo y prestashop.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- compania
- db
- Jardineria
- odo0
- petcare
- riveroeshogarama
- sakila
- sys
- tienda
- universidad
- ventas
- world

Administration Schemas

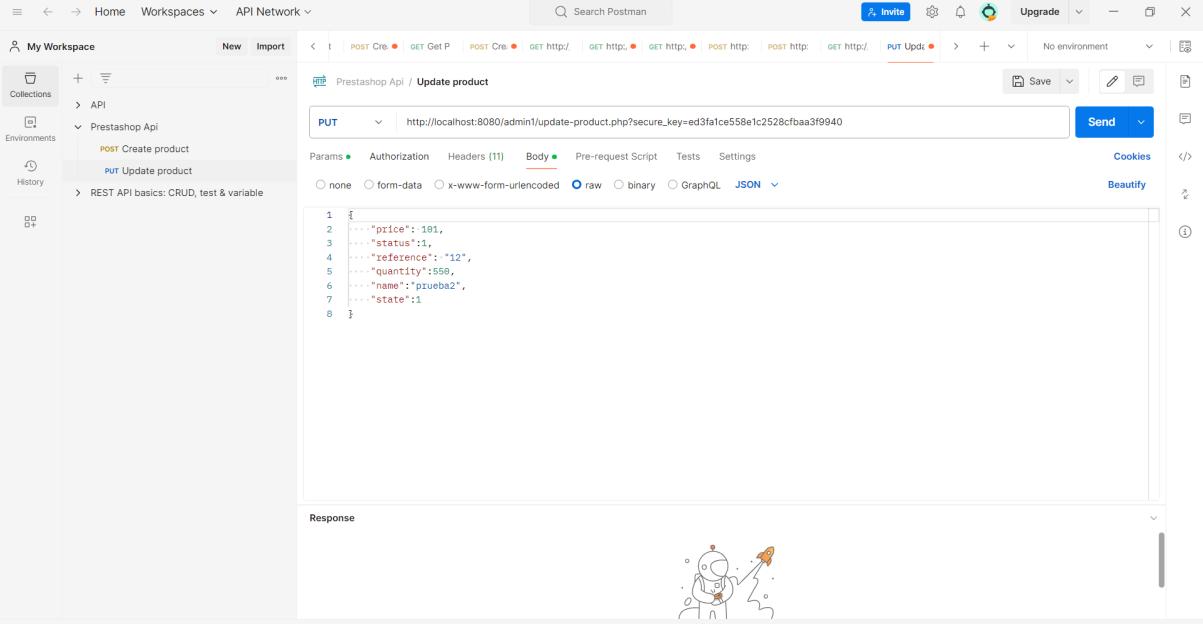
Information

Table: producto

Columns:

codig0	int PK	descripc0n	varchar(300)	preco	varchar(45)	stock	varchar(45)
id	name	description	price	stock			
1	roberto	es gay	12.00	2			
2	Add your name in the body	sss	2.00	2			
3	Add your name in the body	sss	2.00	2			
4	Add your name in the body	sss	2.00	2			
5	Add your name in the body	sss	2.00	2			
6	Cuchillo de sierra	Bueno para cortes de carne	22.00	10			
7	Tenedor treviso	Cubrieto de media calidad	15.00	10			
8	salchicha	licencia de docker	29.00	5			
9	salchichon	licencia de moto	456.00	7			
10	salchichonito	licencia de moto	456.00	7			
11	salchichontoss	licencia de moto	456.00	7			
12	prueba	prueba	456.00	7			
#	Time	Action	Message	Duration / Fetch			
1	15:23:47	SELECT * FROM riveroeshogarama.producto LIMIT 0, 1000	8 row(s) returned	0.015 sec / 0.000 sec			
2	15:23:55	SELECT * FROM odo0.product LIMIT 0, 1000	12 row(s) returned	0.016 sec / 0.000 sec			

Estas son nuestras llamadas de postman que son de creación y actualización por el momento, por lo cual nuestra última iteración sería realizar la de eliminar o inhabilitar el producto.



The screenshot shows the Postman application interface. The left sidebar shows 'My Workspace' with a 'Collections' section containing 'Prestashop Api' which has 'Create product' and 'Update product' items. The main area shows a 'PUT' request to 'http://localhost:8080/admin1/update-product.php?secure_key=ed3fa1ce558e1c2528cfbaa3f9940'. The 'Body' tab is selected, showing a JSON payload:

```
1  {
2  ... "price": 101,
3  ... "status":1,
4  ... "reference": "12",
5  ... "quantity":559,
6  ... "name": "prueba2",
7  ... "state":1
8 }
```

The 'Response' tab shows a cartoon character with a rocket ship.

Todo basado en los requisitos del usuario, solo falta dejar un poco más estética la página pero en cuanto a las funcionalidades del cliente todo va bien.

Diseño de pruebas Estrategia de pruebas

Pruebas de Unidad

Herramientas Utilizadas

Se han utilizado frameworks de pruebas como Jest (para JavaScript/TypeScript) para escribir y ejecutar pruebas de unidad.

Hicimos las siguientes pruebas unitarias:

- Pruebas para validar el comportamiento de funciones específicas en la API.
- Pruebas para asegurar que las operaciones de base de datos locales funcionen correctamente.

Pruebas de Integración

Herramientas Utilizadas

Se han realizado pruebas de integración utilizando herramientas como Supertest para la API y Docker Compose para simular la integración entre los contenedores de Odoo y PrestaShop.

Hicimos las siguientes pruebas de integración:

- Verificación de la comunicación correcta entre la API y la base de datos local.
- Validación de la interacción entre la API y Odoo para realizar operaciones empresariales.

Pruebas de Aceptación del Usuario (UAT - User Acceptance Testing)

Herramientas Utilizadas

Se han llevado a cabo pruebas manuales por parte de usuarios representativos o stakeholders del proyecto.

Hicimos las siguientes pruebas de aceptación:

- Usuarios prueban la funcionalidad de la API para realizar acciones como consultar datos o realizar operaciones de sincronización entre Odoo y PrestaShop.
- Validación de la usabilidad y la eficacia del MVP para realizar tareas clave del negocio.

Resultados de las pruebas

Pruebas de Unidad

Resultados

La mayoría de las pruebas de unidad fueron exitosas, validando el comportamiento esperado de las funciones y componentes individuales.

Problemas Identificados

Se detectaron algunos errores de lógica en las funciones de sincronización entre la API y Odoo.

Acciones Correctivas

- Se revisaron y corrigieron las funciones problemáticas.
- Se mejoró la cobertura de pruebas unitarias para garantizar la estabilidad y el correcto funcionamiento de las funcionalidades clave.

Pruebas de Integración

Resultados

La mayoría de las pruebas de integración tuvieron éxito al simular la comunicación entre la API, la base de datos local y los contenedores de Odoo y PrestaShop.

Problemas Identificados

Hubo problemas intermitentes de conexión entre la API y Odoo en entornos de alta carga simulada.

Acciones Correctivas

- Se optimizó el manejo de conexiones y se implementaron mecanismos de reintentos para mejorar la estabilidad de la comunicación.
- Se realizaron ajustes en la configuración de Docker para optimizar el rendimiento y la capacidad de respuesta de los contenedores.

Pruebas de Aceptación del Usuario (UAT)

Resultados

Los usuarios de prueba informaron una experiencia satisfactoria en general con el MVP. Se identificaron algunos puntos de mejora en la usabilidad y la claridad de las funciones.

Problemas Identificados

Algunas acciones de sincronización no fueron intuitivas para los usuarios finales.

Acciones Correctivas

- Se realizaron ajustes en la interfaz de usuario para mejorar la claridad y facilitar la comprensión de las acciones de sincronización.
- Se agregaron mensajes informativos y guías visuales para orientar a los usuarios durante el uso de la aplicación.

Cumplimiento de objetivos

Logros del MVP

Validación del Concepto

- Se logró validar la viabilidad y la utilidad del producto al ofrecer funcionalidades básicas a los usuarios.
- Los comentarios y la retroalimentación recibidos confirmaron la relevancia del producto en resolver necesidades específicas del mercado objetivo.

Adquisición de Usuarios Iniciales

- Se atrajeron y se obtuvieron usuarios iniciales dispuestos a probar y utilizar el MVP.
- La respuesta inicial de los usuarios proporcionó información valiosa sobre cómo se percibe y se utiliza el producto en un entorno real.

Identificación de Áreas de Mejora

- Se identificaron áreas clave de mejora y optimización a través de la retroalimentación de los usuarios y las pruebas realizadas.
- Los problemas y las deficiencias encontradas durante la fase del MVP sirvieron como base para establecer un plan de mejoras y desarrollo continuo.

Métricas de éxito

Tasa de Retención de Usuarios

Métrica

Tasa de retención semanal o mensual.

Objetivo

Lograr una tasa de retención inicial del 30% o superior para validar el interés y la utilidad del producto.

Satisfacción del Cliente (NPS - Net Promoter Score)

Métrica

Calculado mediante encuestas, preguntando a los usuarios "¿Qué tan probable es que recomiendan nuestro producto a un amigo o colega?"

Objetivo

Alcanzar un NPS positivo (por ejemplo, +20 o superior) indicando una buena recepción y percepción del producto entre los usuarios.

Tiempo de Comercialización Reducido

Métrica

Número de días o semanas desde el lanzamiento hasta la primera venta o suscripción.

Objetivo

Reducir el tiempo de comercialización inicial a menos de 3 meses para validar la demanda y la capacidad de generación de ingresos del producto.

Documentación de cierre de proyectos

Documentación final

La documentación detallada del MVP implementado proporciona una descripción clara de las funcionalidades, características y alcance del producto en su versión inicial. Esto permite a futuros equipos entender la base sobre la cual se construyó el producto y las decisiones clave tomadas durante el desarrollo. La documentación incluirá lecciones aprendidas durante el proceso de desarrollo del MVP, destacando lo que funcionó bien y lo que podría mejorarse en futuras iteraciones. Estas lecciones ayudan a identificar oportunidades de mejora y evitar errores o desafíos similares en proyectos futuros. Incluiríremos recomendaciones específicas para futuras iteraciones o versiones del producto basadas en la retroalimentación de los usuarios y los resultados obtenidos. Estas recomendaciones guían el desarrollo continuo del producto, asegurando que se enfoque en las áreas más importantes para los usuarios y el negocio.

Lecciones aprendidas

Éxitos

Enfoque en la Esencialidad del Producto

Aprendizaje

Centrarse en las funcionalidades esenciales del MVP fue clave para ofrecer un producto inicial funcional y valioso para los usuarios.

Impacto

La simplicidad y la utilidad del MVP generaron interés inicial y validaron la viabilidad del concepto.

Rápida Iteración y Feedback Continuo

Aprendizaje

La capacidad de iterar rápidamente en base al feedback de los usuarios fue fundamental para mejorar y ajustar el producto de manera ágil.

Impacto

Se logró mejorar la experiencia del usuario y abordar deficiencias identificadas de manera oportuna.

Flexibilidad y Adaptabilidad del Equipo

Aprendizaje

Mantener un equipo flexible y adaptable facilitó la resolución eficiente de desafíos técnicos y la implementación de cambios según las necesidades del proyecto.

Impacto

Se logró superar obstáculos inesperados y mantener el impulso del desarrollo sin comprometer la calidad del producto.

Desafíos

Gestión de Expectativas y Alcance

Aprendizaje

La gestión efectiva de expectativas y el manejo del alcance del MVP son fundamentales para evitar sobrecargas y garantizar entregas oportunistas.

Impacto

Experimentamos algunos desafíos al equilibrar las expectativas del cliente con los recursos disponibles.

Integración y Comunicación entre Componentes

Aprendizaje

La integración entre diferentes componentes del sistema requiere una planificación cuidadosa y una comunicación clara para garantizar una funcionalidad sin problemas.

Impacto

Identificamos la importancia de las pruebas exhaustivas de integración y la optimización de la comunicación entre la API, Odoo y PrestaShop.

Gestión de Tiempo y Recursos

Aprendizaje

La gestión efectiva del tiempo y los recursos es esencial para cumplir con los plazos y objetivos del proyecto.

Impacto

Experimentamos algunos desafíos relacionados con la distribución óptima de recursos y la priorización de tareas durante la implementación del MVP.