

CENTRO DE LA INDUSTRIA, LA EMPRESA Y LOS
SERVICIOS

Neiva, Huila

GRÁFICO DE MIRO (CARRITO DE COMPRA)

APRENDIZ:

Patricia Del Rocio Sarmiento Ruiz.

Iván Andrés Murcia Epia.

David Mauricio Flórez Quintero.

Julián David Fierro Casanova.

Erick Daniel Peña Cedeño.

Héctor Fabian Cardoso Morales.

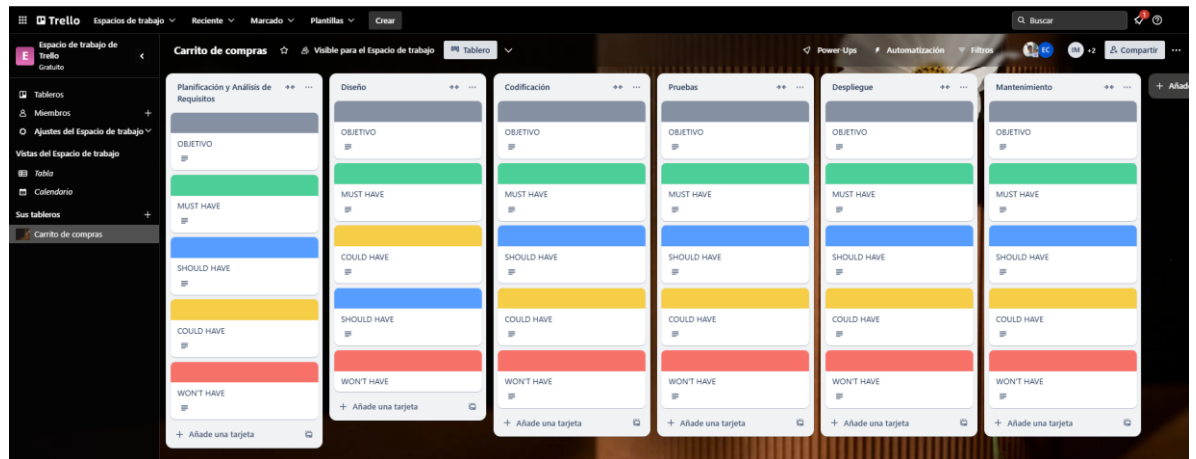
Maria Jose Murcia Martinez

INSTRUCTOR:

Jesús Ariel González

FECHA:

07 de diciembre de 2024



1. Planificación y Análisis de Requisitos:

OBJETIVO: El objetivo es recolectar y priorizar los requisitos del sistema de carrito de compras. En esta fase, se identificarán los requisitos del sistema y se asignarán a las categorías Must Have, Should Have, Could Have y Won't Have.

MUST HAVE:

- **Registro de usuarios:** Los clientes deben poder registrarse en el sistema.
- **Carrito de compras:** Los usuarios deben poder agregar productos al carrito.
- **Proceso de pago:** Debe ser posible realizar pagos de forma segura.
- **Generación de facturas:** El sistema debe generar facturas válidas tras la compra.
- **Seguridad:** El sistema debe proteger la información personal del cliente (uso de HTTPS, encriptación de pagos).
- **Gestión de inventario:** El sistema debe actualizar el inventario cuando un producto se compra.

SHOULD HAVE:

- **Notificación por correo electrónico:** Enviar correos electrónicos de confirmación tras realizar una compra.
- **Historial de compras:** Los usuarios deben poder ver un historial de sus compras anteriores.

- **Soporte para múltiples métodos de pago:** Soporte para tarjetas de crédito, PayPal, transferencias bancarias, etc.
- **Interfaz amigable:** El sistema debería tener una interfaz fácil de usar y amigable.

COULD HAVE:

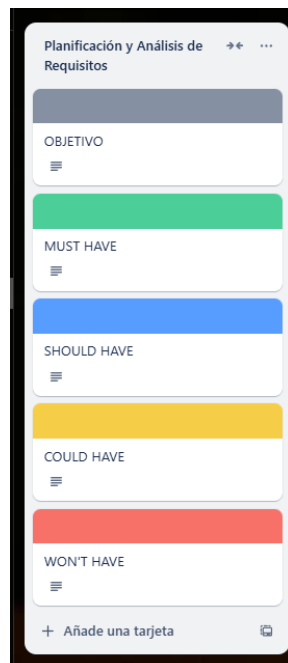
Recomendaciones personalizadas: Basadas en compras anteriores o productos visualizados.

Integración con redes sociales: Los usuarios pueden compartir productos en sus redes sociales.

Funcionalidad de "guardar para después": Los usuarios pueden guardar productos en el carrito para comprarlos más tarde.

WON'T HAVE:

- **Realidad aumentada:** No se incluirá la visualización de productos en 3D.
- **Aplicación móvil en la primera versión:** La aplicación solo estará disponible en la web, no en dispositivos móviles.
- **Función de chat en vivo:** No se implementará un sistema de soporte en vivo en esta fase inicial.



2. DISEÑO:

OBJETIVO: Definir la arquitectura y el diseño del sistema.

En esta fase, se comenzará a definir cómo se estructurará el sistema para cumplir con los requisitos establecidos en la fase anterior.

Arquitectura: El sistema se dividirá en módulos como:

- Módulo de Usuario (autenticación, registro, gestión de cuentas).
- Módulo de Carrito de Compras (gestión de productos, agregar al carrito, eliminar, ver productos).
- Módulo de Facturación (generación de facturas, gestión de pagos).
- Módulo de Inventario (gestión de stock, actualización de inventario).
- Módulo de Seguridad (autenticación, encriptación de datos, protección de pagos).

1. Base de Datos:

- **Usuarios:** Información sobre cuentas, preferencias, historial de compras.
- **Productos:** Datos de productos disponibles en el inventario.
- **Facturas:** Información sobre las transacciones realizadas.

MUST HAVE:

Los módulos Must Have (registro de usuarios, carrito, proceso de pago, facturación) tendrán un diseño detallado primero.

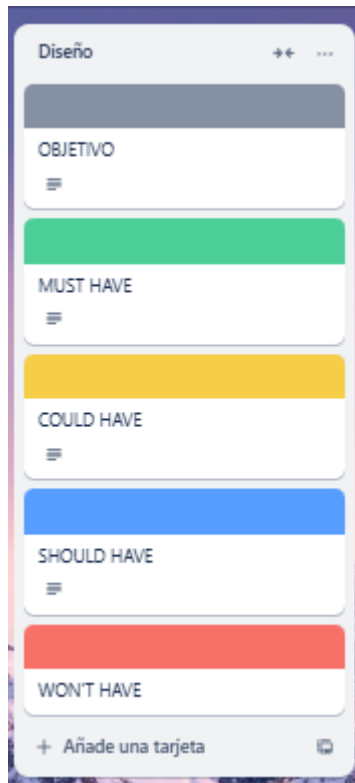
COULD HAVE:

Los módulos Could Have (recomendaciones personalizadas, integración con redes sociales) se desarrollarán más tarde si el proyecto tiene margen para ello.

SHOULD HAVE:

Los módulos Should Have (notificación por correo, historial de compras) se diseñarán si el tiempo lo permite, una vez que los módulos esenciales estén listos.

WON'T HAVE:



3. CODIFICACIÓN:

OBJETIVO:

El objetivo es implementar los requisitos del sistema según el diseño y las prioridades establecidas.

En esta fase, se codifican las funcionalidades del sistema. El equipo de desarrollo se centrará primero en los Must Have.

MUST HAVE:

- Implementación del registro de usuarios.
- Desarrollo del carrito de compras.

- Implementación del proceso de pago seguro.
- Generación de facturas.
- Integración de la gestión de inventario.

SHOULD HAVE:

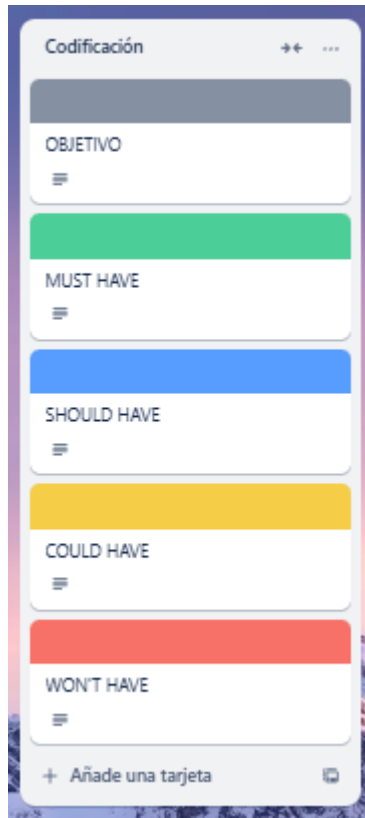
Notificación por correo electrónico.
Historial de compras.

COULD HAVE:

Recomendaciones personalizadas
Integración con redes sociales.

WON'T HAVE:

Los Won't Have (como la realidad aumentada y la app móvil) no se implementarán en esta fase.



4. PRUEBAS:

OBJETIVO:

El objetivo es verificar que el sistema funciona correctamente, cubriendo todos los requisitos Must Have, aunque también hay requisitos should have, could have y don't have.

MUST HAVE:

- Registro de usuarios: ¿El proceso de registro funciona correctamente?
- Proceso de pago: ¿Se realiza correctamente el pago?
- Generación de facturas: ¿Las facturas se generan con la información correcta?
- Gestión de inventario: ¿El inventario se actualiza cuando se hace una compra?
- Realizar pruebas funcionales para validar la experiencia del usuario.

- Registro y cálculo adecuado de pagos y totales.
- Validación de datos en las facturas (evitar campos vacíos o incorrectos).
- ¿Se actualiza correctamente el Stock después de una compra?
- ¿Se genera una alerta cuando los productos de una categoría se agotan?
- ¿Se realiza un inicio de sesión con credenciales válidas?

SHOULD HAVE:

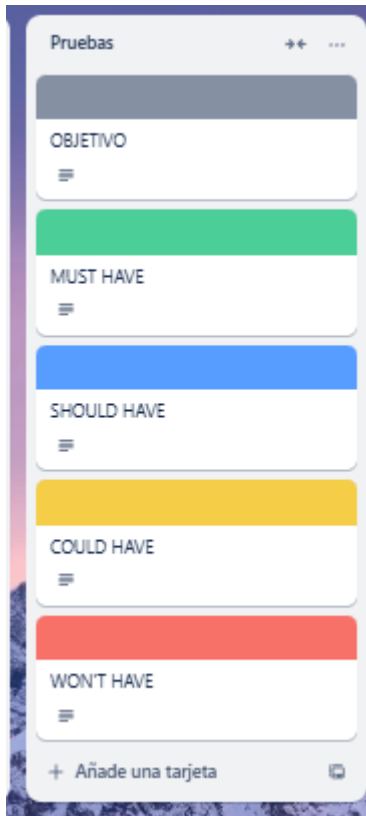
- Notificación por correo electrónico: ¿Se envía correctamente el correo electrónico de confirmación?
- Historial de compras: ¿Los usuarios pueden ver sus compras anteriores?

COULD HAVE:

- ¿El sistema sugiere productos correctamente?
- Integración con redes sociales: ¿Funciona la integración para compartir productos?
- ¿Tiene un análisis predictivo de Stock?
- ¿Tiene pruebas externas con otros componentes de la aplicación?

WON'T HAVE:

Se puede excluir la aplicación móvil, la visualización de productos en 3D y la automatización completa del reabastecimiento del inventario. Ya que son funcionalidades no incluidas en la versión inicial del proyecto.



5. DESPLIEGUE:

OBJETIVO:

El objetivo es desplegar el sistema en la nube de manera segura, eficiente y accesible, garantizando su disponibilidad, protección de datos y un funcionamiento estable para todos los usuarios.

MUST HAVE:

1. Configurar de manera segura el servidor y proteger la información contra accesos no autorizados.
2. Desplegar el sistema en la nube para garantizar acceso y flexibilidad.
3. Realizar pruebas exhaustivas antes de lanzar el sistema para evitar problemas críticos.
4. Asegurar que el sistema siempre esté disponible, incluso si un servidor falla.
5. Cifrar la información para proteger los datos sensibles de los usuarios.

SHOULD HAVE:

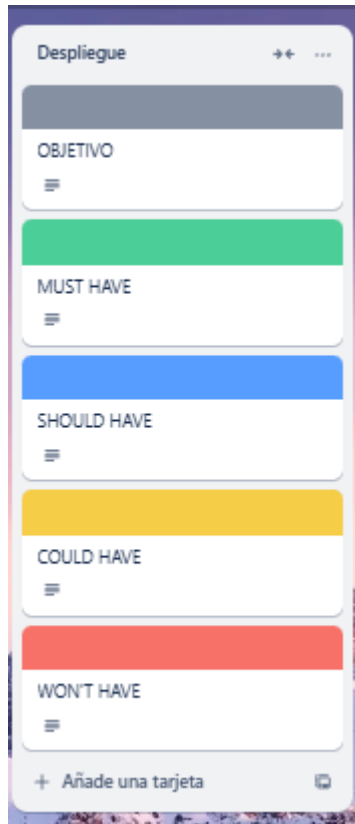
1. Capacitación para los usuarios para que puedan usar el sistema de manera eficiente.
2. Realizar copias de seguridad de manera automática para proteger la información.
3. Optimizar el uso de recursos para asegurar que todo funcione rápido y estable.
4. Probar el sistema para asegurar que pueda manejar la cantidad de usuarios esperada.
5. Planificar actualizaciones periódicas para solucionar errores y mejorar funciones.

COULD HAVE:

1. Brindar una guía básica para que el equipo se familiarice con el uso de herramientas en la nube.
2. Guardar copias de seguridad automáticas para proteger la información importante.
3. Ajustar el espacio de almacenamiento para no gastar recursos de más.
4. Probar todo el sistema en la nube para asegurarse de que funcione bien para todos los usuarios.
5. Tener herramientas para revisar el estado del sistema y detectar problemas rápidamente

WON'T HAVE:

1. No usar servidores físicos para guardar información.
2. No implementar soluciones sin supervisión para evitar problemas importantes.
3. Evitar tecnologías antiguas que no funcionen bien en servicios en la nube.
4. No depender de un solo servidor o lugar para evitar que el sistema se caiga.
5. No guardar información sin cifrar para proteger los datos de los usuarios.



6. MANTENIMIENTO:

OBJETIVO:

El objetivo es realizar el mantenimiento y las actualizaciones del sistema.

Una vez que el sistema esté en producción, se debe realizar mantenimiento para corregir errores, mejorar el rendimiento y agregar nuevas características.

MUST HAVE:

Soporte continuo asegurando que las funciones críticas sigan funcionando correctamente.

SHOULD HAVE:

Actualización de los requisitos Should Have que no se implementaron.

COULD HAVE:

Desarrollo de funcionalidades Could Have que quedaron pendientes.

WON'T HAVE:

- Características que no se implementarán o no se planifican para esta versión
- Rediseño Completo de la Interfaz de Usuario (UI)
- Creación de Nuevas Plataformas
- Integración de Funcionalidades que No Aportan Valor Directo
- Funcionalidades de Alta Complejidad Técnicas

