

Plataforma Web para Gestión de Stock en la venta de Cubremochilas

Grupo notVentas - Jhonatan Friz - Daniel Matamala - Cristian Quilodrán - Jocelyn Tereucán -
Rodrigo Vergara.

Profesor responsable: Ing. Luis Veas.

Instituto de Informática - Universidad Austral de Chile.

Resumen — Este documento presenta una solución tecnológica para optimizar la gestión de “stock” en una empresa dedicada a la venta de Cubremochilas en Valdivia. Actualmente, no existe una comunicación fluida entre los puntos de venta, por lo tanto, no se conocen los inventarios de cada distribuidor, lo que puede producir pérdidas en ventas y mala gestión a largo plazo. La solución propuesta es una plataforma web escalable que permite a los distribuidores controlar su stock, conocer el stock de otros puntos y el total disponible en toda la empresa.

Abstract — This document presents a technological solution to optimize stock management in a company dedicated to selling backpack covers with distribution in three locations in Valdivia. Currently, the lack of communication between these points leads to lost sales due to not knowing the available inventory in other stores. The proposed solution is a scalable web platform that allows distributors to view their stock, the stock of another selected distributor, and the total available across the company. Additionally, additional functions for sales and trend analysis are considered.

Palabras Claves – Gestión ; cubremochilas ; sistema distribuido ; stock

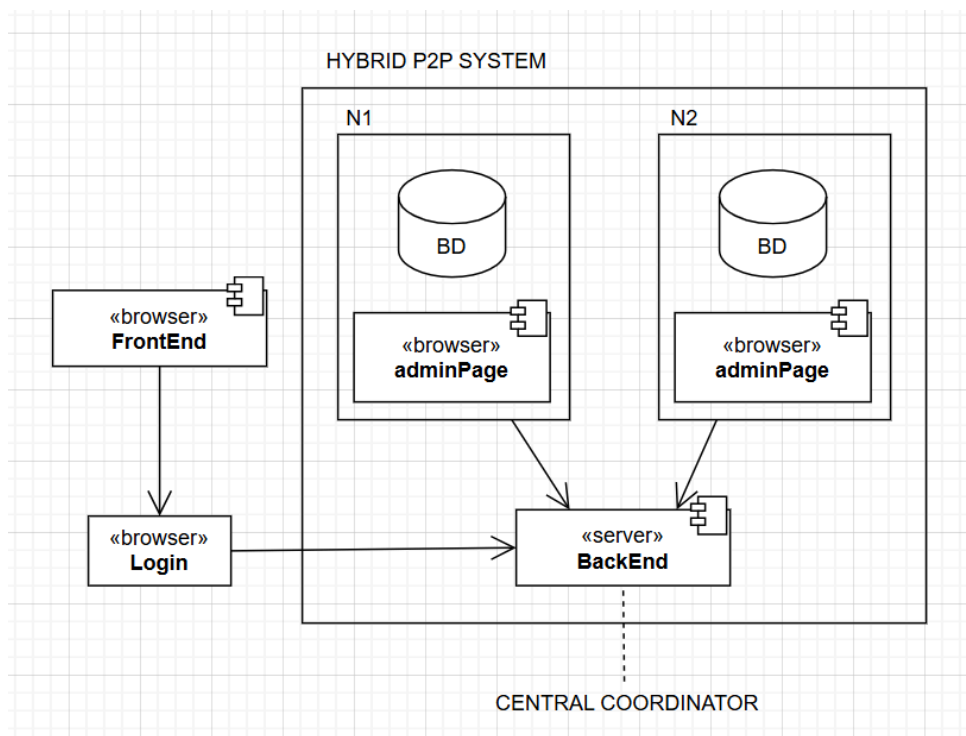
Keywords – Management, backpack covers; distributed system; stock

I. Introducción

Las empresas con múltiples puntos de distribución suelen enfrentar problemas de gestión de inventarios cuando no cuentan con un sistema centralizado. La falta de información actualizada sobre el stock en cada punto puede derivar en oportunidades de venta perdidas y dificultades operativas. En este contexto, la implementación de una plataforma web permitirá mejorar la eficiencia operativa, la disponibilidad de stock y la toma de decisiones basadas en datos en tiempo real.

II. Diseño del sistema

1. Arquitectura de la plataforma
 - Diagrama UML basada en componentes



2. Lista de componentes y descripción

2.1. Frontend

2.1.1. Login:

- Página donde los usuarios pueden iniciar sesión con correo y contraseña.
- Validación de credenciales mediante una API.
- Manejo de errores en caso de credenciales incorrectas.
- Redirección al **Dashboard** tras inicio de sesión exitoso.

2.1.2. Rutas protegidas (React Router)

- Implementación de protección de rutas para restringir el acceso a usuarios autenticados.
- Redirección a la pantalla de **Login** si el usuario no está autenticado.
- Middleware para verificar roles y permisos.

2.1.3. Dashboard

- Panel principal con acceso rápido a secciones clave como productos, tiendas, usuarios y empresas.
- Visualización de métricas y estadísticas (opcional).

2.1.4. Panel de Administración (CRUD)

- CRUD para **Usuarios, Productos y Tiendas**.
- Cada sección contará con su propio formulario de creación y edición.

2.1.5. Formulario (React Hook Form): tenemos que describir la cantidad de formularios que vamos a desarrollar:

2.1.5.1. Crear/editar Usuario:

- Formulario para registrar o modificar usuarios.
- Validación de campos (correo, contraseña, rol, etc.).

2.1.5.2. Crear/editar Producto:

- Formulario para registrar o actualizar productos.
- Campos: nombre, precio, stock, descripción, imágenes.

2.1.5.3. Crear Editar Empresa:

- Formulario para gestionar la información de la empresa.

2.1.5.4. Crear Editar Tienda:

- Formulario para registrar y modificar tiendas.

2.1.5.5.

2.1.6. Alertas:

- Notificaciones para acciones exitosas o errores.

2.1.7. Consultas a APIs (axios, react-query)

- Manejo de peticiones HTTP para obtener y enviar datos.

2.1.8. Mostrar Tienda (Lista de productos y stock)

- Página que muestra los productos disponibles en una tienda específica.

2.1.9. Producto (“[page.cl/product/id_product](#)”)

- Página con información detallada del producto.

2.1.10. Botón de crear productos, compañías, usuarios o tiendas (despliega formulario)

- Al presionar, despliega el formulario para agregar un nuevo dato a la DB y que despliega un formulario.

2.1.11. Componente de Caché (<https://react.dev/reference/react/cache>)

- Implementación de caching para mejorar el rendimiento.

2.1.12.

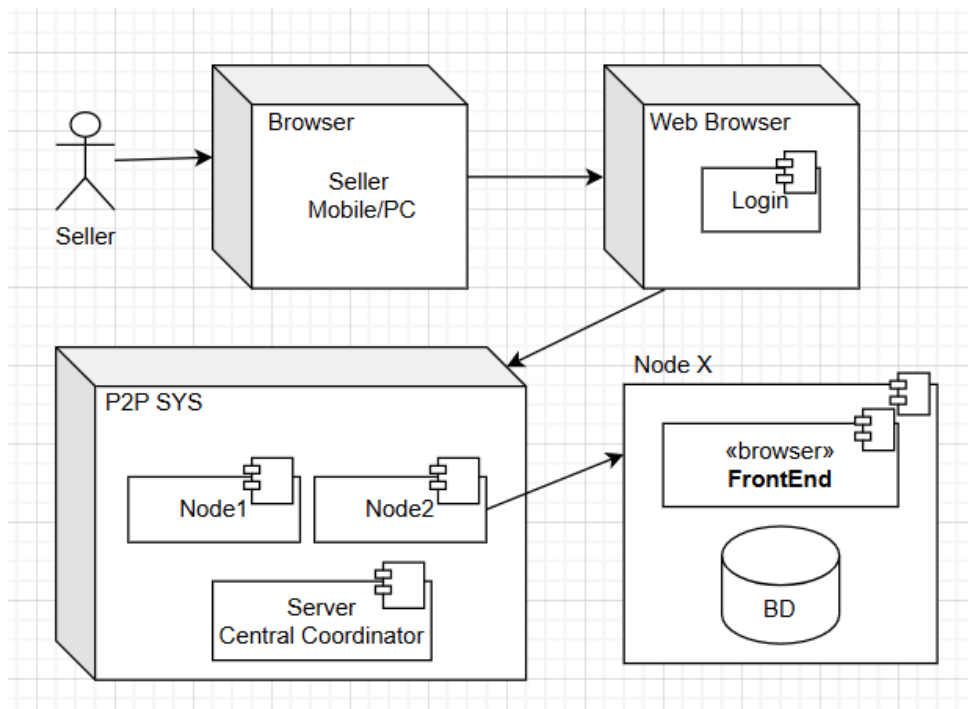
2.2. Backend

2.2.1. autenticación

- Ruta de inicio de sesión para validar usuarios.
- Implementación de JWT para sesiones seguras.
- Middleware para verificar autenticación y permisos.

- 2.2.2. ruta de store - usuario - Empresa - productos
 - Endpoints para administrar tiendas (crear, editar, eliminar, listar).
- 2.2.3. conexión a DB
 - Configuración de la conexión a la base de datos.
 - Definición de modelos de datos.
 - Relaciones entre entidades (usuarios, tiendas, productos, empresa).
- 2.2.4. Configuración
 - Configuración del entorno (variables, base de datos, autenticación).
 - Scripts de inicialización.

3. Modelo de despliegue



4. Modelo fundamental

- 4.1. El sistema P2P Híbrido planificado consta de 3 componentes claves para el éxito del sistema:

Nodo Normal (Peer): Comparten y consumen recursos en la red. Actúan como clientes y servidores a la vez.

Nodo Central (SuperNodo): Nodo con mayor capacidad de procesamiento y almacenamiento, que ayuda a organizar la red, almacena índices de contenido y facilita la búsqueda y conexión entre nodos.

Mecanismo de comunicación: Incluye una combinación de búsqueda centralizada (donde el supernodo ayuda a localizar recursos de manera más eficiente) y búsqueda distribuida (donde los nodos se conectan entre sí sin intermediarios).

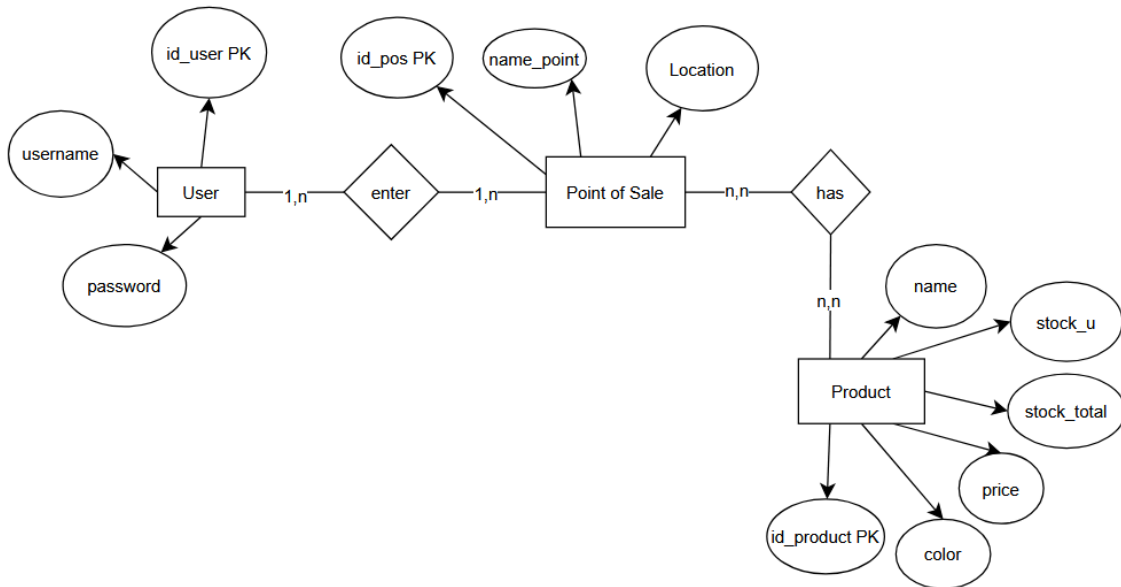
El funcionamiento de un sistema P2P híbrido combina mecanismos centralizados y descentralizados para optimizar la eficiencia. Los peers se registran en el supernodo o servidor central, que mantiene un índice actualizado de los recursos. Cuando un nodo busca un archivo o servicio, consulta al supernodo para localizarlo, y la transferencia de datos ocurre directamente entre los pares, reduciendo la carga en la infraestructura central y mejorando la escalabilidad del sistema.

5. Modelo de base de datos relacional

- Diccionario de datos

Propiedad del dato	Tipo de dato	Descripción	Ejemplo
id_user	int	identificación única del usuario	101
username	varchar(15)	nombre de usuario	León
password	varchar(15)	contraseña de ingreso para el usuario	Leon1001
id_pos	int	identificación única del punto de venta	201
name_point	varchar(15)	nombre del punto de venta	CXP isla teja
Location	varchar(15)	lugar donde está ubicado el punto de venta	Isla teja
id_product	int	identificación única producto	301
color	varchar(15)	color o característica del producto	negro
price	int	precio del producto	5000
stock_total	int	stock total del producto	25
stock_u	int	stock unitario del producto en un punto de venta	5
name	varchar(15)	nombre del producto	cubremochila

- Modelo entidad-relación



6. Tecnologías seleccionadas

6.1. Frontend: Se utilizará **Vite + React + TypeScript y Tailwind**.

Este stack permite:

- Un entorno de desarrollo rápido y eficiente.
- Componentes reutilizables y tipados con **TypeScript**.
- **Tailwind CSS**, que ofrece un sistema de estilos basado en clases utilitarias para un diseño ágil y flexible.

6.2. Backend: Se utilizará **Node.js** con **TypeScript** y las razones principales son:

- Mantener coherencia en el lenguaje de desarrollo.
- Alto rendimiento en aplicaciones web en tiempo real.
- Gran ecosistema de paquetes y soporte comunitario.

6.3. Contenedores: Se utilizará **Docker**.

- Asegurar la portabilidad de la aplicación en distintos entornos.
- Facilitar los despliegues y la escalabilidad del sistema.
- Estandarizar el desarrollo en todos los equipos.

La elección de estas tecnologías se basa en el conocimiento del equipo y su adopción en la industria. Además, cuentan con una comunidad activa y documentación sólida, lo que facilita la resolución de problemas y el aprendizaje continuo.

7. Consideraciones de presupuesto

Para el desarrollo de este trabajo no hemos considerado un presupuesto para el despliegue, utilizaremos nuestros computadores para en una red LAN para el despliegue distribuido de las diferentes “Bodegas” en al menos 3 computadores.

8. Propuesta de preguntas para el cliente.

8.1. ¿Qué dispositivos utiliza para efectuar sus ventas?

Dependiendo de la instancia en la cual me encuentre al momento de realizar una venta, o responder consultas, utilizo mi celular o mi computador.

8.2. ¿Cuáles son los métodos actuales para el seguimiento de ventas?

En estos momentos todo es manual a través de mensajes de texto con la persona en cada punto.

8.3. ¿Las actualizaciones de stock son controladas manualmente?

Sí, actualmente el stock es actualizado de manera manual, anotando en un bloc de notas.

8.4. ¿Se necesitan informes históricos para el seguimiento de stock?

Se podrían implementar ya que tendremos estadísticas que ayudarán al futuro del negocio.

8.5. ¿Hay planes para expandir los puntos de distribución?

A futuro existen planes de poder establecer otros puntos de ventas en toda la ciudad de Valdivia.

8.6. ¿El sistema debe soportar múltiples categorías de productos?

Sí, puesto que espero poder traer otros productos al mercado, esto incluye productos de diversas categorías.

8.7. ¿Cuántos productos y transacciones mensuales esperan manejar?

No demasiadas, alrededor de 10 transacciones mensuales por ahora.

8.8. ¿Cuántos usuarios accederán simultáneamente?

Alrededor de 3 o 4 vendedores al día.