**Descripción del proyecto a desarrollar**

**Requisitos Generales**

**Detalle de los aspectos mínimos que su propuesta de proyecto de portafolio de tiene. debe considerar los siguientes puntos:**

1. El producto debe ser desarrollado como aplicación WEB o Móvil o escritorio, el patrón arquitectónico a utilizar debe ser un modelo de capas.
2. El producto debe ser desarrollado en un lenguaje de programación orientado a los objetos, debe considerar dos aplicaciones una principal y otra secundaria que deberá estar integrada a la aplicación principal y que será utilizada para dar autonomía al usuario administrador del sistema ( caso uso mantención)
3. La aplicación principal tiene contener al menos:
   1. Ocho mantenedores que permitan dar autonomía las peticiones del usuario.
   2. Un módulo completo asociado al negocio y que dará solución a su problemática.
   3. Dos reportes con filtros que entreguen los resultados en pantalla y que permitan exportar a PDF y XLS
   4. Enmascarar claves y sesiones para mantener la seguridad del sistema
   5. Autenticación, manejo de sesiones, perfiles o roles
   6. Las notificaciones al usuario deben ser a través de correo electrónico o mensajes a dispositivos móviles.
   7. El diseño debe ser responsivo para garantizar su adecuada visualización en múltiples dispositivos.
   8. Todos los formularios de ingreso de datos deben contar con las validaciones correspondientes.

**DESCRIPCIÓN**

## Introducción

InventPro es un sistema integral diseñado para pequeñas y medianas empresas (PYMEs) que buscan optimizar la gestión de inventarios y órdenes de compra, asegurando trazabilidad, control de accesos y eficiencia operativa. Actualmente, muchas PYMEs dependen de hojas de cálculo o registros manuales, lo que genera errores, pérdida de información y dificultades para tomar decisiones oportunas.

**El sistema se compone de dos aplicaciones interconectadas:**

1. Aplicación principal (Web App): Gestión del negocio (productos, órdenes, clientes, proveedores, inventario).
2. Aplicación secundaria (Soporte / Notificaciones): Autonomía para administradores y encargados de bodega, envío de alertas y mantenedores de soporte.

La arquitectura está basada en patrón en capas, con backend en Node.js, frontend en React + Tailwind y base de datos PostgreSQL (anteriormente MongoDB), garantizando seguridad, escalabilidad y mantenibilidad.

**2. Contexto**

Actualmente, muchas PYMEs enfrentan problemas graves en la gestión de inventario y órdenes de compra:

* Falta de control de accesos: cualquier empleado con acceso a Excel puede modificar datos, sin registro de quién realizó la acción.
* Errores humanos frecuentes: datos mal anotados, duplicidad de registros y cálculos incorrectos.
* Ausencia de trazabilidad: no se sabe quién descontó productos ni cuándo.
* Tiempos excesivos: contabilizar stock manualmente toma entre 1 y 2 horas semanales por bodega, y verificar stock en planillas dispersas puede tardar 10-15 minutos por producto.
* Pérdidas económicas: estudios internos muestran que PYMEs pierden hasta 10% de inventario anual por errores o movimientos no autorizados. Por ejemplo, si un vendedor descuenta 5 unidades sin registro, se desconoce si el stock real está afectado, generando quiebres de inventario o ventas fallidas.
* Descoordinación interdepartamental: ventas, bodega y compras trabajan con información fragmentada, retrasando la toma de decisiones.
* Riesgos de seguridad: cualquier persona con acceso al archivo Excel puede modificarlo, y usuarios no registrados pueden registrar movimientos sin autorización.

**Impacto cuantificado:**

* Tiempo perdido en tareas repetitivas de conteo y verificación de stock: aprox. 100 horas al año por bodega.
* Pérdida económica directa por errores: si un producto tiene un valor promedio de $10.000 y se pierde 10% del inventario anual en una bodega de 1.000 unidades, la pérdida asciende a $1.000.000 al año.
* Riesgo de ventas fallidas por quiebre de stock: al no haber alertas automáticas, se pierde hasta 5% de ventas potenciales por producto.

### Beneficios Cuantificables de InventPro

1. Reducción de errores de stock
   * Problema actual: 3-5% de los productos por semana son registrados incorrectamente.
   * Con InventPro: reducción a 0.5% o menos, gracias a trazabilidad, control de roles y validación de datos.
2. Ahorro de tiempo en conteo y verificación de inventario
   * Problema actual: 1-2 horas semanales por bodega para contar stock.
   * Con InventPro: se reduce a 10-15 minutos semanales, gracias a reportes automáticos y actualizaciones en tiempo real.
3. Reducción de duplicidad de órdenes y errores de compras
   * Problema actual: 5-10% de órdenes generan errores por control manual.
   * Con InventPro: reducción al 1%, gracias a actualización automática de inventario y control de permisos.
4. Disminución de pérdidas económicas por inventario mal registrado
   * Problema actual: pérdidas estimadas de hasta 10% del inventario anual.
   * Con InventPro: se proyecta reducir pérdidas a 2% o menos, con historial de movimientos, alertas y trazabilidad de usuarios.
5. Ahorro de tiempo en consultas y reportes
   * Problema actual: 10-15 minutos por producto para verificar stock o historial de movimientos.  
     Con InventPro: consultas instantáneas gracias a base de datos centralizada, reducción del tiempo a menos de 1 minuto por producto.
6. Reducción de riesgo de quiebre de stock y ventas fallidas
   * Problema actual: hasta 5% de ventas potenciales perdidas por quiebres de stock no detectados.
   * Con InventPro: disminución de ventas perdidas a 1%, gracias a alertas automáticas de stock crítico.
7. Mejora en la coordinación interdepartamental
   * Problema actual: La descoordinación genera retrasos y duplicidad de tareas.
   * Con InventPro: reducción de tiempos de coordinación en 50-70%, gracias a información centralizada y reportes compartidos en tiempo real.

## 3. Procesos de Negocio

La empresa simulada representa una PYME típica que actualmente gestiona sus inventarios y órdenes de compra de forma manual, lo que genera múltiples ineficiencias y riesgos operativos.

### **3.1. Registro de Productos**

* Actualmente se registran los productos en hojas de cálculo Excel, sin control de acceso ni validación de datos.
* Problemas identificados:
  + Cualquier persona con acceso a la hoja puede modificar, eliminar o duplicar registros, incluso sin autorización.
  + No hay un historial de quién realizó cambios ni cuándo se hicieron.
* Impacto estimado:
  + Se pierden aproximadamente 5-10% de los productos al año por errores de registro o duplicidad.
  + La verificación manual de cada producto toma 5-10 minutos por ítem, generando retrasos semanales de 1 a 2 horas por bodega.

### **3.2. Actualización de Stock**

* El stock se actualiza manualmente, sin trazabilidad ni control de movimientos.
* Problemas identificados:
  + No se registra quién realiza cada ajuste.
  + Los descuentos de stock por ventas pueden ser registrados incorrectamente. Ej.: un vendedor descuenta 5 unidades sin autorización → no hay registro del responsable.
  + Existe riesgo de sobreventa o quiebre de stock.
* Impacto estimado:
  + Promedio de errores de stock: 3-5% de los productos por semana.
  + Retrasos en la actualización generan pérdida de ventas y descoordinación entre áreas de ventas y bodega.

### **3.3. Gestión de Órdenes de Compra**

* Las órdenes se generan en documentos independientes, desconectadas del inventario.
* Problemas identificados:
  + No se actualiza automáticamente el stock al generar órdenes.
  + Difícil control de órdenes pendientes, entregadas o canceladas.
  + Falta de trazabilidad sobre quién creó o modificó la orden.
* Impacto estimado:
  + Cada verificación manual de stock vs. orden toma 10-15 minutos.
  + Errores frecuentes generan duplicidad de órdenes o faltantes de productos, afectando la planificación de compras.

### **3.4. Control de Accesos y Roles**

* La empresa no cuenta con autenticación diferenciada, por lo que cualquier persona puede acceder y modificar información crítica.  
  Problemas identificados:
  + No hay distinción entre administradores, vendedores o bodegueros.
  + Se generan registros inconsistentes y falta de responsabilidad.
* Impacto estimado:
  + Riesgo de accesos no autorizados o uso de credenciales compartidas.
  + Dificultad para auditar movimientos y responsabilizar a un usuario específico.

### **3.5. Gestión de Proveedores y Clientes**

* Los datos de proveedores y clientes se mantienen en registros dispersos, en hojas de cálculo o documentos físicos.
* Problemas identificados:
  + Información desactualizada y redundante.
  + Difícil consolidación para análisis de compras, ventas o pagos.

* Impacto estimado:
  + Tiempos prolongados en búsquedas: 5-10 minutos por consulta.
  + Errores en contacto con proveedores o clientes → retrasos en entrega de productos y servicios.

### **3.6. Comunicación Interna y Coordinación**

* La comunicación entre áreas de ventas, bodega y compras se realiza de manera manual, generalmente por teléfono, correo o mensajes dispersos.
* Problemas identificados:
  + Falta de visibilidad en tiempo real de stock, órdenes y disponibilidad de productos.
  + Errores de coordinación entre áreas: productos sobrevendidos, retrasos en despacho, compras duplicadas.
* Impacto estimado:
  + Incremento del tiempo de operación: cada actualización requiere 5-15 minutos de coordinación adicional.
  + Pérdida económica estimada: hasta 10% de inventario anual por errores humanos y movimientos no registrados.

### **3.7. Riesgos Generales del Proceso Actual**

* Falta de trazabilidad de movimientos = imposible auditar quién realizó cambios o ajustes.
* Errores frecuentes y duplicidad de registros = pérdida de stock y ventas.
* Procesos lentos y manuales = aumento de horas trabajadas y riesgo de errores.
* Información fragmentada = toma de decisiones lenta y poco confiable.

## 4. Problema

Las pequeñas y medianas empresas (PYMEs) que gestionan su inventario y órdenes de compra de forma manual o mediante hojas de cálculo enfrentan graves ineficiencias operativas, riesgos de seguridad y pérdidas económicas significativas.

### **4.1 Impactos Principales**

1. Tiempo de gestión elevado y procesos ineficientes  
   * La verificación manual del stock y la actualización de inventarios puede tomar entre 1 y 2 horas semanales por bodega, dependiendo del tamaño del inventario.
   * La generación de órdenes de compra requiere cruzar información de inventario y ventas manualmente, lo que retrasa la planificación de pedidos y aumenta la carga laboral del personal.
2. Errores frecuentes y duplicidad de registros  
   * El registro manual en Excel o documentos físicos genera duplicidad de productos, entradas erróneas de stock o descuentos de unidades no autorizadas.
   * Se estima que 3-5% de los productos por semana pueden ser registrados incorrectamente, provocando inconsistencias en el inventario y errores en las ventas.
3. Falta de trazabilidad y control de accesos  
   * Cualquier persona con acceso a las planillas puede modificar los datos sin que exista un historial de cambios.
   * No es posible identificar quién realizó cada ajuste de inventario o registro de orden, lo que impide auditar movimientos y responsabilizar a usuarios.
   * Riesgo de uso de credenciales compartidas o usuarios no registrados, aumentando la posibilidad de fraudes internos o mermas no detectadas.

1. Ausencia de reportes confiables y visibilidad en tiempo real  
   * La información fragmentada entre hojas de cálculo y documentos independientes genera retrasos en la toma de decisiones.
   * Consultas sobre stock, movimientos de inventario o historial de ventas pueden tardar 10-15 minutos por producto, dificultando la planificación y control de operaciones.
2. Pérdidas económicas directas  
   * Los errores de registro, ajustes no autorizados y falta de visibilidad generan pérdidas estimadas de hasta 10% del inventario anual, incluyendo mermas, robos internos y productos mal contabilizados.
   * La descoordinación entre áreas de ventas, bodega y compras provoca sobrecostos y quiebres de stock, afectando la rentabilidad y la satisfacción de los clientes.

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 5. Solución Propuesta

El proyecto propone InventPro, un sistema web integral con dos aplicaciones diferenciadas pero integradas, que solucionan los problemas actuales:

### **5.1 Aplicación Principal – Gestión de Inventario y Órdenes**

* Frontend: React + Tailwind (responsivo, usable en desktop y móviles).
* Backend: Node.js + Express.
* Base de datos: PostgreSQL, centralizada para normalizar información y garantizar consistencia.
* Funcionalidades:
  + CRUD completo de productos, categorías, proveedores, clientes y stock.
  + Control de órdenes de compra con actualización automática de inventario.
  + Reportes filtrados y exportables a PDF/XLS para análisis administrativo.
  + Historial de movimientos de stock con trazabilidad de usuario y fecha.
  + Notificaciones internas para alertas de stock crítico.

### **5.2 Aplicación Secundaria – Soporte y Notificaciones**

* Frontend: Micro-frontend en React + Tailwind, integrable al mismo backend RESTful.
* Funcionalidades:
  + Mantenedores de usuarios, roles y permisos.
  + Registro de ajustes manuales de inventario con justificación.
  + Notificaciones automáticas por correo electrónico a usuarios específicos (stock crítico, órdenes pendientes, alertas de vencimiento de productos, etc.).
* Objetivo: permitir a administradores y encargados de bodega operar de forma autónoma, desde cualquier dispositivo.
* Integración: ambas aplicaciones usan la misma API RESTful y base de datos PostgreSQL, garantizando información centralizada y actualizada en tiempo real.

### 

### 

### 

### **5.3 Beneficios para el negocio**

* Reducción de errores humanos y pérdida de inventario.
* Mejora en tiempos de operación y validación de stock.
* Trazabilidad completa de movimientos y acciones por usuario.
* Automatización de reportes para decisiones estratégicas.
* Alertas y notificaciones que permiten anticiparse a quiebres de stock o problemas de inventario.
* Seguridad reforzada mediante JWT, roles, sesiones y auditoría de accesos.

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 6. Requerimientos Generales

### **6.1 Aplicaciones diferenciadas**

Aplicación Principal (Web App – Gestión del Negocio)

* Plataforma principal para la gestión integral de inventarios, órdenes de compra, proveedores, clientes y productos.
* Accesible desde navegador web (desktop y móvil) con diseño responsivo usando React + Tailwind.
* Permite la visualización en tiempo real del estado del inventario y las órdenes, centralizando la información.
* Funcionalidades principales:
  + Gestión completa de productos, categorías, proveedores, clientes e inventario.
  + Control de stock automático al crear o actualizar órdenes.
  + Reportes administrativos filtrables y exportables en PDF y XLS.
  + Historial detallado de movimientos de inventario, incluyendo usuario, fecha, tipo de movimiento y justificación.
  + Integración con la aplicación secundaria para recibir notificaciones y registrar ajustes manuales.

Aplicación Secundaria (Soporte / Notificaciones – Micro-frontend)

* Plataforma orientada a administradores y encargados de bodega para mantención y control en tiempo real.
* Arquitectura de micro-frontend web, desplegada junto a la aplicación principal pero funcionalmente independiente.
* Funcionalidades principales:
  + Mantenedores de usuarios internos, roles y permisos.
  + Registro rápido de ajustes manuales de inventario con justificación y trazabilidad.
  + Notificaciones automáticas vía correo electrónico o push web, dirigidas a usuarios específicos.
  + Gestión de seguridad: control de sesiones, limitación de intentos de login y auditoría de accesos.
* Objetivo: otorgar autonomía y movilidad a administradores sin afectar la operación de la aplicación principal.

### 

### **6.2 Backend y arquitectura**

* Lenguaje: Node.js (ES Modules), orientado a objetos.
* Patrón arquitectónico: modelo de capas, incluyendo:
  + Routes: definición de endpoints REST para cada módulo.
  + Controllers: lógica de entrada/salida de datos.
  + Services / Lógica de negocio: gestión de inventarios, órdenes y reportes.
  + Models: definición de entidades y relaciones en Sequelize (PostgreSQL).
  + Middleware: autenticación, roles, validaciones, rate limit, logging y auditoría.
* Comunicación entre aplicaciones a través de API RESTful centralizada, garantizando sincronización en tiempo real.

### **6.3 Base de datos relacional (PostgreSQL)**

* Centraliza toda la información del sistema: productos, categorías, proveedores, clientes, inventarios, usuarios, roles, órdenes y reportes.
* Migración desde MongoDB para normalización de datos y mejor manejo de relaciones (1-n, n-m).
* Funcionalidades principales de la base de datos:
  + Registros de auditoría de movimientos de inventario y accesos de usuarios.
  + Integridad referencial entre entidades críticas (p.ej. Order → Product → Inventory).
  + Soporte para consultas avanzadas para generación de reportes y estadísticas.

### 

### 

### 

### 

### 

### 

### **6.4 Requerimientos de la Aplicación Principal**

* Mantenedores (mínimo 8): productos, categorías, proveedores, clientes, órdenes, inventario manual, reportes y ajustes de stock.
* Módulo de negocio completo: control automático de stock, historial de movimientos y generación de reportes.
* Reportes: mínimo dos reportes filtrables, paginados y exportables a PDF y XLS.
* Seguridad:
  + Enmascaramiento de claves.
  + Gestión de roles y permisos.
  + Control de sesiones y autenticación con JWT.
* Validaciones:
  + Todos los formularios deben incluir validaciones estrictas de datos (tipo, formato, obligatoriedad).
  + Validaciones del lado del cliente (frontend) y servidor (backend).
* Diseño responsivo: adaptación automática a dispositivos móviles, tabletas y desktop.

**6.5 Requerimientos de la Aplicación Secundaria**

* Mantenedores de soporte: usuarios, roles y permisos, ajustes manuales de inventario.
* Notificaciones automáticas: envíos push o por correo a usuarios específicos sobre:
  + Stock crítico.
  + Órdenes pendientes.
  + Movimientos o ajustes no autorizados.
* Control de seguridad avanzado:
  + Limitación de intentos de login.
  + Registro de IP y control de sesiones activas.
  + Auditoría de accesos y acciones de usuarios.
* Autonomía: permitir operar y actualizar información sin afectar la aplicación principal.

## 7. Requerimientos del Sistema

### 7.1 Requerimientos Funcionales

1. Gestión de usuarios
   * Crear, leer, actualizar y eliminar usuarios internos con roles asignados.
   * Autenticación mediante JWT y control de sesiones.
2. Gestión de productos
   * CRUD completo de productos con stock inicial y precios.
   * Validaciones estrictas de datos (tipo, formato, obligatoriedad).
   * Filtros, búsqueda avanzada y paginación.
3. Gestión de categorías, proveedores y clientes
   * CRUD completo con relaciones entre productos, proveedores y clientes.
4. Gestión de órdenes de compra
   * Crear y actualizar órdenes automáticamente ajustando el inventario.
   * Control de estado: pendiente, entregada, cancelada.
   * Registro de historial de movimientos y trazabilidad por usuario.
5. Inventario manual
   * Registrar aumentos o disminuciones de stock con justificación.
   * Auditoría completa de los movimientos realizados.
6. Reportes
   * Generar reportes filtrados y paginados del inventario y movimientos.
   * Exportación a PDF y XLS para análisis administrativo.
7. Notificaciones automáticas
   * Alertas por correo electrónico o push web sobre stock crítico, órdenes pendientes o ajustes no autorizados.
8. Integración entre aplicaciones
   * Sincronización en tiempo real entre la aplicación principal y la secundaria mediante API RESTful.

### **7.2 Requerimientos No Funcionales**

1. **Seguridad**
   * Control de roles y permisos estrictos.
   * Autenticación mediante JWT y limitación de intentos de login.
   * Encriptación de contraseñas con bcrypt y enmascaramiento de datos sensibles.
   * Auditoría de accesos y movimientos para trazabilidad completa.
2. **Rendimiento**
   * Respuesta del sistema menor a 2 segundos para consultas de inventario y reportes.
   * Capacidad de manejar simultáneamente al menos 50 usuarios activos sin degradación de desempeño.
3. **Escalabilidad**  
   * Arquitectura modular y basada en capas, permitiendo agregar nuevos módulos o usuarios sin afectar la operación.
   * Base de datos PostgreSQL preparada para crecimiento de datos y consultas concurrentes.
4. **Disponibilidad**
   * Sistema disponible 24/7 con tiempo de inactividad mínimo.
   * Tolerancia a fallos en el backend y respaldo regular de la base de datos.
5. **Mantenibilidad**
   * Código organizado en capas (routes, controllers, services, models, middleware).
   * Documentación Swagger por módulo para fácil entendimiento y mantenimiento.
   * Pruebas unitarias y de integración automatizadas para asegurar calidad de software.
6. **Usabilidad**
   * Diseño responsivo adaptable a desktop, tablet y móviles.
   * Interfaz intuitiva, clara y consistente para distintos tipos de usuarios (administrador, bodeguero, vendedor).
7. **Compatibilidad**
   * Funciona en navegadores modernos (Chrome, Edge, Firefox) y dispositivos móviles actuales.
   * Integración con sistemas de correo electrónico y push notifications estándar.

**8. Casos de uso**

**Listado de los casos de uso iniciales, solo propuesta, que justifican el desarrollo del producto, puede guiarse por la tabla de ejemplo:**

| **ID** | **Caso de Uso** | **Descripción** | **Actores** | **Tipo de caso** |
| --- | --- | --- | --- | --- |
| **CU01** | **Mantenedor de usuarios** | **Crear, leer, actualizar y eliminar usuarios internos con control de roles.** | **Administrador** | **Mantención – aplicación secundaria** |
| **CU02** | **Mantenedor de productos** | **CRUD de productos con stock inicial y control de precios.** | **Administrador** | **Negocio – aplicación principal** |
| **CU03** | **Mantenedor de categorías** | **Crear y administrar categorías de productos.** | **Administrador** | **Mantención – aplicación secundaria** |
| **CU04** | **Mantenedor de proveedores** | **CRUD de proveedores y datos de contacto.** | **Administrador** | **Negocio – aplicación principal** |
| **CU05** | **Mantenedor de clientes** | **CRUD de clientes para registro de ventas y pedidos.** | **Administrador** | **Negocio – aplicación principal** |
| **CU06** | **Gestión de órdenes de compra** | **Crear y administrar órdenes, actualizando inventario automáticamente.** | **Administrador** | **Negocio – aplicación principal** |
| **CU07** | **Ajustes manuales de inventario** | **Registrar aumentos o disminuciones de stock con justificación y trazabilidad.** | **Administrador** | **Negocio – aplicación principal** |
| **CU08** | **Login y seguridad** | **Autenticación con JWT, control de sesiones, limitación de intentos fallidos.** | **Usuario interno** | **Soporte – aplicación secundaria** |
| **CU09** | **Reporte de inventario** | **Generar reportes filtrados y paginados del stock actual, exportables a PDF/XLS.** | **Administrador** | **Negocio – aplicación principal** |
| **CU10** | **Historial de movimientos** | **Consultar historial de entradas, salidas y ajustes de stock con filtros por usuario y fecha.** | **Administrador, Bodeguero** | **Negocio – aplicación principal** |
| **CU11** | **Búsqueda avanzada** | **Buscar productos por nombre, categoría, proveedor y filtros combinados.** | **Administrador, Bodeguero, Vendedor** | **Soporte – aplicación secundaria** |
| **CU12** | **Gestión de perfil** | **Actualizar datos personales y contraseña.** | **Usuario interno** | **Mantención – aplicación secundaria** |
| **CU13** | **Notificaciones automáticas** | **Enviar alertas de stock crítico, órdenes pendientes o vencimientos a usuarios específicos.** | **Sistema** | **Soporte – aplicación secundaria** |
| **CU14** | **Integración de datos en tiempo real** | **Sincronizar inventario, órdenes y movimientos entre aplicaciones principal y secundaria.** | **Administrador, Sistema** | **Negocio/Soporte** |
| **CU15** | **Reporte de pérdidas y quiebres de stock** | **Mostrar productos con movimientos no autorizados o errores detectados.** | **Administrador** | **Negocio – aplicación principal** |