Logotipo

El contenido generado por IA puede ser incorrecto.

Nombre del Proyecto:

Nombre del Estudiante: Sebastián Ignacio Cerda Fuentes  
  
Fecha: 11/07/2025

**Informe final Anteproyecto**Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**1. Fundamentación del Problema/Oportunidad**

En el contexto actual de las pequeñas y medianas empresas (PYMES) chilenas, una de las áreas más vulnerables dentro de la gestión operativa es el control de pedidos a proveedores. Muchas empresas aún dependen de métodos manuales o decisiones intuitivas para determinar cuándo y cuánto reponer, lo que da lugar a **rupturas de stock**, **pedidos innecesarios**, **sobrestock** o **pérdidas económicas por falta de sincronización con la demanda real**.

Durante las fases diagnósticas del curso se identificó una tendencia común: las PYMES están dispuestas a adoptar soluciones tecnológicas, siempre y cuando estas sean accesibles, simples y no requieran conocimientos técnicos avanzados. Esta realidad convierte el problema en una oportunidad concreta: no es necesaria una solución compleja, sino una herramienta **eficiente, práctica y orientada a la acción**, que ayude a planificar mejor la reposición de inventario y optimice la relación con los proveedores.

Actualmente, la ausencia de un sistema que permita prever la necesidad de reposición y generar pedidos de manera sistemática genera varios efectos negativos: decisiones reactivas, pérdida de ventas por falta de productos, errores humanos en el pedido, baja capacidad de negociación con proveedores y, en muchos casos, **desconocimiento del consumo real por producto**. Esto impacta directamente en la eficiencia logística y la sostenibilidad del negocio.

Frente a esta situación, surge *ControlRepro*, un sistema inteligente de gestión de pedidos y reposición diseñado especialmente para pequeñas y medianas empresas. El objetivo de *ControlRepro* es **automatizar y profesionalizar el proceso de reposición**, generando sugerencias de pedido basadas en consumo histórico, control de stock mínimo y tiempos de entrega, todo desde una plataforma amigable y de fácil implementación.

La propuesta responde así a una necesidad concreta del mercado: dotar a las PYMES de una solución confiable que **anticipe los quiebres de stock** y **mejore la toma de decisiones operativas**, permitiéndoles ahorrar tiempo, evitar pérdidas y operar con mayor eficiencia.

**2. Alcance del Proyecto**

El proyecto *ControlRepro* contempla el diseño, desarrollo e implementación de una aplicación multiplataforma para la **gestión automatizada de pedidos a proveedores**, orientada a pequeñas y medianas empresas (PYMES) que buscan mejorar su eficiencia operativa y reducir el impacto de los quiebres de stock. El sistema estará enfocado en la reposición inteligente de productos, utilizando información histórica de consumo y reglas de negocio simples para emitir sugerencias automáticas de pedido.

**Alcance funcional**

El sistema cubrirá los siguientes aspectos clave:

* **Login seguro con autenticación local**: El acceso está restringido mediante el uso de RUT y PIN, garantizando que cada operación quede asociada a un trabajador identificado.
* **Módulo de productos y proveedores**: Permite el registro, edición y vinculación de productos con uno o varios proveedores, incluyendo tiempos de entrega (lead time) y visualización estructurada de la información.
* **Exportación de información**: Los productos registrados pueden ser exportados en formato Excel de manera estructurada y legible, facilitando su revisión o posterior uso externo.
* **Alertas visuales de stock crítico**: El sistema resalta automáticamente los productos cuyo stock actual es igual o inferior al punto de reposición, permitiendo una acción preventiva.
* **Interfaz amigable y validaciones integradas**: Se incluyen mensajes de error, formularios controlados, y confirmaciones para cerrar sesión, garantizando una experiencia de usuario clara y segura.

**Alcance técnico**

* **Tecnologías**: El sistema será desarrollado utilizando tecnologías modernas y accesibles como React.js (o Vue.js) para el frontend, Node.js para el backend y una base de datos relacional local (SQLite o PostgreSQL).
* **Compatibilidad**: La aplicación será multiplataforma (Windows, Linux, Mac) y estará diseñada para ejecutarse localmente, sin necesidad de conexión a internet.
* **Arquitectura**: Se implementará un enfoque de Clean Architecture, permitiendo una clara separación de responsabilidades entre la capa de interfaz, lógica de negocio y acceso a datos.

**Alcance temporal**

El desarrollo del proyecto se planifica para un periodo de **12 semanas**, dividido en fases de planificación, implementación modular, validación funcional, documentación y entrega. Al finalizar este ciclo, el sistema deberá contar con al menos un 80% de sus funcionalidades principales implementadas, y haber sido validado mediante pruebas internas.

**Limitaciones fuera de alcance**

* No se contempla la integración con servicios en la nube ni sincronización en tiempo real entre dispositivos.
* No se incluirá facturación electrónica ni conexión directa con plataformas tributarias.
* El motor de sugerencias se basará en reglas simples y no utilizará algoritmos de inteligencia artificial en esta versión inicial.
* La autenticación será local y no se manejarán roles múltiples avanzados.

**3. Objetivos del Proyecto**

A continuación, se definen los **objetivos específicos** del proyecto *ControlRepro*, estableciendo el resultado esperado para cada uno y el compromiso concreto que se busca alcanzar durante el ciclo de desarrollo. Estos objetivos están alineados con la necesidad de **automatizar y optimizar el proceso de reposición de productos en PYMES**, mejorando la toma de decisiones y reduciendo errores operativos.

**Objetivo Específico 1**

**Desarrollar un sistema de autenticación local basado en RUT y PIN para el acceso seguro al sistema.**  
**Resultado esperado:** Garantizar que cada operación registrada dentro del sistema esté asociada a un trabajador identificado.  
**Compromiso:** Lograr un 100% de trazabilidad en el acceso y uso del sistema por parte de los usuarios.

**Objetivo Específico 2**

**Diseñar un módulo de gestión de productos y proveedores que permita vincularlos correctamente, incluyendo tiempos de entrega (lead time).**  
**Resultado esperado:** Facilitar la asociación eficiente entre cada producto y sus proveedores válidos.  
**Compromiso:** Alcanzar una cobertura del 100% de productos registrados con al menos un proveedor asociado.

**Objetivo Específico 3**

**Registrar el consumo histórico de productos como base para la generación de pedidos sugeridos.**  
**Resultado esperado:** Disponer de un historial de consumos que permita calcular la necesidad de reposición en base al comportamiento real.  
**Compromiso:** Lograr un registro funcional y estable de consumos con visualización por producto y rango de fechas.

**Objetivo Específico 4**

**Exportar los pedidos sugeridos en formatos Excel o PDF listos para ser enviados a proveedores.**  
**Resultado esperado:** Facilitar la exportación con un clic, con datos completos y estructurados.  
**Compromiso:** Generar archivos exportables en menos de 3 segundos con formato legible y profesional.

**Objetivo Específico 5**

**Generar alertas cuando un producto esté cerca de su stock mínimo o exista un patrón de retraso de proveedores.**  
**Resultado esperado:** Informar de forma proactiva sobre riesgos operativos, permitiendo actuar con anticipación.  
**Compromiso:** Detectar y alertar el 100% de los casos simulados en pruebas funcionales.

**4. Métricas para Evaluación de Objetivos**

Las siguientes métricas serán aplicadas durante la etapa de pruebas funcionales del sistema, utilizando datos simulados de productos, consumos y proveedores. Su objetivo es verificar el cumplimiento de los objetivos específicos establecidos y asegurar el correcto funcionamiento del sistema desde una perspectiva práctica, cuantificable y verificable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo Específico** | Métrica | Valor meta | Tipo de métrica | Criterio de éxito |
| 1. Login seguro con RUT y PIN | % de accesos autenticados correctamente | 100% | Cuantitativa / Binaria | Todos los accesos deben validarse correctamente y registrar al usuario responsable. |
| 2. Gestión de productos y proveedores | % de productos con proveedor asociado | 100% | Cuantitativa | Todos los productos registrados deben tener al menos un proveedor válido vinculado. |
| 3. Registro de consumo histórico | % de operaciones correctamente almacenadas | ≥ 95% | Cuantitativa | Las operaciones de consumo deben quedar registradas y disponibles para visualización. |
| 4. Generación automática de pedidos sugeridos | % de coincidencia con necesidad real de reposición | ≥ 90% | Cuantitativa | La sugerencia del sistema debe reflejar adecuadamente los productos que requieren reposición. |
| 5. Exportación de pedidos | |  | | --- | |  |  |  | | --- | | Tiempo promedio de exportación del pedido | | ≤ 3 segundos | Cuantitativa / Temporal | El sistema debe generar correctamente el archivo exportable, con formato limpio y usable. |
| 6. Alertas por bajo stock o retraso de proveedor | % de alertas generadas en casos críticos simulados | 100% | Cuantitatiwva | El sistema debe emitir alertas visuales cuando se cumplan las condiciones definidas. |

**5. Propuesta de Solución**

**L**a solución propuesta es una aplicación multiplataforma denominada **ControlRepro**, orientada a resolver el problema de reposición reactiva y poco estructurada de productos en pequeñas y medianas empresas (PYMES). El sistema permitirá **automatizar la generación de pedidos a proveedores**, utilizando reglas simples basadas en el consumo histórico, el stock mínimo y los tiempos de entrega (lead time), todo desde una interfaz amigable y funcional que no requiere conocimientos técnicos avanzados por parte del usuario. **Tabla de Controles Implementados**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Flujo | Tipo de Control | | Herramienta/Implementación | Objetivo |
| Inicio de sesión | Preventivo | | Validación de RUT + PIN local | |  | | --- | |  |  |  | | --- | | Asociar cada acción a un usuario autenticado | |
| Gestión de productos y proveedores | Preventivo | | |  | | --- | |  |  |  | | --- | | Validación cruzada y asociación obligatoria | | Asegurar que cada producto tenga proveedores válidos registrados |
| Registro de consumo | Correctivo | | |  | | --- | |  |  |  | | --- | | Formulario estructurado y registro histórico | | Permitir cálculos basados en datos reales de egresos |
| Generación de pedidos sugeridos | Automatizado | | Motor interno basado en stock mínimo, consumo y lead time | Reponer productos de forma anticipada y precisa |
| Exportación de pedidos | Automatizado | | |  | | --- | |  |  |  | | --- | | Generación de Excel o PDF con librerías específicas | | Facilitar el envío de pedidos a proveedores |
| Alertas | Preventivo | Evaluación continua de stock actual vs. mínimo y retrasos | | Anticiparse a quiebres de stock o problemas con proveedores |

**Funcionamiento general de la solución**

1. **Login**: El sistema se inicia con una pantalla de autenticación donde el trabajador ingresa su RUT y PIN. Solo usuarios verificados pueden operar el sistema.
2. **Gestión de productos y proveedores**: Se permite registrar productos y asociarlos a proveedores definidos, incluyendo precios, lead time y condiciones de compra.
3. **Ingreso de consumos**: Los egresos de productos (ventas o uso) se registran de forma estructurada, alimentando el historial de consumo por producto.
4. **Motor de sugerencia**: Periódicamente, o bajo demanda, el sistema evalúa los consumos recientes y el stock actual para sugerir pedidos automáticos, considerando cuántas unidades pedir y a qué proveedor.
5. **Exportación de pedidos**: Con un clic, el pedido sugerido puede exportarse como archivo Excel o PDF con formato profesional, listo para ser enviado por correo u otros medios.
6. **Alertas visuales**: Se notificará al usuario si algún producto está cerca de su stock mínimo o si un proveedor presenta demoras frecuentes, permitiendo anticiparse a riesgos logísticos.

Esta solución busca equilibrar **automatización y simplicidad**, entregando a las PYMES una herramienta útil, liviana y de rápida implementación, sin necesidad de conexión a internet ni infraestructura costosa. Su diseño modular también permitirá futuras mejoras o integración con sistemas de ventas y contabilidad.

**Diagrama de contexto StockSeguro:**

Diagrama

El contenido generado por IA puede ser incorrecto.

**6. Plan de Proyecto**

El proyecto **StockSeguro** será desarrollado en un período de **12 semanas**, dividido en dos fases: una fase de planificación y diseño, y otra de implementación técnica y validación. El plan considera actividades claves, entregables parciales, instancias de control y depuración, utilizando una metodología ágil e incremental.

**Plan de Proyecto (resumen por semana)**

| **Semana** | **Actividad principal** | **Entregable clave** |
| --- | --- | --- |
| Semana 1 | Definición del problema, objetivos y alcance | Documento base del anteproyecto |
| Semana 2 | Diseño técnico: arquitectura, módulos, entidades | Boceto técnico y diagrama de contexto |
| Semana 3 | Desarrollo del login con autenticación RUT + PIN | Módulo funcional de inicio de sesión |
| Semana 4 | Módulo de productos y proveedores (alta, edición, validación) | Registro y asociación producto-proveedor |
| Semana 5 | Módulo de registro de consumos históricos | Registro funcional con historial por producto |
| Semana 6 | Implementación del motor de sugerencias de reposición | Lógica funcional basada en stock mínimo y consumo |
| Semana 7 | Módulo de generación de pedidos automáticos | Lista sugerida con cantidades y proveedor asignado |
| Semana 8 | Exportación de pedidos en Excel/PDF | Archivos exportables correctamente formateados |
| Semana 9 | Sistema de alertas por bajo stock o retrasos | Notificaciones visuales en pantalla |
| Semana 10 | Pruebas integradas del sistema completo | Versión funcional 1.0 con pruebas internas |
| Semana 11 | Corrección de errores y ajustes finales | Sistema refinado, libre de errores críticos |
| Semana 12 | Documentación técnica y manual de usuario | Informe final + entregable completo del software |

**Metodología de desarrollo**

Se empleará una metodología **ágil e iterativa**, basada en ciclos semanales de construcción-validación. Las herramientas principales serán:

* **Trello** para planificación visual de tareas.
* **Git + GitHub** para control de versiones.
* **Checklist funcional** para validar cada módulo.
* **Revisión cruzada** (simulada) para control visual y de usabilidad.

**7. Plan de Monitoreo y Control**

Para asegurar el cumplimiento de los objetivos y la calidad del desarrollo del sistema *ControlRepro*, se implementará un plan de monitoreo y control basado en **revisiones periódicas**, **herramientas de gestión de tareas**, **control de versiones** y **pruebas funcionales**. Este enfoque permitirá detectar desviaciones a tiempo, ajustar entregables y garantizar un producto funcional y estable

**Frecuencia de monitoreo**

* **Semanal**, durante las 12 semanas del desarrollo.
* Cada semana finalizará con una **revisión del entregable funcional correspondiente** y su validación según criterios definidos.

**Tipo de controles aplicados**

| **Tipo de control** | **Descripción** | **Herramienta utilizada** |
| --- | --- | --- |
| Control de avance | Revisión de tareas completadas vs. cronograma semanal | Trello (tablero de tareas) |
| Control de versiones | Registro y respaldo de cambios en el código fuente | Git + GitHub |
| Control funcional | Verificación de funcionamiento de cada módulo mediante pruebas internas | Checklist funcional |
| Control visual | Validación del diseño de la interfaz y experiencia de usuario | Criterios predefinidos + revisión cruzada |
| Control documental | Revisión de redacción, estructura y claridad del manual y del informe | Rúbrica + lector externo simulado |

**Mecanismo de retroalimentación**

* Cada semana se completará una **lista de validación interna**, confirmando si los requisitos funcionales de ese módulo han sido satisfechos.
* Se mantendrá una **bitácora de errores y soluciones** para registrar cualquier desviación detectada y su corrección.
* En caso de bloqueos técnicos o desajustes de planificación, se ajustará el cronograma de manera flexible, priorizando el cumplimiento de las funcionalidades clave.

**8. Materiales, Métodos y Herramientas**

Esta sección detalla los recursos técnicos, metodológicos y tecnológicos que serán utilizados durante el desarrollo e implementación del sistema *ControlRepro*. Se privilegian herramientas multiplataforma, de libre acceso y de bajo requerimiento técnico, ideales para un entorno PYME y un desarrollo llevado por una sola persona.

**Materiales y herramientas de desarrollo**

| **Elemento** | **Herramienta/Descripción** | **Finalidad** |
| --- | --- | --- |
| **Lenguaje principal** | JavaScript | Desarrollo de la lógica de negocio y la interfaz |
| **Framework frontend** | React.js o Vue.js | Creación de la interfaz gráfica del usuario (SPA) |
| **Framework backend** | Node.js + Express.js | Gestión de rutas, controladores y lógica de negocio |
| **Base de datos** | SQLite o PostgreSQL | Almacenamiento local de productos, consumos y proveedores |
| **Exportación de pedidos** | Librerías como xlsx o pdfkit | Generación automática de archivos Excel o PDF |
| **Control de versiones** | Git + GitHub | Historial de cambios, respaldo y seguimiento del proyecto |
| **Gestión de tareas** | Trello | Organización visual y secuencial del desarrollo por semanas |
| **Editor de código** | Visual Studio Code | Programación, pruebas y depuración de módulos |
| **Estilo visual** | Tailwind CSS o Bootstrap | Desarrollo rápido de interfaces limpias y responsivas |

**Modelo de desarrollo**

Se utilizará una **arquitectura limpia (Clean Architecture)**, que permite separar claramente las responsabilidades entre:

* **Capa de presentación** (interfaz de usuario)
* **Capa de aplicación** (reglas de uso)
* **Capa de dominio** (modelo de negocio)
* **Capa de infraestructura** (base de datos, archivos)

Este enfoque mejora la mantenibilidad, facilita pruebas unitarias y permite la futura expansión del sistema (por ejemplo, agregar IA para sugerencias más complejas).

**Método de trabajo**

El método será **ágil e incremental**, con entregas funcionales semanales. Cada iteración incluirá:

1. Implementación del módulo según cronograma.
2. Validación funcional mediante pruebas manuales.
3. Registro del avance y errores en Trello y bitácora.
4. Control de cambios con Git.
5. Revisión visual y funcional antes del cierre semanal.

**9. Bibliografía**

1. Bragg, S. M. (2020). *Inventory Best Practices* (2nd ed.). Wiley.
2. Guía técnica con buenas prácticas para la gestión de inventario y control de stock.
3. Gackenheimer, C. (2014). *Introduction to Electron: Build cross-platform desktop apps with JavaScript, HTML, and CSS*. Apress.
4. Introducción al desarrollo de aplicaciones multiplataforma de escritorio.
5. Ramírez, J. (2022). *Transformación digital en pequeñas empresas: desafíos y oportunidades*. Revista Iberoamericana de Innovación, 15(2), 89–104.
6. Artículo sobre cómo las PYMES pueden adaptarse tecnológicamente de forma gradual y eficiente.
7. Instituto Nacional de Estadísticas de Chile (INE). (2021). *Radiografía de las PYMES en Chile*. Recuperado de <https://www.ine.cl>
8. Datos contextuales sobre la realidad operativa y tecnológica de las pequeñas empresas chilenas.
9. Singh, V. (2018). *Lean Inventory Management*. Lean Enterprise Academy.
10. Enfoque de gestión ajustada aplicado al manejo de inventarios y reposiciones estratégicas.
11. Chopra, S., & Meindl, P. (2019). *Supply Chain Management: Strategy, Planning, and Operation* (7th ed.). Pearson.
12. Fundamentos sobre planificación de la reposición y relación con proveedores dentro de la cadena logística.
13. Microsoft. (2024). *Node.js and JavaScript developer documentation*. Recuperado de <https://learn.microsoft.com/en-us/javascript/>
14. Documentación técnica sobre el uso de JavaScript y Node.js para aplicaciones web y backend.
15. Vue.js. (2023). *Introduction — Vue.js*. Recuperado de <https://vuejs.org/guide/introduction.html>
16. Documentación oficial para desarrollo frontend con Vue.js.
17. Tailwind CSS. (2024). *Documentation*. Recuperado de <https://tailwindcss.com/docs>
18. Guía de diseño para construir interfaces limpias y modernas con Tailwind.