



Departamento de Informática
Universidad Técnica Federico Santa María



Informe de Proyecto – INF-225-2018-1- CSJ
Proyecto: Plataforma de Inventario GPI

29-Agosto-2018

Integrantes:

Nombres y Apellidos	Email	ROL USM
Leonardo Astudillo Villalón	leonardo.astudillov@sansano.usm.cl	201573598-0
Juan Pablo Jorquera Zapata	juan.jorqueraz@sansano.usm.cl	201573533-6
Eliecer Zambrano Reyes	eliecer.zambrano@sansano.usm.cl	201573540-9

1. Requisitos clave (Final)

1.1-Historias de usuario.

Respecto a las historias de usuario, se mantienen las anteriormente descritas en el Entregable 2, con leves modificaciones respecto a las implementaciones llevadas a cabo.

Bodeguero de obra solicita materiales:

Yo como bodeguero de obras quiero enviar solicitudes de materiales al bodeguero central para terminar el trabajo.

Conversación: El bodeguero de obra está constantemente necesitando materiales para completar dicha obra, por lo que es necesario que pueda enviar solicitudes al bodeguero central.

Criterios de aceptación:

- Se debe poder enviar solicitudes
- Bodeguero central recibe notificación de solicitudes

*Comentario extra: No se ha implementado la lógica tras la confirmación de recepción mediante mensajes.

Bodeguero recibe notificaciones:

Yo como bodeguero quiero que, al momento de agotarse el stock de materiales en bodega, pueda recibir una notificación indicando qué materiales faltan, para así conseguir nuevos y mantener el stock lleno.

Conversación: El bodeguero necesita para una manera de mantenerse informado respecto a la situación actual que se encuentran los materiales en bodega. En caso de encontrarse sin stock algún material, y no se notifica, el bodeguero podría eventualmente retrasar sus funcionalidades.

Criterios de aceptación:

- Si la bodega se encuentra con un stock de cero elementos en algún material, se envía notificación a el bodeguero.

Bodeguero central actualiza bodegas:

Yo como bodeguero central quiero enviar materiales a bodegas que los requieran, para mantenerlas abastecidas.

Conversación: El bodeguero central es informado o recibe solicitudes sobre la falta materiales para alguna obra, por lo que es necesario que, dadas las facultades que posee dentro de GPI, pueda enviar y así distribuir los materiales hacia las bodegas.

Criterios de aceptación:

- Dado una determinada bodega que requiera materiales, el bodeguero central puede enviar materiales con nombre y cantidad.

Bodeguero revisa inventario:

Yo como bodeguero central (o de obra) quiero revisar los materiales disponibles en el inventario para mantener al tanto del stock de materiales, y así avisar tomar medidas de manera oportuna cuando sea necesario pedir nuevos insumos.

Conversación: El bodeguero necesita poder mantenerse al tanto de la cantidad de materiales en el inventario cuando se soliciten en las obras, comprobar el stock de este y pedir nuevos materiales si se requiere.

Criterios de aceptación:

- Se debe mostrar un listado de los materiales con sus respectivas cantidades.
- Debe ser capaz de actualizar el stock a medida que van llegando.

Bodeguero modifica inventario:

Yo como bodeguero quiero modificar los materiales ingresados con anterioridad al sistema, para actualizar si es necesario los datos referidos a cada uno.

Conversación: El bodeguero puede, además de agregar nuevos materiales, modificar aquellos inventariados con antelación. Ya sea agregando o eliminado del sistema, se debe entregar la opción de manera efectiva para él, actualizando el sistema.

Criterios de aceptación:

- Dado un listado de materiales, si el bodeguero selecciona modificar materiales, se despliega una opción con datos a modificar la cantidad de materiales.

Orden de materiales:

Yo como encargado de compras quiero un respaldo de las órdenes de compra para poder guardar un historial de los materiales solicitados.

Conversación: Uno de los principales problemas evidenciados es la incongruencia respecto de los materiales enviados por los proveedores vs los comprados, por lo cual, el encargado de compras necesita una manera de mantener un registro en el sistema de lo obtenido.

Criterios de aceptación:

- Se debe poder agregar una nueva orden, incluyendo los materiales asociados.
- Puede visualizar las distintas órdenes realizadas.

*Comentario extra: Se elimina los criterios de aceptación “Se debe poder informar sobre los materiales recibidos” y “Se debe notificar en caso de haber errores o inconsistencias en el pedido” ya que escapa del alcance del software implementado (relación con proveedores externos).

Petición de compra:

Yo como bodeguero central quiero mandar solicitud de nuevos materiales para poder rellenar mi stock de productos.

Conversación: Al requerir materiales, el bodeguero central debe enviar una solicitud al encargado de compra incluyendo dichos materiales, quien se encarga después de la compra.

Criterios de aceptación:

- Debe poder llenar un formulario con los materiales requeridos.
- Se muestra el estado de la petición.

Tabla 1: Requisitos funcionales.

Req. funcional	Descripción
Pedir insumos en obra.	El bodeguero de obra debe permitir solicitar más materiales, a bodega central o proveedores.
Bodeguero de obra revisa materiales	Se debe mostrar el inventario de obra al bodeguero encargado de ella cuando lo solicite.
Bodeguero central ve todos los materiales	Se debe mostrar los items de bodega central al bodeguero centra.
Notificar petición de materiales.	Al pedir materiales a terreno se debe avisar al bodeguero central de la nueva solicitud.
Notificar stock de materiales.	Al acabarse algún material, hay que notificar al bodeguero correspondiente.
Respaldo recepción de productos.	En obra hay que dejar constancia al momento de recibir los productos.
Agregar materiales de órdenes de trabajo.	El encargado de compra debe listar los materiales solicitados en una orden.
Identificar bodega del material.	Se permite ver en el inventario en cuál bodega se encuentran los materiales en caso de haber.
Modificar materiales inventariados	El bodeguero debe poder modificar los materiales ingresados en bodega.

Tabla 3: Requisitos extra-funcionales (antiguos)

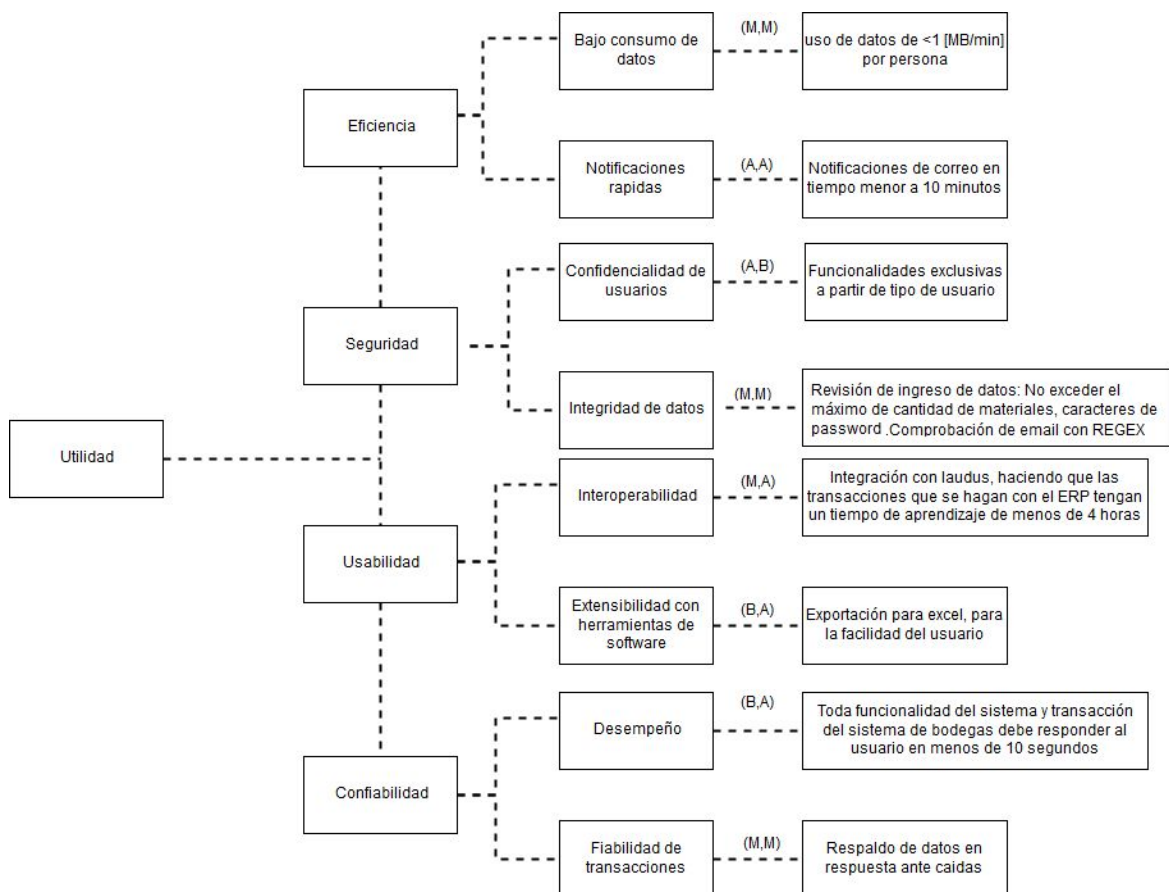
Req. extra-funcional	Descripción y medición
[Interoperabilidad] Compatibilidad con el ERP Laudus	El software en desarrollo estará coordinado con el ERP Laudus a partir de un API que genera este, haciendo que el software pueda ser de fácil lectura para el usuario que maneje el ERP, haciendo que el periodo para adaptarse sea menor a 4 horas.
[Confidencialidad] Encapsulamiento de funcionalidades dependiendo del usuario	Cada usuario identificado podrá usar sus funcionalidades exclusivas a partir de su puesto.
[Eficiencia] Plataforma de bajo consumo	La plataforma requerirá de pocos uso de datos (<1 [MB/min] por persona) .
[Fiabilidad] Respaldo de datos cuando no haya conexión estable	La plataforma tendrá un respaldo de datos, en caso de que la conexión esté inestable.
[Extensibilidad] Herramientas actuales de trabajo	Capacidad para extenderse al software que se usa actualmente (Excel).
[Eficiencia] Notificaciones rápidas	Notificar adecuadamente al hacer nuevas peticiones por correo (<10min)

[Desempeño] Rápida	Respuesta	Las Transacciones hechas por el usuario se demorarán menos de 10 segundos
-----------------------	-----------	---

Tabla 2: Requisitos extra-funcionales (actualizados)

Req. extra-funcional	Descripción y medición
[Interoperabilidad] Compatibilidad con el ERP Laudus	El software en desarrollo estará coordinado con el ERP Laudus a partir de un API que éste genera, haciendo que el software pueda ser de fácil lectura para el usuario que maneje el ERP, haciendo que el periodo para adaptarse sea menor a 4 horas.
[Confidencialidad] Encapsulamiento de funcionalidades dependiendo del usuario	Cada usuario identificado podrá usar sus funcionalidades exclusivas a partir de su puesto.
[Eficiencia] Plataforma de bajo consumo	La plataforma requerirá de pocos uso de datos (<1 [MB/min] por persona) .
[Fiabilidad] Respaldo de datos cuando no haya conexión estable	La plataforma tendrá un respaldo de datos, en caso de que la conexión esté inestable.
[Extensibilidad] Herramientas actuales de trabajo	Capacidad para extenderse al software que se usa actualmente (Excel).
[Eficiencia] Notificaciones rápidas	Notificar adecuadamente al hacer nuevas peticiones por correo (<10min)
[Desempeño] Respuesta Rápida	Las Transacciones hechas por el usuario se demorarán menos de 10 segundos
[Integridad] Revisión de datos	Los datos, como cantidad de caracteres de password y cantidad de materiales a ingresar, no pueden exceder el límite impuesto. Para el registro de usuarios, el email se comprobará si es válido con REGEX .

2. Árbol de Utilidad (Actualizado)



3. Modelo de Software

Modelo de dominio (Actualizado)

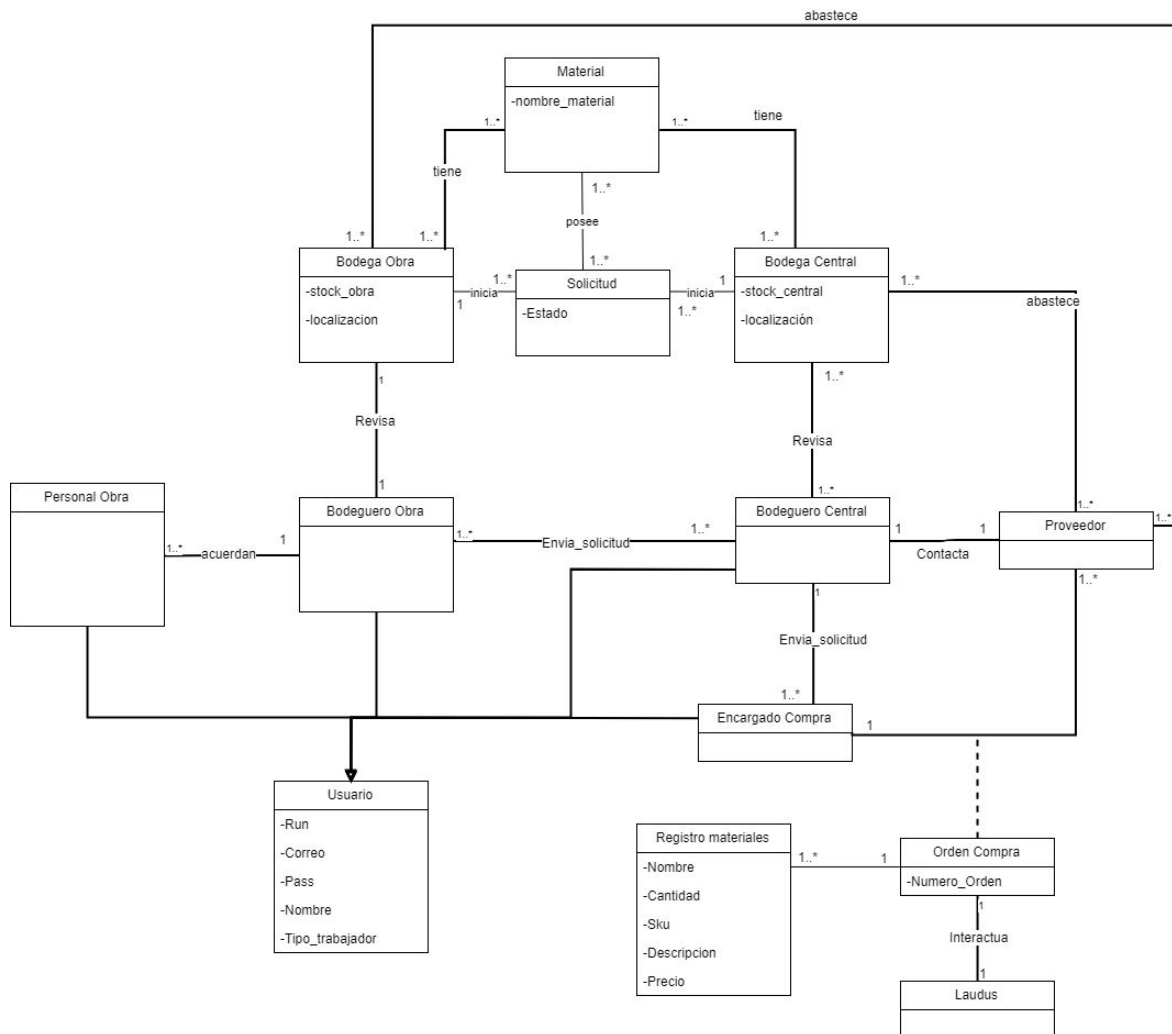
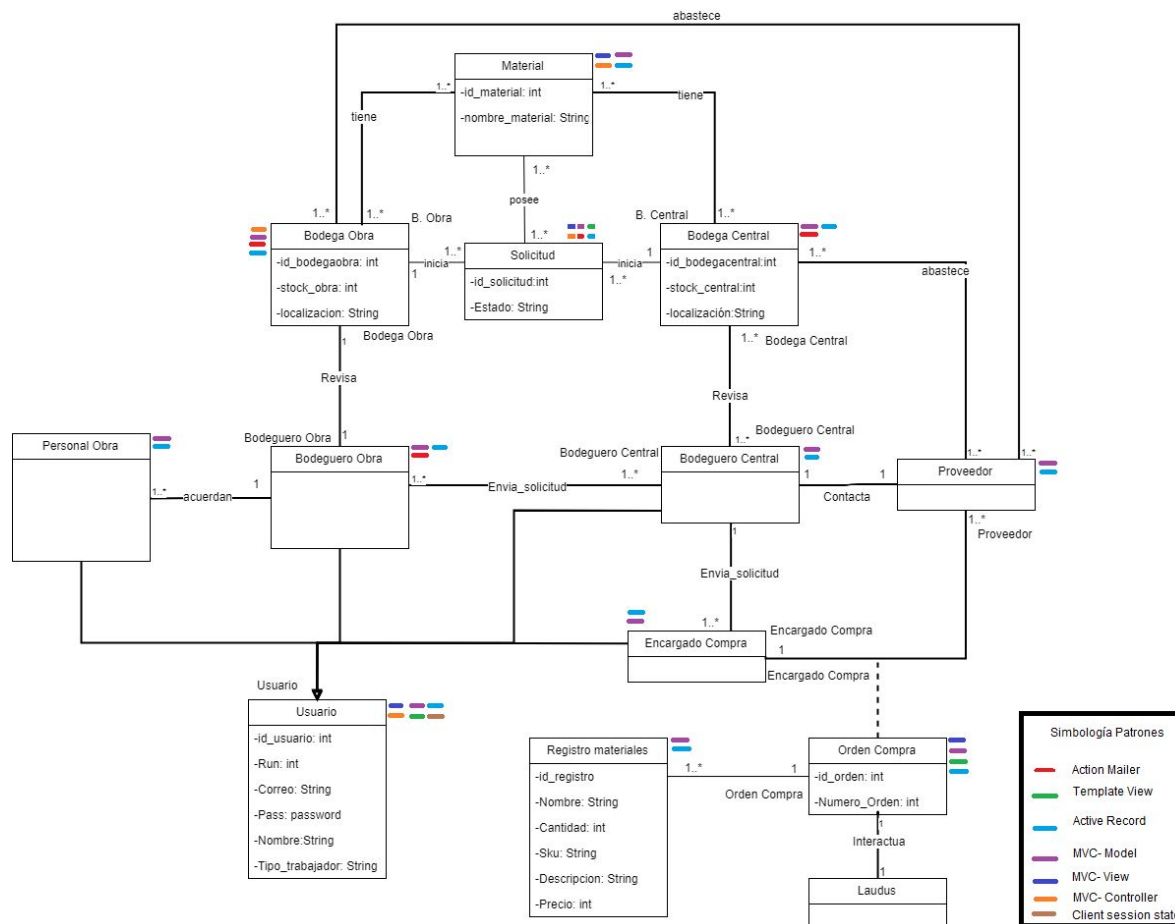


Diagrama de clases



Consideraciones: Dado que el patrón de diseño MVC puede ser evidenciado con diferentes niveles según la clase, se dividió en tres partes: Model, View y Controller. Así se especifica cada clase con que parte de MVC corresponde. Para la clase Laudus, no se muestra patrón de diseño dado que sólo es una representación del agente externo que interactúa con el sistema implementado, en el cual, como se menciona más adelante, se utiliza una API auxiliar para establecer la comunicación externa con el software.

Tabla 3: Selección de Patrones

Intención	Patrón de Diseño	Razonamiento
Presentar una interfaz agradable, sencilla, con un manejo eficaz de cada componente	Model-View-Controller (MVC)	MVC permite dividir eficazmente el problema en modelo, vista y controlador,

		permitiendo usar Ruby on Rails para su implementación. Se realiza la división en dichos componentes, para visualizar en mayor profundidad el patrón de diseño al momento de analizar el software.
Acceso fácil al modelo de la base de datos	Active Record	Active Record permite hacer CRUD con el modelo de negocio (base de datos) de manera sencilla e instantánea
Insertar elementos en un documento HTML.	Template View	Template View permite editar un HTML insertando datos que de otra forma sería complejo de realizar
Enviar notificaciones a través de correos, como por ejemplo, en Bodega_obra.	Action Mailer	Action Mailer permite configurar el envío de correos a quien se especifique (bodeguero central por ejemplo).
Evitar el reingreso de datos para un usuario frecuente en la aplicación.	Client Session State	Client Session State permite mantener una sesión abierta para el usuario cuando quiera reingresar al sistema.

4. Trade-offs entre tecnologías

A continuación se presenta el diagrama de trade-offs.

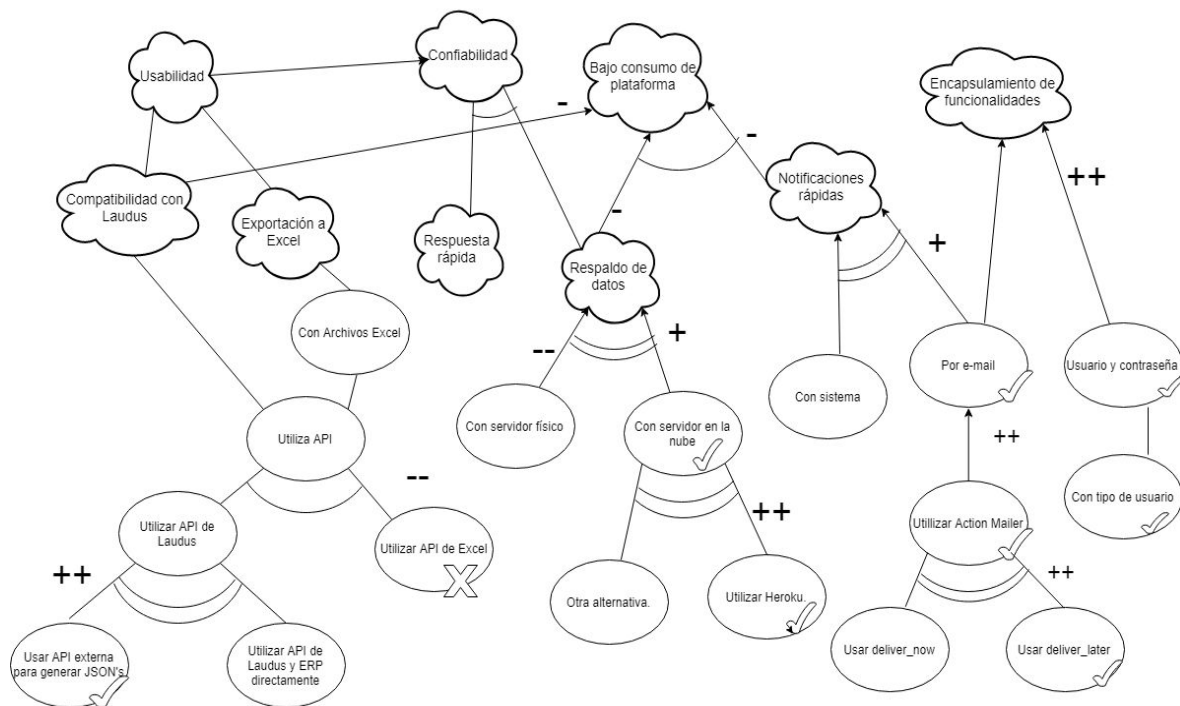


Ilustración 2: Ejemplo de Softgoal Interdependency Graph (SIG)

Tabla 9: Trade-offs entre opciones tecnológicas

A continuación se presentan las principales justificaciones respecto a las decisiones tomadas según los softgoals a cumplir. Se omiten algunos razonamientos para decisiones tomadas en el entregable anterior y se dejan los más relevantes junto a nuevas decisiones tomadas:

Decisión	Softgoal	Evaluación	Razonamiento
Utilizar servidor en la nube	Respaldo de datos.	+	Un servidor para guardar datos en internet otorga mayor flexibilidad y bajo costo para la empresa. Se utiliza

			Heroku para implementar la aplicación
Utilizar e-mail para las notificaciones	Notificaciones rápidas.	+	Se observa que en GPI, la forma usual de recibir información es a través de e-mail, por lo que es más favorable para su estilo de trabajo utilizar dicho método.
Utilizar Action Mailer	Notificaciones rápidas	++	Dado el uso de Ruby on Rails en la aplicación, y a la importante utilidad de los correos, se utiliza Action Mailer para proveer el servicio de envío de notificación..
Notificaciones rápidas	Bajo consumo de plataforma.	-	Podría generar un alto consumo si se realizan muchas notificaciones, por lo que sólo se evidencia un ”-”.
Compatibilidad con Laudus	Bajo consumo de plataforma.	-	Laudus podría generar un alto consumo de datos, por lo que se debe analizar realmente su uso en la aplicación
Usar API externa para generar JSON's	Compatibilidad con Laudus	++	Debido a que se tiene poco conocimiento y acceso a Laudus, se utiliza una API para simular dicho ERP, que generará JSON's con los datos necesarios y que serán interpretados por la aplicación.
Utilizar API de Excel	Exportación a Excel	--	Se desecha la idea de manipular Excel. Dado al previo análisis de requisitos extrafuncionales, posee una dificultad alta y prioridad baja, por lo que se puede descartar para entregar mayor esfuerzo a otras tareas.
Utilizar deliver_later	Notificaciones rápidas	++	Se escoge el método deliver_later para que el tiempo en enviar la notificación no sea notorio para el usuario, encolando el trabajo de enviar notificación hasta cuando se tenga la oportunidad de hacerlo.

5. Deuda Técnica

Ítem deuda técnica	Razonamiento	Impacto
Ingreso materiales de a un sólo elemento	Fácil de utilizar sin necesidad de manejar Javascript	La configuración será poco dirigida al problema actual de GPI.
Utilización de API para simular Laudus	Permite simular el trabajo con un ERP similar a Laudus, logrando interactuar con un agente externo y sin necesidad de relacionarse con Laudus.	Se necesitará comprender y analizar el ERP Laudus nuevamente cuando se necesite implementar, cambiando configuraciones internas.
Registro de usuario no se realiza al regresar a página anterior	Solo se debe ingresar un nuevo usuario a la base de datos si no se regresa a la página anterior, dado a la falta de conocimientos en formularios,	Se creará una lógica de negocio poco consistente.
Alertas mediante Javascript al momento del envío de materiales por el bodeguero central.	Mucho más simple de implementar respecto al flash de Rails para facilitar el manejo de errores con Javascript.	Se observa inconsistencia respecto al resto de la página, no respetando la línea general del código.