



**Departamento de Informática**  
Universidad Técnica Federico Santa María



**Informe de Proyecto – INF-225-2018-1-CSJ**  
**Proyecto Talaveras**  
**05/08/2018**

Integrantes:

Nombres y Apellidos	Email	ROL USM
Patricio Aguirre Cornejo	patricio.aguirre@sansano.us.cl	201573575-1
Jose Chavez	jose.chavezg@sansano.usm.cl	201573589-1
Ignacio Cisternas	ignacio.cisternasn@sansano.usm.cl	201573544-1

Contenido del Informe

<i>Portada</i>	<i>1</i>
<i>Contenido del Informe a Entregar</i>	<i>2</i>
<b>1. Requisitos clave (Actualizado)</b>	<b>3</b>
<b>2. Árbol de Utilidad (Actualizado)</b>	<b>3</b>
<b>3. Modelo de Software</b>	<b>4</b>
<b>4. Trade-offs entre tecnologías</b>	<b>5</b>

## 1. Requisitos clave (Actualizado)

En la entrega 1 se presentaron los siguientes requisitos funcionales y extra-funcionales:

**Tabla 1: Requisitos funcionales entregable anterior**

<b>Req. funcional</b>	<b>Descripción y medición (máximo 2 líneas)</b>
Registro	Podrán crearse cuentas asociadas a cada trabajador de GPI, diferenciando el cargo de este, min 50 cuentas.
Log-in	Al ingresar con una cuenta registrada, el sistema mostrará las vistas correspondientes al cargo en 10 seg o menos.
Solicitud de materiales	Cada petición especificada tendrá un trabajo asociado, materiales pedidos y un estado. Todos visibles.
Solicitud de compra	Indicarán los materiales pedidos y el bodeguero solicitante.
Actualización estado	Cuenta de bodegueros podrán actualizar estado de solicitud de material dentro de las primeras 24 hrs de llegada.
Verificación de material	Al procesar una solicitud de material, verificar si existen los materiales dentro de la bodega de la empresa en tiempo real.
Log-out	Cada cuenta que ingresa tendrá la opción de cerrar la sesión.

**Tabla 2: Requisitos extra-funcionales entregable anterior**

<b>Req. extra-funcional</b>	<b>Descripción y medición (máximo 2 líneas)</b>
Desempeño	Las actualizaciones de estado de las Solicitudes de material sean inmediatas o dentro de los primeros 10 min.
Disponibilidad	El software sea capaz de soportar 10 usuarios simultáneamente sin caída.
Escalabilidad	Podrá crecer hasta 100 usuarios simultáneos.
Usabilidad	El envío de una solicitud de alta prioridad llegue con 3 hrs o más tiempo de retraso.

Dado lo anterior, se realizaron unos cambios a medida que avanzaba el desarrollo del software requerido por el cliente, los cuales se presentan a continuación separado en 2 tablas nuevamente, uno para los requisitos funcionales y otra para los extra-funcionales:

**Tabla 3: Requisitos funcionales (actualizados)**

<b>Req. funcional</b>	<b>Descripción y medición (máximo 2 líneas)</b>
Administración de usuario	Un nuevo usuario podrá registrarse en la página. Un usuario ya registrado podrá realizar Log-in y Log-out.
Solicitud de materiales	Cada petición especificada tendrá un trabajo asociado, materiales pedidos y un estado. Todos visibles.
Solicitud de compra	Indicarán los materiales pedidos con su precio y el bodeguero solicitante, dentro de la orden de compra.
Actualización estado	Cuenta de bodegueros podrán actualizar estado de solicitud de material dentro de las primeras 24 hrs de llegada.
Verificación de material	Al procesar una solicitud de material, verificar si existen los materiales dentro de la bodega de la empresa en tiempo real.
Comprar Material	El encargado de adquisiciones, enviará la orden de compra a LAUDUS, encargado de procesar la compra.

En comparación a los requisitos funcionales presentados en el entregable 1, se realizaron específicamente 2 cambios, se combinaron los requisitos de Registro, Log-in y Log-out en uno solo llamado Administración de usuario, debido a que estos no son tan fundamentales en el proceso de solución al problema del cliente y representan un solo requisito funcional. Además se creó el Requisito funcional Comprar material, ya que como grupo nos dimos cuenta que con los existentes solo se estaba generando un solicitud de compra, pero no era enviada a nadie en específico para que se efectuara la compra dentro del programa de los materiales. Por último se cambió un poco la descripción de Solicitud de material, para dejar más en claro a que se refiere.

Tabla 4: Requisitos extra-funcionales (actualizados)

Req. extra-funcional	Descripción y medición (máximo 2 líneas)
Rendimiento	Las actualizaciones de estado de las Solicitudes de material sean visibles en < 10 min.
Disponibilidad	El software sea capaz de soportar 10 usuarios simultáneamente sin caída.
Escalabilidad	Podrá crecer hasta 100 usuarios simultáneos dentro de 6 meses.
Rendimiento	Al solicitar materiales, la solicitud se demore < 30 min. en llegar a la bodega central.
Confiabilidad	La base de datos debe permanecer activa para el sistema 99% del tiempo.
Usabilidad	La creación de una nueva cuenta de usuario sea fácil y demore < 5 min.
Confiabilidad	El sistema debe estar activo las 24 hrs del día para la realización y visualización de pedidos de material.
Seguridad	El administrador del sistema será el único que podrá modificar partes o sectores de este.

En los requisitos extra-funcionales se realizaron varios cambios, debido a que mientras avanzaba el desarrollo del software, las preguntas aumentaban sobre las fallas que podría poseer el sistema. Se eliminó el requisito de usabilidad anterior y se agregó otro más acorde al problema. Además se mejoraron las descripciones de los demás requisitos para una mejor comprensión del problema a tratar. En la parte de Rendimiento se agregaron los tiempos de demora que debería poseer el sistema al momento de procesar solicitudes y datos, a parte del crecimiento que debería poseer el sistema requerido por la clienta. Por otra parte se agregaron requisitos de Confiabilidad, debido a que son desafíos importantes que debería cumplir el sistema, debido a que este debe estar disponible todo el tiempo para así poder planificar de un mejor forma las obras que necesitan de nuevos materiales. Por último se agregó un ítem de seguridad que debe cumplir el software, ya que personas con conocimientos deberían ser capaces de actualizar o cambiar partes de este en un futuro.

## 2. Árbol de Utilidad (Actualizado)

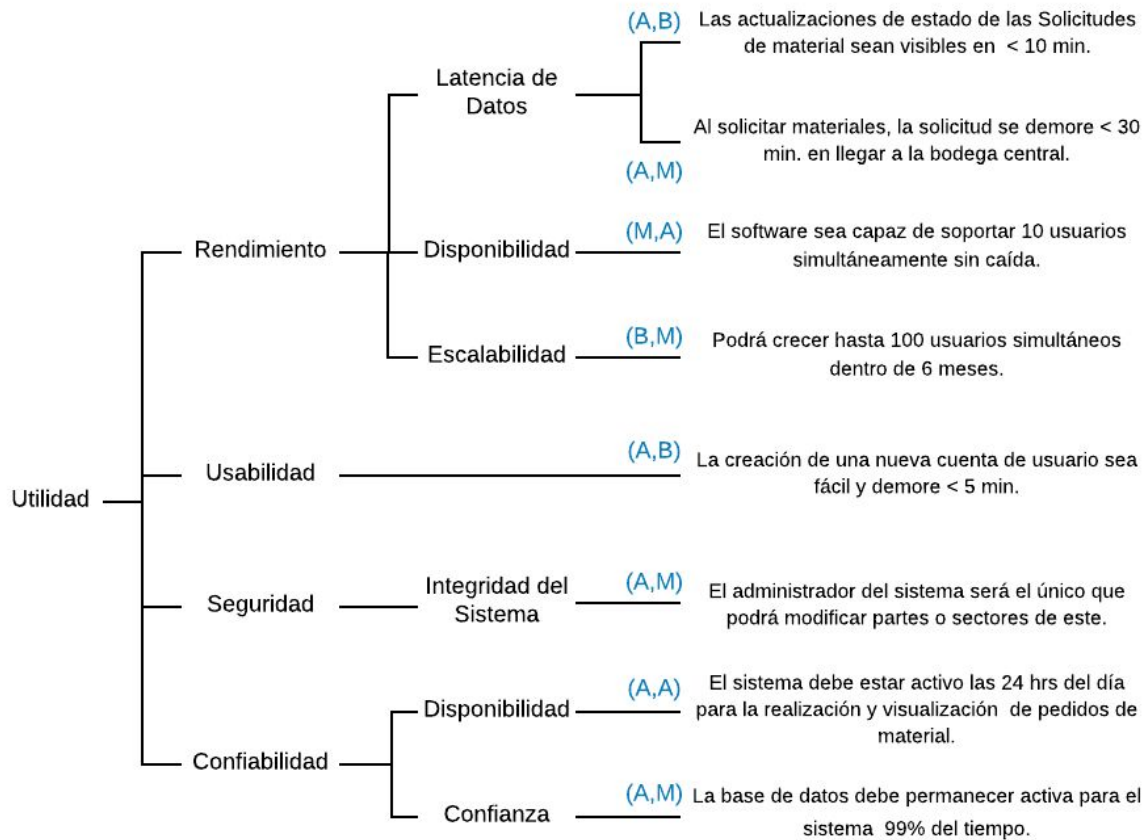


Ilustración 1: Árbol de utilidad (Actualizado)

Como se puede observar en el árbol de utilidad presentado en la Ilustración 1, se clasificó los requisitos extra-funcionales de la tabla 4 presentada en el ítem anterior, a los cuales se les agregó un nivel de importancia y dificultad. Este árbol es una actualización del entregable anterior y demuestra de mejor forma lo que se ha ido observando durante el desarrollo del software.

## 3. Modelo de Software

A continuación se presenta el modelo de software pensado para dar solución al problema del cliente, el cual se deriva del modelo de dominio del entregable 1 mejorado según nuevas consideraciones a las que se llegaron a acuerdo dentro del grupo.

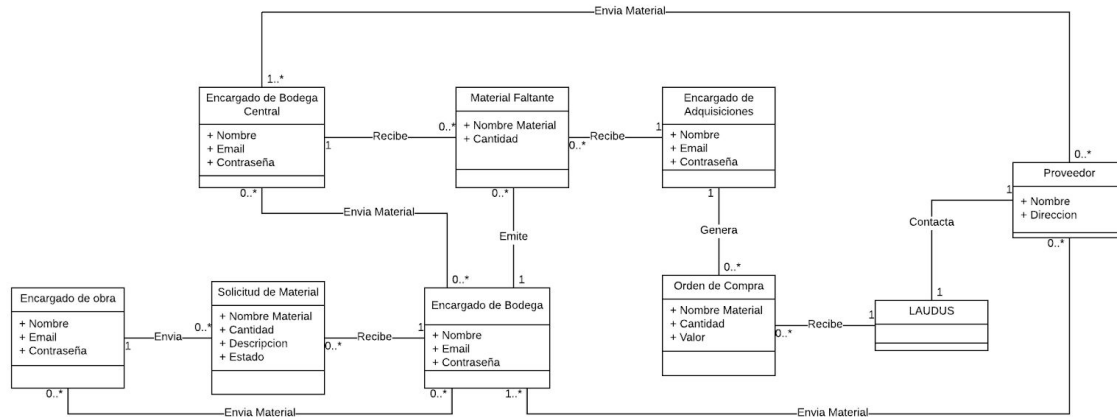


Ilustración 2: Modelo de Software

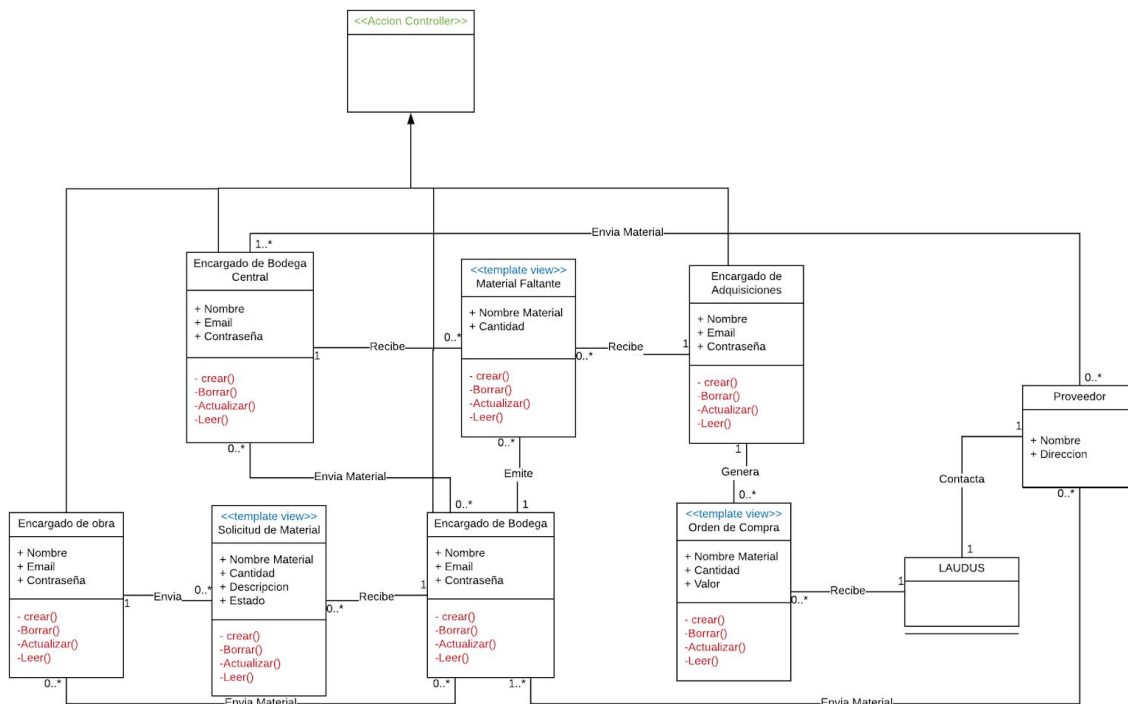


Ilustración 3: Modelo de Software (Implementación de patrones)

Tabla 5: Selección de Patrones

Intención	Patrón de Diseño	Razonamiento
Generar el Software a través de Ruby on Rails para solucionar el problema de solicitud de materiales dentro de la empresa GPI.	Model-View-Controller (MVC)	MVC permite aplicar una solución genérica a este problema planteado por GPI.
Interacción con la base de datos sin la necesidad explícita de consultas SQL.	Active Records	A través de este patrón las consultas SQL no son necesarias y facilitan las operaciones CRUD dentro de la base de datos para cada clase que necesite almacenar datos. Representa la M del MVC utilizado.
Dentro del software hay solicitudes web que necesitan de una plantilla o cierta acción específica a realizar	Action Controller	Ruby on Rails trabaja ya con este controlador, por lo que su uso es más accesible y simple dentro de este MVC.
Dentro del problema se solicitan vistas de datos dinámicos, por lo que las vistas se deben generar al momento de solicitar un dato y no estar previamente definidas o estáticas.	Template View	Las vistas dentro de ruby on rails están sujetas a Templates de HTML, por lo que el proceso se vuelve mucho más simple dentro de la aplicación web si se utiliza esto.



#### 4. Trade-offs entre tecnologías

En esta sección se presenta el SIG que nosotros consideramos de nuestro proyecto, además de una tabla explicativa del razonamiento del porqué las opciones favorecen o desfavorecen el desarrollo de nuestro software a realizar.

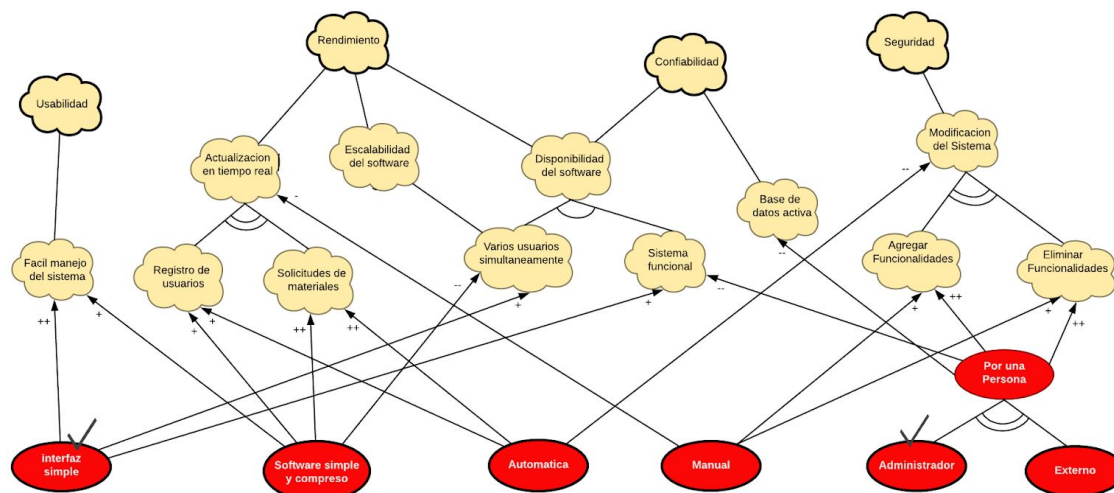


Ilustración 4: Softgoal Interdependency Graph (SIG)

Tabla 6: Trade-offs entre opciones tecnológicas

Decisión	Softgoal	Evaluación	Razonamiento
Interfaz Simple	Fácil manejo del Sistema	++	Una interfaz simple ayudaría a que los usuarios naveguen de forma fácil y rápida a través del sitio.
	Varios usuarios simultáneamente	+	A través de una interfaz con pocos datos y carga, ayuda a que varios usuarios a la vez puedan solicitar datos sin sobrecargar el sistema.
	Sistema funcional	+	El sistema estaría disponible y funcional la mayoría del tiempo y con menos riesgo de caída o lentitud al poseer interfaz simple.
Software simple y compacto	Fácil manejo del sistema	+	Al ser simple y compacto los datos y el sistema se vuelven muy simple de usar para cualquier usuario.

	Registro de usuario	+	Al ser compreso los datos se guardan más rápido dentro de la base de datos y presentaría un registro muy sencillo.
	Solicitudes de Materiales	++	A través de un software simple la petición de solicitudes serán muy rápidas, prácticamente en tiempo real.
	Varios usuarios simultáneamente	--	Un software simple no soporta a muchos usuarios simultáneamente, debido al que ser compreso muchas solicitudes enviadas al mismo tiempo, o mucho usuarios conectados sobrecargan al sistema, volviéndolo lento y sin disponibilidad para otros que requieran su uso.
Automática	Registro de Usuarios	+	Si registrar a un usuario se hace de forma automática y sin la confirmación de alguien, el proceso de los datos es mucho más rápido y expedito.
	Solicitudes de Materiales	++	Al mandar una solicitud de forma automática al sistema el tiempo se ve reducido considerablemente, ya que esta se genera en el instante y facilita la aprobación de esta.
	Modificación del sistema	--	Querer actualizar el software de forma automática, no es muy factible hoy en día, por su alto costo en hardware y software y no aseguran el éxito en las nuevas funcionalidades o al borrar algunas de estas.
Manual	Actualización en tiempo real	-	Desfavorece al software debido a que le agrega tiempo al almacenamiento de datos nuevos y petición de datos, porque una persona o varias de estas deberían estar revisando cada solicitud o registro nuevo, ralentizando al sistema.

	Agregar funcionalidades	+	Favorece al querer agregar nuevas funcionalidades, además de hacerse de forma segura y confiable por una persona que entiende del software.
	Eliminar Funcionalidades	+	Favorece la eliminación de funcionalidades obsoletas por parte del sistema, ya que una persona con conocimientos podría reemplazarla o simplemente sacar alguna de forma segura y sin afectar el rendimiento del software y hardware
Por una persona	Sistema Funcional	--	Que una persona se haga cargo de todo el software, ya sea externa o un administrador de la empresa dificulta la restauración en caso de una caída del sistema.
	Base de datos activa	--	Al igual que en el caso anterior que una persona sea la encargada de este punto, dificulta el mantener activa todo el tiempo un abase de datos y en caso de caída de esta complica en tener solo a una persona a cargo.
	Agregar Funcionalidades	++	Tener a cargo una persona responsable de actualizar funcionalidades del sistema, protege tanto los datos como la integridad misma del sistema, debido a que ya lo conoce y sabe exactamente su funcionamiento.
	Eliminar Funcionalidades	++	Como en el punto anterior se mencionó esto protege los datos y al sistema mismo de no eliminar puntos vitales para el funcionamiento del software.