



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



**Informe de Proyecto – INF-225-2018-1-CSJ**  
**Proyecto Talaveras**  
**2018-08-26**

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>Contenido del Informe</i>	<i>2</i>
<b>1. Requisitos clave (Final)</b>	<b>3</b>
<b>2. Árbol de Utilidad (Final)</b>	<b>3</b>
<b>3. Modelo de Software (Final)</b>	<b>3</b>
<b>4. Trade-offs entre tecnologías (final)</b>	<b>3</b>
<b>5. Deuda técnica incurrida</b>	<b>3</b>

## 1. Requisitos clave (Final)

A continuación se presentan las tablas finales de requisitos del proyecto de ingeniería de software 2018-1, para el cliente GPI.

Los requisitos funcionales que se desarrollaron dentro del proyecto son los que se presentan en la tabla 1, los cuales no presentan cambios respecto al entregable 2, ya que no se encontró necesario agregar o quitar funcionalidades del software para dar solución al problema de la clienta.

Req. funcional	Descripción y medición (máximo 2 líneas)
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.

**Tabla 1: Requisitos funcionales Finales**

En la tabla 2 se presentan los Requisitos extra-funcionales finales que se identificaron para este proyecto y la situación planteada por la cliente al principio del semestre. Durante el desarrollo del software no nos encontramos con más requisitos para incluir desde la entrega 2, por lo que no presenta cambios respecto a la presentada en ese entregable.

<b>Req. extra-funcional</b>	<b>Descripción y medición (máximo 2 líneas)</b>
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.

Tabla 2: Requisitos extra-funcionales Finales

## 2. Árbol de Utilidad (Final)

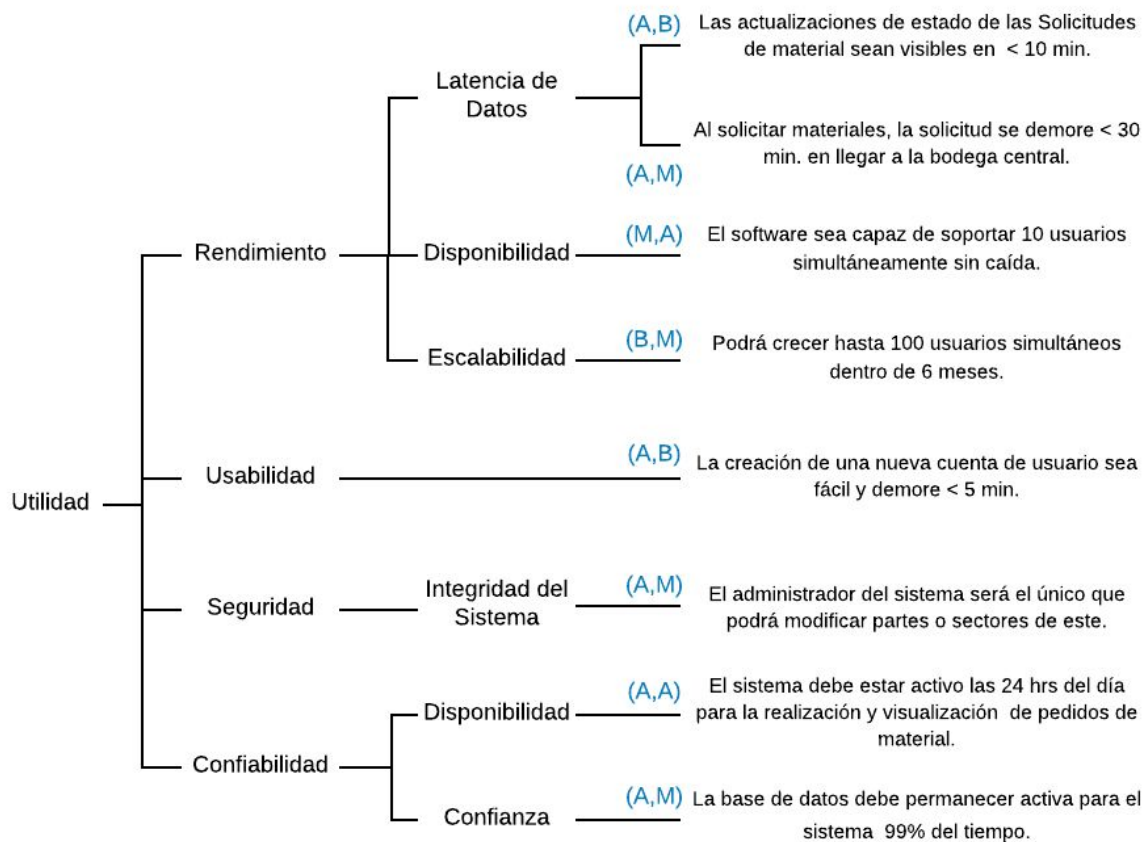
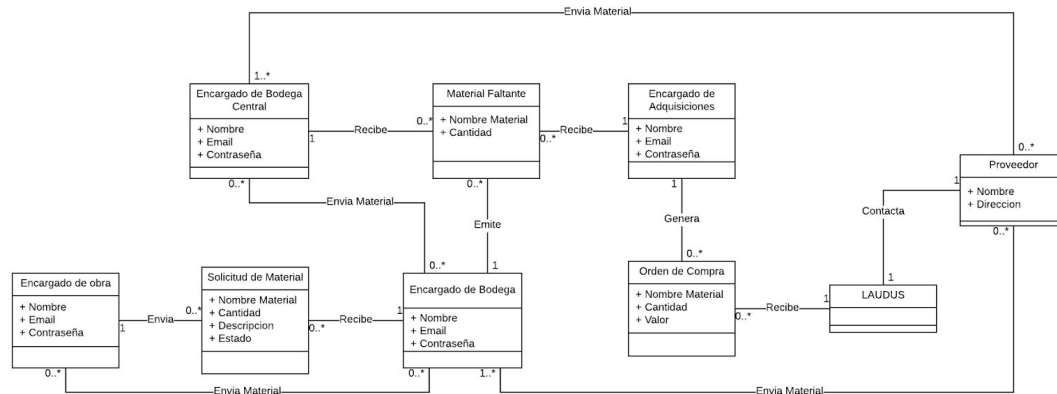


Ilustración 1: Árbol de utilidad

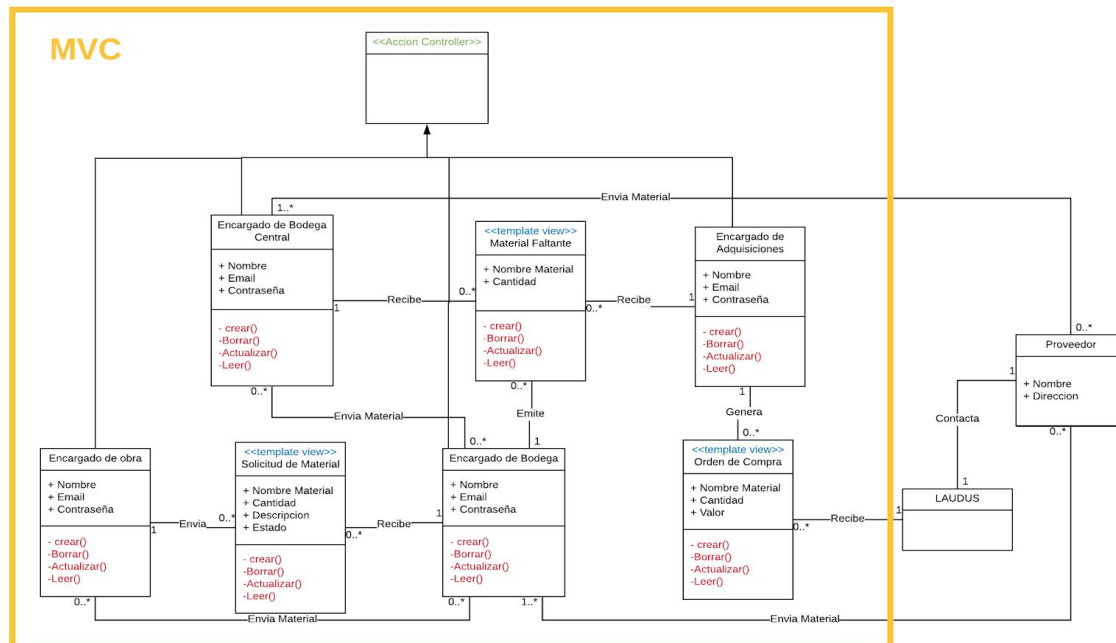
La ilustración 1 muestra el Árbol de utilidad Final del proyecto el cual no tuvo modificaciones debido a que no encontramos cambios en los requisitos extra-funcionales respecto al entregable anterior.

### 3. Modelo de Software (Final)

A continuación se muestra el Modelo de Software Final (Ilustración 2) que se planteó para solucionar el problema que presenta la empresa GPI. En la Ilustración 3 se muestra el mismo modelo pero con la aplicación de los patrones usados en el desarrollo del software, el cual en comparación al de la entrega 2 presenta un mínimo cambio, el detalle en donde está implementado MVC dentro del software. Además se presenta una tabla que explica el motivo y razonamiento de aplicar esos patrones dentro del modelo.



## Ilustración 2: Modelo de Software Final



### Ilustración 3: Modelo de Software Final (patrones aplicados)

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.	<b>Model-View-Controller (MVC)</b>	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.	<b>Active Records</b>	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	<b>Action Controller</b>	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.	<b>Template View</b>	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.

Tabla 3: Selección de Patrones

#### 4. Trade-offs entre tecnologías (final)

En esta sección se presenta el SIG Final (ilustración 4) que nosotros consideramos de nuestro proyecto, además de la tabla 3, la cual es explicativa del razonamiento del porqué las opciones favorecen o desfavorecen el desarrollo de nuestro software a realizar.

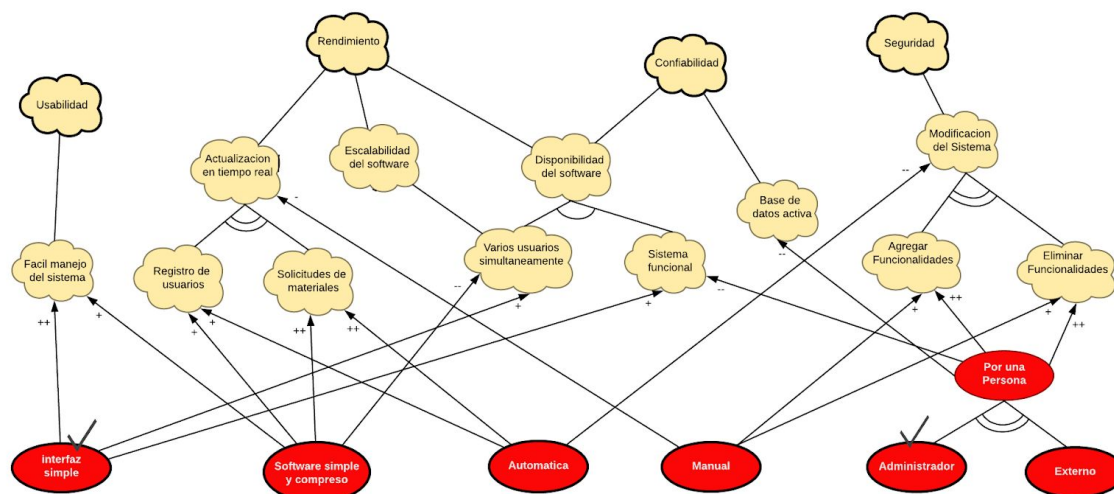


Ilustración 4: Softgoal Interdependency Graph (SIG)

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 compacto 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.

Tabla 4: Trade-offs entre opciones tecnológicas

## 5. Deuda técnica incurrida

En esta sección se presenta la tabla 5, con la cual se pretende dar a conocer la deuda técnica que tendrá el software final a crear por el grupo.

Ítem deuda técnica	Razonamiento	Impacto
Software inseguro	Facilita la programación y por el tiempo no se considera mayor seguridad al implementar funcionalidades.	Agentes externos tienen fácil accesibilidad para alterar datos.
Uso de Base de datos local	Fácil creación y uso al ya venir incorporada en el Framework (RoR).	Dificultades si se quiere asociar a otras APP o uso externo de los datos (ERP).
Simulación de ERP de empresa con API externa insegura	Fácil de implementar y probar en el software.	Al querer asociar la APP con la API real de la empresa, podrían ocurrir problemas de ejecución o rendimiento debido al tamaño y/o estructura del ERP de GPI (LAUDUS).
Inexistente filtro para creación de Cuenta	Se asume que solo colaboradores de GPI tendrán acceso a creación de cuenta dentro del software.	Cualquiera puede acceder y crear una cuenta, por lo que estaría entregando datos incorrecto y alteraría información. Por lo que debiese existir un filtro previo para comprobar que son trabajadores de GPI.

**Tabla 5: Deuda técnica**