



Departamento de Informática
Universidad Técnica Federico Santa María



Informe de Proyecto – INF-225-2018-1-<CSJ>

Proyecto “JIP”

2018-08-05

Integrantes:

Nombres y Apellidos	Email	ROL USM
Pedro Chacón	pedro.chacon.14@sansano.usm.cl	201473561-8
Jonathan Mendoza	jonathan.mendoza.14@sansano.usm.cl	201473573-1
Ian Mora	ian.mora.14@sansano.usm.cl	201473609-6

Tabla de contenidos.

1. Requisitos clave (Actualizado).....	3
2. Árbol de Utilidad (Actualizado).....	4
3. Modelo de Software.....	5
4. Trade-offs entre tecnologías.....	7

1. Requisitos clave (Actualizado)

Nº	Req. funcional	Descripción y medición (máximo 2 líneas)
1	Notificar de órdenes pendientes	El sistema debe avisar al bodeguero central de las órdenes de compra que aún están pendientes.
2	Generar solicitudes de materiales	El sistema debe permitir la creación de solicitudes para el envío de materiales a terreno.
3	Generar solicitudes de compra.	El sistema debe poder generar las órdenes de compra necesarias para la adquisición de materiales.
4	Mostrar solicitudes de materiales.	El sistema debe ser capaz de mostrar las solicitudes de materiales que se han realizado.
5	Mostrar solicitudes de compra.	El sistema debe ser capaz de mostrar las solicitudes de compra que el bodeguero ha realizado.
6	Control de tipos de usuarios	Cada usuario creado solo debe acceder a las funcionalidades que se le son permitidas.
7	Avisar falta de stock	Cuando se solicite un material este debe mostrar si tiene o no stock
8	Acceso a la plataforma	El acceso debe hacerse a través de un usuario y contraseña asignado por algún encargado.

Tabla 1: Requisitos funcionales (actualizados)

Req. extra-funcional	Descripción y medición (máximo 2 líneas)
Persistencia de datos	El sistema debe ser capaz de mantener de manera persistente los datos que se le ingresan.
Escalabilidad	Las funcionalidades del sistema deben funcionar de la misma manera tanto para 1 como para 1000 solicitudes que se realicen.
Multiplataforma	El sistema será accesible vía navegador web, tanto en PC de escritorio como en Smartphones.

Tabla 2: Requisitos extra-funcionales (actualizados)

A través del análisis y desarrollo del proyecto, en conjunto con conversaciones con el cliente, se decidió la actualización de requisitos funcionales. Esta actualización consiste en añadir tres requisitos funcionales nuevos, los cuales están mayoritariamente enfocados a aumentar la seguridad e integridad de los datos de la empresa, limitando la cantidad de funciones por tipo de usuario dentro de la plataforma. Otro requisito está enfocado en aumentar la facilidad para detectar stocks de materiales bajos o nulos al momento de su solicitud para aumentar la información entregada por la petición.

2. Árbol de Utilidad (Actualizado)

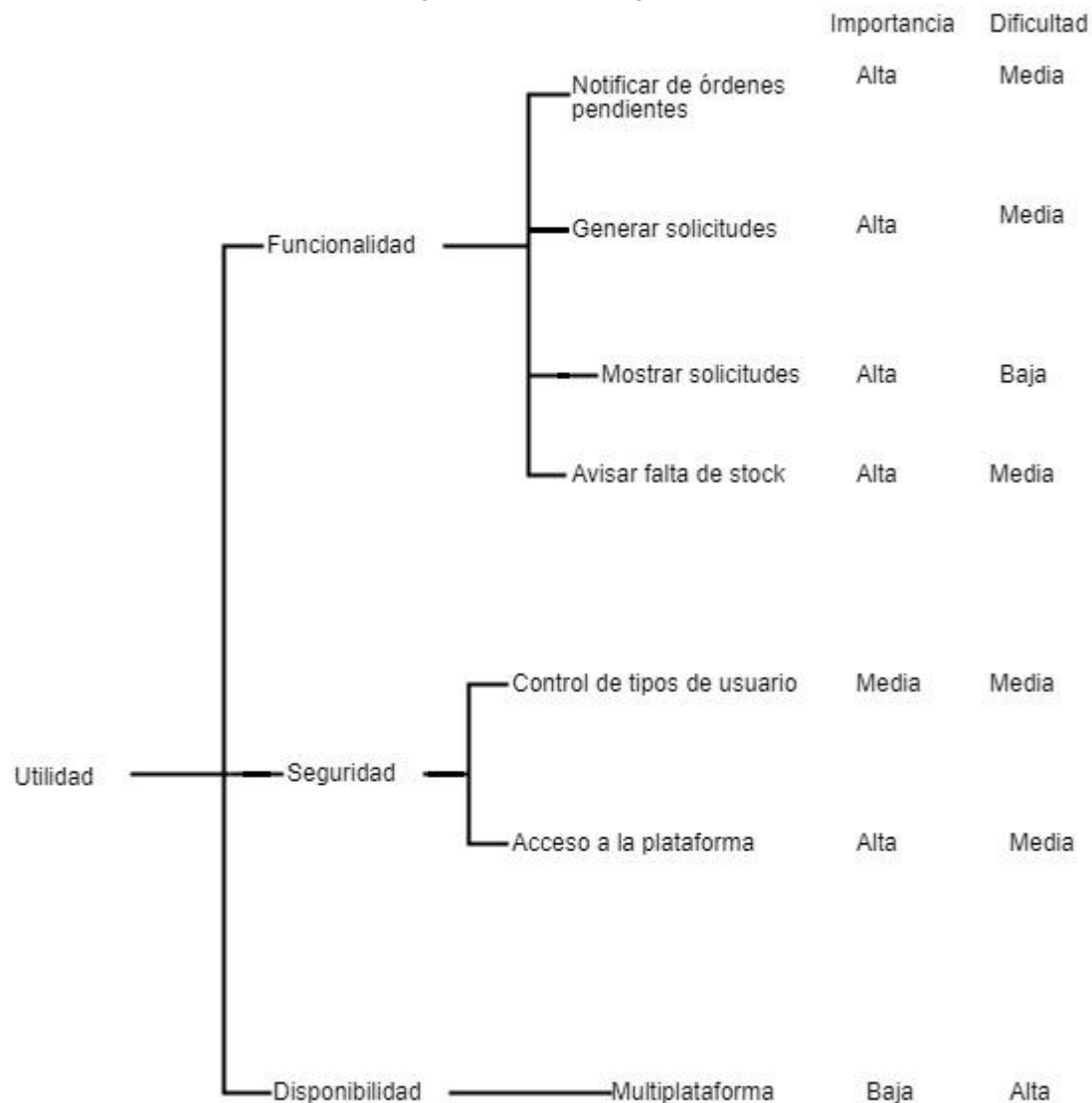


Ilustración 1: árbol de utilidad

Debido a la reducción del tiempo disponible por razones extra-programadas y al poco tiempo disponible antes de las entregas oficiales, como equipo se decidió focalizar el desarrollo de la plataforma en su funcionalidad más que en otros parámetros como seguridad o disponibilidad. De esta forma se puede lograr una exposición más clara del avance realizado sin la necesidad de recurrir a 'falsas promesas'. Ahora, si bien se quitarán esfuerzos dentro de las otras áreas, no significa que quedarán sin desarrollar, sino que será un desarrollo más lento y paulatino a través del tiempo.

3. Modelo de Software

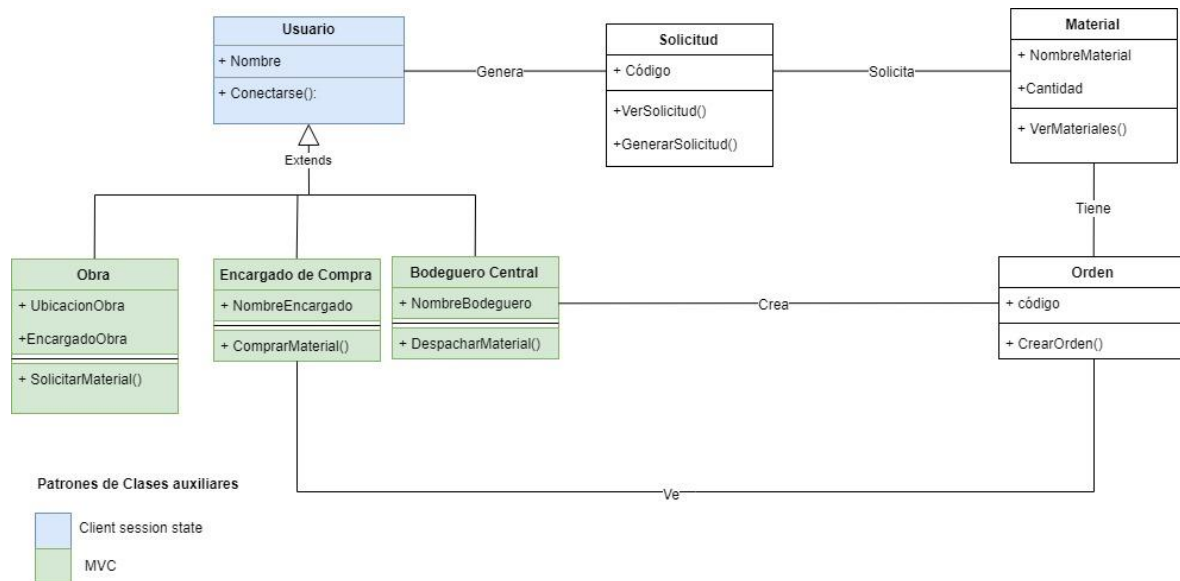


Ilustración 2: Modelo de Software

Intención	Patrón de Diseño	Razonamiento
Conservar los datos de la sesión activa	Client session state	Se utiliza Client Session State para poder conservar los datos del usuario que inicia sesión y así poder verificar los permisos que tiene disponibles para cada acción que desee realizar sin tener la necesidad de consultar múltiples veces la base de datos, evitando así el overhead. Con esto se permite poder tener un registro constante de lo que modifica cada usuario.
Que el encargado de obra solicite materiales	Model-View-Controller (MVC)	Model view controller se utiliza para poder comunicar a externos con la aplicación, en este caso son los tres tipos de usuario que se tienen, los cuales podrán acceder a las funcionalidades que entrega la aplicación, las cuales están representadas en el modelo como los métodos: <i>SolicitarMaterial()</i> , <i>ComprarMaterial()</i> , <i>DespacharMaterial()</i> de esta manera a través de una página web y con esos métodos logramos comunicar usuarios externos con la aplicación.
Que el encargado de compra cree una solicitud de compra		
Que el encargado de compra vea las órdenes de compra.		
El bodeguero central despacha materiales.		

Tabla 3: Elección de patrones

4. Trade-offs entre tecnologías

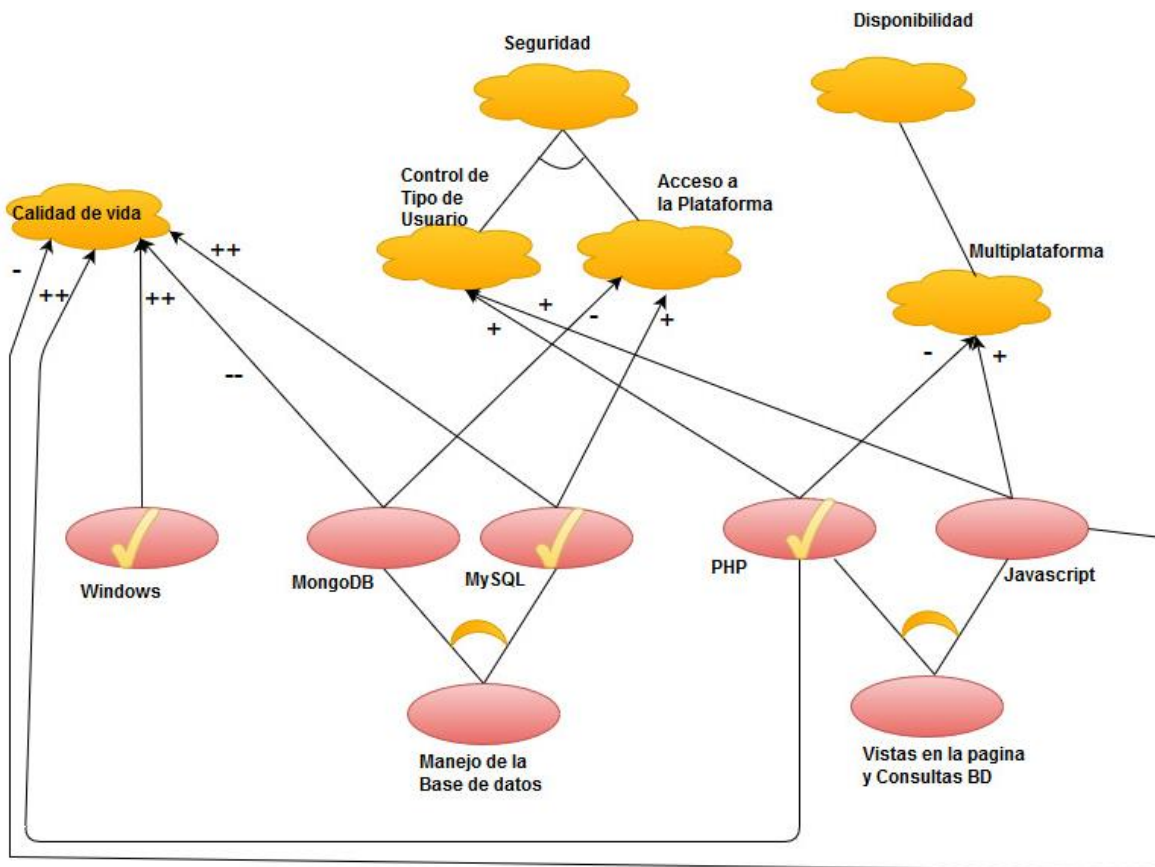


Ilustración 3: Softgoal Interdependency Graph

Decisión	Softgoal	Evaluación	Razonamiento
Manejo de Bases de datos con MySQL	Disponibilidad	++	La utilización de una base de datos que no sea <i>Open Source</i> entrega un mejor soporte y documentación en caso de fallas.
Utilización PHP	Multiplataforma	++	De acuerdo a los conocimientos que tiene el equipo la meta puede ser alcanzada de manera más rápida y eficiente utilizando PHP en vez de JavaScript debido a la curva de aprendizaje de ambos lenguajes.
Utilización de MySQL	Seguridad	++	La utilización de una base de datos que no sea <i>Open Source</i> entrega más confianza en temas de seguridad. Además, en el caso de MongoDB, por defecto no se encuentra

			configurado con encriptación de datos o protocolos lo cual lo hace un objetivo fácil en cuanto a seguridad.
Tecnología utilizada para el proyecto	Calidad de vida	++	La utilización de la tecnología WAMP, facilita la realización del proyecto debido a que es una tecnología que el equipo desarrollador ha utilizado y manejado bien en ocasiones anteriores, además de contar con una curva de aprendizaje más rápida que otros entornos de desarrollo.

Tabla 4: Trade-offs entre opciones tecnológicas