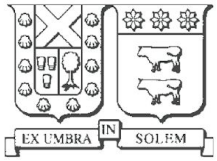
Departamento de Informática
Universidad Técnica Federico Santa María



Informe de Proyecto – INF-225-2018-1-<CSJ>
Proyecto <Aplicación web para optimización del flujo de materiales en GPI>
<martes 05-08-2018>

Integrantes:

Nombres y Apellidos	Email	ROL USM
María Apolo	maria.apolo@sansano.usm.cl	201573509-3
Nicolás Rosas	nicolas.rosasg@sansano.usm.cl	201573608-1

Contenido del Informe a Entregar

1. Requisitos clave (Actualizado)

Req. funcional	Descripción y medición (máximo 2 líneas)
Registro de materiales en bodega	El sistema debe contar con una base de datos que tenga todos los materiales existentes en todas las bodegas.
Envío de solicitudes de materiales	El sistema debe permitir que el personal de obra envíe solicitudes de materiales a la bodeguera central.
Búsqueda de materiales	El sistema debe permitir a la bodeguera central buscar disponibilidad de materiales específicos en la base de datos.
Exportar solicitudes en formato excel	El sistema deberá permitir a los encargados de adquisición exportar las solicitudes recibidas en formato excel.
Registro de usuarios en sistema	El sistema deberá permitir a los usuarios registrarse para poder acceder a su perfil y a las funcionalidades.
Restricción personal de obra	Solo el usuario con cargo personal de obra puede acceder a la funcionalidad de envío de solicitud de materiales.

Tabla 1: Requisitos funcionales (actualizados)

Req. extra-funcional	Descripción y medición (máximo 2 líneas)
Disponibilidad usuarios	El sistema podrá atender a 50 usuarios simultáneos.
Desempeño búsquedas	El sistema deberá ejecutar la búsqueda de un material en específico en un tiempo <= 0.5[s]
Escalabilidad	Se podrán agregar hasta 100000 materiales a la base de datos.
Confidencialidad	Los precios de los materiales, sus costos y facturas deben ser confidenciales. Restricción de vistas según cargos.
Integridad de datos	No se deben permitir accesos a los datos fuera de los métodos definidos.

2. Árbol de Utilidad (Actualizado)

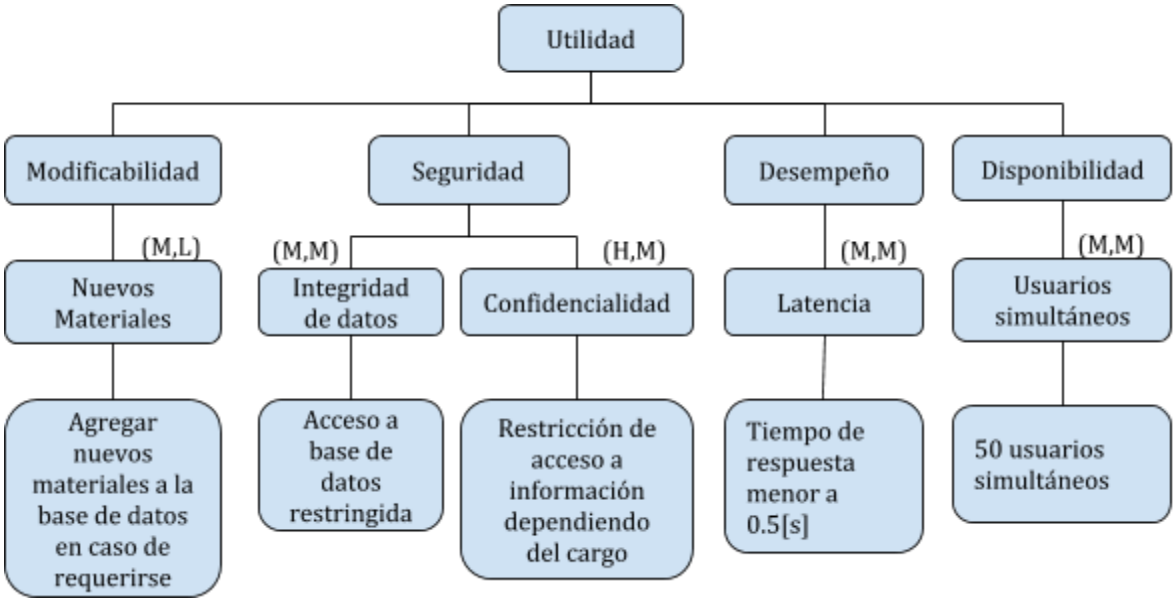
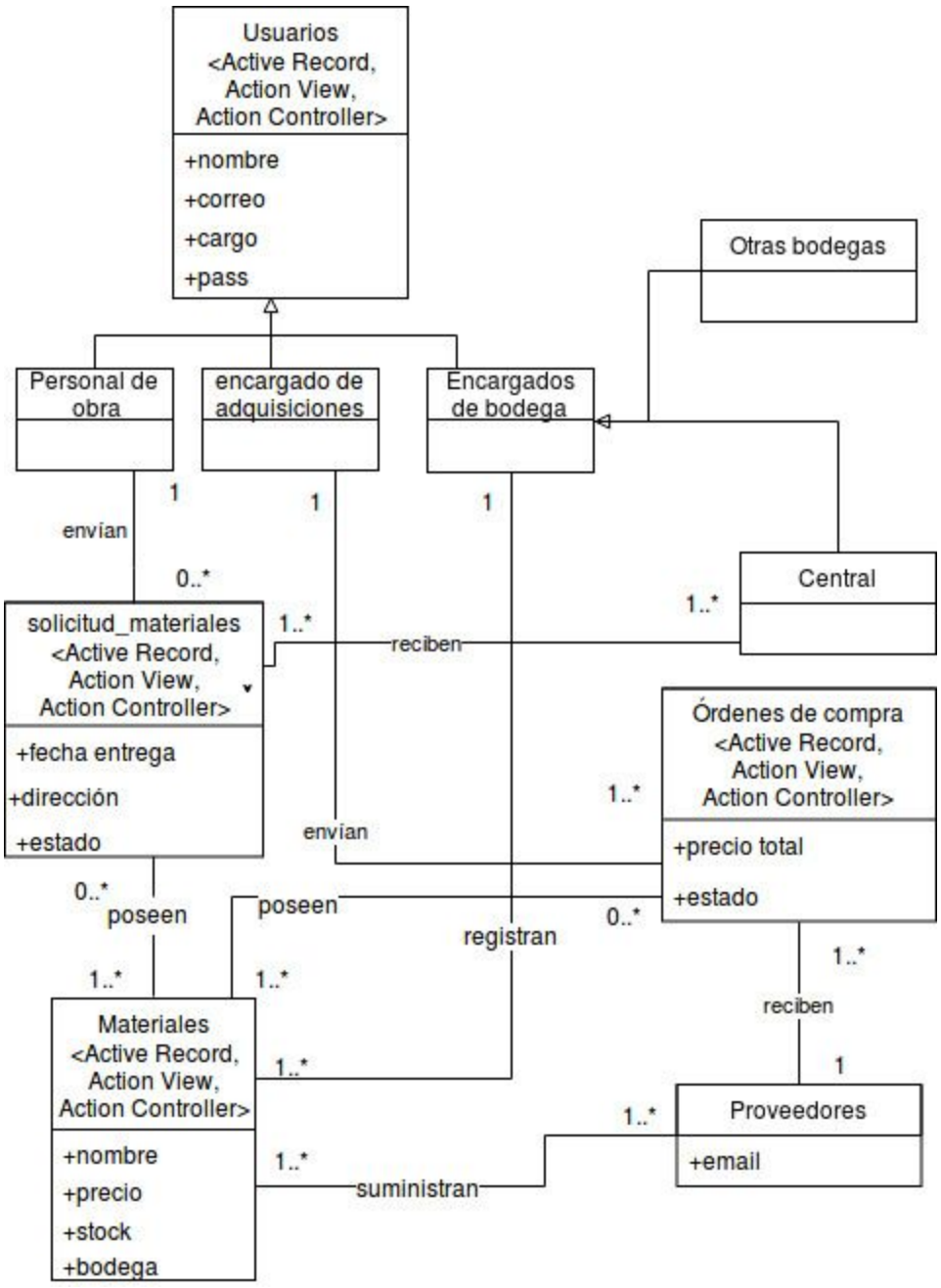


Ilustración 1: Árbol de utilidad

3. Modelo de Software



En el modelo de software se hizo distinción entre el cargo de encargado de bodega central y encargados de otras bodegas, ya que, pese a que ambos perfiles tienen acceso a la función de registrar materiales y actualizar su estado, el perfil de encargado de la bodega central tiene además otras funcionalidades, como lo son el recibir las órdenes por parte del personal de obra.

No se modela al proveedor como un usuario, ya que este no ingresará al sistema, solamente se necesita guardar el email de contacto para solicitar los materiales.

A la clase "Materiales" se le agregó el atributo "bodega", ya que, dado que en la base de datos se encontrará el material disponible en **todas** las bodegas de gpi, es importante que la encargada de la bodega central sepa en qué bodega se encuentra el stock (de haberlo) para poder gestionar las solicitudes.

Tabla 3: Selección de Patrones

Intención	Patrón de Diseño	Razonamiento
Realizar operaciones de tipo CRUD dentro de la base de datos.	Active Record	Se selecciona por sobre otro ORM debido a que al tener un modelo de dominio simple, el mapeo *one-to-one* proporcionado por este patrón simplifica el desarrollo de la aplicación. Además, se encuentra implementado en el framework. MVC utilizado para el desarrollo del software.
Manejar las solicitudes que ocurren en el software.	Action Controller	Permite manejar las solicitudes web que ocurren dentro de la aplicación, como también la comunicación entre el modelo y las vistas del software, del mismo modo que controla las rutas para todas las vistas del software. Además, se encuentra incluido en el patrón MVC del framework utilizado.
Mostrar a usuario los resultados de las consultas que se realizan en la aplicación.	Action View	Compila las respuestas que entrega Action Controller a las solicitudes que se hacen en la aplicacion web. Se encuentra incluido en el framwork MVC utilizado.

4. Trade-offs entre tecnologías

Esta sección describe los *trade-offs* (compromisos) entre requisitos que surgen cuando una solución tecnológica favorece alguno pero perjudica otro (algo muy usual, por cierto). P.ej. elegir Hadoop para almacenar instancias de Asignatura favorece la *disponibilidad* (porque no se necesita esperar un unlock para leer) pero afecta la *integridad* (porque Hadoop usa consistencia eventual y se podría leer valor obsoleto).

Ya que el proyecto debe entregar *software funcional* en cada iteración, Ud ya ha tomado ciertas decisiones tecnológicas (p.ej. quizás está usando Scala o Python/Heroku). Para este informe, deberá describir los trade-offs con respecto a requisitos entre la tecnología que está usando y otra alternativa (a elegir).

Los trade-offs de su proyecto serán descritos usando *softgoals* ("metas blandas") y operacionalizaciones. Instanciando los conceptos vistos en clase: los *softgoals* serán los requisitos extra-funcionales y escenarios (descritos en el Árbol de Utilidad): las *operacionalizaciones* serán las opciones tecnológicas; y los *trade-offs* serán las ventajas/desventajas de cada opción. Describa los softgoals, operacionalizaciones y trade-offs de su proyecto en la figura "Trade-offs entre decisiones tecnológicas" (un SIG similar al de la figura ejemplo).

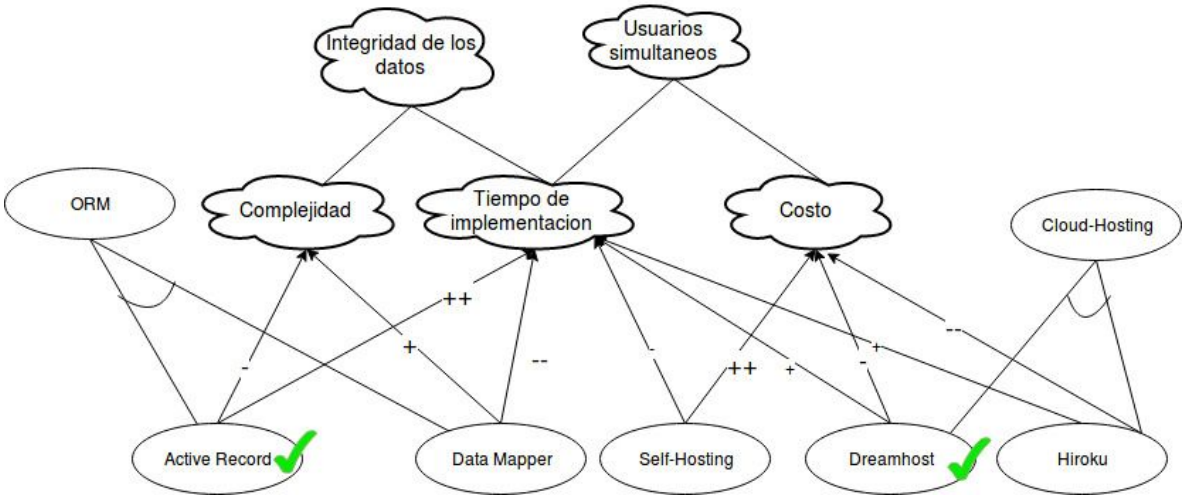


Ilustración 2: Ejemplo de Softgoal Interdependency Graph (SIG)

En su figura, indique claramente (con + y ++) cuando una opción favorece (algo o mucho) un requisito, e igualmente (con – y --) cuando lo desfavorezca (poco o mucho). Finalmente, indique claramente (con circulito u overlay) la decisión tomada (¡que bien podría no ser la que Ud está usando ahora!).

Describa el razonamiento detrás de cada ++/+/-/-- en la tabla "Trade-offs entre Opciones Tecnológicas".

Decisión	Softgoal	Evaluación	Razonamiento
Utilizar Active record	Integridad de datos	+	Al estar integrado nativamente en el software elegido para realizar el proyecto, el tiempo de implementación de esta característica es mucho menor que si se usara algún otro patrón de diseño ORM, e.g Data Mapper. Además, al trabajar en un modelo simple, el problema relacionado con la complejidad de la base de datos se puede ignorar.

Utilizar DreamHost	Usuarios simultáneos	+	Si bien utilizar Dreamhost conlleva un costo asociado, las facilidades que entrega para el despliegue de la aplicación la hacen la mejor opción una vez la aplicación sea entregada al cliente.

Tabla 9: Trade-offs entre opciones tecnológicas