



## Entregable 2 The All Mightys

Abdel Sandoval Moreno 201573504-2 - abdel.sandoval@sansano.usm.cl  
Alejandro Vilches Cornejo 201573554-9 - alejandro.vilches@sansano.usm.cl  
Vanessa Guzman Canaval 201573528-k - vanessa.guzman@sansano.usm.cl

### 1. Contexto del Proyecto

#### 1.1. Objetivo del Proyecto

El enfoque principal de este proyecto es la creación de un sistema que permita establecer mejoras en la comunicación entre trabajadores a la hora de solicitar materiales mediante la automatización de este proceso.

#### 1.2. Resumen del Proyecto

El Problema principal que busca resolver este proyecto, es la falta de comunicación entre los diferentes componentes humanos en la solicitud de materiales para una obra. Este problema se origina debido a que en el momento que los proveedores envían un pedido, no existe un rastreo del mismo, y en ocasiones no se informa a la persona que solicita el material que estos llegaron a bodega, lo que provoca pedidos atrasados o materiales que quedan almacenados hasta que alguien más los necesite.

Los componente humanos que se relacionan en este proceso son los siguientes:

- Personal de obra: Es el que realiza una solicitud de materiales al bodeguero, este será el final del recorrido para el material solicitado.
- Bodeguero: Es el que recibe la solicitud de un material, revisa la bodega en busca de este para su despacho, de no existir debe iniciar una orden de compra por medio del encargado de compras. Además debe poder llevar de forma eficiente el control del inventario.
- Encargado de Compras: Es el que recibe una nueva orden desde el bodeguero, y realiza la compra a los proveedores por medio de Laudus. Además notifica el tiempo en el que llegará el material, para que tanto el bodeguero como el personal de obra puedan planificar sus trabajos.

Con respecto al camino que siguen los materiales, se tiene en consideración la creación de un sistema de rastreo o "*checking*" que permita tanto al bodeguero y el personal de obra el poder dar confirmación de la llegada de los materiales solicitados, para su despacho o uso respectivamente.

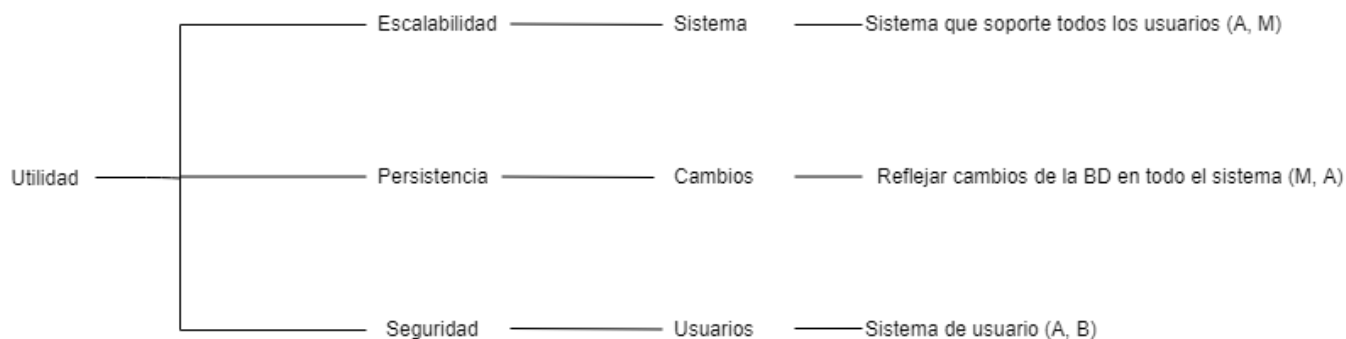


## 2. Requisitos Funcionales y Extra Funcionales

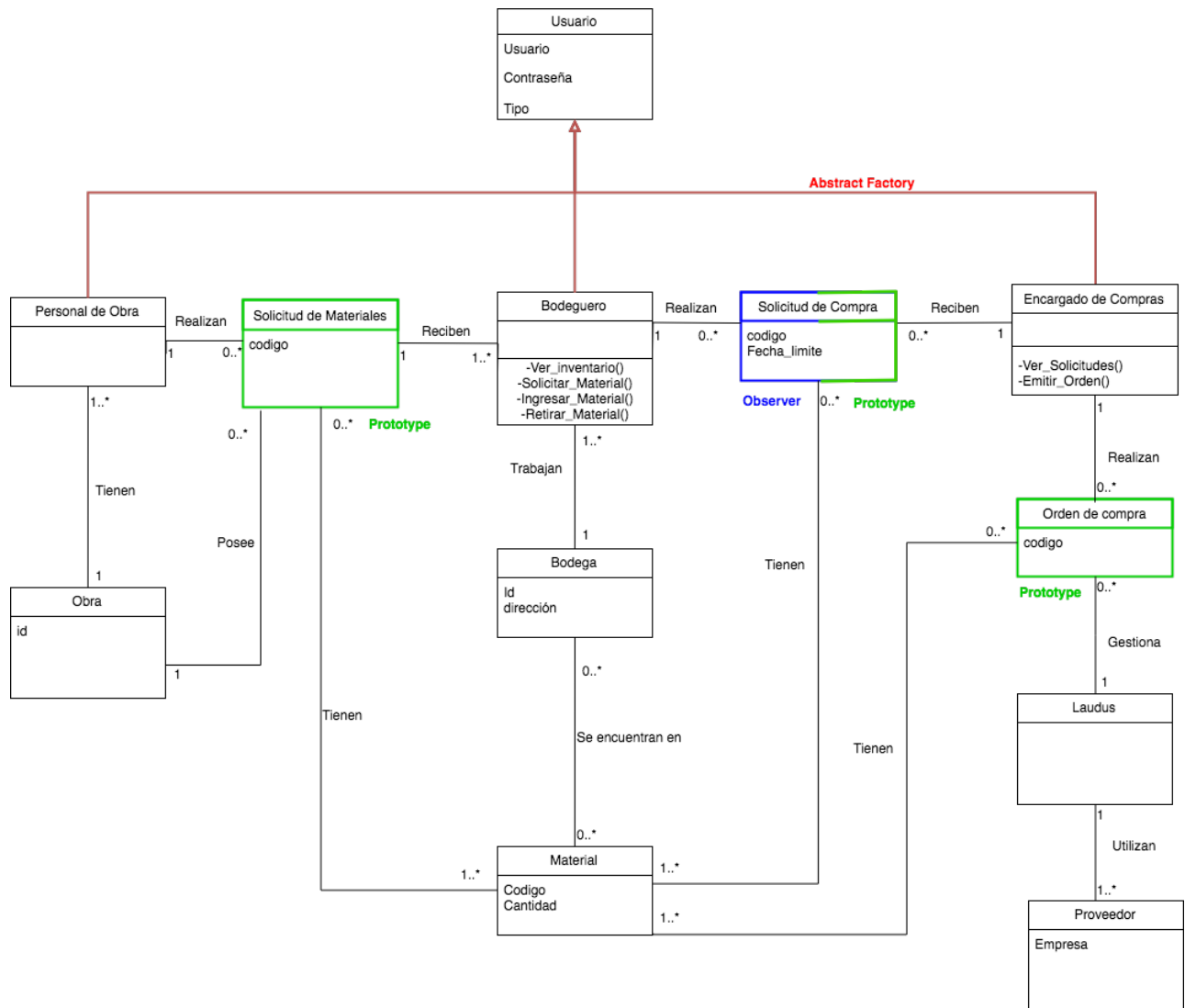
Requisitos Funcionales	
Ingresar material	Que el bodeguero pueda agregar al inventario los materiales que llegan a bodega.
Retirar material	Que el bodeguero pueda retirar del inventario los materiales que se retiran de la bodega.
Ver inventario	Que el bodeguero pueda ver los materiales en el inventario y su cantidad.
Solicitar material	Que el bodeguero pueda realizar solicitudes de compra al encargado de compras.
Ver solicitudes de material	Que el encargado de compras sea capaz de ver las solicitudes de material
Emitir orden de compra	El encargado de compras debe ser capaz de emitir una orden de compra a sus proveedores.
Rastreo de Pedidos	Mantener un rastreo de los productos, y/o tener un sistema de notificación para los pedidos emitidos, de tal manera de poder tener conocimiento del estado en el que se encuentran.
Tiempos Limites	Recibir una notificación cuando se acerque la fecha límite de un pedido, para poder estar al tanto de su posible retraso.
Usabilidad	Mostrar de distintos colores las órdenes dependiendo de su prioridad, el cual depende de su fecha límite.

Req. Extra Funcional	Descripción y medición
Seguridad	El software deberá tener un sistema de usuarios con acciones limitadas según el tipo de usuario que sea.
Escalabilidad	La cantidad de usuarios que soporta el sistema debe dar abasto para la cantidad de trabajadores que lo utilizarán ahora o en un futuro.
Persistencia de datos	Si ocurre un cambio en la base de datos, este debe verse reflejado para todos los usuarios luego de realizado el cambio.

## 3. Árbol de utilidad



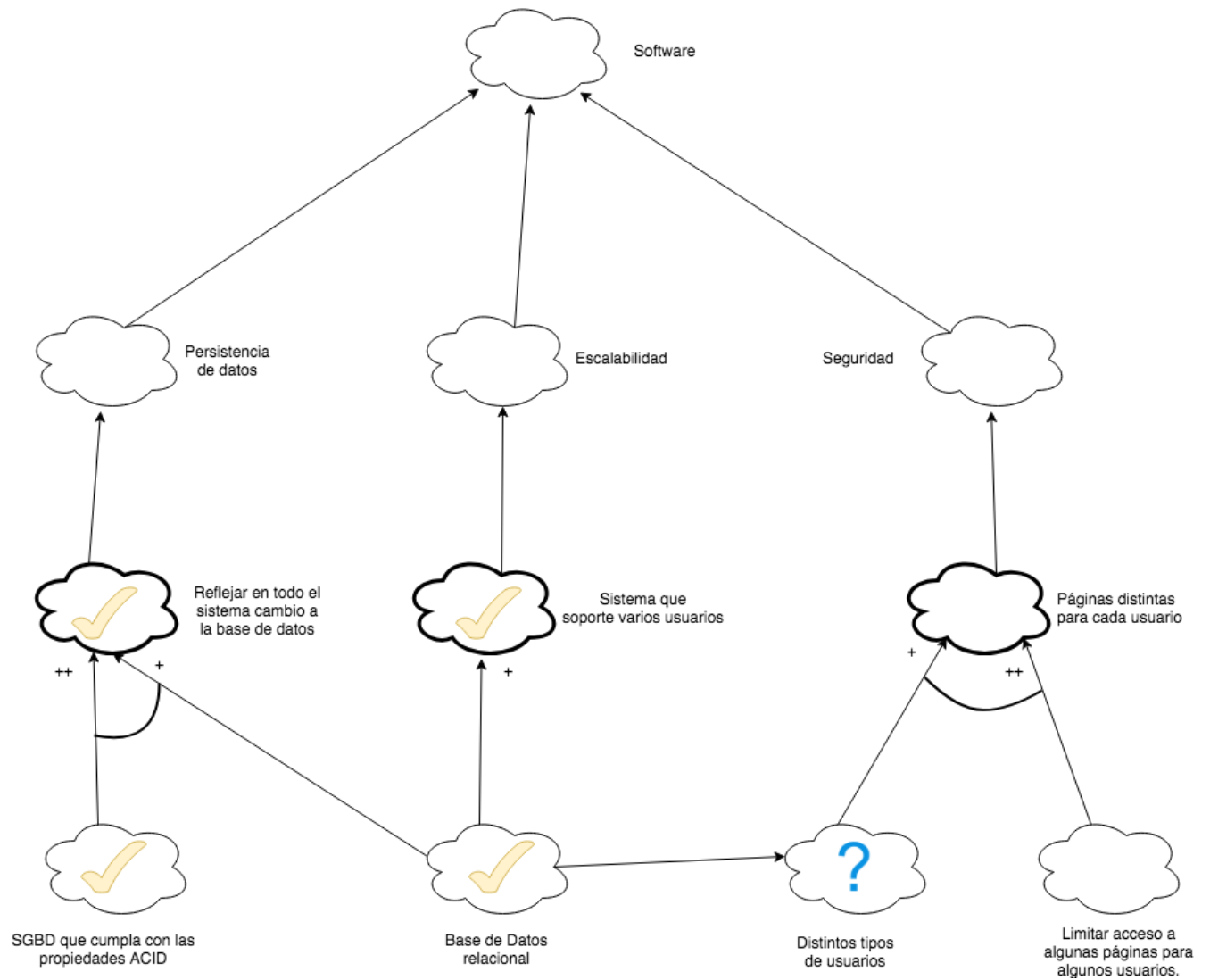
## 4. Modelo de Dominio





Intención	Patrón de diseño	Razonamiento
Crear usuarios similares pero diferenciándolos según su función	Abstract Factory	Ya que <i>AbstractFactory</i> te permite crear clases abstractas que tengan atributos bases y al extenderlas crear nuevas para diferenciarlas.
Utilizar la Herencia de Plantillas HTML en Django	Prototype	Django permite definir una plantilla HTML y poder heredar el formato definida en esta al resto de plantillas, por lo que se reduce la repetición de código y se puede mantener una estructura similar en todas las aplicaciones del software, realizando la menor cantidad de cambios necesarios a la plantilla.
Enviar correos con fechas límites cercanas	Observer	<i>Observer</i> les avisa a los objetos cuando eventos ocurran, esto sirve para enviar las notificaciones de las fechas límites

## 5. Trade-offs entre tecnologías





Decisión	Softgoal	Evaluación	Razonamiento
Usar sistema de SGBD que cumpla con propiedades ACID	Persistencia de Datos	++	Esto ayuda a que no hayan colisiones entre consultas, ni se generen inconsistencias en la base de datos como por ejemplo retirar más material del que hay en una bodega.
Usar Base de Datos relacional	Escalabilidad	+	Como estamos trabajando con relaciones entre clases, este tipo de base de datos se ajusta más a las necesidades que tenemos en comparación a, por ejemplo, una base de datos no SQL.
Usar Base de Datos relacional	Persistencia de Datos	+	Necesitamos una base de datos para guardar la información que cambiará en el tiempo.
Crear distintos tipos de usuario	Seguridad	+	Esto nos ayudará a diferenciar los distintos permisos que se les asignará a cada trabajador.
Limitar acceso a algunas páginas a ciertos usuarios	Seguridad	++	Esto sirve para que ciertos usuarios no hagan cosas que nos les corresponde.