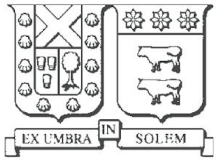
Departamento de Informática
Universidad Técnica Federico Santa María



Informe de Proyecto – INF-225-2018-1-CSJ
Proyecto "Sistema de Insumos GPI"
2018-08-29

Integrantes:

Nombres y Apellidos	Email	ROL USM
Montserrat Figueroa	monserrat.figueroa@sansano.usm.cl	201573525-5
Ignacio Tampe Palma	ignacio.tampe@sansano.usm.cl	201573514-k
Franco Zalavari Palma	franco.zalavari@sansano.usm.cl	201573501-8

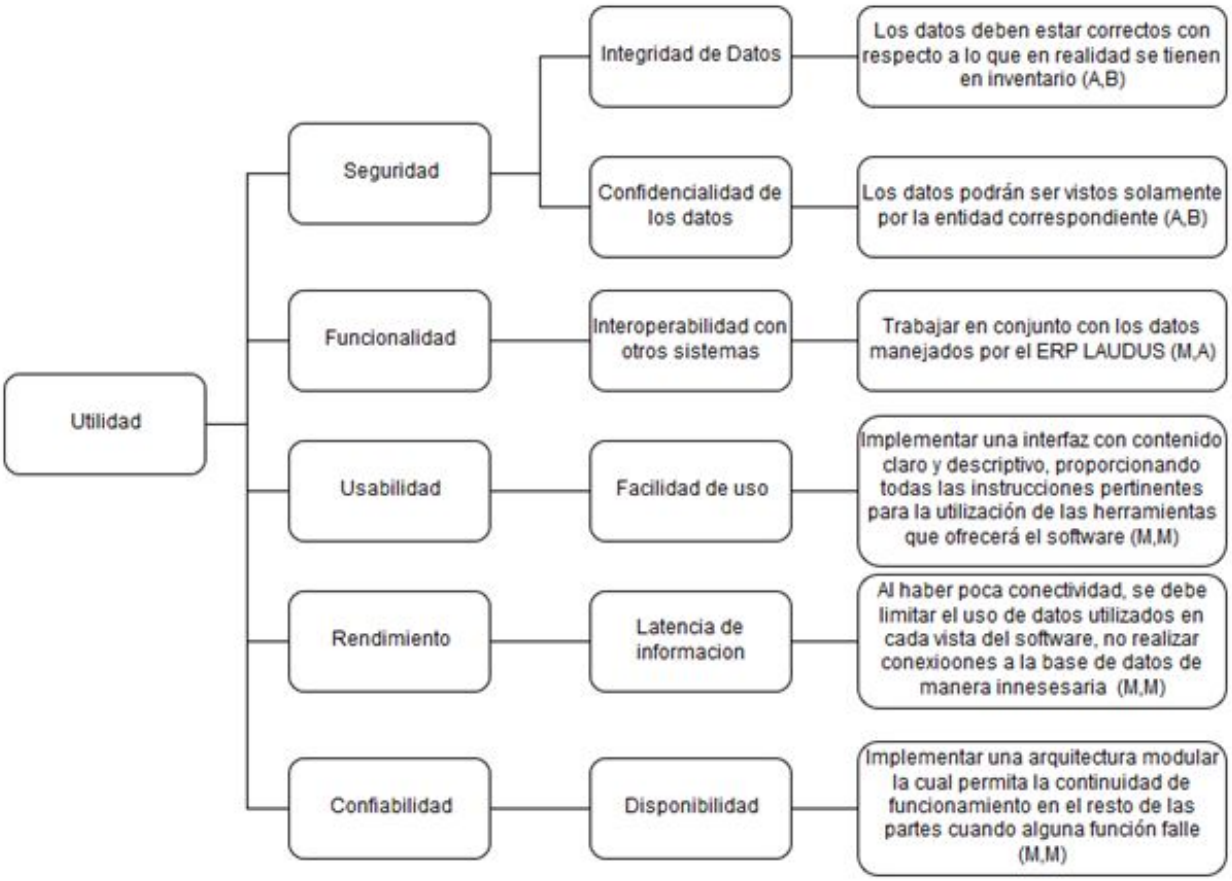
Contenido del Informe

1. Requisitos clave (Final)

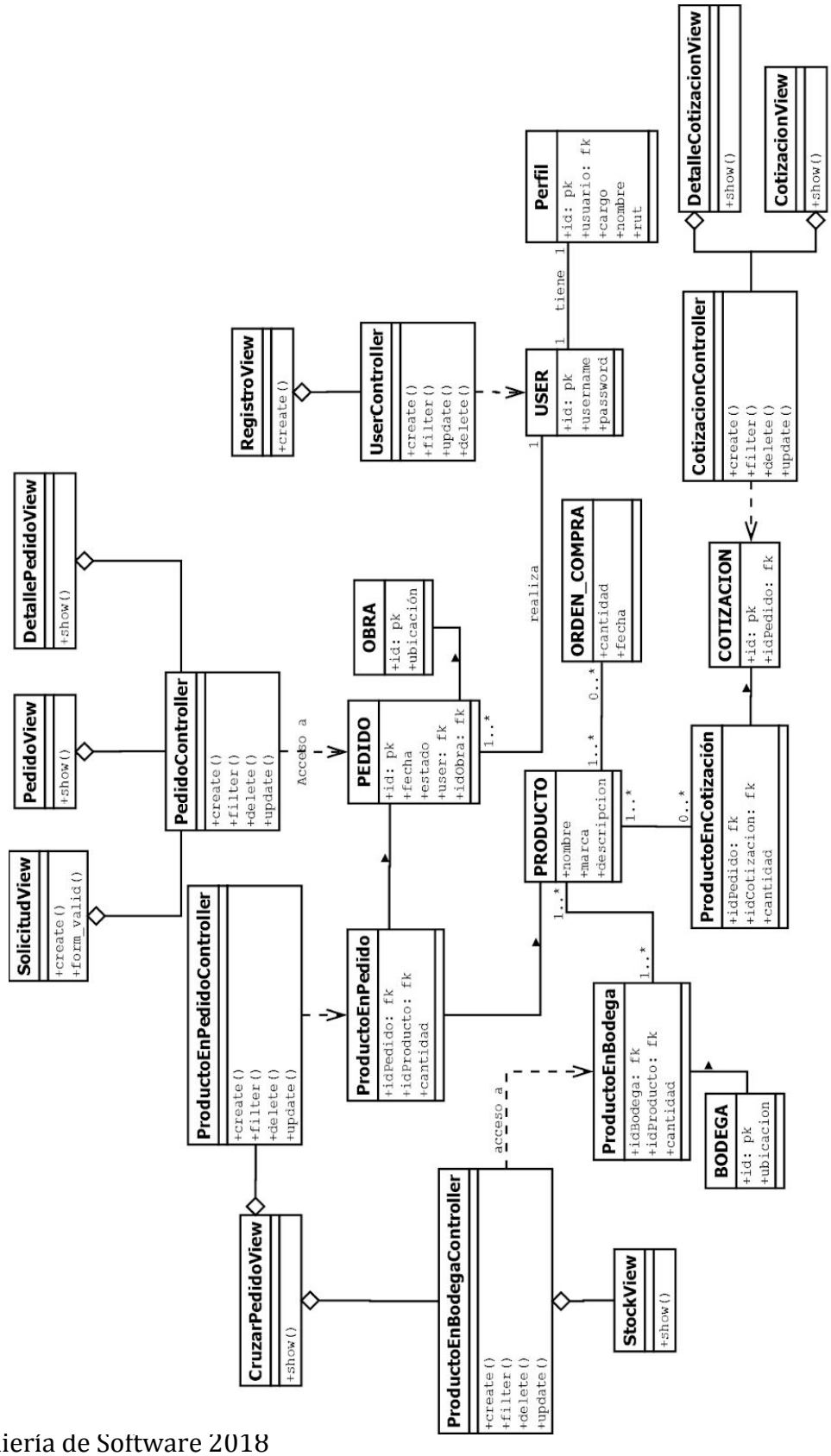
Req. funcional	**Descripción y medición**
Realizar solicitud de insumo	Un personal de obra solicita un insumo faltante a través del sistema enviando un formulario a bodega.
Revisar solicitudes de insumos	Los bodegueros deben revisar nuevas solicitudes de personal de obra para revisar stock
Revisar órdenes de compra	los encargados de bodega revisan el estado las órdenes de compra para asegurar su envío en el día especificado.
Envío de cotización	Los proveedores deben hacer su llegar su cotización al sistema para gestionar el pago y pronto envío del pedido
Generar órdenes de compra	El encargado de compra genera órdenes para pedir productos no disponibles en bodega según bodegueros
Revisar Stock de un producto	Los bodegueros revisan la plataforma para asegurar la disponibilidad de productos para asignar o solicitarlos.

Req. extra-funcional	**Descripción y medición**
Desempeño	Se debe poder acceder al sistema y modificar datos aún utilizando conexiones lentas e inestables
Integridad de datos	Asegurar stock fidedigno a la realidad
Interoperabilidad	Integración de datos de aplicaciones de terceros

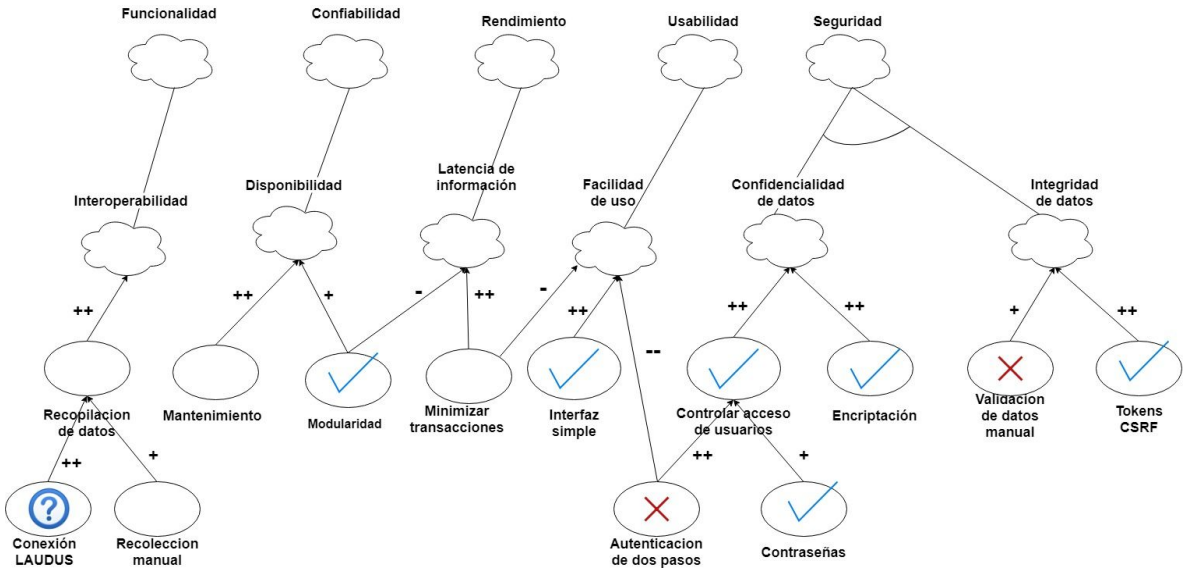
2. Árbol de Utilidad (Final)



3. Modelo de Software (Final)



4. Trade-offs entre tecnologías (final)



Decisión	Softgoal	Evaluación	Razonamiento
Modularidad	Disponibilidad	+	Permite la mantención de manera eficaz sin comprometer la disponibilidad completa del software.
Mantenimiento	Disponibilidad	++	Corrige los errores y fallas que pueda presenciar el software al momento de utilizarlo.
Modularidad	Latencia de información	-	Aumenta los tiempos de ejecución
Minimizar transacciones	Latencia de información	++	Disminuye los tiempos de ejecución al realizar menos consultas innecesarias a la base de datos. Se requiere conocimiento y experiencia en consultas
Controlar acceso de usuarios	Confidencialidad de los datos	++	Controlar el acceso de usuarios a los distintos datos permite que estos solo sean vistos por la persona adecuada y se evitan modificaciones erróneas
Tokens CSRF	Integridad de los datos	++	Aseguran que toda consulta POST fue hecha por un usuario dentro del

			sistema sin permitir inyecciones de elementos a través de forms.
Interfaz Simple	Facilidad de Uso	+	Permite mejor la comprensión del usuario al utilizar el software, ya que no se ve sobrecargado con información en cada vista
Minimizar transacciones	Facilidad de Uso	-	Para minimizar las transacciones se tendría que asegurar que el usuario que realiza la solicitud arme toda la request completa para así solo realizar un envío lo que implica más trabajo para desarrollar un cliente complejo que permita esto.
Encriptación	Confidencialidad de datos	++	Se utiliza SSL para asegurar cifrado punto a punto entre el usuario y el servidor.
Conectar a Laudus	Interoperabilidad	++	Integración con APIs permite tener un sistema más flexible donde no nos tendríamos que preocupar de la implementación directa, solo de cómo parsear la información entregada por la API
Recolección manual de datos	Interoperabilidad	+	Permita la obtención de datos a partir de otros sistemas pero de manera poco eficiente

5. Deuda técnica incurrida

Tabla 5: Deuda técnica

Ítem deuda técnica	Razonamiento	Impacto
Carencia de verificación de token CSRF en cruzar stock	Se manejó sin el framework de form validation dado a que es un form read_only	Propenso a posibles inyecciones de datos a la form.
Rutinas de JQuery poco óptimas	Se requería realizar varias cosas con Jquery y jamás se había usado el framework antes	Lentitud en ciertas respuestas de vistas como la de crear un pedido.
Mal manejo de relaciones Many-to-Many	No se conocía cómo funcionaban estas relaciones por lo que se implementaron a mano	Se requieren hacer queries un poco más complejas para obtener los datos de esas relaciones.
API ERP solamente funciona con Oodo	Fue más fácil implementar la comunicación de API directamente con Oodo que con LAUDUS	Requerirá migrar el modelo diseñado a la API de Laudus