



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



## Entregable IV

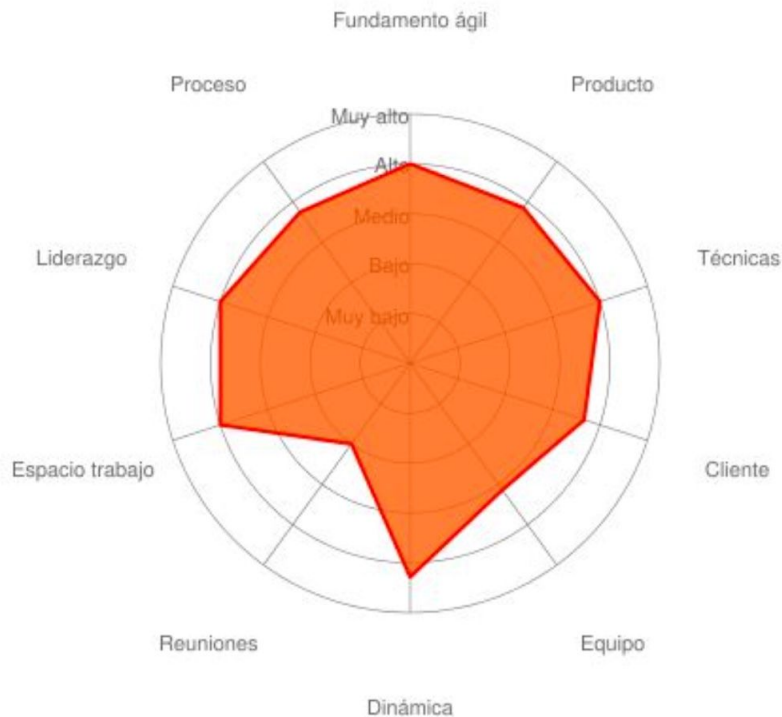
### Proyecto: Seth Project

Integrantes:

Nombres y Apellidos	Email	ROL USM
Anghelo Carvajal	anghelo.carvajal.14@sansano.usm.cl	201473062-4
Gabriela Gallardo	gabriela.gallardo.13@sansano.usm.cl	201373095-7
Nicolás Lagos	nicolas.lagos.12@sansano.usm.cl	201204213-5
Cristopher Jimenez [Consultor QA]	cristopher.jimene.13@sansano.usm.cl	201373039-6

## Post-Mortem Metodológico

Imagen de resultados:



### Evaluación nivel de agilismo:

El gráfico entregado por el análisis realizado demuestra el nivel que el equipo posee. Se denota como un resultado bastante positivo por parte de los integrantes, no obstante, sería mucho un análisis mucho más certero si existiese un personaje externo al proyecto hiciera un análisis más detallado, preciso y real para ver en qué nivel de agilidad se encuentra el equipo, por mismo se da que existen diferentes empresas evaluadoras.

Si bien las estadísticas son buenas, aun queda mejorar ciertos aspectos que son relevantes en el campo laboral, como lo son reuniones y comunicación entre miembros del equipo.

### Fortaleza

- Entregas puntuales.
- Uso de técnicas disponibles.
- Abierto a sugerencias y críticas por parte del grupo.
- Buen ambiente de trabajo.
- Alta disponibilidad.
- Solución de problema y demas en equipo.

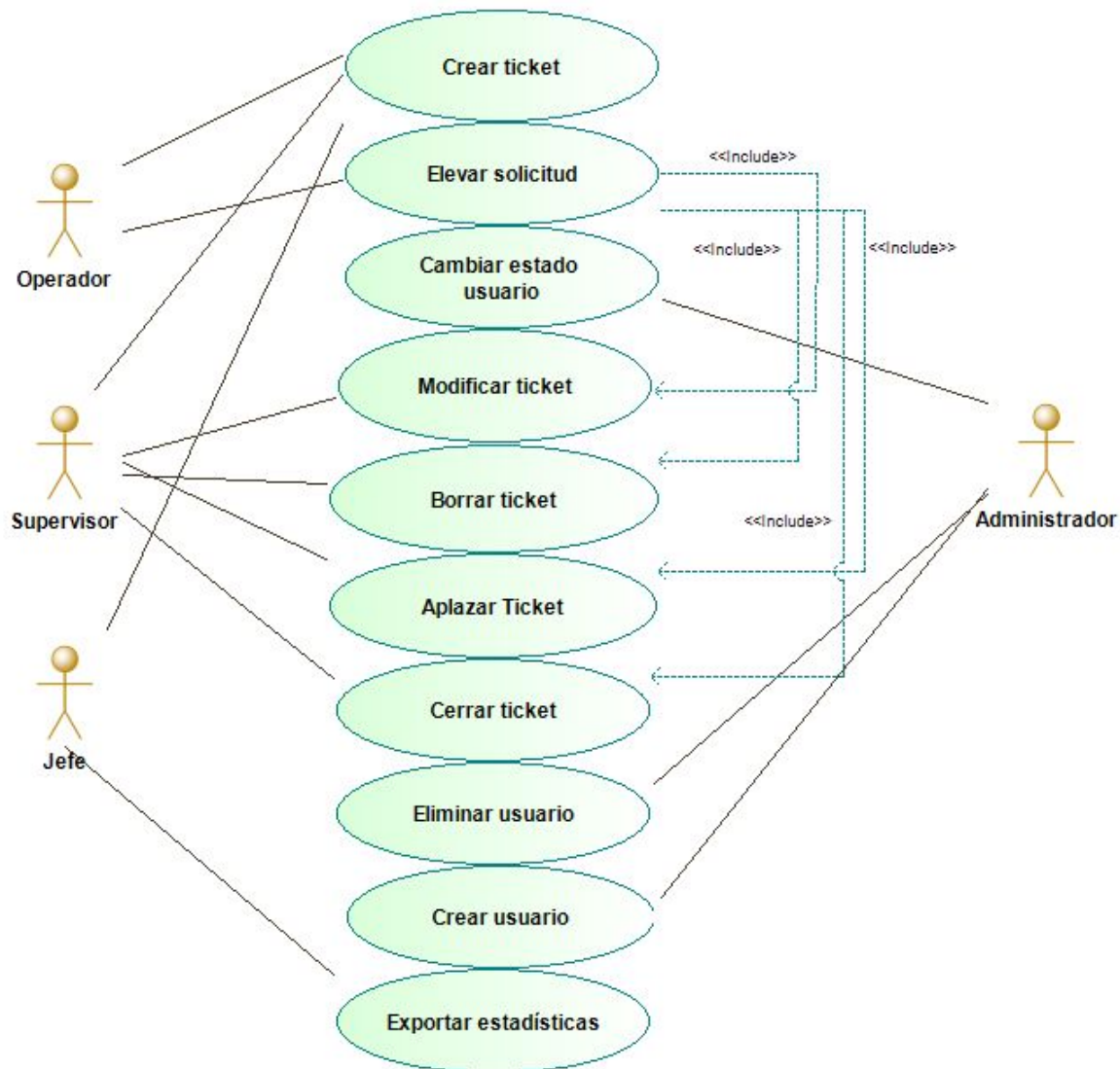
### Debilidades

- Falta de comunicación por parte de los integrantes del grupo.
- Falta de comunicación del grupo con los clientes para hacer consultas, por ejemplo.
- Falta de organización de los integrantes del proyecto en términos de una empresa, dado que este era un proyecto pequeño, nunca se considero necesario.
- Falta de análisis de calidad del producto.
- Dejar partes importantes del trabajo a último instancias considerando que no son suficientemente importantes.
- Subestimación de las tareas a realizar.

Como equipo se aprecia que el tener un canal oficial como por ejemplo para fijar reuniones y tareas es indispensable para mejorar aún más la productividad y de esta manera poder contar con un mejor producto en el tiempo requerido, Además el fijar un tiempo entre cada reunión para ver avances beneficiaría la productividad grupal. No obstante todo esto debe ocurrir luego de que la comunicación entre integrantes del equipo mejore, lo cual se puede lograr mediante más interacción en las vías de comunicación tanto formales como informales, realizando más reuniones de equipo fuera de las instancias de trabajo, entre otras.

Finalmente cabe destacar que la comunicación con el cliente debe enfatizarse más y de mejor manera, ya que el no contar con la retroalimentación de este puede generarse una visión del producto distinta a la del cliente, lo cual perjudica el producto a entregar.

## Diagrama de Casos de Uso

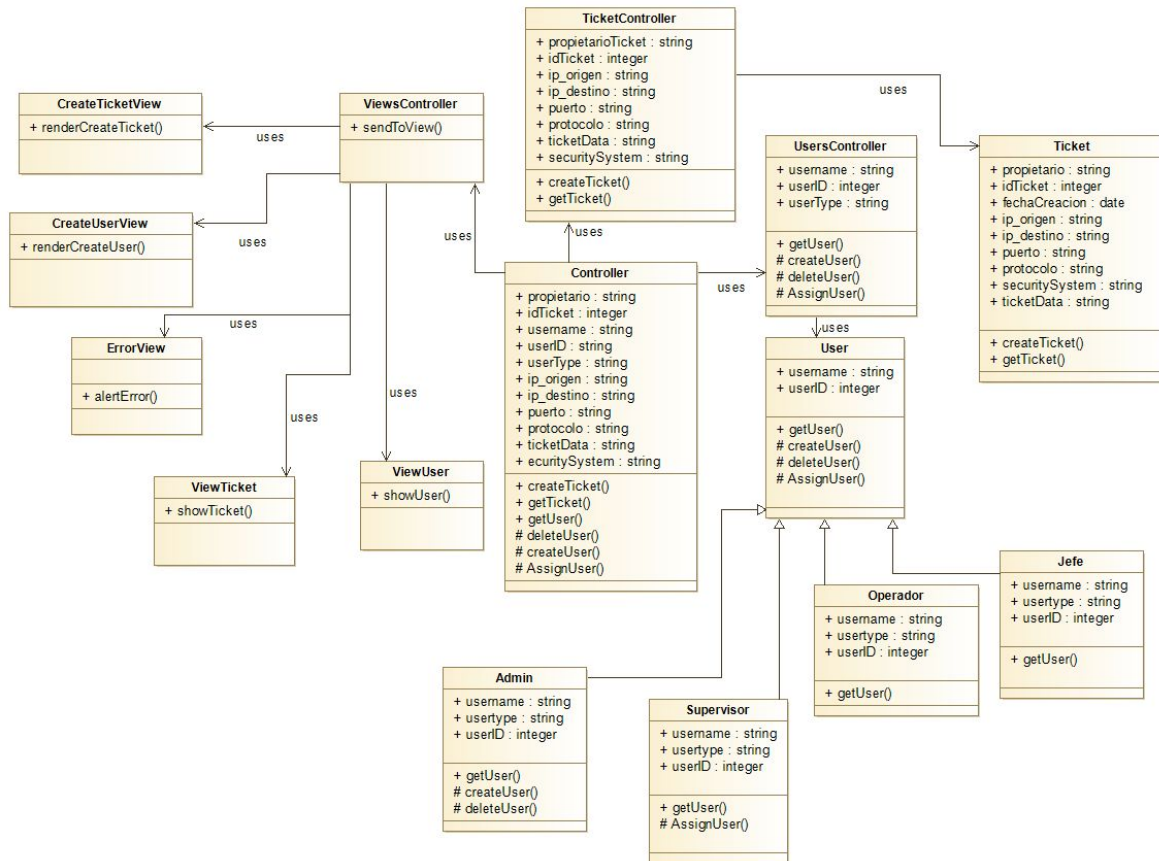


La mayoría de las funcionalidades solicitadas por los clientes fueron implementadas en el producto, como se detallaron en los casos de uso entregados en el primer informe. No se hicieron inclusión de nuevas funcionalidad, porque se consideró que el producto estaba siendo desarrollado con lo que era necesario, por lo que hacer modificaciones para agregar más cosas, generaría que se perdiera la idea principal del proyecto.

## Patrones de diseño y Frameworks

Intención	Patrón de Diseño	Razonamiento
Se desea mostrar en el Diagrama de Clases la desconexión de un usuario después de la media hora.	Observer	La clase usuarios deberá ser capaz de avisarle a la clase controladora de logs del tiempo que lleva sin actividad un determinado usuario, con el fin de que esta pueda cerrar su sesión de forma oportuna en el caso que haya excedido el límite de tiempo especificado por el cliente y además de registrar el tiempo que permaneció en línea el usuario.
Dentro del diagrama de clases se aprecian el modelo separado del controlador y las vistas de cada entidad (por ejemplo usuarios)	Modelo vista controlador	Debido a que la estructura de mvc es un mecanismo fácil de manipular (Esto no implica que sea fácil), y dado a la experiencia del grupo usando este patrón en el paso, las maneras existentes en las que se puede usar y que crea una capa de seguridad adicional por tener la lógica del negocio separada del front end, llegamos a la conclusión que era una buena opción y decidimos usarla.
Dar acceso a ciertos métodos dependiendo del tipo de usuario que está conectado.	Private Class Data	La clase usuario se encuentra constituida por diferentes tipos de usuarios, dada esta diferencia, hay métodos que son permitidos para ciertos usuarios según el nivel dentro de la organización.

## Modelo de Dominio y Diagrama de Clases



## Pruebas de Software

Defecto encontrado	Mitigación	Resultado obtenido	Observaciones
Vínculo vacío al crear un ticket nuevo, generando una caída del sistema.	Agregando una condición en caso que en la creación de un ticket no se cree un vínculo a otro ticket.	Se crea el ticket sin problema.	Al momento de crear una nueva funcionalidad se tiene que tener en cuenta el funcionamiento del programa antes que esta haya sido implementada.
No se crea el ticket luego de haber sacado campos considerados como no relevantes.	Volver a dejar los campos que se quisieron sacar hasta que se pueda localizar el error	Se crea el ticket sin problema.	
Exportación de información con error	No se logró localizar el error al momento de descargar el documento	No se puede descargar informes por ahora.	
Problemas con la dependencia bcrypt	Instalar la dependencia de forma específica.	La encriptación por medio de este método funciona, no afectando el funcionamiento del programa.	En algunas ocasiones, las dependencias de un sistema pueden fallar al ser probadas en otro entorno