



Departamento de Informática
Universidad Técnica Federico Santa María



Entregable IV

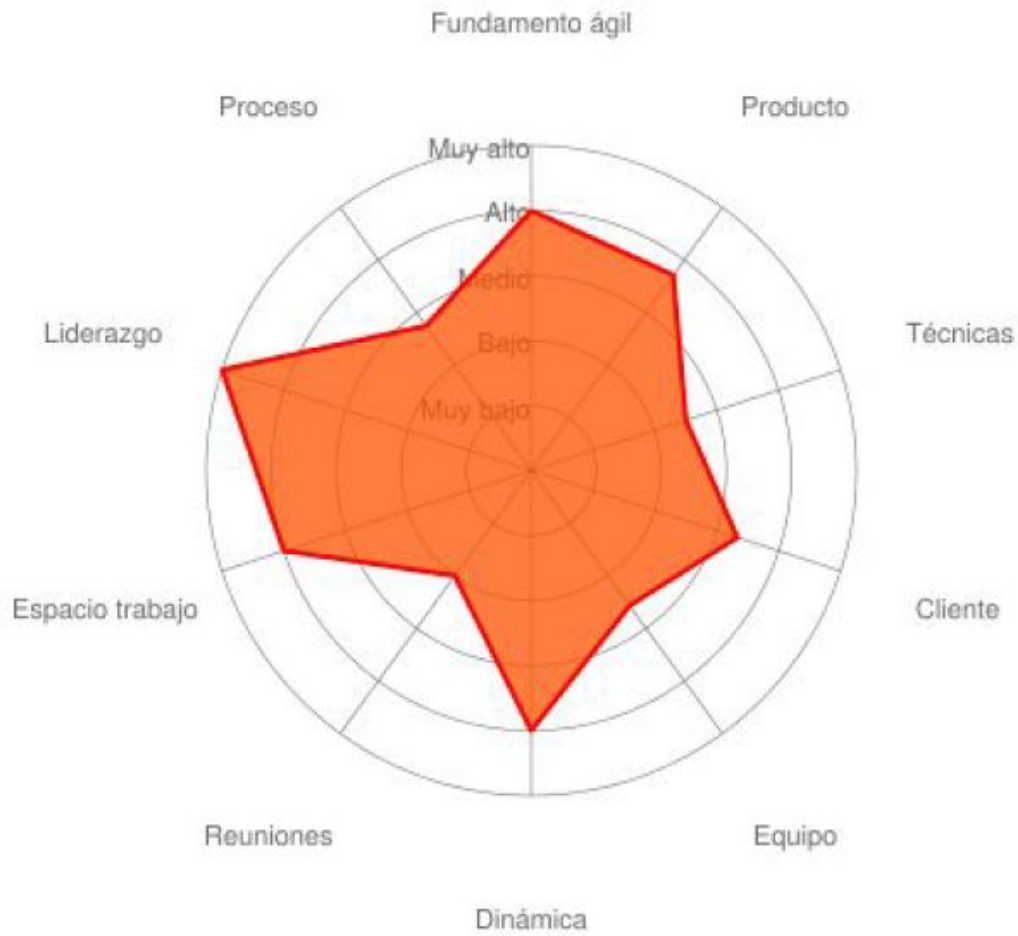
Proyecto: Sistema de Ticket para Gestión Administrativa

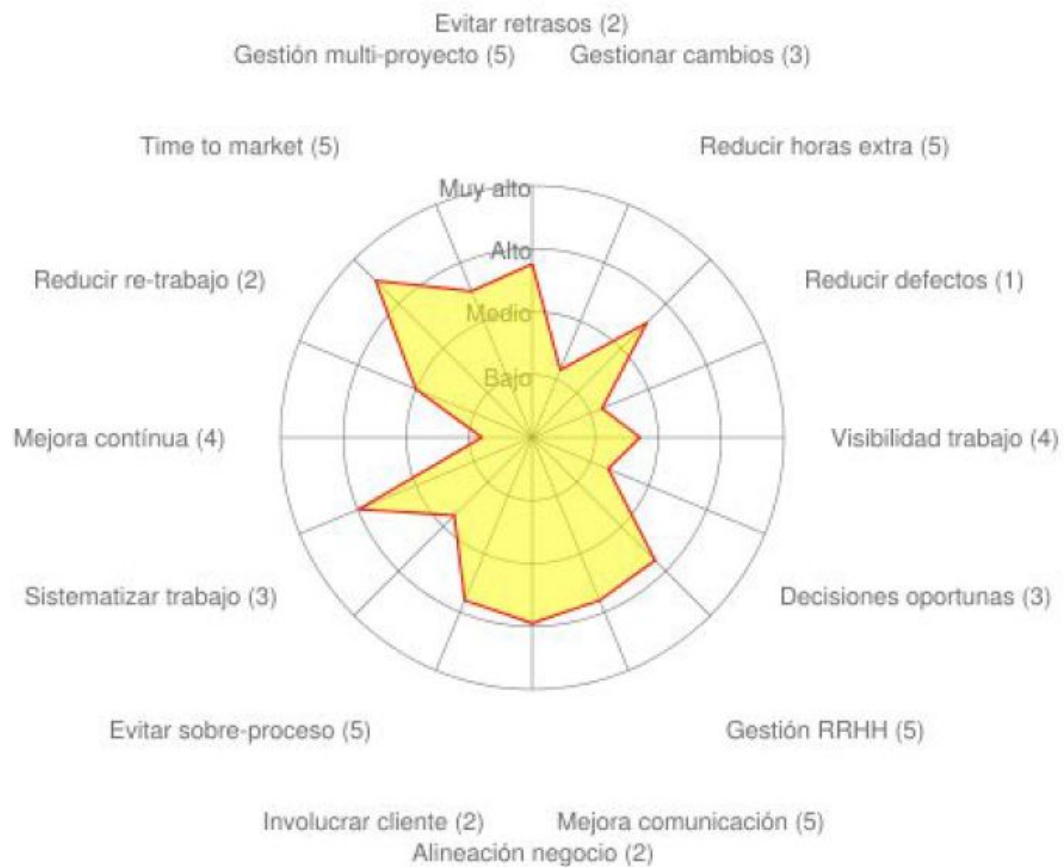
Integrantes:

Nombres y Apellidos	Email	ROL USM
Juan Pablo León León	juan.leonl.14@sansano.usm.cl	201473047-0
Rodolfo Jaramillo Guzmán	rodolfo.jaramillo.14@sansano.usm.cl	201473024-1
Félix Urtubia Carrasco	felix.urtubia.13@sansano.usm.cl	201303050-5
Sebastián Acevedo	sebastián.acevedo.13@sansano.usm.cl	201373052-3

Post-Mortem Metodológico

Resultado evaluación por áreas:



Resultado evaluación por objetivos:**Fortalezas:**

1. Se promovió la sencillez en el grupo, buscando e implementando las soluciones más simples que cumplieren con los requerimientos solicitados.
2. Se organizó el trabajo en múltiples entregas con 2 a 4 semanas de separación cada una, fomentando un trabajo incremental.
3. Existió una visión clara en el grupo sobre cómo debía ser el proyecto final respecto a sus capacidades y limitaciones.
4. Se pudo contar con la presencia de un representante del cliente en todas las presentaciones para aclarar dudas y escuchar sugerencias.

Debilidades:

1. Desarrollo poco continuo. Se trabajó en el proyecto a intervalos, generalmente se abandonaba por algunas semanas hasta que se acercaba la fecha de entrega.
2. Falta de experiencia en la planificación del proyecto.
3. Poca organización para garantizar un flujo de trabajo.

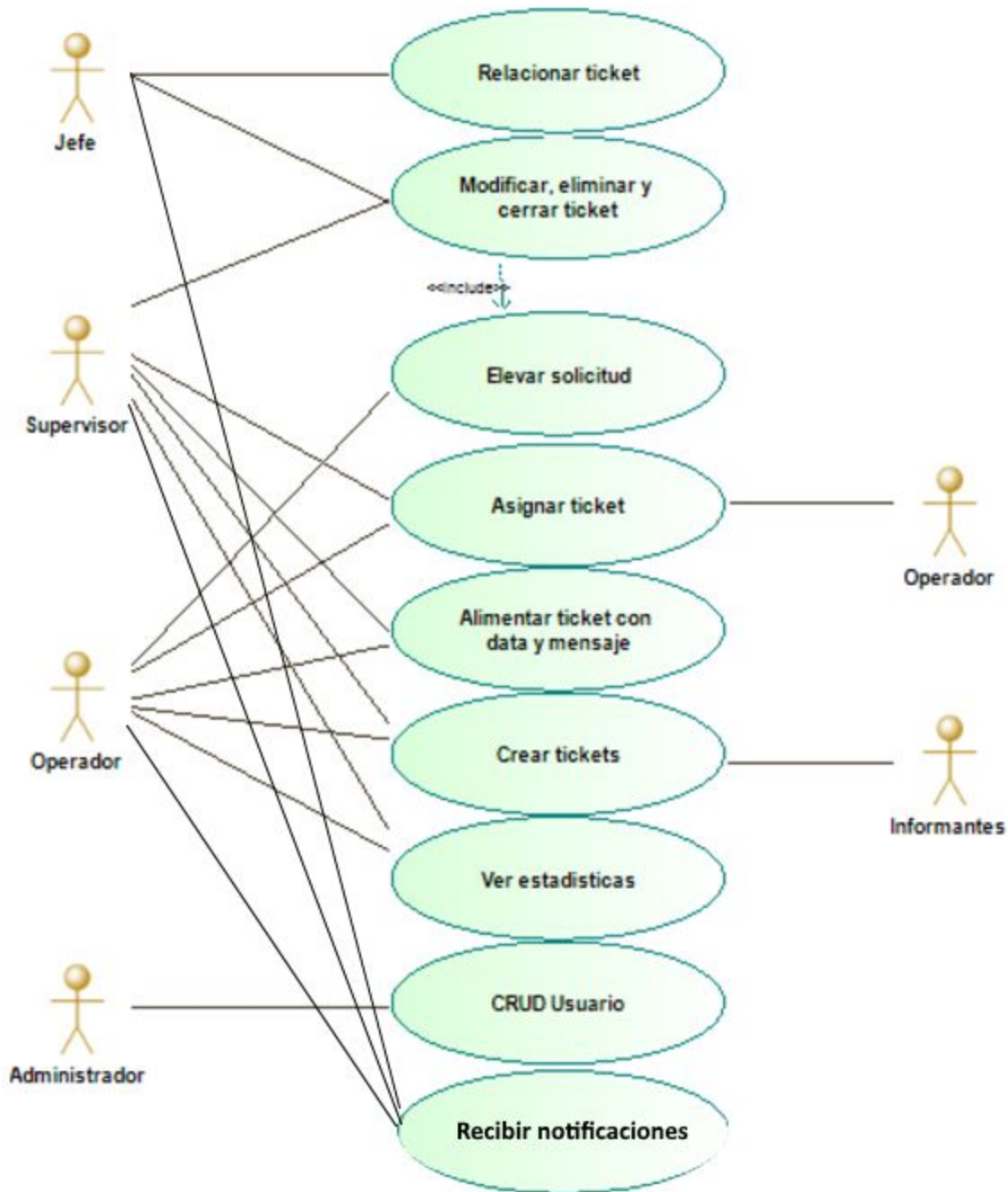
4. Se tendía a priorizar otros deberes u obligaciones por sobre el desarrollo del proyecto.
5. Existió dificultad para aprender a usar las herramientas y lenguaje para realizar el desarrollo del proyecto.
6. No hubo distribución de los deberes. Cada uno trabajaba por su cuenta y unos considerablemente más que otros.
7. Falta de motivación por desarrollar el proyecto por parte de algunos miembros.

Mejoras:

1. Trabajar exclusivamente en el proyecto, sin distracciones de otros ramos o pruebas.
2. Realizar reuniones cada dos o tres días para discutir el avance realizado hasta el momento y lo que queda por hacer.
3. Distribuir claramente las tareas y requisitos a cumplir entre los integrantes del grupo.
4. Llevar a cabo reuniones iniciales para discutir y aprender las herramientas y lenguaje que se utilizará para desarrollar el proyecto.
5. Fijar metas claras con fechas inamovibles para fomentar el cumplimiento del trabajo.
6. Tener más comunicación con el cliente vía correo electrónico.

Diagrama de Casos de Uso (final)

No hubo muchos cambios en el diagrama de casos de uso comparados con el inicial en el primer entregable, solo se agregó la funcionalidad de las notificaciones que avisan a los usuarios respecto a los cambios en los tickets en distintas situaciones.



Patrones de diseño y Frameworks (final)

Intención	Patrón de Diseño	Razonamiento
Se desea mostrar en el Diagrama de Clases la notificación en el Sistema cuando un usuario crea un ticket, se modifica o un operador envía una solicitud para modificar el estado del ticket y avisar al usuario correspondiente	Observer	Seleccionamos este patrón de diseño ya que diferentes usuarios deben recibir notificaciones de distintas acciones o registros, siendo más fácil controlar estos así. También ofreciendo flexibilidad al usuario por si no desea recibir notificaciones de cierto evento.
En el diagrama de clases se desea mostrar la interacción modelo-vista-controlador a través de paquetes separados, que permiten el entendimiento y un desarrollo más ordenado, permitiendo escalabilidad. También se desean mostrar los cambios en el modelo hechos por un usuario a todos los demás, cambiando la vista.	Modelo - Vista - Controlador	Se utiliza este patrón de diseño para separar lo que es datos de la vista, siendo ambos conectados por el controlador, esto se hace para facilitar el desarrollo y el mantenimiento del software
	Template View	Se desea hacer más entendible el código del sistema, facilitando el desarrollo y también reutilizar el código que se pueda, para así reducir tiempos de desarrollo y aumentar la cohesión del sistema.

Modelo de Dominio y Diagrama de Clases (final)

Modelo de Dominio:

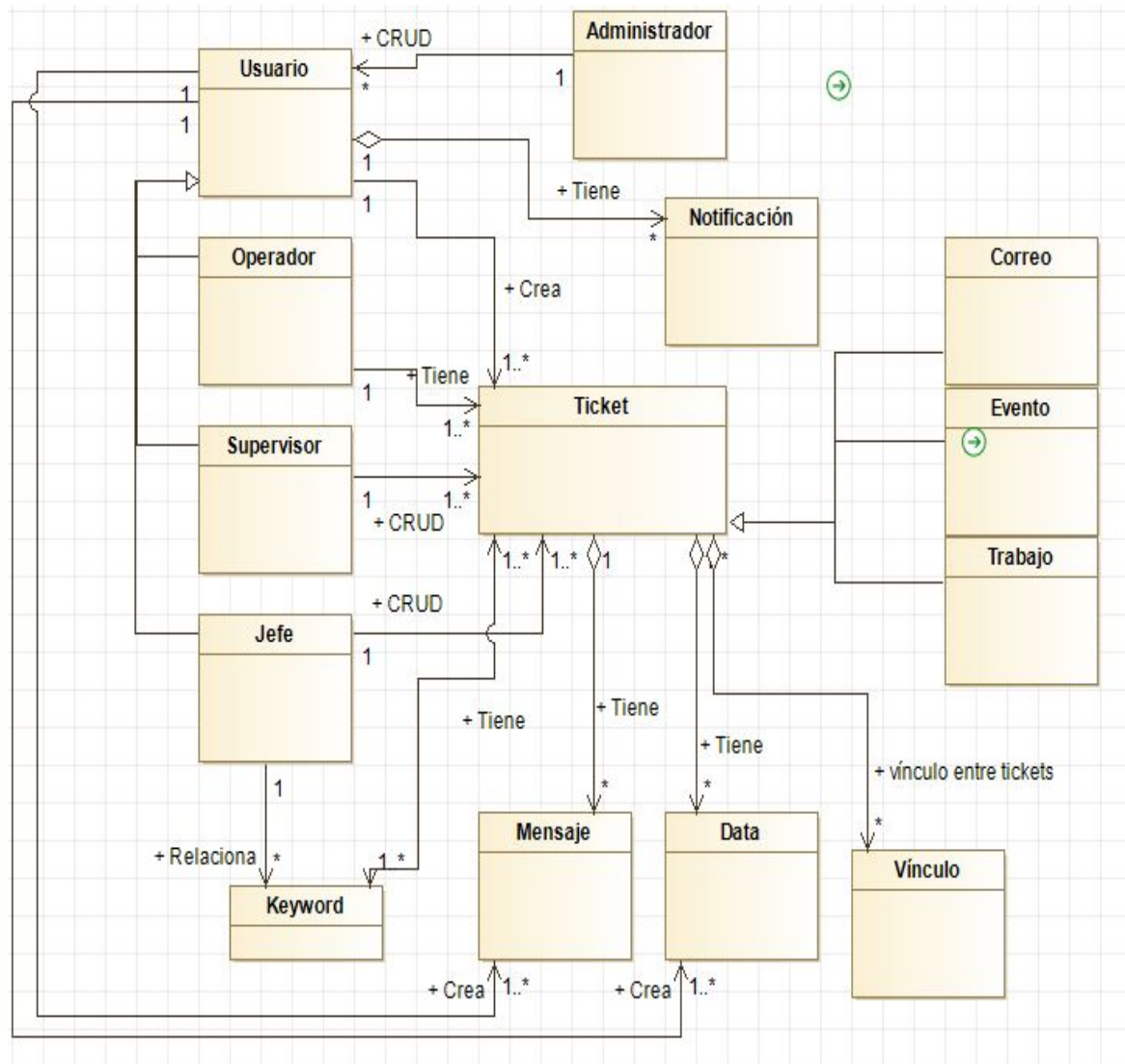
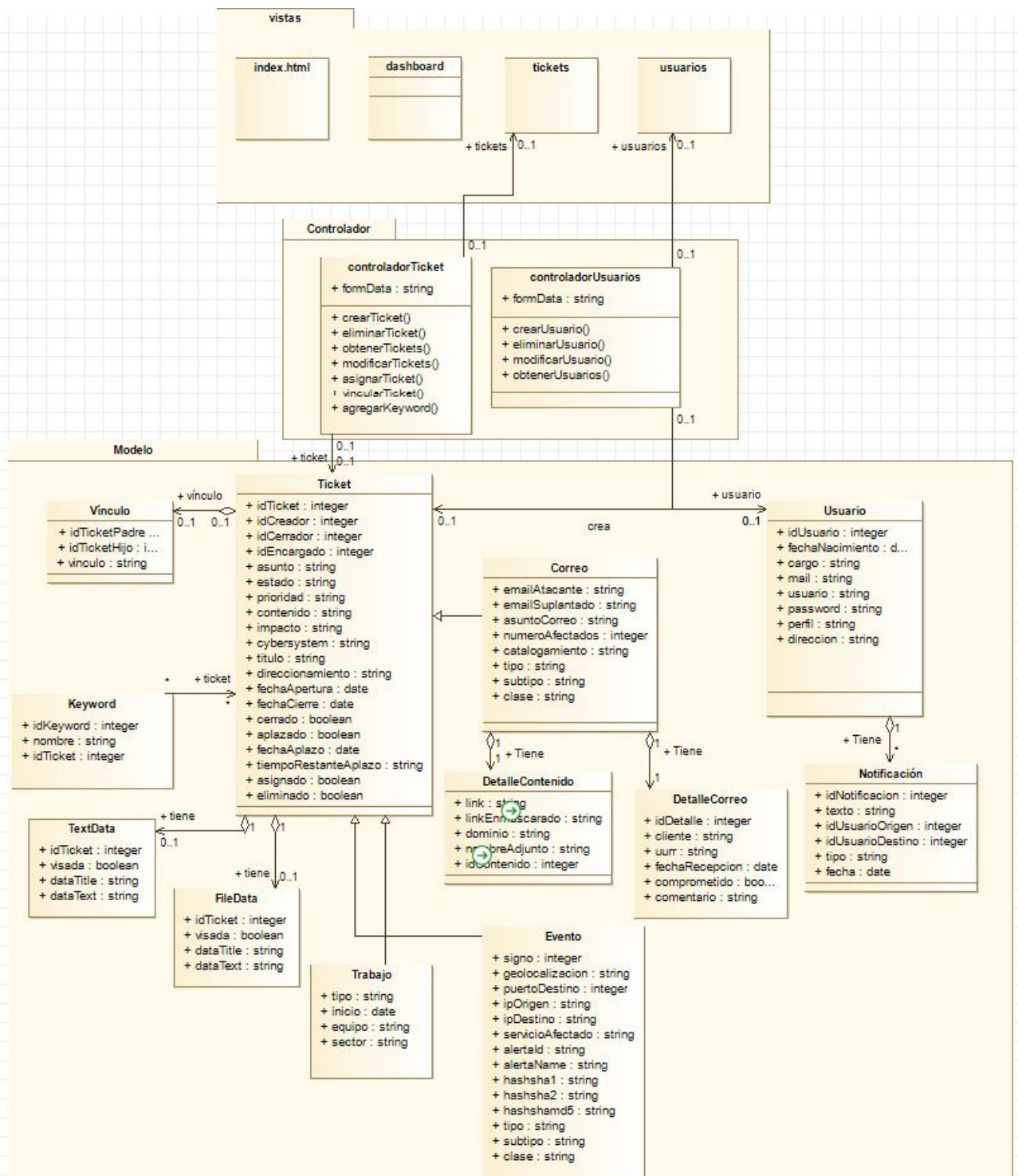


Diagrama de Clases:



Pruebas de Software (actualización)

Defecto encontrado	Mitigación	Resultado obtenido	Observaciones
Todos los ticket aparecen como urgentes al ser creados sin importar la prioridad asignada.	Cambio en el código de la función de crear ticket.	Tickets se crean con la prioridad asignada correctamente	Para evitar este tipo de errores, se debe revisar el código minuciosamente
Los tickets creados por los usuarios de tipo Jefe son agregados pero no pueden ser vistos por este usuario.	Agregar permisos dentro de la vista de "tickets" para que Jefe pueda ver todos los tickets	Jefe puede ver todos los tickets incluso los creados por él mismo	Tener claros los requerimientos facilita estos detalles
Los usuarios tipo Jefe no pueden ver los tickets que no tienen asignados un usuario encargado.	Añadir excepción dentro de la vista de "tickets" para Jefe, para poder visualizar tickets sin asignación	Jefes pueden ver todos los tickets, sin importar si están asignados o no	Tener claros los requerimientos facilita estos detalles
Existe una vista de inicio donde el usuario puede ver los tickets creados por ellos. No existe botón para volver a esta vista.	Vista eliminada.	Aquella vista ya no es accesible debido a que no tenía propósito útil.	Es necesario tener una vista estándar para cada acción y una forma de acceder a estas vistas.
Luego de que un ticket sea eliminado los usuarios de tipo Supervisor y Jefe pueden seguir modificándolos.	Aún sin solución	Aún sin solución	Aún sin solución