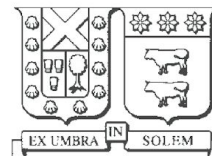




Departamento de Informática
Universidad Técnica Federico Santa María



Entregable III

Proyecto: SGA

Sistema de Gestión Académica

Equipo: LionBit

Integrantes:

Nombres y apellidos	Email	ROL USM
Benjamín Gautier Ortiz	benjamin.gautier.14@sansano.usm.cl	201460036-4
Francisco Reyes Gómez	francisco.reyesg.14@sansano.usm.cl	201473117-5
Alonso Zamorano Ponce	alonso.zamorano@sansano.usm.cl	201673056-7

Tabla de Contenidos:

1) Objetivo	3
2) Resumen ejecutivo	3
3) Árbol de utilidad	4
4) Identificación de soft goals	5
<i>Elecciones y trade-offs</i>	6,7
5) Pruebas de software	8
<i>Tabla de requisitos</i>	8,9
<i>Precondiciones</i>	9
<i>Relaciones entre requisitos y Historias usuario</i>	10
<i>Detalles casos de prueba</i>	11
6) Modelo de Calidad	12

Objetivos

En este informe se presenta el árbol de utilidad, se identifican soft-goals y se muestran las pruebas de software de la aplicación web SGA del equipo LionBit.

Resumen ejecutivo

El problema planteado por el cliente es la falta de mecanismos para evaluar la situación actual de cada ramo de los alumnos de la carrera prevención de riesgos de la sede Concepción de la USM. Ante esto se propone el “Sistema de Gestión Académica”, un software dedicado a la gestión de una carrera en términos educativos, el cual busca entregar una mayor autonomía y las herramientas necesarias a los jefes de carrera y profesores para apoyar la toma de decisiones.

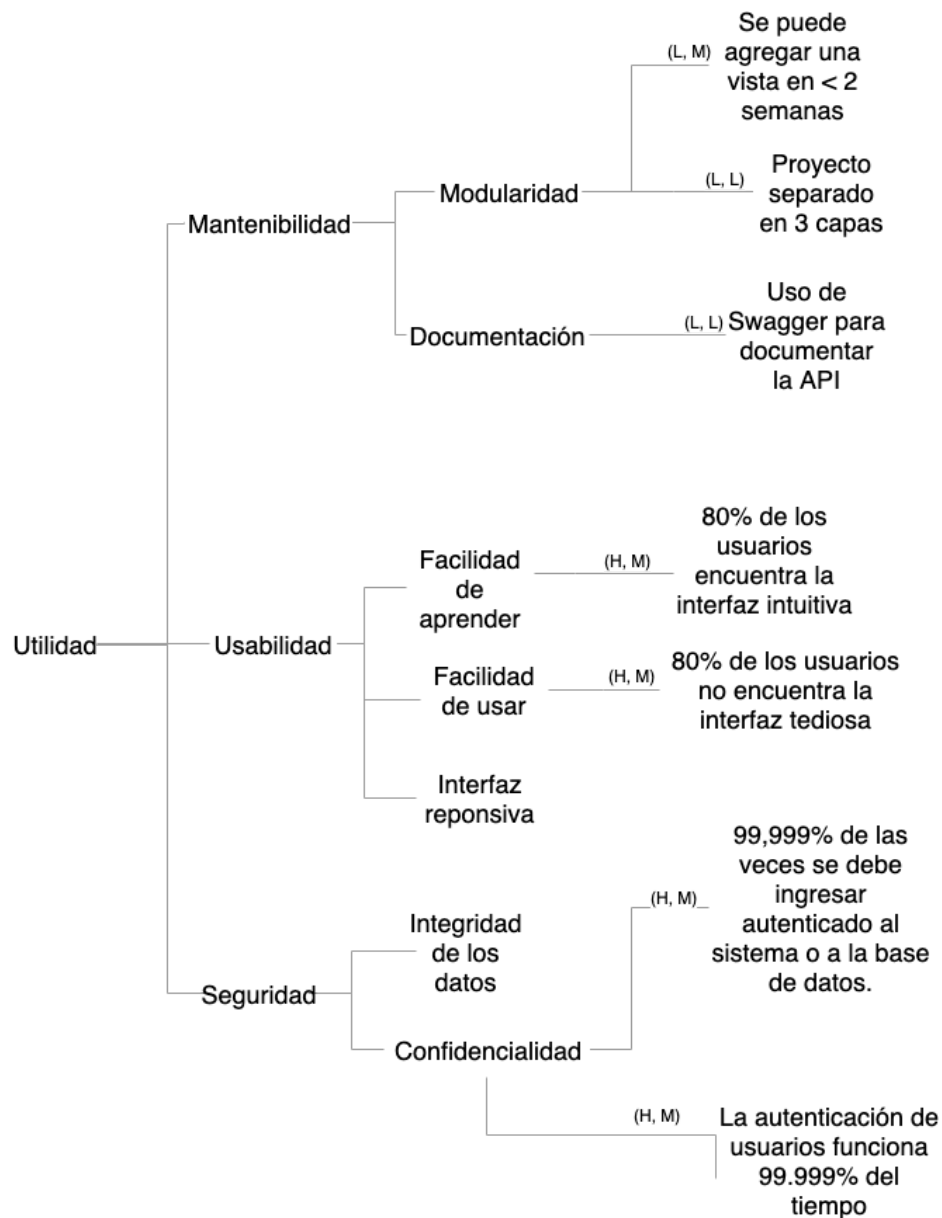
El objetivo principal de los stakeholders es conocer la situación académica de los alumnos en todo momento, y también tener herramientas e indicadores para apoyar la toma de decisiones.

El objetivo a cumplir es valioso para el cliente dado que actualmente existe una alta tasa de reprobación y de abandono de la los alumnos de la carrera Prevención de Riesgos, por lo que los indicadores generados con el sistema les darán la información necesaria para estudiar las causas de sus problemas y las posibles acciones a tomar.

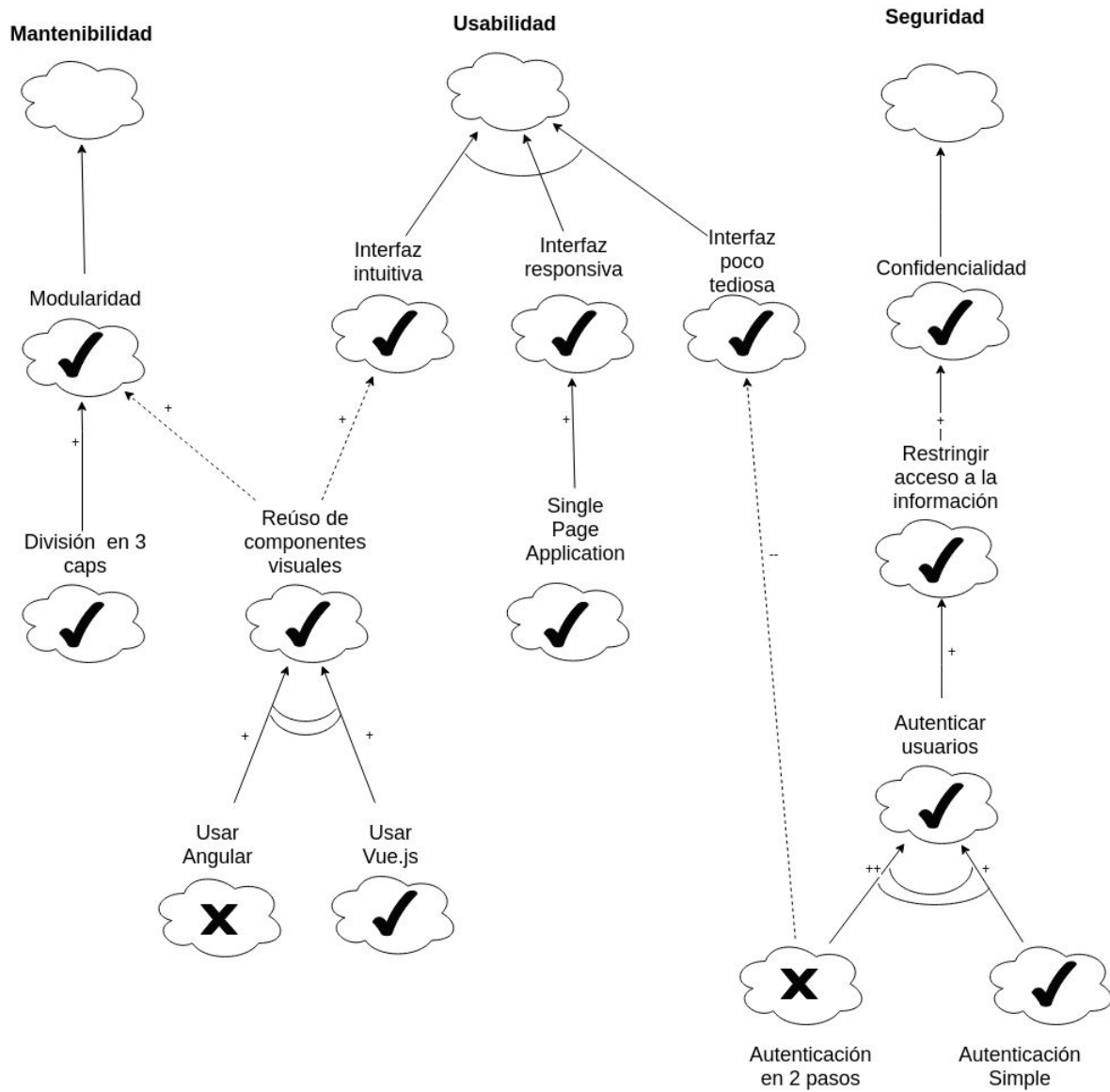
En esta primera aproximación se propone registrar la asistencia a clases y las evaluaciones de los alumnos.

I. Árbol de utilidad

En esta sección se muestra el árbol de utilidad, el cual representa gráficamente los atributos de calidad del software “SGA”, además se especifica el nivel de prioridad (para el cliente) y de dificultad (para el equipo) por cada escenario.



II. Identificación de soft goals



Sobre las elecciones:**- ¿Por qué Vue.js y no Angular?**

Cabe mencionar que ambos son frameworks de front-end, es decir para construir toda interfaz del sistema con la interactuara el usuario. En una primera instancia el desarrollo estaba orientado a los microservicios por lo que decidimos que tener un servicio para el lado del cliente también era algo bueno para la arquitectura del software, por lo que buscamos un framework acorde a nuestras necesidades, dentro de los que conocíamos estaban Angular y Vuejs.

Para decidirnos en cuanto a cuál de los dos usar, comparamos nuestra experiencia de desarrollo y la re-utilización de los componentes disponibles desarrollados ya por otros frameworks como bootstrap o material design.

Finalmente decidimos usar Vue.js, marco de trabajo bastante fácil de aprender a usar y donde encontramos muchas piezas o componentes reutilizables en donde debíamos adaptarnos al caso de uso dado.

Cabe mencionar que no utilizamos angular debido a que la curva de aprendizaje de este y la configuración del servidor del cliente es mucho más complicada, a comparación con vue.js.

- ¿Por qué Autenticación simple y no en dos pasos?

Primero mencionar que existe un costo mayor a la hora de implementar la autenticación con dos pasos, ya que se debe usar otro tipo de confirmación del usuario que quiere entrar a la aplicación, como por ejemplo un código enviado al teléfono. Lo anterior nos generaba costes en la implementación, en el uso de un proveedor de mensajes telefónicos, coste en el servicio de almacenamiento, e invertir más tiempo (acotado) en la historia de autenticación, además el cliente no especificó nada sobre el tema.

Por otro lado, la autenticación en dos pasos por lo general es usada para temas sensibles (sistemas críticos o donde los datos deben ser confidenciales al extremo). Por lo anterior, decidimos usar una autenticación simple, con contraseñas cifradas en un algoritmo de cifrado.

- **Single Page Application:**

Al desarrollar el proyecto utilizando tecnologías como vue.js o angular, estas se basan en la ideal del Single Page, esta consiste en que la web app no carga todos los recursos al moverse de una página a otra, sino que utiliza y reutiliza los componentes que son necesarios para cargar visita, permitiendo mejoras en los tiempos de carga (que consecuentemente, aporta bastante en términos de usabilidad al sistema y utilizará menos recursos puesto que se hacen menos peticiones al servidor), a su vez crean una estructura sólida y fácil de mantener (puesto que las páginas ahora se construyen a partir de componentes y módulos).

- **Aplicación construida en 3 capas:**

Al desacoplar el proyecto en tres capas aumentamos la modularización del código y por ende, facilita la mantención del mismo. Para ello, se busca establecer un “bounded context”, esto consiste que todas las funciones relacionadas entre sí están inmersas dentro de la misma unidad de negocio o entidad, es decir, cada entidad tendrá un dominio dependiendo de su rol dentro del sistema; como cliente, servicios y capa de datos. Y de esta forma hay una perfecta alineación entre las unidades de negocio y la estructura del sistema.

En este proyecto, este se separa en las capas de datos, de presentación y de negocios, cada una representada según la tecnología que se esté utilizando para su construcción, es decir, para la capa de negocios, se utilizara Spring, donde se generara las entidades correspondientes y las relaciones entre estas.

III. Pruebas de software

Tabla de de requisitos funcionales:

ID	Nombre del Requisito	Roles asociados	Descripción
RF1	Ingreso y salida del sistema	1) Admin 2) Jefe de carrera 3) Profesor	<p><u>Usuarios de la plataforma</u>: Cada usuario registrado en la plataforma puede ingresar a la misma usando sus credenciales.</p> <p>Una vez que estos ingresen, el sistema deberá mostrar el inicio dependiendo del tipo de usuario que haya ingresado.</p>
RF2	El sistema debe ser capaz de generar nuevos usuarios	1) Admin	<p><u>Admin</u>: Luego de presentar sus credenciales en la zona de ingreso del sistema.</p> <p>El sistema debe mostrar un menú que indique que es posible administrar usuarios, dentro de este apartado, el sistema deberá proveer de un formulario para la creación de usuarios. Se requiere como mínimo los siguientes datos: Nombre completo del usuario, correo electrónico, contraseña y rol de usuario (Admin, jefe de carrera o Profesor).</p>
RF3	El sistema debe permitir el crear nuevas asignaturas	1) Admin 2) Jefe de carrera	<p><u>Admin</u>: El administrador puede ingresar al sistema usando sus credenciales.</p> <p><u>Jefe de carrera</u>: Un jefe puede ingresar al sistema al sistema usando sus credenciales.</p> <p>Una vez que estos usuarios ingresen al sistema, el sistema debe mostrar un menú que indique es posible administrar a las asignaturas asociadas a la carrera. Una vez dentro de este apartado, el sistema deberá proveer un formulario para crear una nueva asignatura. Se requiere como mínimo los siguientes datos: Nombre asignatura, Sigla USM, Créditos asociados.</p>

RF4	El sistema debe permitir el crear nuevos semestres	1) Admin 2) Jefe de carrera	<p><u>Admin</u>: El administrador puede ingresar al sistema usando sus credenciales.</p> <p><u>Jefe de carrera</u>: Un jefe puede ingresar al sistema al sistema usando sus credenciales.</p> <p>Una vez que estos usuarios ingresen al sistema, el sistema debe mostrar un menú que indique es posible administrar los semestres que utilizaran a lo largo de la carrera. Una vez dentro de este apartado, el sistema deberá proveer un formulario para crear un nuevo semestre. Se requiere como mínimo los siguientes datos: Sigla del semestre a crear (1S,2S,3S), año donde ocurrirá.</p>
RF5	El sistema debe notificar si una acción fue realizada satisfactoriam ente	1) Admin 2) Jefe de carrera 3) Profesor	<p><u>Usuarios de la plataforma</u>: Cada usuario registrado en la plataforma puede ingresar a la misma usando sus credenciales.</p> <p>Una vez que estos ingresen y realicen alguna acción tales como la creación de un curso (solo administradores y jefe de carrera) el sistema debera arrojar una notificación visual que evidencie que lo realizado se es satisfactorio o no.</p>

Precondiciones (PC) que serán utilizados en los casos de prueba:

- PC1: El correo del usuario no debe estar en blanco
- PC2: La contraseña no debe estar en blanco
- PC3: Si el correo del usuario y la contraseña son válidos. Ingresar.
- PC4: El campo nombre del usuario no debe estar en blanco.
- PC5: El campo apellido del usuario no debe estar en blanco.
- PC6: El campo correo electrónico del usuario no debe estar en blanco.
- PC7: El campo de contraseña del usuario no debe estar en blanco.
- PC8: El dropdown de asignación de roles de usuario no debe estar nulo.
- PC9: Los campos deben indicar que son obligatorios (cuando así lo sea).
- PC10: Si los datos están correctamente ingresados y son válidos en cada uno de los campos, aceptar la creación de la entidad seleccionada.
- PC11: El campo nombre de la asignatura no debe estar en blanco.
- PC12: El campo de sigla de la asignatura no debe estar en blanco.
- PC13: El campo de créditos de la asignatura no debe estar en blanco.
- PC14: El campo de sigla asignatura no debe estar en blanco.
- PC15: El campo de año donde se dictara la asignatura no debe estar sin seleccionar.

Relaciones con historias de usuario:

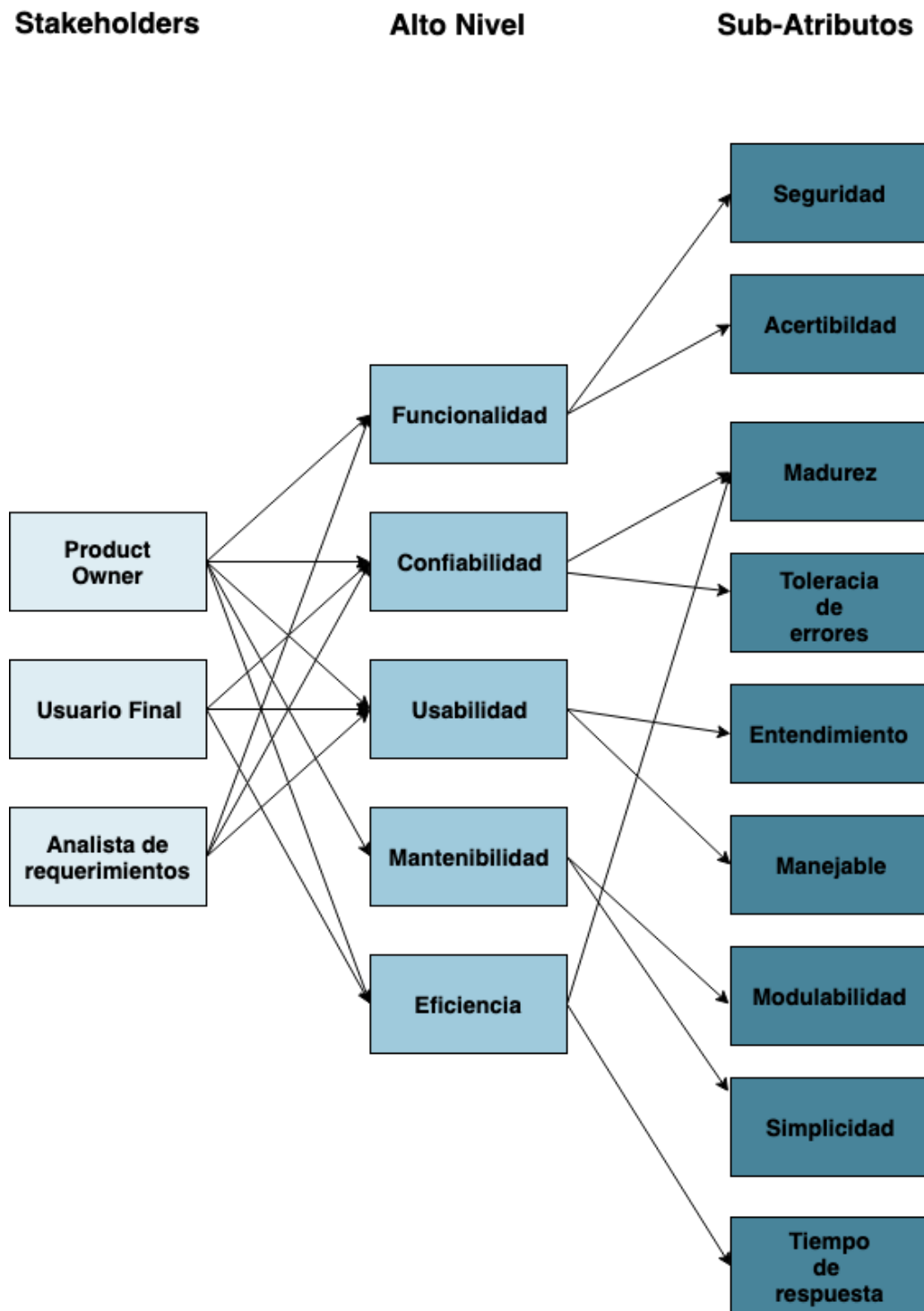
- **RF1** satisface la historia de usuario *“Como usuario registrado de la plataforma quiero ingresar al sistema haciendo uso de mi correo y contraseña.(HU1)”*
- **RF2** satisface la historia de usuario *“Como Administrador General, quiero inscribir usuarios al sistema con roles específicos para que puedan entrar al sistema. (HU2)”*
- **RF3** satisface la historia de usuario *“Como Administrador General, quiero inscribir asignaturas al sistema para relacionarlos posteriormente con los profesores correspondientes.(HU3)”*
- **RF4** satisface la historia de usuario *“Como Administrador General, quiero inscribir semestres a través de los años para crear mallas semestrales a ser rendidas en el semestre dado.(HU4)”*
- **RF5** satisface la historia de usuario *“Como Usuario, quiero notificaciones que demuestran que la operación computada es correcta o fallida.(HU5)”*

Detalles de Casos de Prueba:

Caso de Prueba #	RF#	PC#	Caso de Prueba	Pasos del caso	Datos utilizados	Resultados Obtenidos
CP1	RF1	PC1, PC2, PC3	Verificar ingreso	1.Ingresar a la página de Ingreso 2.Ingresar Correo 3.Ingresar contraseña 4.Clic en “Ingresar”	Correo: francisco.reyegom@gmail.com Contraseña: Hola!123456	Ingreso exitoso, pero la plataforma aún no hace distinción del rol del usuario ingresado. Por lo tanto, aún el sistema no filtra las características de esta, según el rol.
CP2	RF2	PC4, PC5, PC6, PC7, PC8, PC9, PC10	Crear un usuario del sistema	1.Hacer ingreso al sistema con las credenciales necesarias 2.Hacer clic en el apartado “Usuarios” en el menú de la página principal. 3.Hacer clic en crear un “añadir nuevo usuario” 4.Ingresar nombre del usuario 5.Ingresar apellido 6.Ingresar correo electrónico	Nombre: Francisco Apellidos: Reyes Correo: francisco.reyesg.14@sanano.usm.cl Contraseña: Qweqwe!123456 Rol: Administrador	El sistema cierra el formulario y refresca la tabla de usuarios existentes, donde ahora figura el usuario recién ingresado. <i>Falta lanzar una notificación para indicar que se guardó correctamente.</i>

				7.Ingresar contraseña provisoria 8.Ingresar rol del usuario 9. Clic en "Guardar"		Este problema se mitigo al realizar la historia de usuario HU5.
CP3	RF3	PC9, PC10, PC11, PC12, PC13	Crear una asignatura correspondiente a la carrera	1.Hacer ingreso al sistema con las credenciales necesarias 2.Hacer clic en el apartado "Asignaturas" en el menú de la página principal. 3.Hacer clic en crear un "añadir nueva asignatura" 4.Ingresar nombre de la asignatura 5.Ingresar sigla USM 6.Ingresar créditos asociados 7.Clic en "Guardar"	Nombre asignatura: Ing. de software Sigla USM: INF-225 Créditos: 3	El sistema cierra el formulario y refresca la tabla de asignaturas existentes en el sistema, donde ahora figura la asignatura recién ingresada. Falta lanzar una notificación para indicar que se guardó correctamente. Este problema se mitigo al realizar la historia de usuario HU5.
CP4	RF4	PC10, PC14, PC15	Crear un semestre a efectuarse durante la carrera	1.Hacer ingreso al sistema con las credenciales necesarias 2.Hacer clic en el apartado "Semestre" en el menú de la página principal. 3.Hacer clic en crear un "añadir nuevo semestre" 4.Ingresar año a dictarse 5.Ingresar sigla/número de semestre 7.Clic en "Guardar"	Año a dictarse: 2019 Sigla/Número de semestre: 1S	El sistema cierra el formulario y refresca la tabla de semestres existentes en el sistema, donde ahora figura el semestre recién creado.
CP5	RF5	PC10, PC14, PC15	Notificar si es satisfactorio el crear un semestre a efectuarse durante la carrera	1.Hacer ingreso al sistema con las credenciales necesarias 2.Hacer clic en el apartado "Semestre" en el menú de la página principal. 3.Hacer clic en crear un "añadir nuevo semestre" 4.Ingresar año a dictarse 5.Ingresar sigla/número de semestre 7.Clic en "Guardar"	Año a dictarse: 2019 Sigla/Número de semestre: 1S	El sistema cierra el formulario y arroja una notificación satisfactoria.

IV. Modelo de Calidad



Para explicar el modelo anterior lo haremos por tipo de stakeholder:

1. Product Owner: Este tipo de stakeholder se encuentra preocupado por:
 - a. Funcionalidad: El dueño del producto posee una visión del producto, la cual tiene funcionalidades que se deben cumplir con la visión, por lo que a este le concierne que las funcionalidades del sistema estén completas y funcionen de forma óptima.
 - b. Confiabilidad: La aplicación debe realizar las operaciones que aseguren que no se perderá información, que el sistema será capaz de sobreponerse a errores y que las operaciones serán realizadas de forma oportuna y bajo demanda.
 - c. Usabilidad: Al dueño del producto le preocupa que el sistema sea fácil de usar, que los usuarios no se pierdan en el sistema y puedan navegar por la aplicación de forma fluida.
 - d. Mantenibilidad: Uno de las principales preocupaciones del dueño del producto, es el como se va a mantener el sistema durante el tiempo. Por lo que idear una estrategia basada en modulos y simplificar el código, ayudara bastante en la documentación y mantenibilidad del código.
 - e. Eficiencia: Al dueño del producto le preocupa que el sistema sea eficiente en términos de recursos dado que busca minimizar gastos con la implementación del sistema, por lo que necesita una baja carga al servidor.
2. Usuario final:
 - a. Confiabilidad: El usuario busca realizar operaciones que persistan en el tiempo, que no sean interrumpidas por errores de la aplicación que esta utilizando.
 - b. Usabilidad: El usuario necesita una aplicación fácil de usar dado que no dispone de mucho tiempo ni tiene la experticie técnica para configurar entradas complejas.
 - c. Eficiencia: El usuario final busca minimizar su tiempo usando el sistema, por lo que necesita que las operaciones realizadas en el mismo sean eficientes y eficaces, que respondan de forma rápida a las entradas del usuario.

3. Analista de requerimientos:

- a. Funcionalidad: El analista de requerimientos se preocupa de las funcionalidades de la aplicación, ya que éste en una primera iteración es quien las define y las desgloza para tener un backlog que esté orientado hacia la visión del Product Owner.
- b. Confiabilidad: El analista de requerimientos le preocupa este ámbito dado que necesita consistencia y coherencia entre los requisitos, es decir, relacionar los requerimientos de forma correcta el pos de ir madurando el sistema a través de las iteraciones.
- c. Usabilidad: El analista de requerimientos está involucrado y le concierne este punto dado que necesita que la aplicación sea usable, por lo que necesita ser claro y detallado en la definición del product backlog para que los desarrolladores realicen un sistema intuitivo y amigable con el usuario final.