

2015/2016



Plan SQA “Equidata”

Abdellah Absaoui

Alejandro Alexiades

Rubén Barrado

Pablo Márquez

Pablo Martín

Tomás Muñoz

Jaime Tamames

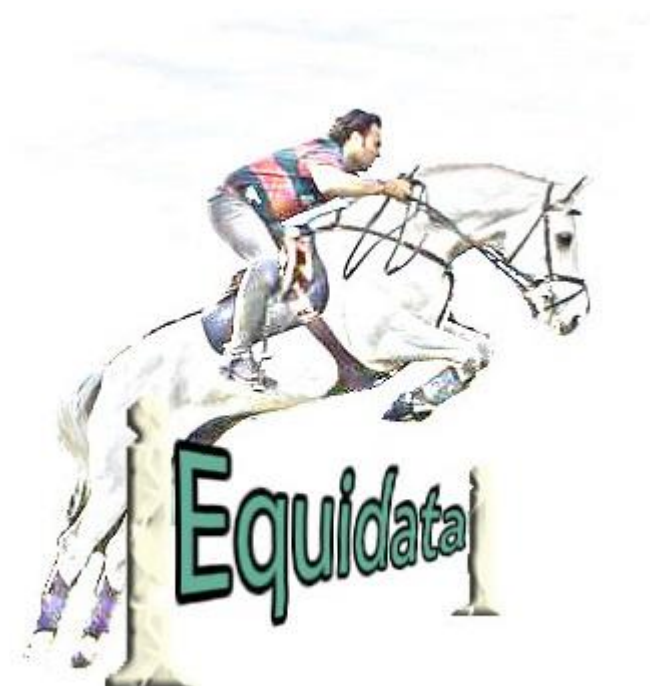
Eduardo Vela

Ismael Ventura

Carlos Villasur

Universidad Complutense

Facultad de Informática





Historial de versiones

Fecha del cambio	Identificador del cambio	Causa del cambio	Personas encargadas
01/04/2016	SQA-V1.0	Creación del documento	Carlos Villasur.
15/05/2016	SQA-V1.1	Revisión e incorporación de los puntos creados por Carlos Villasur, Alejandro Alexiades, Pablo Márquez y Pablo Martín.	Tomás Muñoz Testón.



ÍNDICE

1- Propósito	5
2- Documentos de referencia	6
3- Gestión.....	6
3.1 Organización	6
3.2 Actividades	7
3.3 Funciones y responsabilidades	8
3.4 Estimación de recursos	9
4- Documentación	10
4.1 Propósito	10
4.2 Requisitos mínimos de documentación	10
4.3 Otra documentación	12
5- Estándares, prácticas, convenciones y métricas.....	12
5.1 Propósito	12
5.2 Contenido	13
6- Revisiones de software	17
6.1 Propósito	17
6.2 Requisitos mínimos	17
6.3 Otras revisiones o auditorías.....	20
7- Pruebas.....	21
8- Informe de errores y acciones correctoras	21
9- Herramientas, técnicas y metodologías	22
10- Control de medios	23
10.1 Identificación de medios	23
10.2 Protección de medios	24



11- Control de proveedor	25
12- Colección de registros, mantenimiento y conservación	27
13- Formación	28
13.1 Funciones.....	28
14- Gestión del riesgo	29
15- Glosario.....	30
16- Procedimiento de cambio e historial del plan SQA	31
17- Apéndices	32
17.1 Revisiones técnicas formales-Find&fun.....	32
17.2 Revisiones técnicas formales-Linker	35



1- PROPÓSITO

Como ingenieros, nuestro objetivo a la hora de construir productos software, es construir software de calidad.

El Plan SQA o Plan de control de calidad es un conjunto de procedimientos organizativos, que se realizan para asegurar que los procesos y productos de software cumplen con los requisitos y estándares de IS que le han sido asignados. Todo producto de software busca satisfacer el propósito para el cual fue creado, estar en sintonía con el ambiente donde se va a aplicar, cumplir con los requisitos y necesidades de los clientes y usuarios, además de ser sostenible, de confianza y estable. El logro de estos factores permite decir que se trata de un software de alta calidad.

Dicho esto, el propósito de este plan es tratar de asegurar y garantizar la calidad de nuestro proyecto, así como de establecer las pautas y actividades que deben desarrollarse para garantizar la calidad del producto “Equidata”.

Este plan abarca todas las partes del ciclo de vida del producto, relacionadas con las fases de elaboración, construcción, evaluación y transición. De esta manera se irá adquiriendo una visión más profunda y segura de que la calidad del producto está cumpliendo sus objetivos y podremos progresar semanalmente con el trabajo.

Como equipo de proyecto tenemos los siguientes objetivos:

- Definir los requisitos de calidad a ser verificados.
- Indicar las partes del ciclo de vida cubierto por el plan SQA así como las líneas de trabajo que serán contempladas en el mismo.
- Describir las tareas de calidad a realizar.
- Especificar los documentos involucrados en el desarrollo del plan.



2- DOCUMENTOS DE REFERENCIA

- Material didáctico de la asignatura Gestión de Proyectos Software.
- 730-2002 - IEEE Standard for Software Quality Assurance Plans.
(Este estándar está más centrado en el plan de garantía de calidad y en cómo implementarlo).

3- GESTIÓN

La gestión de calidad describe la estructura organizativa que influye y controla la calidad del producto software. Mediante esta gestión se identifican las tareas asociadas al plan SQA, los roles y responsabilidades dentro del propio plan, así como identificar a los responsables de preparar y mantener el plan. Las tareas asociadas al plan SQA, estarán a su vez relacionadas con la planificación temporal.

3.1 ORGANIZACIÓN

Las actividades serán realizadas por todos los miembros del grupo, repartiendo las tareas entre los integrantes y supervisadas por los subjeses de proyecto y en su medida posteriormente validadas por el jefe del proyecto.

La organización del equipo del proyecto seguirá una estructura descentralizada controlada explicada en el documento <<Plan de Gestión de Configuración (apartados 4.2 y 4.3)>> y las tareas descritas en este plan no tienen a personal asignado sino que se reparten entre todo el equipo.

Cabe destacar que, aunque la responsabilidad del correcto trabajo de los subgrupos y del grupo en general recae sobre los subjeses y jefe del proyecto, cualquier miembro del grupo puede participar de forma activa y dar ideas para el correcto mantenimiento del plan SQA.



3.2 ACTIVIDADES

Básicamente vamos a llevar a cabo dos acciones destinadas al mantenimiento de la calidad en nuestro proyecto:

- RTFs (Revisiones Técnicas Formales).
- Revisiones por parte de los subjeses y jefe del proyecto.
- Tareas secundarias.

Una RTF (revisión técnica formal, o inspección formal) es un conjunto de procedimientos llevado a cabo por los desarrolladores cuyo principal objetivo es detectar errores antes de que se conviertan en defectos. Para cumplir el objetivo general se deberán alcanzar una serie de objetivos:

- Descubrir errores.
- Verificar que el software alcanza los requisitos exigidos.
- Garantizar que el software se desarrolla de acuerdo con unos ciertos estándares predefinidos.
- Conseguir un software desarrollado de manera uniforme.
- Hacer que los proyectos sean más manejables.

Nos dividimos el trabajo en revisores (personas que revisaron el trabajo de otros grupos) y revisados (personas que defendieron nuestro proyecto ante otros grupos).

Las revisiones por parte de los subjeses y jefe del proyecto tienen un carácter menos formal, y con ellas se busca que los documentos sean lo más completos y precisos posibles.

En el caso de ser detectado algún error, se comunicará a todos los miembros del equipo para su siguiente corrección.

Como tareas secundarias realizamos:



- Reuniones semanales: A través de las reuniones con los miembros podremos determinar los errores y poder tratarlos para su corrección, así como la diferente asignación de las tareas por parte de los Subjefes y el Jefe del proyecto. Dichas reuniones se realizan los martes en el laboratorio de la facultad de informática de la Universidad Complutense de Madrid de 11:00 a 13:00.
- Consulta expertos: solicitaremos el asesoramiento a los profesores Gonzalo Méndez y Susana Bautista para resolver posibles dudas o problemas que nos surjan.

3.3 FUNCIONES Y RESPONSABILIDADES

En este apartado del plan de SQA se nombran a las funciones descritas en el apartado anterior (3.2 Actividades de calidad a realizarse) y se identifican a los responsables del equipo de proyecto que serán los encargados de realizarlas.

Por lo tanto, las RTFs realizadas hasta el momento se han llevado a cabo por los siguientes integrantes:

- Los revisores fueron: Todos los miembros del grupo.
- Los revisados fueron: Todos los miembros del grupo.

Las revisiones semanales realizadas por los subjefes de su trabajo semanal y posteriormente la revisión general del jefe de todo el trabajo elaborado por el equipo de proyecto en esa misma semana es realizada por los siguientes componentes del equipo:

- Subjefes:
 - Equipo ACAI: Carlos Villasur.
 - Equipo TRE: Tomás Muñoz.
 - Equipo PJP: Pablo Márquez.
- Jefe: Tomás Muñoz Testón.



Las tareas secundarias observables en el apartado anterior serán realizadas por los siguientes miembros del equipo:

- Reuniones semanales: Todo el equipo de proyecto.
- Consulta Experto: Será el profesor de la asignatura, Gonzalo Méndez, quien nos resolverá dudas y problemas a cualquiera de los miembros del equipo.

Además, Como se puede observar en las tareas secundarias, al ser un equipo de calidad dinámico es labor de todo el equipo de proyecto el mantener el plan SQA siempre revisado y actualizado, cualquier incidencia o irregularidad que se observe deberá de ser comunicado cuanto antes mediante los diferentes medios de comunicación que tiene disponible el equipo y será solucionado en la mayor brevedad posible.

3.4 ESTIMACIÓN DE RECURSOS

Para poder realizar las actividades de calidad citadas anteriormente se deben destinar recursos humanos para garantizar que dicha actividad se pueda realizar de manera eficiente y eficaz. Además, las personas destinadas a este fin deben poder acceder a Google Drive y Word para que todo nuestro proyecto se encuentra alojado en un sitio común a todos los integrantes del mismo.

Recursos necesarios:

- Recursos humanos: integrantes del proyecto “Equidata”.
- Recursos materiales: Ordenadores con acceso a internet y Microsoft Office instalado.



4- DOCUMENTACIÓN

4.1 PROPÓSITO

En este punto especificaremos cada uno de los documentos que deberán ser revisados durante el proyecto los cuales, estarán relacionados con el desarrollo del mismo de una forma periódica para asegurar que se cumplen todos los estándares de calidad pre establecidos.

4.2 REQUISITOS MÍNIMOS DE DOCUMENTACIÓN

La forma en que aseguramos la calidad del proyecto se encuentra a lo largo de los siguientes documentos con el objetivo de satisfacer las necesidades del cliente de la mejor forma posible:

- Especificación de Requisitos (SRS)
- Plan de Gestión de Configuración (PGC)
- Plan del Proyecto (PdP)
- Plan de Gestión de Riesgos (PGR)
- Plan de Calidad (SQA)
- Revisiones Técnicas Formales (RTF'S)
- Documentación generada a consecuencia de RTF'S (Actas de cambios realizados).

4.2.1 DESCRIPCIÓN DE REQUISITOS SOFTWARE

Los requisitos del proyecto Equidata se detallan en la versión del documento de «Especificación de Requisitos» que se ha realizado en consecuencia con el estándar IEEE 830-1998.



4.2.2. DESCRIPCIÓN DE DISEÑO SOFTWARE

El diseño de nuestra página web se ha realizado con C++ builder versión de 2010. El estándar que utilizaremos en el diseño de la aplicación ha sido creado por el equipo Equidata y se especifica más adelante en este mismo documento (apartado 5.2).

4.2.3 PLAN DE VERIFICACIÓN Y VALIDACIÓN

El proyecto de Equidata no dispone del documento de verificación y validación, dado que este proyecto no se llevará a cabo en la fase de codificación por lo que no habrá pruebas para dicho documento. En el documento de Plan de Gestión de Configuración (PGC), recogemos información correspondiente a este tema.

4.2.4 INFORME DE RESULTADOS DE VERIFICACIÓN Y VALIDACIÓN

Tal y como explicamos en el apartado anterior, nuestro proyecto no cuenta con la documentación de las pruebas de verificación y validación. A causa de esta razón no podemos obtener informes de resultados.

4.2.5 DOCUMENTACIÓN DE USUARIO

En este apartado se debería de reflejar un manual destinado al usuario que daría el soporte necesario para saber utilizar el programa de forma correcta y también para saber si las necesidades del usuarios se satisfarían con las operaciones que ofrece el programa.

Dado que la asignatura no requiere de este apartado, lo omitimos.

4.2.6 PLAN DE GESTIÓN DE CONFIGURACIÓN SOFTWARE

Equidata dispone de un documento «Plan de Gestión de Configuración Software» (PGC) en el cual se puede consultar toda la información de este campo con detalle.



4.3 OTRA DOCUMENTACIÓN

La documentación añadida de forma complementaria a nuestro proyecto es:

- Los correspondientes informes de personalidad MBTI y KIVIAT realizado al comienzo del proyecto a cada miembro.

5- ESTÁNDARES, PRÁCTICAS, CONVENCIONES Y MÉTRICAS

5.1 PROPÓSITO

En este apartado explicaremos los estándares, prácticas, convenciones y métricas a utilizar sobre nuestro proyecto para que éste tenga la calidad deseada.

El proyecto Equidata utilizará:

- Estándares: IEEE 1058-1998 (Plan de proyecto), IEEE 830-1998 (Especificación de requisitos), IEEE 1540-2001 (Plan de gestión de riesgos), IEEE 730-2002 (Plan de Garantía de Calidad), IEEE 828-2012 (Plan de la Gestión de la Configuración).
- Convenciones: Plantilla oficial de la documentación del equipo Equidata (Explicada con detalle en el apartado 5.2).
- Métricas:
 - Índice de productividad = tamaño / esfuerzo = líneas de código generado / horas trabajadas.
 - Tasa de defectos = defectos / tamaño = número de errores / líneas de código generadas.



- **Facilidad de mantenimiento:** Es la métrica con la que se mide la facilidad para corregir un defecto o un cambio en los requisitos realizado por el cliente. La corrección será la actividad que más esfuerzo necesite
- **Facilidad de uso:** La facilidad de uso es un intento de cuantificar lo amigable que puede ser el programa con el usuario. Se puede medir en función de cuatro características:
 - Tiempo en el que un individuo se acostumbra al sistema y aprende a usarlo de forma fluida.
 - Conocimiento y habilidad necesaria para conocer el entorno.
 - Aumento en relación a la productividad provocada por el uso del sistema de forma eficiente..
 - Opinión de los primeros usuarios a cerca de la aplicación.
- **Índice de fallo en documentación:** número de errores detectados en un documento tras una revisión técnica formal por hoja.

5.2 CONTENIDO

En este apartado describimos el diseño de la documentación y algunos estándares a seguir durante el proyecto.

- **Estándares de documentación:** En cuanto a los documentos del proyecto Equidata, existe una plantilla de diseño oficial y de carácter obligatorio para todos los documentos. En dicha plantilla el formato establecido incluye:
 - La portada con el logo de la aplicación, el título y los nombres de los miembros del equipo y compañía del documento (Figura 1).
 - La cabecera y el pie de páginas con el escudo de la UCM y los datos del documento (Figura 2).
 - La fuente para el cuerpo será Arial 12, justificado e interlineado 1.5, para los títulos será Arial 14 en negrita y títulos de apartados será Arial



11 con sombreado azul o con bordes dependiendo de la importancia del título (Figura 3).

- Los índices, indexados para facilitar el acceso y con el número de página en el que está ese apartado (Figura 2).

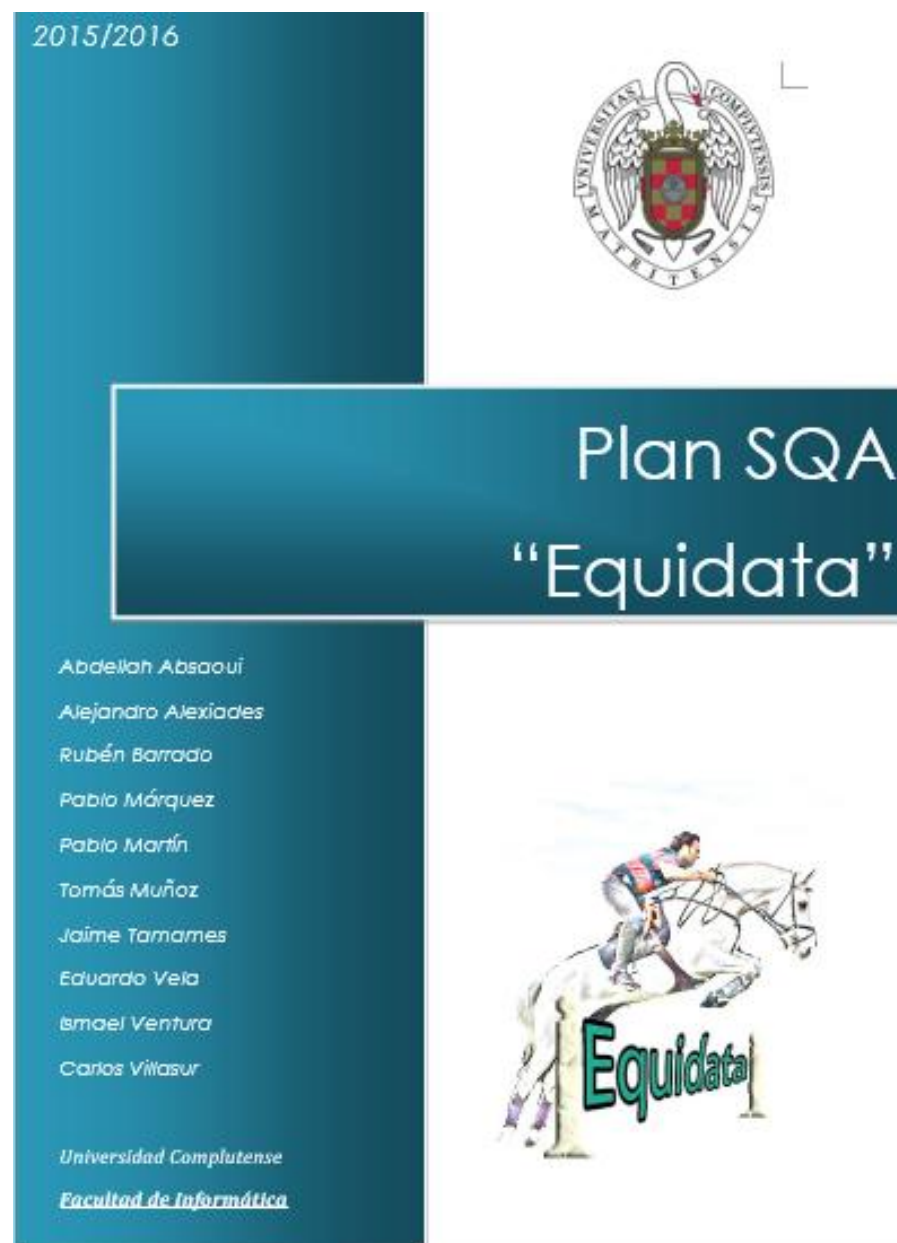


Figura 1: Estilo portada



INDICE	
1- Propósito.....	4
2- Documentos de referencia	6
3- Gestión	7
3.1 Organización.....	7
3.2 Actividades.....	7
3.3 Funciones y responsabilidades	9
3.4 Estimación de recursos.....	10
4- Documentación.....	11
4.1 Propósito.....	11
4.2 Requisitos mínimos de documentación	11
4.3 Otra documentación.....	13
5- Estándares, prácticas, convenciones y métricas.....	14
5.1 Propósito.....	14
5.2 Contenido.....	15
6- Revisiones de software.....	19
6.1 Propósito.....	19
6.2 Requisitos mínimos.....	19
6.3 Otras revisiones o auditorías	21
7- Pruebas	22
8- Informe de errores y acciones correctoras	23
9- Herramientas, técnicas y metodologías	24
10- Control de medios.....	25
10.1 Identificación de medios	25
10.2 Protección de medios	25

Gestión de proyecto software	Página 3
------------------------------	----------

Figura 2: Índice indexado



6- REVISIONES DE SOFTWARE

6.1 PROPÓSITO

El propósito de este punto por parte de nuestro equipo de proyecto es garantizar unos documentos de calidad, bien desarrollados, así como proporcionar una práctica disciplinada de nuestro trabajo para detectar y corregir defectos en el Software que posteriormente deberíamos implementar al finalizar todos los documentos necesarios, Calidad, Gestión de la Configuración, Plan de proyecto, etc. Cuanto antes se detecten los errores antes podrán ser resueltos, porque es más costoso solucionar un problema encontrado a destiempo.

6.2 REQUISITOS MÍNIMOS

Este apartado se centra en la comprobación del documento que contiene los requisitos mínimos, que recoge el conjunto de requerimientos del proyecto. Para esta revisión todos los integrantes del proyecto deberían tener acceso a los entregables al profesor, y haber asistido a las diferentes reuniones para haber obtenido información de estos requisitos. Semanalmente se irá revisando el trabajo terminado para su posterior entrega y corrección. Las siguientes revisiones serán realizadas por todos los miembros del grupo según indicamos en la Agenda dentro de este apartado.

6.2.1 REVISIÓN DE REQUERIMIENTOS

Para realizar esta revisión se llevarán a cabo reuniones periódicas entre los miembros del proyecto para discutir sobre los diferentes puntos y llegar a un acuerdo entre los jefes y el resto de integrantes sobre cuáles de los requerimientos son los correctos. Además, en estas reuniones, tendremos la opción de preguntar al profesor de la asignatura para garantizar la correcta realización de los requerimientos y que están bien definidos.

Figura 3: Formatos de fuentes



- Estándares de diseño: El diseño de la aplicación sigue un estándar creado por el equipo de Equidata. Los detalles del diseño están definidos en el apartado 3.1.1 INTERFACES DE USUARIO del documento *Especificación de documentos software de Equidata*, donde se muestra la funcionalidad de la aplicación.

6- REVISIONES DE SOFTWARE

6.1 PROPÓSITO

Las revisiones del software son un conjunto actividades que suceden como resultado de un análisis para detectar errores y defectos de ingeniería de software con el fin de poderlos corregir. Por ello, la revisión es aconsejable realizarla en una fase temprana del proyecto. Así se detectan errores que hubieran pasado inadvertidos y que generarían nuevos errores si las revisiones no se llevaran a cabo. Además, esto nos permitirá realizar una documentación del proyecto con una calidad mejor y más uniforme.

6.2 REQUISITOS MÍNIMOS

Todos los documentos son revisados semanalmente por el equipo SQA. Estos se basan en los estándares y prácticas definidas. Se controla quién hace las revisiones y se anota en el historial de las mismas. Además, se redacta el documento de entrega semanal al profesor.

6.2.1 REVISIÓN DE REQUERIMIENTOS

La revisión de requerimientos se realiza para asegurar que se cumplió con los requerimientos especificados. Para hacer esta revisión será fundamental



reunirse cada cierto tiempo en grupo para debatir si todos los puntos se realizan de forma correcta y si hay que hacer alguna modificación.

6.2.2 REVISIÓN DE DISEÑO PRELIMINAR

Esta revisión se realiza para asegurar la consistencia y suficiencia técnica del diseño preliminar del software y así evitar futuros fallos de diseño.

6.2.3 REVISIÓN DE DISEÑO CRÍTICO

Esta revisión se llevará a cabo para asegurar la coherencia del diseño detallado en el documento de especificación de requerimientos.

6.2.4 REVISIÓN DEL PLAN DE VERIFICACIÓN Y VALIDACIÓN

La revisión del Plan de Verificación y Validación se realizará para comprobar que se cumple con todos los métodos especificados en dicho plan. Debido a la falta de implementación, no realizaremos esta revisión.

6.2.5 AUDITORÍA FUNCIONAL

La auditoría física, se encarga de comprobar que los requisitos especificados en el documento de requerimientos fueron cumplidos, y que se produjeron previamente a la liberación del software. No la realizaremos debido a la falta de implementación.

6.2.6 AUDITORÍA FÍSICA

La auditoría física, se realiza para verificar que el software y la documentación son consistentes y están preparados para que pueda salir nuestro producto software al mercado. Por falta de implementación no realizaremos la auditoría.



6.2.7 AUDITORÍAS INTERNAS AL PROCESO

Estas, tienen por objeto verificar la consistencia:

- Del código frente al documento de diseño.
- Especificaciones de interfaces e implementaciones de diseño frente a requerimientos funcionales.
- Requerimientos funcionales frente a descripciones de testeo.

No se llevará a cabo debido a la escasa implementación del proyecto.

6.2.8 REVISIONES DE GESTIÓN

Estas revisiones se realizarán de forma periódica para asegurar la ejecución de todas las actividades identificadas en este Plan de Gestión. Se procurará que sea realizada por una persona externa al grupo de trabajo.

6.2.9 REVISIÓN DEL PLAN DE GESTIÓN DE CONFIGURACIÓN

Esta revisión se realiza para asegurar que se cumple la consistencia y completitud de los métodos especificados en el Plan de Gestión de configuración.

6.2.10 REVISIÓN POST MORTEM

Esta revisión se realiza al terminar el proyecto para especificar las actividades que se han desarrollado e implementado durante el proyecto y para proveer recomendaciones; esta labor corresponderá al profesor.

6.2.11 AGENDA

En este apartado se estimarán las semanas en que realizaremos revisiones. En función de esta estimación se realizará un ajuste de recursos humanos, prioridades e iteraciones.



Revisión	Semanas en las que se realizan.
Revisión de Requerimientos	1, 2, 3, 4
Revisión de Diseño Preliminar	4, 5
Revisión de Diseño Crítico	NO
Revisión de Plan de V&V	NO
Auditoria funcional	NO
Auditoria física	NO
Auditorías internas al proceso	NO
Revisión de gestión	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
Revisión del plan de gestión de configuración	10, 11, 12

6.3 OTRAS REVISIONES O AUDITORÍAS

Carecemos de otras revisiones o auditorías debido al tiempo disponible para realizar el proyecto.



7- PRUEBAS

En lo que respecta a este apartado, no podemos rellenarlo puesto que no se va a desarrollar la parte de código para el programa y no podremos realizar pruebas.

En el caso de que si se desarrollara el código, se deberían realizar una serie de controles a lo largo del desarrollo del proyecto y al final del mismo a la hora de lanzarla al mercado. Los controles serían periódicos realizándose dependiendo de los avances del proyecto, aunque se intentaría hacer un control semanal para poder encontrar un numero de errores con los que se pudiera ir haciendo frente.

Los controles se realizarían para comprobar que el código cumpla con los requisitos funcionales establecidos.

Nos apoyaríamos en software de control de errores para realizar el control de estos test e ir indicando que debemos de revisar.

8- INFORME DE ERRORES Y ACCIONES CORRECTORAS

En este apartado, expondremos los casos ante los que nos podemos encontrar a la hora de desarrollar software y en las correcciones correspondientes realizadas por el equipo de calidad:

- En el caso de que el error detectado sea leve, el equipo de SQA podrá tomarse la libertad y asumir la responsabilidad de corregir el error siempre y cuando se sigan los criterios acordados en el grupo de Equidata y respetando los estándares impuestos



- En el caso de que el error detectado sea grave, el producto pasará a fase de producción para seguir las indicaciones para la resolución del error de la manera más rápida y eficiente.

9- HERRAMIENTAS, TÉCNICAS Y METODOLOGÍAS

En este apartado explicaremos las herramientas que hemos utilizado para desarrollar nuestro Software, además de añadir más información sobre ellas en los siguientes documentos:

- Plan de Gestión de la Configuración Software.
- Plan de Gestión de Proyectos Software.

Las herramientas utilizadas son las siguientes:

- **Google Drive:** Herramienta perteneciente a Google que nos sirve para organizar todos los documentos del proyecto, así como su creación y modificación. Además, es la herramienta que usaremos para realizar las entregas al cliente.
- **Microsoft Office:** Herramienta utilizada para la creación de la documentación del proyecto. Utilizaremos Microsoft Word ya que es una herramienta muy potente como editor de texto.
- **TRELLO:** Herramienta web utilizada para organizar el desarrollo del proyecto en cuanto a la organización para elaborar los diferentes documentos o realización de código.
- **IBM RSA:** Herramienta utilizada para la realización de los diferentes diagramas, tanto de clases, de actividad, etc. Asegurándonos en cierto modo de la realización de unos diagramas de calidad.
- **Sublime-Text 2++:** Es un editor de texto que nos permite editar los archivos HTML5 y CSS3.



- **Herramienta de desarrollador Chrome:** Es herramienta integrada en el explorador google Chrome que nos permite ver y hacer cambios en nuestro código HTML al momento. Además, nos indica los errores de nuestro código.
- **C++ builder 2010:** Se emplearán para crear la interfaz de nuestra aplicación.

Estas herramientas se caracterizan porque:

- Son accesibles a todos los miembros del grupo.
- No supone para los integrantes del proyecto un esfuerzo extra para utilizarlas.
- Son fiables e intuitivas para trabajar con ellas.

10- CONTROL DE MEDIOS

Desarrollaremos en esta sección, los medios que llevaremos a cabo para identificar los lugares de almacenamiento de los diferentes documentos durante el desarrollo del proyecto.

10.1 IDENTIFICACIÓN DE MEDIOS

Entregables	Medio	Documentación	Copia
Documentos	-Generados con Microsoft Word -Almacenados en Google Drive	-Google Drive.	-El jefe del equipo se encarga de mantener al día un back up de todos los documentos del



	<ul style="list-style-type: none">-Convertidos a PDF-Entregados mediante Google Drive y aviso mediante correo electrónico al cliente.		repositorio almacenados.
Código de la aplicación	-Implementado en Sublime Text 2++ y ejecutado y testeado con el navegador.	<ul style="list-style-type: none">-Google Chrome.-Sublime Text 2++.	-El jefe del equipo se encarga de mantener al día un back up de todos los documentos del repositorio almacenados.

10.2 PROTECCIÓN DE MEDIOS

Mediante este apartado se describirá cuáles serán las políticas de seguridad que seguiremos a lo largo del proyecto. De esta manera trataremos evitar posibles filtraciones y/o plagios que pueda sufrir nuestro proyecto tanto antes como después de terminarlo.

Al ser un proyecto de corta duración en la que solamente se realizarán la documentación, el almacenamiento y la seguridad del diseño y del código no se verá reflejado por igual la seguridad de los mismos.

Al usar un servicio en la nube como por ejemplo Google Drive, sabemos que esos documentos no nos pertenecen y asumimos el control que tiene la empresa sobre ellos, Al igual que ocurre al usar el correo institucional.



En cualquier caso, tomamos algunas medidas para proteger nuestro proyecto:

- El propietario de la carpeta compartida es el jefe del proyecto. Solo él puede añadir miembros a la carpeta.
- Cualquier persona externa que tenga que tener acceso, por cualquier motivo a nuestro proyecto lo hará tan solo con permiso de lectura, como por ejemplo: el cliente y las entregas.

Al ser un proyecto académico, no se utilizan medidas de cifrado para “encriptar” información ya que este proyecto nunca se implementará y no existe nadie que quiera aprovecharse de nuestro trabajo.

11- CONTROL DE PROVEEDOR

Dispondremos fundamentalmente de tres proveedores, uno que nos proporcionara la memoria suficiente para almacenar nuestra aplicación, otro que nos facilitara base de datos donde se almacenara los datos que se proporcionarán a los usuarios. Y por último un navegador web, que será imprescindible para usar nuestra aplicación.

Para los dos primeros proveedores llevaremos una serie de medidas de control, ya que el navegador es un software suficientemente fiable.

Las medidas que llevaremos a cabo son:

- Aprobación de resultados de ensayos de proveedor:

El proveedor debe entregar un informe de validación con los ensayos realizados de acuerdo al plan de ensayos acordado. Este informe será analizado por equipo de calidad SQA para la aceptación de los prototipos previo a la instauración de la base de datos y el servidor web.



Cualquier desviación con respecto a las especificaciones debe ser discutida y validada por el equipo de calidad SQA.

- Un Informe de pruebas funcionales y calidad del dato:

Las muestras iniciales deben pasar una prueba inicial para confirmar que la calidad del dato que recibe el usuario desde la base de datos. Esta prueba es esencial para garantizar que los usuarios reciban al instante todos los datos que son, además comprobar que la estructura de los documentos CSV están bien definidos. Por ultimo hay que verificar su integración al proceso, la aplicación debe pasar una prueba funcional para garantizar que cumplen de manera satisfactoria la función para la que fue diseñada.

- Un informe de validación del equipo de calidad SQA:

En el caso de que el componente esté bajo requerimientos de validación por parte del equipo de desarrollo, será necesario un informe positivo por parte del equipo de calidad SQA.

En relación a la interfaz software, nuestra aplicación es independiente y autosuficiente para que no tenga que depender de terceras aplicaciones. Esto nos proporcionara un funcionamiento más rápido a la hora de cambiar partes del software para que se adecue a los requisitos de calidad.



12- COLECCIÓN DE REGISTROS, MANTENIMIENTO Y CONSERVACIÓN

La documentación que nos encargaremos de conservar serán los siguiente:

- **Especificación de requisitos (SRS):** Es muy importante ya que tendremos que cumplir los requisitos establecidos. Para ello lo consultaremos varias veces a lo largo del proyecto.
- **Plan de Gestión de Riesgos:** Lo tendremos muy en cuenta para poder solucionar cualquier riesgo que nos surja a lo largo del desarrollo del software, para saber cómo hay que actuar y que riesgos son más prioritarios si surgen varios.
- **Plan de calidad (SQA):** Se debe conservar para poder verificar en todo momento que se están realizando las actividades de calidad reflejadas en el plan.
- **Plan de Gestión de la Configuración:** Lo utilizaremos para minimizar las confusiones a la hora de desarrollar el software, ya que controla todas las modificaciones.
- **Plan de Proyecto:** Se utilizará como guía para desarrollar el software ya que contiene toda la información relativa a la gestión del proyecto.
- **Revisiones Técnicas Formales (RTF):** Se conservará para poder aplicar todos los cambios sugeridos en las RTFs.
- **Actas Semanales:** Se tendrán muy en cuenta para saber en todo momento que es lo último acordado, además de saber quiénes estuvieron presentes por si surge alguna duda con el documento.

El método principal que usaremos para archivar, salvaguardar y mantener la documentación será el repositorio de Google Drive, ya que es la forma más fácil para que cada uno de los miembros del grupo tenga su propia copia en su



ordenador. Además, nos permite modificar, consultar y guardar un documento de forma simultánea.

Para llevar un control sobre la documentación que vamos realizando a lo largo del proyecto usaremos la herramienta Trello, donde también se especificarán las tareas de revisión y back-up de cada documento, así nos aseguraremos que todos los documentos estén en su última versión y cumplan los requisitos establecidos.

El periodo de conservación de los documentos será mínimo de un año, pero se conservará en el repositorio de google drive el tiempo que Google te permita almacenarlos, por si alguien desea utilizarlo para crear una aplicación similar.

13- FORMACIÓN

Desde el comienzo de esta asignatura hemos ido adquiriendo una serie de habilidades y conocimiento sobre la gestión de proyectos software y metodologías de desarrollo que nos van a permitir elaborar un plan de calidad aceptable.

13.1 FUNCIONES

A continuación, se expone las funciones a través de los distintos puntos que deberán cumplir los integrantes de nuestro grupo, será en todo caso riguroso ya que son fundamentales para conseguir unos documentos que cumplan con la calidad exigida:

- **Garantía de calidad:** Realizar una serie de procedimientos organizativos que nos generen un software de calidad como la realización de RTFs.



- **Análisis del coste de la calidad:** Esto incluye la planificación de la calidad, la inspección y pruebas durante los procesos y los costes derivados de los fallos.
- **Auditoría:** Revisiones para verificar que la documentación cumpla los estándares y procedimientos establecidos. Estas auditorías podrán ser realizadas por personas externas al proyecto.
- **Control de calidad:** Serie de inspecciones, revisiones y pruebas realizadas en torno al desarrollo del software que nos aseguran que nuestro producto cumple los requisitos establecidos.

14- GESTIÓN DEL RIESGO

La realización de la gestión de riesgos ha sido llevada a cabo por tres fases y de esta manera garantizamos el éxito de nuestro proyecto:

- **Fase de reducción**
- **Fase de Supervisión**
- **Fase de gestión del riesgo**

Y en relación con la priorización de riesgos nos hemos basado en los niveles de riesgos SQAS-SEI. Donde hemos organizado cada riesgo por su orden de prioridad en función de nivel de gravedad, probabilidad y consecuencia.

Todos los riesgos estarán detallados en el documento *Plan de gestión del Riesgo*.



15- GLOSARIO

- **Aseguramiento de la calidad del Software (SQA, Software Quality Assurance):** Conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza para que el software satisfaga los requisitos dados de calidad.
- **Auditoría:** Examen crítico y sistemático que realiza una persona o grupo de personas independientes del sistema auditado, que puede ser una persona, organización, sistema, proceso, proyecto o producto, con el objeto de emitir una opinión independiente y competente.
- **Gestión de la configuración del software (SCM, Software Configuración Management):** Proceso de identificar y definir los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.
- **Revisión:** Forma de revisión de software en la cual un diseñador o programador lidera a los miembros de un equipo de desarrollo y otra de las partes involucradas a través de un producto de software, y los participantes hacen preguntas y comentarios acerca de posibles errores, violación de estándares de desarrollo, y otros problemas.
- **MBTI:** Test de personalidad diseñado y creado por Katharine Cook Briggs y su hija Isabel Briggs Myers para ayudar a una persona a identificar algunas de sus preferencias personales más importantes.
- **KIVIAT-** Gráfico de radar, en estrella que se utiliza para representa característica cuantificada. Esto facilita un análisis detallado de varios objetos simultáneamente.
- **Interfaces de usuario-** Medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.



- **HTML5 (HyperTextMarkupLanguage, versión 5):** Lenguaje de marcado para la elaboración de páginas web.
- **CSS3 (Cascading Style Sheets):** Hoja de estilo en cascada o CSS es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML2 .
- **Back-up:** En informática es una copia de los datos originales que se realiza con el fin de disponer de un medio para recuperarlos en caso de su pérdida
- **SQAS-SEI (Software Quality Assurance Subcommittee - Software Engineering Institute):** Mecanismo descrito en la tabla SQAS_SEI para calcular el nivel de riesgo de un proyecto.
- **Prueba (Testing):** Una investigación técnica de un producto bajo prueba con el fin de brindar información relativa a la calidad del software, a los diferentes actores involucrados en un proyecto.

16- PROCEDIMIENTO DE CAMBIO E HISTORIAL DEL PLAN SQA

Seguiremos un determinado procedimiento para realizar cambios en el plan SQA. Para ello, lo primero será informar a nuestro jefe de proyecto (Tomas Muñoz Testón) para que evalúe los cambios propuestos, y luego a través de Trello asigne los cambios a algún miembro del equipo. Una vez realizado el cambio, se reflejará en la tabla que está a continuación, que guardará el historial de cambios. Este procedimiento será fundamental para llevar un control de todos los cambios y modificaciones que se vayan realizando, así se facilitaran las revisiones del documento en el futuro. Estas revisiones se harán de forma periódica antes de cada entrega o modificación de algún documento relacionado con el plan SQA.



Fecha del cambio	Identificador del cambio	Causa del cambio	Personas encargadas
12/05/2016	SQA-v1.0	Revisión y creación de diversos puntos del plan SQA	Alejandro Alexiades Abdellah Absaoui Pablo Márquez Pablo Martin

17- APÉNDICES

17.1 REVISIONES TÉCNICAS FORMALES-FIND&FUN

Los miembros de Equidata encargados de recibir y decidir qué hacer con los cambios de dicha revisión han sido

- Carlos Villasur
- Jaime Tamames
- Pablo Márquez
- Rubén Barrado
- Los miembros de Find&Fun encargados de dicha revisión han sido:
 - Lucas Rodríguez Fernández
 - José Luis Sánchez García



17.1.1 REVISIONES TÉCNICAS FORMALES-PGR

- Estilo:
 - Formato de la portada.
 - SE ACEPTA EL CAMBIO.
 - En el índice riesgo 20 falta ponerlo en mayúsculas.
 - SE ACEPTA EL CAMBIO.
- Fallos formato:
 - Control de versiones nulo.
 - NO SE ACEPTA EL CAMBIO: Solamente versión con cambios drásticos.
 - No se entiende o poco clara la priorización de riesgos la probabilidad.
 - La tabla de riesgos no cuadra ni en colores ni en valores.
 - SE ACEPTA EL CAMBIO.
 - Poner un mismo idiomas en la tabla de riesgos (inglés o castellano).
 - SE ACEPTA EL CAMBIO.
 - La forma de mostrar los riesgos es poco útil.
 - SE ACEPTA EL CAMBIO.

17.1.2 REVISIONES TÉCNICAS FORMALES-SRS

- Historial de versiones:
 - NO SE ACEPTA EL CAMBIO: Solamente versión con cambios drásticos.
- Introducción mejorable:
 - SE ACEPTA EL CAMBIO.
- Usuarios (no están claros):
 - Punto 1.2: empresas organizadoras concursos hípicas (gestionar/administrar) y participantes (consultas públicas).
 - SE ACEPTA EL CAMBIO.
 - Punto 2.1: aparece el término jurado como usuario.
 - SE ACEPTA EL CAMBIO.
 - Punto 3.1: aparece el término invitado como usuario (consultas públicas)



y dos roles (administrador y juez).

- SE HA ACEPTA EL CAMBIO.
- En los casos de usos: que representa el término operador del sistema.
 - SE ACEPTA EL CAMBIO.
- Casos de uso (mucho copia y pega sin modificar):
 - Modificar administrador: flujo de evento punto 2.
 - SE ACEPTA EL CAMBIO.
 - Alta concurso: descripción.
 - SE ACEPTA EL CAMBIO.
 - Modificar concurso: pre y post condición y acción. Flujo evento 2 y 4.
 - SE ACEPTA EL CAMBIO.
 - Listar concursos: flujo evento 3 a).
 - SE ACEPTA EL CAMBIO.
 - Listar participantes prueba: flujo de evento: falta la comprobación de los datos de entrada.
 - SE ACEPTA EL CAMBIO.
 - Generar orden salida es igual a generar clasificación de prueba.
 - SE ACEPTA EL CAMBIO.
 - Listar premios jinete: entrada seria el ID_Jinete.
 - SE ACEPTA EL CAMBIO.
 - En los altas, la entrada seria los atributos de cada entidad
 - SE ACEPTA EL CAMBIO.
- Diagrama Entidad-Relación
 - La prueba tiene el premio pero pero el juez evaluaría la participación de cada jinete otorgando el atributo puntos.
 - SE ACEPTA EL CAMBIO.



17.2 REVISIONES TÉCNICAS FORMALES-LINKER

Los miembros de Equidata encargados de recibir y decidir qué hacer con los cambios de dicha revisión han sido:

- Abdellah Absaoui
- Pablo Martin
- Tomas Muñoz
- Eduardo Vela
- Ismael Ventura

Los miembros de Linker encargados de dicha revisión han sido:

- David de los Santos
- Pedro Sánchez
- Eloy González

17.2.1 REVISIONES TÉCNICAS FORMALES-PGR

- Riesgos sin clasificar.
 - SE ACEPTA EL CAMBIO.

17.2.2 REVISIONES TÉCNICAS FORMALES-SRS

- Alcance
 - Separar “estadístico más” y “más visual”
 - SE ACEPTA EL CAMBIO.
- Definiciones, acrónimos y abreviaturas
 - No veo necesaria la definición de “Jinete” en este apartado ya que es una palabra de uso común.
 - SE ACEPTA EL CAMBIO.
 - Las palabras “Binomio” y “ODBC” no aparece en toda la SRS por lo que no es necesaria su definición.
 - NO SE ACEPTA EL CAMBIO.



- Separar “Lahoja” (en la definición de CSS3).
 - SE ACEPTA EL CAMBIO.
- Resumen:
 - Corregir errata en “secciónse”.
 - SE HA ACEPTA EL CAMBIO.
- Descripción general:
 - Un participantes puede cambiar la prueba para el día siguiente
 - SE ACEPTA EL CAMBIO.
 - Icono de familia(cambiar por icono de jinete).
 - SE ACEPTA EL CAMBIO.
 - ¿No se especifica el orden de salida de los binomios?
 - SE ACEPTA EL CAMBIO.
 - ¿Donde se inscriben los jinetes?
 - SE ACEPTA EL CAMBIO.
- Perspectiva del producto: Interfaces del usuario
 - Deberían describir aquí las distintas ventanas que se pueden abrir dependiendo del usuario que sea el que usa la aplicación funcionalidades etc..
 - NO SE ACEPTA EL CAMBIO.
 - No dependería de ninguna funcionalidad del sistema operativo, como la hora? o del propio sistema operativo?.
 - NO SE ACEPTA EL CAMBIO.
- Perspectiva del producto: Operaciones
 - Completar:
 - b) Periods of interactive operations and periods of unattended operations;
 - c) Data processing support functions;
 - d) Backup and recovery operations.?
 - NO SE ACEPTA EL CAMBIO.
- Perspectiva del producto: Requisitos de adaptación
 - No debería aparecer esto aquí, debería aparecer en interfaces software.



- NO SE ACEPTA EL CAMBIO.
- Funciones del producto
 - Falta de organización en las funciones de los gestores, Podrían organizar las funciones por gestores más específicos, para que quedara más claro.
 - SE ACEPTA EL CAMBIO.
 - Faltan funciones de mostrar, cosas que un usuario puede querer consultar, Añadirlas en admin, prueba, resultados prueba...
 - NO SE ACEPTA EL CAMBIO.
- Requisitos funcionales
 - No existen alta jinete, ni alta caballo
 - NO SE ACEPTA EL CAMBIO.
 - Podrían ponerse los atributos de cada una de las entidades de los módulos al comienzo para facilitar la comprensión de la posterior referencia a dichos atributos en cada una de las funciones.
 - NO SE ACEPTA EL CAMBIO.
- Modulo administrador; Modificar administrador
 - Errata en la precondition.
 - SE ACEPTA EL CAMBIO.
 - Errata en la postcondición.
 - SE ACEPTA EL CAMBIO.
 - Errata en la acción y flujo de eventos.
 - NO SE ACEPTA EL CAMBIO.
 - ¿Qué datos se consideran correctos?
 - NO SE ACEPTA EL CAMBIO.
- Modulo administrador; Listar administrador
 - Debería especificarse un bucle en el diagrama de actividades
 - SE ACEPTA EL CAMBIO.
- Modulo administrador; Listar administrador
 - Debería especificarse un bucle en el diagrama de actividades
 - SE ACEPTA EL CAMBIO.



- Módulo administrador; Asignar administrador a licencia
 - Queda un poco confuso si debe existir el administrador antes de asignarle la licencia o se da de alta en el proceso, revisar
 - NO SE ACEPTA EL CAMBIO.
- Módulo concurso: Modificar concurso
 - Punto 1.2: empresas organizadoras concursos hípicas (gestionar/administrar) y participantes (consultas públicas).
 - SE ACEPTA EL CAMBIO.
- Usuarios (no están claros):
 - Revisar flujo de evento
 - NO SE ACEPTA EL CAMBIO.
 - Revisar post condición
 - NO SE ACEPTA EL CAMBIO.
- Módulo concurso: Borrar concurso
 - ¿Qué pasa si hay jinetes en el concurso?
 - SE ACEPTA EL CAMBIO.
- Módulo concurso: Listar concurso
 - En el flujo de evento pone: “Si no existe ningún administrador dado de alta en el sistema, se mostrará un mensaje indicando que no hay ningún concurso creado.”
 - SE ACEPTA EL CAMBIO.
 - En el diagrama de actividades representar un bucle que recorra los concursos.
 - NO SE ACEPTA EL CAMBIO.
- Módulo participantes: Listar participantes
 - En el flujo de evento hace referencia a listar participantes.
 - SE ACEPTA EL CAMBIO.
 - Representar en el diagrama un bucle que recorra los participantes.
 - NO SE ACEPTA EL CAMBIO.
- Módulo participantes: Puntuar participantes
 - Se tiene que comprobar si no se ha hecho antes una puntuación, Si se



vuelve a puntuar se modifica la puntuación anterior.

- SE ACEPTA EL CAMBIO.
- Módulo participantes: Inscribir participantes
 - No se comprueba que el participante ya haya sido asignado a la prueba.
 - NO SE ACEPTA EL CAMBIO.
- Módulo participantes: Generar clasificación prueba
 - Hay que revisarlo porque la tabla es la de Generar orden de salida de la prueba.
 - SE ACEPTA EL CAMBIO.
- Módulo juez: Alta juez
 - Falta entrada.
 - SE ACEPTA EL CAMBIO.
- Módulo juez: Borrar juez
 - No se comprueba que el juez ya esté dado de baja lógica.
 - NO SE ACEPTA EL CAMBIO.
- Módulo juez: Listar juez
 - Precondición: Harían falta más datos de precondición, post condición etc...
 - NO SE ACEPTA EL CAMBIO.
 - En el diagrama de actividad se lista al principio., Comprobar que hay datos que listar.
 - NO SE ACEPTA EL CAMBIO.
- Módulo juez: Buscar juez
 - En el diagrama de actividades la recuperación de datos se lleva a cabo después de comprobar si existen los datos, no antes.
 - NO SE ACEPTA EL CAMBIO.
 - Debería llamarse a mostrar juez.
 - NO SE ACEPTA EL CAMBIO.
- Módulo licencias: Alta licencia
 - Faltan datos de entrada.
 - SE ACEPTA EL CAMBIO.



- Módulo licencias: Modificar licencia
 - Faltan datos de entrada.
 - NO SE ACEPTA EL CAMBIO.
 - Podrían de existir dos pruebas con los mismos valores en los atributos exceptuando el id.
 - NO SE ACEPTA EL CAMBIO.
 - Diagrama de actividades corregir concepto anterior.
 - NO SE ACEPTA EL CAMBIO.
- Módulo licencias: Borrar licencia
 - ¿Prioridad ALTA. Borrado físico?
 - SE ACEPTA EL CAMBIO.
- Módulo licencias: Listar licencia
 - Corregir sintaxis del diagrama de actividad.
 - SE ACEPTA EL CAMBIO.
- Módulo licencias: Buscar licencia
 - Debería llamarse Mostrar Licencia.
 - NO SE ACEPTA EL CAMBIO.
 - La recuperación de datos se hace después de comprobar el ID, y solo en caso de que exista.
 - NO SE ACEPTA EL CAMBIO.
- Módulo premio: Alta premio
 - Faltan datos de entrada.
 - SE ACEPTA EL CAMBIO.
- Módulo premio: Borrar premio
 - No se comprueba si ya esta dado de baja lógica.
 - NO SE ACEPTA EL CAMBIO.
- Módulo premio: Buscar premio
 - Se comprueba el ID antes de recuperar los datos.
 - NO SE ACEPTA EL CAMBIO.
- Módulo premio: Listar premio
 - ¿Qué es listar? Hay que coger los datos antes de listarlos.



- NO SE ACEPTA EL CAMBIO.
- Módulo premio: Listar premios por jinete
 - Precondición: existe jinete.
 - NO SE ACEPTA EL CAMBIO.
 - Faltan datos de entrada.
 - SE ACEPTA EL CAMBIO.
- Módulo prueba: Alta prueba
 - Faltan datos de entrada.
 - SE ACEPTA EL CAMBIO.
- Módulo prueba: Borrar prueba
 - No se da el caso en el diagrama de actividades en el que la prueba ya esté dada de baja lógica.
 - NO SE ACEPTA EL CAMBIO.
- Módulo prueba: Lista prueba
 - Precondición: que haya alguna prueba en el sistema.
 - SE ACEPTA EL CAMBIO.
- Módulo prueba: Buscar prueba
 - Comprobar ID antes de recuperar los datos.
 - NO SE ACEPTA EL CAMBIO.
- Módulo prueba: Asignar premio a una prueba
 - En el diagrama de actividad se debe reflejar cómo se asigna el premio a la prueba (mediante una tabla intermedia en la que tengas el ID de la prueba y el ID del premio por ejemplo).
 - NO SE ACEPTA EL CAMBIO.
- Módulo prueba: Listar resultados de prueba
 - Menciona que la prueba debe estar finalizada, pero no se menciona como se comprueba ni si se hace, además ¿debería haber una función de finalizar prueba?
 - NO SE ACEPTA EL CAMBIO.
- Módulo Usuario
 - Errata en el título.



- **SE ACEPTA EL CAMBIO.**
- Módulo Usuario: Buscar estadísticas del jinete
 - Primero habría que comprobar el ID y luego recuperar los datos, en el diagrama de actividad viene al revés.
 - **NO SE ACEPTA EL CAMBIO.**
 - El título del diagrama de actividad es “Buscar estadísticas del caballo” cuando debería ser “Buscar estadísticas del jinete”
 - **SE ACEPTA EL CAMBIO.**
- Módulo Usuario: Buscar resultados de prueba
 - Falta diagrama de actividad.
 - **SE ACEPTA EL CAMBIO.**
- Requisitos de rendimiento
 - Estaría bien contemplar la capacidad de usuarios en línea que podrá soportar la aplicación.
 - **NO SE ACEPTA EL CAMBIO.**
- Requisitos lógicos y de la base de datos
 - No estaría mal, además del diagrama E/R que se definan las tablas surgidas de las relaciones entre las entidades.
 - **NO SE ACEPTA EL CAMBIO.**
- Apéndice
 - Errata en “Descripción de problemas que solucionará el software”.
 - **SE ACEPTA EL CAMBIO.**