

ComAppÑeros

Plan de Garantía de Calidad Software

IEEE Std. 730-2002

Sebastián Águila

Diego del Corral

Miguel Franco

Álvaro Navas

Mario Michiels

Rodrigo de Miguel

Sergio Pino

Cristian Pinto

Emanuel Ramírez

Fernando Rivilla

Cristhian Rodríguez



ComAppÑeros

Universidad Complutense de Madrid

Gestión de Proyectos Software y Metodologías de Desarrollo

Historial de versiones

Versión	Fecha
SQA-ComAppÑeros-v0.0.1	23/04/2016
SQA-ComAppÑeros-v0.0.2	25/05/2016

Tabla 0: Control de versiones

Índice

1	Propósito	5
2	Documentos de referencia	6
3	Gestión	7
3.1	Organización	7
3.2	Actividades de calidad a realizarse	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
4.2.1	Descripción de requisitos software	10
4.2.2	Descripción de diseño software	10
4.2.3	Plan de Verificación y Validación	10
4.2.4	Informe de resultados de Verificación y Validación	11
4.2.5	Documentación de usuario	11
4.2.6	Plan de Gestión de Configuración Software	11
4.3	Otra documentación	11
5	Estándares, prácticas, convenciones y métricas	12
5.0.1	Propósito	12
5.0.2	Contenido	12
6	Revisiones del software	14
6.1	Propósito	14
6.2	Requisitos mínimos	14
6.2.1	Revisión de requerimientos	14
6.2.2	Revisión de diseño preliminar	14
6.2.3	Revisión de diseño critico	15
6.2.4	Revisión del Plan de de Verificación y Validación	15
6.2.5	Auditoría funcional	15
6.2.6	Auditoría física	15
6.2.7	Auditoría internas al proceso	15
6.2.8	Revisiones de gestión	16
6.2.9	Revisiones del Plan de Gestión de Configuración	16
6.2.10	Revisión Post Mortem	16
6.2.11	Agenda	16
6.3	Otras revisiones y auditorías	16
7	Pruebas	17
8	Informe de errores y acciones correctoras	18
8.1	Procedimientos a seguir para informar, realizar seguimiento, y resolver los problemas identificados	18
8.2	Organización específica para resolver los problemas identificados	18

9 Herramientas, técnicas y metodologías	19
9.1 Almacén de documentación del proyecto	19
9.2 Comunicación y coordinación entre los miembros del proyecto (y el profesor)	19
9.3 Desarrollo de documentación	20
10 Control de medios	22
10.1 Identificación de medios	22
10.2 Protección de medios	23
11 Control de proveedor	25
12 Colección de registros, mantenimiento y conservación	26
13 Formación	27
13.1 Funciones	27
14 Gestión del riesgo	28
15 Glosario	29
16 Procedimiento de cambio e historias del plan de SQA	30
17 Apéndices	31
17.1 Revisiones Técnicas Formales	31

1. Propósito

El Plan SQA o Plan de control de calidad es un conjunto de actividades sistemáticas y planeadas para asegurar que los procesos y productos de software cumplen con los requerimientos, estándares y procedimientos. 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 requerimientos que se le imponen, además de ser mantenible (sostenible), confiable y estable, el logro de estos factores permite decir que se trata de un software de calidad.

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 de la aplicación **ComAppÑeros**.

Este plan abarca las partes del ciclo de vida relacionadas con: 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 grupo de proyecto tenemos los siguientes objetivos:

- Definir los requerimientos(requisitos) de calidad a ser verificados.
- Indicar las partes del ciclo 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

Nuestra referencia son:

- Standard de IEEE Std. 730-2002, de 2002 de la *IEEE Computer Society*.
- Apuntes y material audiovisual de la asignatura Gestión de Proyectos Software y Metodologías de Desarrollo (GPS) y sus predecesoras, Ingeniería del Software (IS) y Modelado de Software (MS).
- Proyectos de años anteriores de la asignatura de Gestión de Proyectos Software.

3. Gestión

La gestión de calidad es un trabajo que recae sobre los propios miembros del equipo que se centrará en asegurar la correcta implementación del proceso establecido y que los productos derivados del proceso se ajusten a los criterios de calidad. Se controlará que las actividades realizadas se adapten al plan de calidad.

3.1. Organización

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

El equipo de trabajo está estructurado siguiendo una MANTEI Descentralizada Controlada («**apartados 4.2 y 4.3 del Plan de Proyecto**») y las tareas para el plan SQA se reparten de manera dinámica entre los grupos, de forma que no poseemos un grupo fijo ni una persona que se dedique en exclusiva al plan.

Aunque cabría recordar 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 llamar la atención y dar ideas para mantener el plan SQA.

3.2. Actividades de calidad a realizarse

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

- RTF(Revisión Técnica Formal)
- Revisiones por parte de los subjeses y jefe del proyecto

Una revisión técnica formal es un conjunto de procedimientos cuyo objetivo es:

- Verificar la firmeza de la documentación y su relación con los requerimientos.
- Verificar que se cumple todo lo exigido por el cliente.
- Lectura fácil y comprensible de la documentación aportada.
- Proposición de mejoras, añadidos, patrones, etc...

Se han realizado ocho revisiones técnicas formales. Cuatro de las revisiones técnicas formales han sido realizadas a nuestros documentos por parte de dos grupos de clase y las cuatro restantes las hemos realizado a otros dos grupos de clase, por cada grupo se han hecho dos RTF's: una para el documento de especificación de requisitos y otra para el documento de gestión de los riesgos. Dichas RTF's están adjuntas al apéndice de este plan.

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

Con este procedimiento hemos podido detectar algunos errores en nuestro proyecto y así poder solucionarlos, también hemos realizado revisiones sobre otros grupos y sus proyectos que pueden ayudarnos a solucionar futuros errores nuestros.

Las revisiones por parte de los subjeses y jefe del proyecto tiene un carácter más informal, con ellas se busca que los documentos sean lo más completos y precisos posibles, estas se llevan a cabo semanalmente, dado que cada siete días tenemos que entregar algún tipo de documentación.

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

- **Aplicación de métodos técnicos** : ayudarán a conseguir especificaciones y diseños de alta calidad.
- **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 Subjeses y el Jefe. Dichas reuniones se realizan los Martes en el laboratorio de la facultad de informática de la Universidad Complutense de Madrid.
- **Consultar expertos** : solicitaremos el asesoramiento.

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 :
 - **Linker**
 - **Equidata**
- Los revisados fueron:
 - **Linker**

- **Find&Fun**

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

- **Subjeses :**

- Subgrupo CSM: Mario Michiels
- Subgrupo FND: Fernando Rivilla
- Subgrupo MECS: Miguel Franco

- **Jefe :**

- Rodrigo de Miguel

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

- Aplicación de métodos técnicos : Todo el equipo de proyecto.
- Reuniones semanales : Todo el equipo de proyecto.
- Consulta Experto : Será el profesor de la asignatura, Gonzalo Méndez y Susana Bautista quienes nos resolverá dudas y problemas.

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 al resto y será solucionado.

3.4. Estimación de recursos

Para poder realizar las actividades de calidad citadas anteriormente se deben destinar recursos humanos para garantizar que la actividad de calidad se realiza de manera eficiente y eficaz. Además las personas destinadas a este fin deben poder acceder a Google Drive ya que toda nuestro proyecto se encuentra allí.

Recursos necesarios:

- **Recursos humanos:** los miembros del proyecto **ComAppÑeros**.
- **Recursos materiales:** ordenadores con acceso a internet.

4. Documentación

4.1. Propósito

El propósito es definir toda la documentación que se va a ir generando durante todo el proceso de desarrollo de la aplicación. Se listan los documentos a revisar o que necesitan ser auditados y los criterios de revisión. Así como los objetivos de cada uno de los documentos.

4.2. Requisitos mínimos de documentación

A continuación mostramos una lista de documentos con la cual nos aseguramos de toda la gestión de calidad del proyecto junto con la implementación del software:

- Plan de Gestión de Configuración
- Plan de Proyecto
- Plan de Gestión de Riesgos
- Plan de Calidad (SQA)
- Especificación de Requisitos
- Revisiones Técnicas Formales (RTF's)
- Documentación generada a causa de las RTF's.

4.2.1. Descripción de requisitos software

Todos los requisitos de la aplicación están descritos en la «Especificación de Requisitos Software» realizada a partir del estándar IEEE 8301998.

4.2.2. Descripción de diseño software

Necesario consultar al profesor

4.2.3. Plan de Verificación y Validación

Toda la información está recogida en la Gestión de Configuración. Este proyecto no tiene un documento únicamente dedicado a la verificación y validación ya que no se realizarán dichas pruebas

debido a que no se desarrollará el código de la aplicación.

4.2.4. Informe de resultados de Verificación y Validación

No se dispone de informes de resultado. Como se ha indicado en el apartado anterior no habrá código, y por tanto, no se realizarán pruebas de verificación y validación.

4.2.5. Documentación de usuario

Resulta muy útil para que el usuario tenga mayor información acerca de la aplicación (usuarios que usan por primera vez la aplicación, usuarios que la conocen y desean ampliar sus conocimientos sobre la aplicación o usuarios que únicamente buscan si dicha aplicación es lo que buscan o no).

ComAppñeros no dispondrá de documentación de usuario ya que la asignatura no lo requiere.

4.2.6. Plan de Gestión de Configuración Software

Toda la información de este plan está descrita en el documento «Plan de Gestión de Configuración».

4.3. Otra documentación

El «Informe de personalidad MBTI y KIVIAT» es otro documento adicional perteneciente al proyecto.

5. Estándares, prácticas, convenciones y métricas

5.0.1. Propósito

En esta sección se identifican las normas, prácticas, convenciones, técnicas estadísticas, y métricas que se aplicarán. Las medidas de producto y de proceso se incluirán en las métricas empleadas.

ComAppñeros hace uso de:

- Estándares:
 - IEEE 830-1998, Especificación Requisitos Software.
 - IEEE 1058-1998, Plan de Proyecto.
 - 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 del proyecto ComAppñeros, explicada en el apartado 5.2.
- Métricas:
- Requisitos de calidad: no se aplica.

5.0.2. Contenido

En esta sección se detallan los aspectos de diseño para la documentación del proyecto.

- **Estándares de documentación**
 - La documentación del proyecto se ha escrito utilizando el sistema de composición de textos LaTeX.
 - El tamaño de fuente general es de 11pt.
 - Los documentos son de tipo “article”.
 - Los índices incluyen enlaces para facilitar la navegación al lector.
- **Estándares de diseño**
 - La tipografía por defecto es: Roboto.
 - Tamaño fuente títulos: 20px.
 - Tamaño fuente títulos pequeños: 18px.
 - Tamaño fuente párrafos: 15px.

- El color por defecto es: #000000.

■ Estándares de codificación

- Al tratarse de una aplicación web, son necesarios varios lenguajes para desarrollarla.
 - Lógica de negocio: Java 8.
 - Maquetación web: HTML 5.
 - Lógica de negocio GUI: JavaScript
 - Hoja de estilos GUI: CSS
- También se hace uso de distintos frameworks que ayudan a la generación de código seguro y eficiente, ahorrando horas de trabajo al equipo, lo que se traduce en efectividad y rendimiento.
 - Para la GUI: Twitter Bootstrap, Modernizer y JQuery.
 - Para la persistencia de los datos: JPA.
- Todos los ficheros de código estarán adecuadamente documentados: HTML, CSS, JavaScript y JAVA. Se escribirá cada fichero de código teniendo en cuenta en todo momento las convenciones de Google para dicho código:
 - HTML: <https://google.github.io/styleguide/htmlcssguide.xml>
 - CSS: <https://google.github.io/styleguide/htmlcssguide.xml>
 - JAVA: <https://google.github.io/styleguide/javaguide.html>
 - JS: <https://google.github.io/styleguide/javascriptguide.xml>

6. Revisiones del software

6.1. Propósito

El propósito para nuestro equipo de proyecto es garantizar que los documentos de calidad que estén 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 y Especificación de Requisitos.

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.

6.2.2. Revisión de diseño preliminar

Este apartado no será efectuado en nuestro proyecto debido a la falta de implementación y tiempo para llevarlo a cabo.

6.2.3. Revisión de diseño critico

Esta revisión se llevará a cabo para asegurar la consistencia del diseño detallado con la SRS (Especificación de Requisitos Software).

Este apartado no se realizará en nuestro proyecto, dado que el tiempo marcado por la asignatura no nos lo permite.

6.2.4. Revisión del Plan de de Verificación y Validación

Esta revisión será efectuada para garantizar la consistencia y completud de los métodos que se expresan en el Plan de Configuración Software, ya que para este proyecto no se hará el Plan de Verificación y Validación.

Este apartado no será efectuado en nuestro proyecto debido a la falta de implementación.

6.2.5. Auditoria funcional

Esta auditoría se efectúa antes de la puesta en el mercado del software, para garantizar que todos los requisitos especificados en el documento de requisitos fueron cumplidos.

Este apartado no será efectuado en nuestro proyecto debido a la falta de implementación.

6.2.6. Auditoría física

Esta revisión se efectúa para garantizar que el software y el plan de proyecto son consistentes y están aptos para salir al mercado.

Este apartado no será efectuado en nuestro proyecto debido a la falta de implementación.

6.2.7. Auditoría internas al proceso

Estas auditorías son para garantizar la consistencia del código con el documento de diseño, especificaciones e implementaciones de diseño con los requerimientos funcionales y los requerimientos funcionales con las descripciones de testeo.

Este apartado no será efectuado en nuestro proyecto debido a la falta de implementación.

6.2.8. Revisiones de gestión

Estas revisiones se realizan periódicamente para asegurar la ejecución de todas las actividades identificadas en este Plan. Deben realizarse por una persona ajena al grupo de trabajo (en caso de que sea posible).

Esta revisión es realizada por los subjefes de grupo primero y después por el jefe de proyecto cada cierto tiempo.

6.2.9. Revisiones del Plan de Gestión de Configuración

Esta revisión también es realizada por los subjefes de grupo así como el jefe de proyecto.

6.2.10. Revisión Post Mortem

Este apartado del plan de calidad será llevado a cabo por el profesor de la asignatura de Gestión de Proyectos Software y Metodologías de Desarrollo quien asignará una nota al final del proyecto.

6.2.11. Agenda**6.3. Otras revisiones y auditorías**

No tenemos otras revisiones o auditorías debido a la naturaleza de nuestro proyecto.

7. Pruebas

En el caso de que se desarrolle el código, se realizará un control rutinario y evolutivo durante el desarrollo.

Previamente a la implementación del software se generarán todas las pruebas. Posteriormente, cada vez que implementemos una funcionalidad nueva, nos ayudaremos de los test asociados a esa funcionalidad, de esta forma en todo momento podremos garantizar la calidad del software, el grado en que se cumple el diseño y, sobre todo, los requisitos.

Los tipos de pruebas que realizaremos serán las siguientes:

- Estáticas: revisión de documentos y buenas prácticas de compatibilidad.
- Dinámicas: pruebas en caja negra y caja blanca.
- Pruebas de ejecución: usando para ello pruebas de regresión.
- Pruebas automatizadas: que nos garanticen que las pruebas realizadas anteriormente siguen siendo exitosas

Los encargados de llevar a cabo las pruebas serán además de los integrantes del equipo de desarrollo, personas ajenas al desarrollo implicadas en el proyecto.

8. Informe de errores y acciones correctoras

En el punto de informe de errores y acciones correctoras del plan SQA se elabora un resumen de los problemas que nos podemos encontrar en el desarrollo de software y de sus posibles correcciones.

8.1. Procedimientos a seguir para informar, realizar seguimiento, y resolver los problemas identificados

Estos problemas pueden surgir de las revisiones realizadas semanalmente por el jefe de proyecto y los subjeses, además, se asignará una persona o grupo para la realización de esta revisión.

La gravedad de los errores detectados será analizada y categorizada por el subjefe de grupo que corresponda, asumiendo así todas las responsabilidades relacionadas con dicha decisión.

Para certificar la correcta categorización de un error concreto, como grave o menor, será necesario el pronunciamiento al respecto del jefe superior (En nuestro caso Rodrigo de Miguel).

8.2. Organización específica para resolver los problemas identificados

- Si el error de software es menor y fácil de solucionar, el personal encargado de SQA (asignado por el jefe de equipo), se tomará la libertad y asumirá la responsabilidad de modificar el cambio en el producto, basándose en los criterios que se han fijado por el grupo de ComAppÑeros y el estándar correspondiente si así compete.
- Si el error de software es grave, el producto volverá a fase de producción con las respectivas indicaciones de como resolver el error así como los necesarios procedimientos formales necesarios que garanticen la corrección del error, con el fin de solventarlo lo más rápido y eficientemente posible.

9. Herramientas, técnicas y metodologías

El equipo SQA debe asegurar que las herramientas software, técnicas y metodologías usadas para controlar el proceso de creación y gestión de software son las adecuadas y se usan apropiadamente. Dichas herramientas software, técnicas y metodologías son descritas en los documentos:

- Plan de Gestión de la Configuración Software.
- Plan de Gestión de Riesgos Software
- Plan de Proyecto Software.

Las herramientas software utilizadas para el desarrollo y gestión del proyecto se agrupan de la siguiente manera, según su función principal:

9.1. Almacén de documentación del proyecto

- *Google Drive*: debe ser utilizado para guardar ordenadamente la documentación que se vaya generando. En una carpeta a la que también tendrá acceso el profesor, se realizarán las entregas que se vayan pidiendo, tales como informes semanales, o cualquier otra tarea.

Mientras esta herramienta funcione adecuadamente, se deberá usar para este fin exclusivamente esta herramienta, durante todo el proceso de desarrollo y gestión del proyecto.

Se debe tener en cuenta que las cuentas de los miembros del equipo, al ser de la UCM, disponen de almacenamiento ilimitado en Google Drive, por tanto, la única limitación para el uso de esta herramienta es la conexión necesaria a internet.

- *Dropbox*: debe ser utilizado para el mismo fin que Google Drive, pero únicamente se usará en el caso de que Google Drive falle de manera generalizada.

Las características son equiparables a Google Drive en cuanto al uso que le daríamos. Sin embargo en Dropbox se tiene la limitación de tener una capacidad inicial de 2GB por cuenta.

- *Github*: Se empleará esta herramienta para la conservación del código generado por el proyecto así como su desarrollo en paralelo por los miembros del equipo.

9.2. Comunicación y coordinación entre los miembros del proyecto (y el profesor)

- *Gmail*: debe ser utilizado exclusivamente para comunicarse, ya sea entre los miembros del grupo o con el profesor, en cuyo caso la persona encargada de comunicarse con él, será el jefe de equipo: Rodrigo de Miguel. Los correos enviados/recibidos se tratarán como documentos formales y oficiales, los cuales se pueden utilizar para hacer cualquier tipo de alegación.

Al igual que en Google Drive, se debe tener en cuenta que las cuentas de los miembros del equipo, al ser de la UCM, disponen de almacenamiento ilimitado en Gmail.

En caso excepcional de no disponer de otros medios, se podría utilizar Gmail para copias de seguridad.

- *Outlook*: Únicamente en el caso de fallo generalizado de Gmail se utilizará Outlook. Las características son equiparables a Gmail, en cuanto al uso que le daríamos, excepto que en Outlook, a pesar de tener capacidad ilimitada, en el caso de que nuestra capacidad se incrementase repentinamente, se podría dar el caso de tener que esperar algún día para volver a tener más espacio de almacenamiento.
- *Trello*: debe ser utilizado exclusivamente para comunicación y coordinación (tales como reparto de tareas) durante todo el desarrollo y gestión del proyecto. Será utilizado únicamente por los miembros del equipo.

En el caso de fallo generalizado de la aplicación, simplemente se dejará de usar y se sustituirá su uso mediante las otras herramientas aquí descritas.

Las restricciones por no pagar suplementos no suponen un impedimento para nuestro uso de la aplicación.

- *WhatsApp*: debe ser utilizado exclusivamente para comunicación entre todos los miembros del equipo, para lo cual se dispondrá de un grupo en el que todos los miembros serán incluidos. Debe quedar claro que los mensajes a través de este medio se pueden borrar, y por tanto, no servir para futuras alegaciones (para ese fin se utilizará Gmail).

En el caso de fallo generalizado de la aplicación, se utilizará Telegram, la cual no está aquí descrita por ser totalmente igual en cuanto al uso que le daríamos.

9.3. Desarrollo de documentación

- *Texmaker*: debe ser utilizado exclusivamente para desarrollar la documentación sobre el lenguaje de marcado Latex, usado para dar un formato elegante y de calidad a la documentación generada. Los documentos entregables se harán con esta aplicación.

En el caso de fallo generalizado de la aplicación, se utilizará únicamente Google Docs.

- *Google Docs*: debe ser utilizado exclusivamente para desarrollar la documentación pertinente, ya sea de manera individual o en paralelo.

En el caso de fallo generalizado de la aplicación, se utilizará únicamente Texmaker (o alguna herramienta que permita trabajar con Latex, ya sea individualmente o en paralelo), y posteriormente se juntará todo el trabajo realizado mediante la coordinación de los subjeses y jefe.

- *IBM RSA*: debe ser utilizado exclusivamente para desarrollar la documentación que se deba realizar en forma de diagramas, UML principalmente. Los subgrupos y el grupo en general utilizará GitHub como repositorio para los diagramas y código, para poder trabajar en paralelo.

En el caso de fallo generalizado de la aplicación, se usará cualquier programa estable que permita el diseño gráfico, y preferiblemente en forma de diagramas.

Respecto a todas esas herramientas:

- Todos los miembros del grupo tienen acceso a ellas.
- Todos los miembros del grupo conocen su funcionamiento, bien porque ya las han usado anteriormente o bien porque se les ha enseñado a utilizarlas durante el desarrollo del proyecto.
- En el caso de que algún miembro necesitase más formación para alguna de esas herramientas, el personal del equipo que conociese mejor su funcionamiento, le daría dicha formación.

Para asegurarnos del correcto funcionamiento de dichas herramientas, el software utilizado se comprobará que funciona adecuadamente en cada uso que se le de. Y en el caso de descubrir algún fallo importante en alguna herramienta, se usarán las alternativas aquí descritas.

10. Control de medios

A continuación se detallan la identificación del medio físico para cada producto software, así como las metodologías utilizadas para su conservación y protección frente a los daños que pudieran producirse.

10.1. Identificación de medios

Todos la documentación generada a lo largo del desarrollo y gestión del proyecto, es desarrollada, almacenada y entregada de la misma manera. Por tanto trataremos la documentación como un único producto software generalizado, llamado: “Documentación”.

Trabajo entregable	Medio	Documentación	Copia
Documentación	<ul style="list-style-type: none"> ■ Generado en Google Docs y/o LaTeX ■ Almacenado en Google Drive (Cuentas UCM.es) ■ Convertido en PDF ■ Entregado mediante Google Drive 	<ul style="list-style-type: none"> ■ Google Drive: http://www.google.com/policies/privacy/ ■ Cuentas UCM.es: https://www.ucm.es/soe-electronica/politica-privacidad 	<p>Periódicamente el jefe y subjeses de proyecto se descargan la carpeta compartida, que tenemos en Google Drive, a sus PCs (SO Windows 10) y almacenan las versiones en formato comprimido</p>
Código de la aplicación	<ul style="list-style-type: none"> ■ Generado en Eclipse (Java) ■ Almacenado en servidor privado, accendiendo a él mediante Filezilla ■ Además de los ya mencionados en el apartado anterior. 	<ul style="list-style-type: none"> ■ Eclipse: http://www.eclipse.org/documentation/?sess=32267642b71e61a5209ee01bb85d8fdbb ■ Filezilla: https://filezilla-project.org/privacy.php ■ Además de los ya mencionados en el apartado anterior. 	<p>El equipo de desarrollo del proyecto tiene copias locales del código que va generando, solo sube a la nube versiones completas de código depurado y que compila.</p>

Tabla 1: PF de los casos de uso del módulo Buscar Inquilino

10.2. Protección de medios

En esta sección describiremos la política de seguridad que el equipo de ComAppñeros lleva a cabo para evitar el plagio, sabotaje, filtraciones, o posibles pérdidas de información de nuestro proyecto durante su desarrollo o una vez concluido el mismo.

En primer lugar destacamos que, debido a las características académicas del proyecto (y la consecuente no implementación de la aplicación), la confidencialidad no es de los puntos más importantes del proyecto.

Debido al uso de las aplicaciones externas como las herramientas y servicios que proporciona Google, asumimos el control que esta entidad tiene sobre todos nuestros documentos. Mediante Google Drive, se facilita el acceso a toda nuestra documentación.

Dado que todos utilizamos el correo institucional de la UCM y los servicios en la nube que esta institución nos provee. Sabemos que la UCM tiene derechos y acceso a la información allí depositada.

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

- El propietario de la carpeta compartida con el profesor es el jefe del proyecto (Rodrigo de Miguel). Solo él puede añadir miembros y agregar archivos 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. (Véase el caso del profesor y las entregas o nuestros compañeros de otros proyectos para las Revisiones Técnicas Formales).
- Uso de un GitHub privado, para aseguramos que los únicos conocedores de nuestro código seamos nosotros.
- Por política de privacidad, restringimos el acceso a la información del nombre, apellidos, teléfono, DNI y dirección de vivienda del cliente que se registre en el sistema y que almacenamos en un servidor dedicado contratado para el sistema.

No se utilizan medidas de cifrado para encriptar información.

11. Control de proveedor

Nuestra aplicación interactúa con software proveniente de Google (Google Maps y Google+) y de Facebook. La fiabilidad de ambas aplicaciones está más que demostrada, y el riesgo de fallo en estas es mínimo.

Dado que nuestro equipo ha sido creado este año y ninguno de nuestros miembros ha utilizado antes las APIs de Facebook o Google+ para administrar las cuentas de una aplicación ni tampoco la de Google Maps, no disponemos de código heredado para la realización de la aplicación.

12. Colección de registros, mantenimiento y conservación

La documentación que vamos a conservar es la siguiente:

- Especificación de requisitos: Se conservará para poder comprobar que nuestra aplicación cumple con todos los requisitos establecidos por el cliente.
- Plan de Proyecto: Se conservará para comprobar que la documentación referenciada por este es correcta y que el proyecto se ha desarrollado de la manera correcta.
- Plan de Gestión de Riesgos: Se conservará para poder reaccionar de manera efectiva ante los riesgos y poder comprobar si los riesgos contemplados en este se corresponden con los que han surgido durante el proyecto, así como si las soluciones propuestas en este han sido óptimas.
- Plan SQA: Se conservará para garantizar que se han seguido las medidas de garantía de calidad correspondientes.
- Plan de Gestión de la Configuración: Se conservará para evitar errores y gestionar posibles cambios de una forma en la que se realice el mínimo esfuerzo posible, garantizando siempre la estabilidad y la calidad de la aplicación.
- Revisiones Técnicas Formales: Se conservarán para futuras revisiones y para tener un registro de los cambios y decisiones acordadas en dichas reuniones.
- Informes Semanales: Se conservarán para llevar un registro del trabajo realizado semanalmente y de la evolución del proyecto a través del tiempo, y de si este progreso se corresponde al previsto.

La documentación se guardará en formato digital. Se guardarán copias en una carpeta compartida en “Google drive”. El jefe de proyecto también tendrá una copia de seguridad de todo el proyecto, así como los subjefes. De esta forma se reducirá al mínimo el riesgo de pérdida de información.

Para mantener la documentación actualizada y arreglar posibles fallos, esta se someterá a revisión por todos los miembros del equipo antes de ser validada. Los cambios en la documentación deberán ser informados al resto del equipo antes de ser llevados a cabo para evitar posibles colisiones.

La información se conservará durante un año una vez sea creada la última versión de esta.

13. Formación

Todos los miembros del equipo hemos aprobado y/o cursado las asignaturas de Ingeniería del Software y Modelado del Software donde nos hemos familiarizado con el desarrollo de documentación para proyectos software. Con esto y con lo dado en las clases de Gestión de Proyectos Software y Metodologías del Desarrollo estamos capacitados para hacer un buen plan SQA.

13.1. Funciones

El equipo SQA deberá cumplir las siguientes funciones:

- Identificación de calidad que debe cumplir el producto.
- Verifica la entrega y mantenimiento de la documentación requerida.
- Revisar que las actividades de ingeniería del software se ajusten al proceso del software.
- Mantener una documentación ajustada al procedimiento establecido.
- Normaliza y audita un correcto proceso, cumpliendo los procedimientos establecidos.

14. Gestión del riesgo

Toda la información detallada de la gestión de riesgos se encuentra en el documento “Gestión de Riesgos”.

15. Glosario

- **API (Application Programming Interface):** conjunto de subrutinas, métodos y procedimientos que cumplen una serie de funciones con el fin de poder ser utilizadas por otro software como una capa de abstracción.
- **Auditoria:** el examen crítico y sistemático que realiza una persona o grupo de personas sobre un sistema, proceso, proyecto o producto, con el fin de emitir una opinión independiente y competente.
- **BBDD:** Base de Datos. Conjunto de datos pertenecientes a un mismo contexto almacenados para su posterior uso.
- **Cambio:** modificación o alteración de una cosa.
- **Estándar:** tipo, modelo, norma o patrón establecido que sirve como referencia para ser utilizado posteriormente.
- **Framework:** conjunto de programas, bibliotecas y otras herramientas que puede servir de base para la organización y desarrollo de software.
- **GUI (Graphical User Interface):** Interfaz grafica de usuario. Consiste en proporcionar un entorno visual sencillo que permita la comunicación entre los usuarios y el sistema.
- **Metodología:** conjunto de procedimientos que permiten alcanzar un objetivo.
- **Métrica:** medición del software en relación a parámetros predeterminados (Ej.: número de líneas de código, volumen de la documentación, etc.).
- **Prueba:** hecho, suceso o argumento con que se intenta probar que algo cumple unos determinados aspectos.
- **Requerimiento:** característica que se desea que tenga un sistema.
- **Requisito:** necesidad sobre el contenido, forma o funcionalidad de un producto o servicio. Establece lo que debe hacer el sistema.
- **Revisión:** sometimiento de una cosa a un examen para comprobar su funcionamiento y validez, y en caso de no ser correctos, corregirla y/o repararla.
- **Riesgo:** posibilidad de que se produzca un contratiempo que ponga en peligro la integridad del sistema.
- **SQA (Software Quality Assurance):** Conjunto de actividades sistemáticas y planeadas para asegurar que los procesos y productos de software cumplen con los requerimientos, estándares y procedimientos.

16. Procedimiento de cambio e historias del plan de SQA

Los posibles cambios en el plan SQA deberán de ser comunicados al jefe del proyecto, Rodrigo de Miguel, para que pueda evaluarlos y si fuese necesario, aplicar dichos cambios. Además, habrá que apuntar el cambio en un historial que almacenará todos los cambios realizados hasta el momento de forma detallada (identificador del cambio, motivo del cambio, persona o personas encargadas de realizar dicho cambio y la fecha en la que se realizó).

Se realizaran cuatro revisiones de los documentos: una por cada mes hasta la fecha de entrega del proyecto, más otra unos días antes de la entrega.

17. Apéndices

17.1. Revisiones Técnicas Formales

Las RTF llevadas acabo por el equipo de **ComAppÑeros** quedan detalladas en las «actas de RTF» que se realizaron para dicho fin una vez hechas y aprobadas por los equipos revisado.