

Challenge Accepted

Plan de SQA

Versión 2.2

Historia de revisiones

Fecha	Versión	Descripción	Autor
26/08/2014	1.0	Creación del documento.	Federico Blumetto.
29/08/2014	1.1	Revisión del documento.	Gonzalo Sintas.
30/08/2014	1.2	Corrección del documento.	Federico Blumetto.
30/08/2014	1.3	Revisión y Verificación del documento.	Gonzalo Sintas.
09/09/2014	2.0	Ajuste del plan.	Federico Blumetto.
11/09/2014	2.1	Se agregaron checklists.	Federico Blumetto.
14/09/2014	2.2	Revisión del documento.	Gonzalo Sintas.

Contenido:

- [1. Propósito](#)
- [2. Referencias](#)
- [3. Gestión](#)
 - [3.1. Organización](#)
 - [3.2. Actividades](#)
 - [3.2.1. Ciclo de vida del software cubierto por el Plan](#)
 - [3.2.2. Actividades de calidad a realizarse](#)
 - [3.2.3. Revisar cada producto](#)
 - [3.2.4. Revisar el ajuste al proceso](#)
 - [3.2.5. Realizar Revisión Técnica Formal \(RTF\)](#)
 - [3.2.6. Asegurar que las desviaciones son documentadas](#)
 - [3.2.7. Relaciones entre las actividades de SQA y la planificación](#)
 - [3.3. Responsables](#)
- [4. Documentación](#)
 - [4.1. Propósito](#)
 - [4.2. Documentación mínima requerida](#)
 - [4.2.1. Especificación de requerimientos del software](#)
 - [4.2.2. Modelo de casos de uso](#)
 - [4.2.3. Modelo de dominio.](#)
 - [4.2.4. Pautas para la interfaz de usuario.](#)
 - [4.2.5. Descripción del diseño del software.](#)
 - [4.2.6. Plan de Calidad.](#)
 - [4.2.7. Plan de Verificación & Validación.](#)
 - [4.2.8. Plan de Gestión de configuración.](#)
 - [4.2.9. Plan de Proyecto.](#)
 - [4.3. Otros documentos](#)
- [5. Estándares, prácticas, convenciones y métricas](#)
 - [5.1. Estándar de documentación](#)
 - [5.2. Estándar de verificación y prácticas](#)
 - [5.3. Otros Estándares](#)
- [6. Revisiones y auditorías](#)
 - [6.1. Objetivo](#)
 - [6.2. Requerimientos mínimos](#)
 - [6.2.1. Revisión de requerimientos](#)
 - [6.2.2. Revisión de diseño preliminar](#)
 - [6.2.3. Revisión de diseño crítico](#)
 - [6.2.4. Revisión del Plan de Verificación & Validación](#)
 - [6.2.5. Auditoría funcional](#)
 - [6.2.6. Auditoría física](#)
 - [6.2.7. Auditorías internas al proceso](#)
 - [6.2.8. Revisiones de gestión](#)
 - [6.2.9. Revisión del Plan de gestión de configuración](#)

- [6.2.10. Revisión Post Mortem](#)
 - [6.2.11. Agenda](#)
 - [6.3. Otras revisiones](#)
 - [6.3.1. Revisión de documentación de usuario](#)
- [7. Verificación](#)
- [8. Reporte de problemas y acciones correctivas](#)
- [9. Herramientas, técnicas y metodologías](#)
- [10. Gestión de riesgos](#)
- [11. Checklists](#)
 - [11.1. Especificación de requerimientos.](#)
 - [11.2. Modelo de Casos de Uso.](#)
 - [11.3. Pautas para la interfaz de usuario.](#)
 - [11.4. Arquitectura.](#)
 - [11.5. Descripción de la arquitectura.](#)
 - [11.6. Plan de gestión de la configuración.](#)
 - [11.7. Plan del proyecto.](#)
 - [11.8. Plan de verificación y validación.](#)
 - [11.9. Documentación general a entregar.](#)

1. Propósito

El propósito de este documento es detallar todo lo referente a la planificación del seguimiento de la calidad del proyecto. Abarca tanto la calidad del proceso a seguir como la calidad del producto a construir.

En este plan de calidad se especificará:

1. el detalle de las actividades realizadas.
2. mecanismos para monitorear y medir la calidad del proceso.
3. mecanismos para monitorear y medir la calidad del producto.
4. los niveles de calidad.
5. los objetivos a cumplir.

El producto que desarrollaremos es un juego multiplataforma para Windows Phone y Android, del tipo búsqueda del tesoro, donde el objetivo es superar una serie de desafíos semanales que utilizan las características del teléfono móvil.

Uno de nuestros objetivos es que el juego en cuestión sea publicado en el Windows Phone Store y en la Play Store, por lo que el mismo debe cumplir una serie de requisitos establecidos para los productos de las respectivas tiendas.

2. Referencias

[1]ANSI/IEEE Std 730.1-1989, IEEE Standard for Software Quality Assurance Plans.

3. Gestión

3.1. Organización

Como se menciona en el MUM, las líneas de trabajo están divididas en las disciplinas de trabajo básicas y las disciplinas de gestión.

A continuación vamos a definir cómo interactúa el equipo de trabajo de la gestión de la calidad con las diferentes disciplinas.

Equipo de gestión de calidad:

1. Federico Blumetto - Responsable de SQA.
2. Gonzalo Sintas - Asistente de SQA.

Líneas de trabajo básicas:

- Requerimientos: Los integrantes de esta línea deben informar o hacer partícipe a la línea de gestión de calidad para saber que propiedades de calidad se debe cumplir para obtener la conformidad del cliente.
- Análisis: En el análisis tienen que colaborar tanto los integrantes que saben sobre los requerimientos como la línea de calidad para asegurar la misma.
- Diseño: Se seguirán de cerca las acciones del diseño para que se respeten (en la medida de lo posible) los estilos de las interfaces nativas de cada plataforma.
- Implementación: Para esta disciplina se establecerán procedimientos de codificación a seguir y se revisarán por parte de equipo de SQA que se cumplan las mismas.
- Verificación: Los integrantes del equipo de verificación realizarán su trabajo y pasaran un reporte de las pruebas realizadas.

Líneas de trabajo de gestión:

- Gestión de la configuración: Los integrantes de esta disciplina informaran su progreso y los contratiempos que puedan llegar a ocurrir.
- Gestión de proyecto: Al igual que la otra disciplina de gestión mencionada, esta deberá informar el progreso del proyecto.
- Comunicación: La comunicación es fundamental para el éxito de un proyecto, por lo tanto tiene que ser de forma clara. El responsable de la comunicación deberá hacer que la comunicación sea fluida y encontrar un medio para que todos los integrantes estén al tanto de las actividades a realizar y la situación del proyecto en cuestión.

3.2. Actividades

3.2.1. Ciclo de vida del software cubierto por el Plan

El plan de calidad cubre las siguientes etapas:

- Etapa de Requerimientos y análisis.
- Etapa de Diseño.
- Etapa de Implementación.
- Etapa de Verificación.

Para cada etapa se revisan los documentos que deberán ser entregados y los productos que se vayan generando a lo largo del proyecto, tanto prototipos como versiones beta de cada plataforma en que desarrollaremos.

3.2.2. Actividades de calidad a realizarse

Las tareas a ser llevadas a cabo deberán reflejar las evaluaciones a realizar, los estándares a seguir, los productos a revisar, los procedimientos a seguir en la elaboración de los distintos productos y los procedimientos para informar de los defectos detectados a sus responsables y realizar el seguimiento de los mismos hasta su corrección.

Las actividades que se realizarán son:

- Revisar cada producto.
- Revisar el ajuste al proceso.
- Realizar Revisión Técnica Formal (RTF).
- Asegurar que las desviaciones son documentadas.

3.2.3. Revisar cada producto

En esta actividad se revisan los productos que se definieron como claves para verificar en el Plan de Calidad.

Se debe verificar que no queden correcciones sin resolver en los informes de revisión previos, si se encuentra alguna no resuelta, debe ser incluida en la siguiente revisión. Se revisan los productos contra los estándares, utilizando la checklist definida para el producto.

Se debe identificar, documentar y seguir la pista a las desviaciones encontradas y verificar que se hayan realizado las correcciones.

Como salida se obtiene el Informe de revisión de SQA, este informe debe ser distribuido a los responsables del producto y se debe asegurar de que son conscientes de desviaciones o discrepancias encontradas.

3.2.4. Revisar el ajuste al proceso

En esta actividad se revisan los productos que se definieron como claves para verificar el cumplimiento de las actividades definidas en el proceso. Con el fin de asegurar la calidad en el producto final del desarrollo, se deben llevar a cabo revisiones sobre los productos durante todo el ciclo de vida del software.

Se debe recoger la información necesaria de cada producto, buscando hacia atrás los productos previos que deberían haberse generado, para poder establecer los criterios de revisión y evaluar si el producto cumple con las especificaciones.

Esta información se obtiene de los siguientes documentos:

- Plan del Proyecto.
- Plan de la iteración.
- Plan de Verificación.

Antes de comenzar, se debe verificar en los informes de revisión previos que todas las desviaciones fueron corregidas, si no es así, las faltantes se incluyen para ser evaluadas.

Como salida se obtiene el Informe de revisión de SQA correspondiente a la evaluación de ajuste al proceso, este informe debe ser distribuido a los responsables de las actividades y se debe asegurar de que son conscientes de desviaciones o discrepancias encontradas.

3.2.5. Realizar Revisión Técnica Formal (RTF)

El objetivo de la RTF es descubrir errores en la función, la lógica ó la implementación de cualquier producto del software, verificar que satisface sus especificaciones, que se ajusta a los estándares establecidos, señalando las posibles desviaciones detectadas. Es un proceso de revisión riguroso, su objetivo es llegar a detectar lo antes posible, los posibles defectos o desviaciones en los productos que se van generando a lo largo del desarrollo. Por esta característica se adopta esta práctica para productos que son de especial importancia.

En la reunión participan el responsable de SQA e integrantes del equipo de desarrollo.

Se debe convocar a la reunión formalmente a los involucrados, informar del material que ellos deben preparar por adelantado, llevar una lista de preguntas y dudas que surgen del estudio del producto a ser revisado.

La duración de la reunión no debe ser mayor a dos horas.

Como salida se obtiene el Informe de RTF.

3.2.6. Asegurar que las desviaciones son documentadas

Las desviaciones encontradas en las actividades y en los productos deben ser documentadas y ser manejadas de acuerdo a un procedimiento establecido.

Se debe chequear que los responsables de cada plan los modifiquen cada vez que sea necesario, basados en las desviaciones encontradas.

3.2.7. Relaciones entre las actividades de SQA y la planificación

Se detallan en la siguiente tabla las actividades referentes a la calidad según se establecen en el MUM.

Actividad	Semana cuando se realiza
Identificar las propiedades de calidad.	02,03,04.
Plan de calidad.	02,04.
Evaluar y ajustar el plan de SQA.	06,08,12.
Revisión técnica formal.	05,07,09,11.
Revisar las entregas.	Todas las semanas.
Revisar el ajuste al proceso.	Todas las semanas.
Evaluar la calidad de los productos.	03,05,07,09,11,13,14.
Realizar el informe final de calidad.	14.

3.3. Responsables

En la siguiente tabla se muestran los responsables de las actividades relacionadas a la calidad del proyecto.

Actividad	Responsable
Identificar las propiedades de calidad	Responsable de SQA. Colaboración de asistente y analistas.
Plan de calidad.	Responsable de SQA.
Evaluar y ajustar el plan de SQA.	Responsable de SQA y asistente.
Revision tecnica formal.	Responsable de SQA. En conjunto con los implementadores.
Revisar las entregas.	Responsable de SQA.
Revisar el ajuste al proceso.	Responsable de SQA.
Evaluar la calidad de los productos.	Responsable de SQA y asistente.
Realizar el informe final de calidad.	Responsable de SQA.

4. Documentación

4.1. Propósito

Identificación de la documentación relativa a desarrollo, Verificación & Validación, uso y mantenimiento del software.

Establecer como los documentos van a ser revisados para chequear consistencia: se confirman criterio e identificación de las revisiones.

4.2. Documentación mínima requerida

La documentación mínima es la requerida para asegurar que la implementación logrará satisfacer los requerimientos.

La documentación mínima es la siguiente:

- Especificación de requerimientos de software.
- Modelo de casos de uso.
- Modelo de Dominio.
- Pautas para la interfaz de usuario.
- Descripción y diseño del software (Arquitectura).
- Plan de Calidad.
- Plan de Verificación y Validación.
- Plan de Configuración.
- Plan de Proyecto.

4.2.1. Especificación de requerimientos del software

El documento de especificación de requerimientos deberá describir, de forma clara y precisa, cada uno de los requerimientos esenciales del software además de las interfaces externas.

El cliente deberá obtener como resultado del proyecto una especificación adecuada a sus necesidades en el área de alcance del proyecto, de acuerdo al compromiso inicial del trabajo y a los cambios que éste haya sufrido a lo largo del proyecto, que cubra aquellos aspectos que se haya acordado detallar con el cliente.

La especificación debe:

- Ser completa:
 - Externa, respecto al alcance acordado.
 - Internamente, no deben existir elementos sin especificar.
- Ser consistente. No pueden haber elementos contradictorios.
- Ser no ambigua. Todo término referido al área de aplicación debe estar definido en un glosario.
- Ser verificable. Debe ser posible verificar, siguiendo un método definido, si el producto final cumple o no con cada requerimiento.

- Estar acompañada de un detalle de los procedimientos adecuados para verificar si el producto cumple o no con los requerimientos.
- Incluir requerimientos de calidad del producto a construir.

Los requerimientos de calidad del producto a construir son considerados dentro de atributos específicos del software que tienen incidencia sobre la calidad en el uso y se detallan a continuación:

Funcionalidad

- a. adecuación a las necesidades.
- b. precisión de los resultados.
- c. interoperabilidad.
- d. seguridad de los datos.

Confiabilidad

- a. madurez.
- b. tolerancia a faltas.

Usabilidad

- a. comprensible.
- b. aprendible.
- c. operable.
- d. atractivo.

Eficiencia

- a. comportamiento respecto al tiempo.
- b. utilización de recursos.

Mantenibilidad

- a. analizable.
- b. modificable.
- c. estable, no se producen efectos inesperados luego de modificaciones.
- d. verificable.

Portabilidad

- a. adaptable.
- b. instalable.
- c. co-existencia.

Cada uno de estos atributos debe cumplir con las normas y regulaciones aplicables a cada uno.

4.2.2. Modelo de casos de uso

El Modelo de casos de uso es fundamental para describir detalladamente los requerimientos del sistema.

4.2.3. Modelo de dominio.

El modelo de domino muestra en forma gráfica las relaciones entre los componentes.

4.2.4. Pautas para la interfaz de usuario.

Este documento es fundamental para identificar cómo va a lucir la interfaz de usuario y evaluar la facilidad de uso.

4.2.5. Descripción del diseño del software.

El documento de diseño especifica como el software será construido para satisfacer los requerimientos.

Deberá describir los componentes y subcomponentes del diseño del software, incluyendo interfaces internas. Este documento deberá ser elaborado primero como Preliminar y luego será gradualmente extendido hasta llegar a obtener el Detallado.

El cliente deberá obtener como resultado del proyecto el diseño de un producto de software que cubra aquellos aspectos que se haya acordado con el cliente incorporar al diseño, en función de la importancia que estos presenten y de sus conexiones lógicas.

El diseño debe:

- Corresponder a los requerimientos a incorporar:
 - Todo elemento del diseño debe contribuir a algún requerimiento.
 - La implementación de todo requerimiento a incorporar debe estar contemplada en por lo menos un elemento del diseño.
- Ser consistente con la calidad del producto.

4.2.6. Plan de Calidad.

El Plan de Calidad es relevante porque se tratan todos los temas referentes al seguimiento de la calidad en el proyecto.

4.2.7. Plan de Verificación & Validación.

El Plan de V & V deberá identificar y describir los métodos a ser utilizados en:

- La verificación de que:
 - los requerimientos descritos en el documento de requerimientos han sido aprobados por una autoridad apropiada. En este caso sería que cumplan con el acuerdo logrado entre el cliente y el equipo.
 - los requerimientos descritos en el documento de requerimientos son implementados en el diseño expresado en el documento de diseño.
 - el diseño expresado en el documento de diseño está implementado en código.
- Validar que el código, cuando es ejecutado, se adecua a los requerimientos expresados en el documento de requerimientos.

4.2.8. Plan de Gestión de configuración.

El Plan de gestión de configuración debe contener métodos para identificar componentes de software, control e implementación de cambios, y registro y reporte del estado de los cambios implementados.

4.2.9. Plan de Proyecto.

El Plan de Proyecto debe contener todas las actividades relacionadas con la gestión del proyecto para que el mismo pueda realizarse de manera controlada.

4.3. Otros documentos

Todo lo que se elabora, en mayor o menor medida, afecta a la calidad del proyecto. Los que se consideran de mayor incidencia se destacan en la sección anterior.

5. Estándares, prácticas, convenciones y métricas

5.1. Estándar de documentación

Como estándares de documentación se definirán:

- Estándar de documentación técnica.
- Estándar de los entregables (Las pautas se definen a continuación).

La documentación técnica del producto debe:

- Ser adecuada para que un grupo independiente del de desarrollo pueda encarar el mantenimiento del producto.
- Incluir fuentes, Modelos de Casos de Uso, Objetos.

Para la escritura de documentos se han definido plantillas para ser utilizadas en la elaboración de entregables.

Todas las plantillas, y un documento adicional creado para llevar control de quien es el responsable de cada documento, son colocados al inicio de la semana en Google Drive, en una carpeta compartida con el grupo.

En estas plantillas se definen:

- No contiene encabezados y pies de páginas.
- El texto normal se realiza en Verdana tamaño 11 y justificado.
- Los títulos son acorde a los que provee Google Drive, también en la fuente Verdana.
- Deben contener además como mínimo: fecha, versión, responsables y un historial de las revisiones del mismo.

Se definen además una serie de pautas para trabajar con la documentación de forma adecuada:

- Se utiliza como herramienta Google Docs para procesar texto y Google Sheets para planillas de cálculo. Definimos estas herramientas por la facilidad que proveen de trabajo concurrente y almacenamiento en la nube.
- El nombre del archivo deberá respetar el formato que aparece en el MUM.
- Utilizar como mínimo una herramienta de corrección ortográfica.
- Verificar mínimamente el índice de contenidos.
- Debe existir una primera revisión básica con los integrantes del área al cual corresponda el documento.
- Cuando un documento es finalizado se debe marcar en el documento de control (llamado "Nombres de documentos") para dejarlo al equipo de SQA. La línea correspondiente a ese archivo debe resaltarse con color amarillo y marcar la columna "Listo para la revisión".

- Cuando un documento no se entrega se resalta la fila correspondiente con color rojo.

5.2. Estándar de verificación y prácticas

Se utilizan las prácticas definidas en el Plan de Verificación y Validación.

Como estándar se utiliza el documento de:

- Std 1012-1986 IEEE Standard for Software Verification and Validation Plans.

5.3. Otros Estándares

En futuras versiones especificaremos estándares para la codificación, así como estándares que deberá cumplir la aplicación para formar parte de las tiendas de Windows Phone y Android.

6. Revisiones y auditorías

6.1. Objetivo

Definición de las revisiones y auditorías técnicas y de gestión que se realizarán.

Especificación de cómo serán llevadas a cabo dichas revisiones y auditorías.

6.2. Requerimientos mínimos

6.2.1. Revisión de requerimientos

Esta revisión se realiza para asegurar que se cumplió con los requerimientos especificados por el Cliente.

6.2.2. Revisión de diseño preliminar

Esta revisión se realiza para asegurar la consistencia y suficiencia técnica del diseño preliminar del software.

6.2.3. Revisión de diseño crítico

Esta revisión se realiza para asegurar la consistencia del diseño detallado con la especificación de requerimientos.

6.2.4. Revisión del Plan de Verificación & Validación

Esta revisión se realiza para asegurar la consistencia y completitud de los métodos especificados en el Plan de V & V.

6.2.5. Auditoría funcional

Esta auditoría se realiza previa a la liberación del software, para verificar que todos los requerimientos especificados en el documento de requerimientos fueron cumplidos.

6.2.6. Auditoría física

Esta revisión se realiza para verificar que el software y la documentación son consistentes y están aptos para la liberación.

6.2.7. Auditorías internas al proceso

Estas auditorías son para verificar la consistencia:

- del código versus el documento de diseño.
- especificaciones de interfaz.
- implementaciones de diseño versus requerimientos funcionales.
- requerimientos funcionales versus descripciones de testeo.

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).

6.2.9. Revisión del Plan de gestión de configuración

Esta revisión se realiza para asegurar 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 concluir el proyecto para especificar las actividades de desarrollo implementadas durante el proyecto y para proveer recomendaciones.

6.2.11. Agenda

Las revisiones y auditorías se piensan realizar de la siguiente forma, como muestra la tabla:

Revisión	Semanas
Revisión de Requerimientos.	03,04,05,06,07.

Revisión de Diseño preliminar.	04,05,06.
Revisión de Diseño Crítico.	07,08,09,10.
Revisión de Plan de V&V.	05,07,10.
Auditoria funcional.	12.
Auditoria física.	12.
Auditorías internas al proceso.	04,06,08,10,12.
Revisión de gestión.	04,06,08,10,12.
Revisión del plan de gestión.	04,06,08,10,12.
Revisión Post Mortem.	14.

6.3. Otras revisiones

6.3.1. Revisión de documentación de usuario

Se revisa la completitud, claridad, correctitud y aplicación de uso.

7. Verificación

Sobre cuestión de la verificación se debe consultar el Plan de Verificación y Validación, donde se tratan todas las cuestiones referentes a esta disciplina.

8. Reporte de problemas y acciones correctivas

El equipo de SQA es el responsable de la revisión de las entregas, pero además cada entregable tiene un responsable que lo elabora. En caso de que el entregable no pase la revisión de calidad se le informará al responsable de el entregable en cuestión para que realice las acciones correctivas correspondientes.

9. Herramientas, técnicas y metodologías

En principio se realiza un revisión manual de las entregas por el equipo de SQA. Otras herramientas serán definidas a futuro para el apoyo de la revisión de los entregables.

10. Gestión de riesgos

Sobre cuestiones de riesgos hay que consultar el documento de riesgos.

11. Checklists

En esta sección se especifican Checklists que se utilizarán en la corrección de los entregables. Las mismas fueron obtenidas de la memoria organizacional, donde seleccionamos las que se adaptan a nuestras necesidades.

11.1. Especificación de requerimientos.

Nº	Pregunta	Si	No	NA
1	¿Se utiliza un lenguaje conocido por el cliente?			
2	¿Se utiliza un lenguaje conocido por los desarrolladores?			
3	¿Cada requerimiento está definido una única vez? (no hay requerimientos redundantes)			
4	¿Los requerimientos están especificados con un nivel apropiado de detalle? (ni más, ni menos)			
5	¿Los requerimientos funcionales cubren todas las situaciones anormales? (ejemplo: caída del sistema)			x
6	¿El documento contiene sólo detalles de implementación necesarios?			
7	¿Cada característica del producto final es descripta con una única terminología?			
8	¿Se corresponde con las necesidades explicitadas en los documentos referentes a las reuniones de requerimientos que se mantuvieron con el cliente?			
9	¿Hay un índice?			
10	¿Este documento ha sido validado por el cliente?			
11	¿Los requerimientos están priorizados?			
12	¿Todas las unidades de medida están definidas?			
13	¿Esta definido perfectamente el alcance del sistema?			
14	¿Algún requerimiento debería estar especificado con más detalle?			

15	¿Algún requerimiento debería estar especificado con menos detalle?			
16	¿Se ha definido qué información falta si es que falta alguna?			
17	¿Están descritas claramente las principales interfaces externas e internas del producto?			
18	¿Están incluidos los requerimientos de calidad? (como ser robustez, confiabilidad)			
19	¿Están incluidos todos los requerimientos relacionados con las restricciones de Diseño?			
20	¿Están incluidos todos los requerimientos relacionados con el hardware?			
21	¿Están incluidos todos los requerimientos relacionados con entradas?			
22	¿Están incluidos todos los requerimientos relacionados con salidas?			
23	¿La estructura del documento se adhiere a un estándar aceptado?			
24	¿El documento de requerimientos está organizado clara y lógicamente?			
25	¿El documento de requerimientos está libre de contradicciones?			
26	¿Cada requerimiento es relevante al problema y a su solución?			
27	¿Hay algún requerimiento que pueda tener más de una interpretación?			
28	¿Hay un glosario en el cual el significado específico de cada término está definido?			
29	¿Hay algún requerimiento que es imposible de verificar?			
30	¿Cada requerimiento puede ser trazado hacia el documento donde se describe el problema o hacia una necesidad específica del cliente?			
31	¿Todos los requerimientos son trazables hacia un documento específico de diseño?			
32	¿Todos los requerimientos son trazables hacia un módulo específico?			
33	¿Algunos de los requerimientos definidos son en realidad detalles de diseño?			
34	¿Algunos de los requerimientos definidos son en realidad detalles de verificación?			
35	¿Algunos de los requerimientos definidos son en realidad detalles de gestión del proyecto?			
36	¿Está definida la plataforma de software requerida?			

37	¿Están definidas las herramientas de software requeridas?			
38	¿Los requerimientos de desempeño mínimos de la red están especificados?			

11.2. Modelo de Casos de Uso.

Nº	Pregunta	Si	No	NA
1	¿Todos los actores del modelo son exactamente los que se desprenden del Documento de Requerimientos?			
2	¿Los actores tienen los nombres intuitivos y descriptivos?			
3	¿Todos los actores están claramente definidos y son consistentes con el Documento de Requerimientos?			
4	¿Se puede ver claramente desde el diagrama de casos de uso y sus descripciones qué actores están involucrados en cada caso de uso?			
5	¿Todos los actores están conectados con los casos de uso correctos de acuerdo al Documento de Requerimientos?			
6	¿Todos los casos de uso del modelo son exactamente los que se desprenden del Documento de Requerimientos?			
7	¿Todos los casos de uso llevan a cumplir un sólo objetivo claramente definido?			
8	¿Todos los casos de uso tienen nombres que transmiten claramente cuál es su objetivo?			
9	¿Todas las interacciones de los actores con el sistema son consistentes con las descripciones de los actores?			
10	¿Todas las descripciones de los casos de uso son consistentes con el Documento de Requerimientos?			
11	¿Todas las entradas y salidas están correctamente definidas para cada caso de uso?			
12	¿Todos los flujos alternativos han sido cubiertos?			
13	¿Todas las pre y postcondiciones para cada caso de uso están especificadas?			
14	¿Todos los diagramas de los casos de uso concuerdan con las descripciones de los mismos?			

15	¿Todos los casos de uso están escritos como casos de uso esenciales?			
16	¿Todos los casos de uso están libres de detalles de implementación?			

11.3. Pautas para la interfaz de usuario.

Nº	Pregunta	Si	No	NA
1	¿Definió los formatos de los distintos tipos de pantallas?			
2	¿Definió los controles y elementos de diseño de interfaz de pantalla sus características, su disposición, y cómo se gestionan los eventos relacionados con ellos?			
3	¿Indicó la forma de navegar dentro de la pantalla y entre ellas?			
4	¿Definió los distintos tipos de formatos de impresión que se utilizarán en el sistema?			

11.4. Arquitectura.

Nº	Pregunta	Si	No	NA
1	¿La arquitectura está compuesta de capas?			
2	¿Los paquetes están bien definidos, incluyen su funcionalidad e interfaz para otros módulos?			
3	¿Todas las estructuras de datos importantes están descritas y justificadas?			
4	¿Todas las estructuras de datos importantes están ocultas mediante funciones de acceso?			
5	¿La organización y contenido de la base de datos están especificados?			
6	¿Todos los objetos importantes están descritos y justificados?			
7	¿La interfaz con el usuario está modularizada tal que los cambios en ella no afecten el resto del programa?			
8	¿Los aspectos claves de la interfaz de usuario están definidos?			
9	¿Se ha estimado el uso de memoria, se ha descrito y justificado una estrategia para la gestión de la memoria?			
10	¿Se ha incluido una estrategia coherente de manejo de errores?			

11	¿Se ha definido un nivel de robustez?			
12	¿La arquitectura está diseñada para acomodar cambios?			
13	¿Los objetivos principales del sistema están claramente establecidos?			
14	¿El diseño de alto nivel es independiente de la máquina y lenguaje que se usará para implementarlo?			
15	¿Se dan motivaciones para todas las decisiones importantes?			

11.5. Descripción de la arquitectura.

Nº	Pregunta	Si	No	NA
1	¿Identificó los casos de uso y actores más significativos para definir la Arquitectura del sistema (tomando en cuenta los más importantes para el cliente y que mitiguen los riesgos más serios)?			
2	¿Se han considerado varios estilos arquitectónicos diferentes antes de la definición de la arquitectura resultante?			
3	¿Realizó la trazabilidad entre el Modelo de Casos de Uso y el Modelo de Diseño, identificando los objetos y subsistemas de diseño que intervienen en el caso de uso y sus relaciones (Diagrama de paquetes)?			
4	¿Se especifican en el documento las correspondencias entre los requerimientos y el Modelo de Diseño?			
5	¿La arquitectura seleccionada ha sido ejercitada en escenarios reales?			
6	¿Realizó un diagrama mostrando la descomposición del Modelo de Diseño en Subsistemas, sus interfaces y las dependencias entre ellos, mostrando para cada Subsistema la Capa de la Arquitectura en la que se encuentra?			
7	¿Realizó una breve descripción del Subsistema, incluyendo un diagrama conteniendo las clases y los subsistemas que lo componen y para cada uno de ellos una breve descripción?			
8	¿Identificó las Clases de Diseño claves del Sistema?			
9	¿Describió las clases del Diseño y los subsistemas que corresponden a la Arquitectura del Diseño de cada Caso de Uso?			
10	¿Realizó la trazabilidad entre las entidades de diseño del Modelo de Diseño y los componentes del Modelo de Implementación?			
11	¿Indicó para cada entidad de diseño qué elemento en el ambiente de implementación le corresponde, por ejemplo para cada subsistema el			

	módulo que lo implementa; para una funcionalidad importante cuál es el conjunto de objetos que la implementan, a qué módulo pertenece?			
12	¿Especificó cómo están organizados los componentes de acuerdo a la estructura y mecanismos de modularización disponibles en el ambiente de implementación y en el(los) lenguaje(s) de programación, y cuáles son las dependencias entre los distintos componentes?			
13	¿Muestra la correspondencia entre la Arquitectura del Software y la Arquitectura del Hardware?			
14	¿Se tienen en cuenta todas las propiedades de calidad que debe tener el sistema?			
15	¿Se ha alcanzado un grado adecuado de modularidad?			
16	¿Se ha diseñado para el cambio?			

11.6. Plan de gestión de la configuración.

Nº	Pregunta	Sí	No	NA
1	¿Los productos de software a controlar están identificados claramente en el plan?			
2	¿Existe una regla para nombrar a cada producto de software?			
3	¿Se especifica cuándo se crea una nueva línea base?			
4	¿Se especifica qué productos son incluidos en la nueva línea base?			
5	¿Se especifica quienes son responsables por los productos incluidos en la línea base?			
6	¿Existe una regla para identificar las líneas base (versión)?			
7	¿Existe un procedimiento de control de cambios bien definido (qué debe hacer un integrante del grupo cuando quiere modificar cualquier producto presente en la línea base)?			
8	¿Se especifica en el plan si se reportará el estado de implementación de los cambios permitidos?			
9	¿El plan asigna a cada actividad SCM un responsable?			
10	¿El plan describe qué herramientas son utilizadas para implementar las actividades SCM?			

11.7. Plan del proyecto.

Nº	Pregunta	Sí	No	NA
1	¿Las dependencias entre las actividades están claramente identificadas?			
2	¿Se ha calculado el camino crítico del proyecto?			
3	¿El camino crítico del proyecto es revisado regularmente en busca de cambios?			
4	¿Se han realizado planes de contingencia para manejar retrasos de actividades específicas?			
5	¿Se utiliza el análisis de valor ganado para calcular el progreso?			
6	¿Se han definido mecanismos para rastrear los errores durante el proyecto?			

11.8. Plan de verificación y validación.

Nº	Pregunta	Sí	No	NA
1	¿Todos los métodos de Verificación y Validación aseguran completitud y compatibilidad de los requerimientos funcionales y no funcionales?			
2	¿La configuración del Software a ser testeado, como así también los elementos de Software y Hardware de soporte están definidos y se adecuan a los requerimientos?			
3	¿Las pruebas están definidas, son completas y son consistentes con los requerimientos?			
4	¿Los casos y procedimientos de prueba tienen instrucciones definidas, claras y concisas?			
5	¿La agenda del Plan de Verificación y Validación está definida, indicando que prueba se harán a que, cuándo y por qué?			

11.9. Documentación general a entregar.

Nº	Pregunta	Sí	No	NA
1	¿Se verificó que el documento siga el formato estándar?			

2	¿Todas las unidades de medida están definidas?			
3	¿Todas las figuras, tablas, y diagramas tiene referencias cruzadas?			
4	¿Están todas las figuras, tablas y diagramas necesarios?			
5	¿Todas las figuras, tablas y diagramas están rotulados?			
6	¿Hay un índice?			