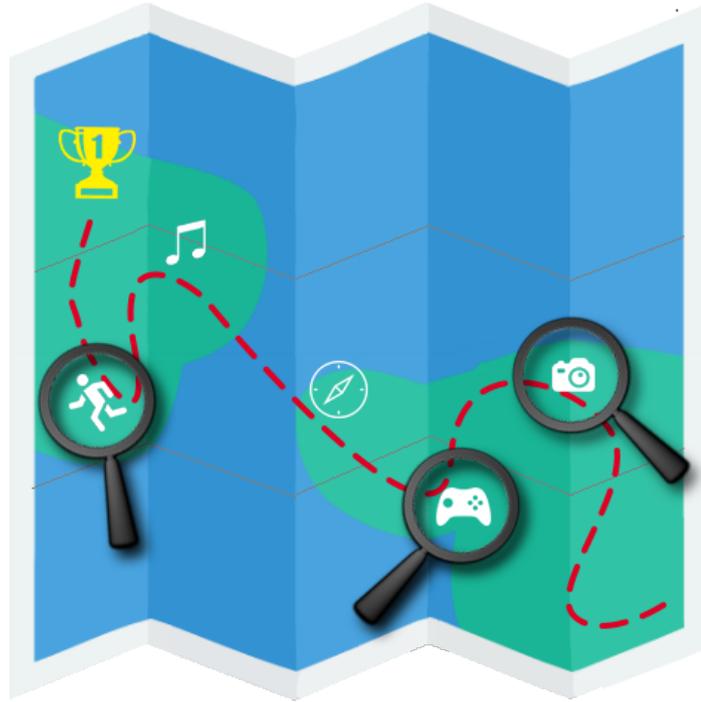


BeatIt!

Plan de SQA

Versión 1.2.2



Historia de revisiones

Fecha	Versión	Descripción	Autor
18/08/2014	1.1.1	Creación del documento	Pablo Olivera
29/08/2014	1.1.2	Actualización del documento	Pablo Olivera
30/08/2014	1.1.3	Revisión SQA	Emiliano Vázquez
10/09/2014	1.2.1	Actualización del documento	Pablo Olivera
14/09/2014	1.2.2	Revisión SQA	Emiliano Vázquez

Contenido

[Propósito](#)

[Referencias](#)

[Gestión](#)

[Organización](#)

[Actividades](#)

[Ciclo de vida del software cubierto por el Plan](#)

[Actividades de calidad a realizarse](#)

[Revisar cada producto](#)

[Revisar el ajuste al proceso](#)

[Realizar Revisión Técnica Formal \(RTF\)](#)

[Asegurar que las desviaciones son documentadas](#)

[Relaciones entre las actividades de SQA y la planificación](#)

[Responsables](#)

[Documentación](#)

[Propósito](#)

[Documentación mínima requerida](#)

[Especificación de requerimientos del software](#)

[Descripción del diseño del software](#)

[Plan de Verificación & Validación](#)

[Reportes de Verificación & Validación](#)

[Documentación de usuario](#)

[Plan de Gestión de configuración](#)

[Otros documentos](#)

[Estándares, prácticas, convenciones y métricas](#)

[Estándar de documentación](#)

[Estándar de verificación y prácticas](#)

[Otros Estándares](#)

[Revisiones y auditorías](#)

[Objetivo](#)

[Requerimientos mínimos](#)

[Revisión de requerimientos](#)

[Revisión de diseño preliminar](#)

[Revisión de diseño crítico](#)

[Revisión del Plan de Verificación & Validación](#)

[Auditoría funcional](#)

[Auditoría física](#)

[Auditorías internas al proceso](#)

[Revisiones de gestión](#)

[Revisión del Plan de gestión de configuración](#)

[Revisión Post Mortem](#)

[Agenda](#)

[Otras revisiones](#)

[Revisión de documentación de usuario](#)

[Verificación](#)

[Reporte de problemas y acciones correctivas](#)

[Herramientas, técnicas y metodologías](#)

[Gestión de riesgos](#)

[Apéndices](#)

[Checklists](#)

[Revisión de requerimientos](#)

[Revisión del modelo de casos de usos](#)

[Revisión de las Pautas para la Interfaz de Usuario](#)

[Revisión de la Interfaz de Usuario](#)

[Revisión de la Arquitectura](#)

[Revisión de Descripción de la Arquitectura](#)

[Revisión del Modelo de Diseño de los Casos de Uso](#)

[Revisión para Diseño Orientado a Objetos](#)

[Revisión de la Gestión de Configuración del Software](#)

[Revisión del Plan de Gestión de Configuración del Software](#)

[Revisión de la Implementación](#)

[Revisión del Plan de Proyecto](#)

[Revisión de la Estimaciones y Mediciones](#)

[Revisión de la Documento de Riesgos](#)

[Revisión del Plan de Verificación y Validación](#)

[Revisión de la Documentación](#)

1. Propósito

El propósito de este documento es especificar las pautas a seguir durante el proceso de desarrollo con el objetivo de asegurar la calidad de los productos elaborados.

Se definirán propiedades de calidad y deberá asegurarse que se cumplan, además se realizará un seguimiento y control del sistema en desarrollo, del proceso y de los procedimientos seguidos.

El objetivo es el desarrollo de un juego para smartphones con plataformas

Windows Phone y Android que permita al usuario cumplir 10 desafíos al estilo "Scavenger Hunt".

El plan cubre el ciclo de desarrollo, dividido en cuatro fases, inicial, elaboración, construcción y transición. El plan no cubre el ciclo de mantenimiento.

Será necesario del compromiso por la calidad de todos los integrantes del equipo para que este plan cumpla su objetivo.

2. Referencias

- [1] ANSI/IEEE Std 730.1-1989, IEEE Standard for Software Quality Assurance Plans.
- [2] Modelo de proceso para desarrollo de software - [MUM](#)
- [3] Documento de especificación de requerimientos.
- [4] Memoria organizacional

3. Gestión

En esta sección se detalla la organización, actividades y responsables para el aseguramiento de la calidad.

3.1. Organización

Las líneas de trabajo dentro de la organización que tienen influencia y controlan la calidad del software se basan en el MUM [2] que agrupa las disciplinas en dos grupos.

1. Las disciplinas Básicas son las que involucran las actividades de ingeniería "tradicionales" de desarrollo de software:

- Requerimientos
- Diseño
- Implementación
- Verificación
- Implantación

2. Las disciplinas de Gestión están constituidas por actividades que brindan "soporte" a las Básicas y se realizan en forma paralela a estas en cada iteración, dividiéndose en:

- SCM
- SQA
- Gestión de Proyecto
- Comunicación

La Gestión de Calidad estará a cargo del responsable y del asistente de SQA.

Además se relacionará con otras disciplinas de la siguiente forma:

- La disciplina Requerimientos hace partícipe al Responsable de SQA de las reuniones de relevamiento de requerimientos para que éste tenga conocimiento del Alcance del sistema que se quiere desarrollar y de las propiedades de calidad que el Cliente requiere que se cumplan.
- La disciplina Gestión de Proyecto informa de las actividades que se realizarán y los resultados obtenidos referentes a la Gestión de Proyecto.
- La disciplina Verificación informa de las actividades que se realizarán y los resultados obtenidos referentes a la Verificación.
- La disciplina Gestión de Configuración y Control de Cambios informa de las actividades que se realizarán y los resultados obtenidos referentes a la Gestión de Configuración y Control de Cambios.

Distribución del equipo de trabajo:

Combinación de roles	Nombre
Administrador - Asistente de Verificación - Resp. de Comunicación	Raúl Speroni
Analista - Implementador	Alejandro Brusco, Martín Steglich
Responsable de SQA – Asistente de Verificación	Pablo Olivera
Analista - Diseñador de Interfaz de Usuario - Implementador	Martín Alayón, Martin Berguer
Responsable de Verificación - Asistente de SQA	Emiliano Vázquez
Arquitecto - Asistente de Verificación - Coordinador de Desarrollo	Gonzalo Javiel
Especialista Técnico - Implementador - Responsable de Integración	Cristian Bauza, Joaquin Velazquez, Luciana Martínez
Responsable de SCM - Especialista Técnico - Implementador	Felipe García

3.2. Actividades

3.2.1. Ciclo de vida del software cubierto por el Plan

Los ciclos de vida contemplados por el plan son los siguientes:

- Requerimientos
- Diseño
- Implementación
- Verificación
- Implantación

Todos los entregables del proyecto serán revisados por el responsable y asistente de SQA.

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.

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 concientes de desviaciones o discrepancias encontradas.

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.

3.2.4. 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 desarrollo 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 consientes 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 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 lo estipulado.

3.2.7. Relaciones entre las actividades de SQA y la planificación

La siguiente tabla detalla las tareas que se realizarán durante el proyecto para gestionar la calidad. Se incluye la semana en la cual se va a realizar la actividad. Las fechas son estimadas por lo que pueden variar durante el desarrollo del proyecto.

Actividad	Semana cuando se realiza
Q1 Identificar las propiedades de calidad	1,2,3,4
Q2 Plan de calidad	2,4
Q3 Evaluar y ajustar el plan de SQA	9,11
Q4 Revisión técnica formal	5,7,9,11
Q5 Revisar las entregas	1,2,3,4,5,6,7,8,9,10,11,12,13,14
Q6 Revisar el ajuste del proceso	3,5,7,9,11,13,14
Q7 Evaluar la calidad de los productos	3,5,7,9,11,13,14
Q8 Realizar el informe final de SQA	14

3.3. Responsables

El seguimiento, control de calidad y las indicaciones de acciones correctivas estarán a cargo del responsable de SQA y del asistente de SQA.

Se definieron además los siguientes responsables por área.

Rol	Nombre
Requerimientos (Analistas)	Alejandro Brusco
Diseño	Gonzalo Javiel
Verificación	Emiliano Vázquez
Implantación	Joaquin Velazquez
Gestión de Configuración	Felipe García
Gestión de Proyecto	Raúl Speroni
Gestión de Calidad	Pablo Olivera

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.

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 este haya sufrido a lo largo del proyecto, que cubra aquellos aspectos que se haya acordado detallar con el cliente.

La especificación debe:

- Ser completa :
 - a. Externa, respecto al alcance acordado.
 - b. 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
- c. recuperabilidad

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.

Se pone especial énfasis en los siguientes atributos:

1. Usabilidad: El juego deberá tener un bajo costo de aprendizaje, comprensión y operatividad. Debe cumplir que contenga una navegación simple, intuitiva y dinámica.
2. Eficiencia: Es fundamental tener una buena performance, el tiempo de respuesta es muy importante para la experiencia de uso en el juego.

4.2.2. 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:
 - a. Todo elemento del diseño debe contribuir a algún requerimiento
 - b. 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

4.2.3. Plan de Verificación & Validación

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

- La verificación de que:
 - a. 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.
 - b. los requerimientos descritos en el documento de requerimientos son implementados en el diseño expresado en el documento de diseño.

c. el diseño expresado en el documento de diseño esta implementado en código.

- Validar que el código, cuando es ejecutado, se adecua a los requerimientos expresados en el documento de requerimientos.

4.2.4. Reportes de Verificación & Validación

Estos documentos deben especificar los resultados de la ejecución de los procesos descritos en el Plan de V & V.

4.2.5. Documentación de usuario

Dado que la usabilidad deberá ser uno de los principales atributos, la documentación será mínima, solo se incluirá un breve tutorial en los desafíos que se considere necesario.

4.2.6. Plan de Gestión de configuración

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

4.3. Otros documentos

Otros documentos que inciden en la calidad del producto a desarrollar:

- Plan de desarrollo
- Plan de proyecto
- Plan de iteración
- Pautas para la interfaz de usuario
- Documento de riesgos
- Gestión de cambios
- Documento de validación con el cliente
- Estimaciones y mediciones
- Alcance del sistema
- Descripción de la arquitectura
- Modelo de diseño
- Modelo de casos de uso

5. Estándares, prácticas, convenciones y métricas

5.1. Estándar de documentación

Como estándares de documentación se definirán dos documentos:

- Estándar de documentación técnica y
- Estándar de documentación de usuario.

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.

En estas plantillas se definen:

- encabezado y pie de página.
- fuente y tamaño de fuente para estilo normal
- fuente y tamaño de fuente para los títulos a utilizar
- datos mínimos que se deben incluir: fecha, versión y responsables.

1. Las plantillas a utilizar son las sugeridas por el MUM, las cuales tienen definidos los siguientes formatos:
 - a. Carátula
 - i. Nombre del proyecto, título: verdana 18, negrita.
 - ii. Versión del documento: verdana 16, negrita.
 - iii. Tabla de la historia de revisiones: verdana 10.
 - b. Título nivel 1: verdana 11, negrita.
 - c. Títulos niveles 2 y 3: verdana 10, negrita.
 - d. Título nivel 4: verdana 10, cursiva.
 - e. Texto normal: verdana 10.
 - f. Los títulos siguen una lista jerárquica (nivel 1: 1, nivel 2: 1.1, nivel 3: 1.1.1).
 - g. Pie de página.
2. Los documentos generados se guardan en la carpeta PIS/Documentos distribuidos en subcarpetas de acuerdo a la [disciplina](#).
3. Al hacer cambios en los documentos:
 - a. actualizar la tabla de la historia de revisiones.
 - b. actualizar el índice cuando corresponda.
4. Para editar las plantillas se utilizará las herramientas que provee google.
5. Cuando un documento que corresponde a la entrega semanal no va a ser entregado, notificar al responsable de SQA indicando una breve descripción de la causa y una fecha estimada para la entrega.
6. Plazo para entregar los documentos para SQA: sábados 24:00hs.
7. Los responsables de cada área serán los encargados de notificar al responsable de SQA cuando el documento esté terminado y listo para ser revisado.

El responsable y el asistente de SQA revisarán los documentos entregables con el fin de verificar que se cumpla lo mencionado en los puntos anteriores.

En caso de tener dudas sobre el contenido se contactará a el/los responsable/s. Luego de pasar por la revisión se guardan en formato pdf y se notificará al administrador que será el encargado de realizar la entrega.

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

Pendiente, se definirá en próximas versiones del documento.

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 contra el documento de diseño, especificaciones de interfaz, implementaciones de diseño contra requerimientos funcionales y requerimientos funcionales contra 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

A continuación se detalla una agenda tentativa de las revisiones y auditorías indicando la semana que se llevará a cabo tal actividad. La misma está sujeta a la disponibilidad de los recursos humanos y las prioridades del equipo.

Auditoría funcional y física están sujetas a las fechas de liberaciones parciales.

Revisión	Semanas
Revisión de requerimientos	3,4,5,6,7,8
Revisión de diseño preliminar	4,5,6

Revisión de diseño crítico	7,8,9,10
Revisión del Plan de V&V	5,7,10
Auditoría funcional	8,12
Auditoría física	8,12
Auditorías internas al proceso	4,6,8,10,12
Revisiones de gestión	4,6,8,10,12
Revisión del Plan SCM	4,6,8,10,12
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

Todo lo referente a V&V se encuentra en el Plan de Verificación y Validación.

8. Reporte de problemas y acciones correctivas

El responsable de SQA y el asistente de SQA revisarán los productos, en caso de detectar errores menores serán corregidos y para los potenciales defectos que se consideren mayores se notificará al responsable del área (Ver pto. 3.3.) para que evalúe si es un error, en tal caso deberá ser corregido, de lo contrario se descartará la observación.

Lo referente a los documentos ver 5.1. donde se detalla la metodología a seguir.

9. Herramientas, técnicas y metodologías

Para realizar las revisiones de producto se utilizará los checklists definidos en el apéndice.

10. Gestión de riesgos

Los riesgos identificados, la estrategia de mitigación, monitoreo y plan de contingencia a ser llevados a cabo, serán descritos en el Documento de Gestión de Riesgos (GPDRIG6).

11. Apéndices

11.1. Checklists

11.1.1. Revisión 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)			
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 descriptas 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?			

11.1.2. Revisión del modelo de casos de usos

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?			

11.1.3. Revisión de las 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?			

11.1.4. Revisión de la Interfaz de Usuario

Nº	Pregunta	Si	No	NA
1	¿La interacción es intuitiva de usar?			
2	¿Es fácil de aprender?			
3	¿Es correcta la rapidez de respuesta a las acciones del usuario?			
4	¿Se usó el color de forma efectiva?			
5	¿El usuario recibe un correcto feedback de la aplicación?			
6	¿Se usó el color de forma efectiva?			

11.1.5. Revisión de la Arquitectura

Nº	Pregunta	Si	No	NA
1	¿La arquitectura esta 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 descriptos 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 esta 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?			

11.1.6. Revisión de Descripción de la Arquitectura

Nº	Pregunta	Si	No	NA
----	----------	----	----	----

1	¿Identificó los casos de uso y actores más significantes 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?			

11.1.7. Revisión del Modelo de Diseño de los Casos de Uso

Nº	Pregunta	Si	No	NA
1	¿Especificó los casos de uso, mediante subsistemas u objetos de diseño que interactúan, y determinó las operaciones e interfaces de los distintos subsistemas u objetos de diseño?			
2	¿El modelo de diseño permite cumplir con todos los requerimientos?			
3	¿Para cada objeto se definió parámetros, reglas, operaciones que realiza, requerimientos de implementación, interfaz de usuario y relaciones con otros objetos?			
4	¿Esta especificación se realiza mediante Diagrama de Paquetes, Diseño de flujo de eventos, Diagramas de interacción y Requerimientos especiales o de implementación?			

5	¿Identificó los objetos y subsistemas de diseño que intervienen en el caso de uso y sus relaciones?			
6	¿Realizó un diagrama de interacción para representar la realización de cada Caso de Uso?			
7	¿Se cuenta con una descripción escrita en términos de objetos y subsistemas de diseño, que explica y complementa el diagrama de interacción y sus niveles?			
8	¿Describió los requerimientos no funcionales en la realización de un caso de uso?			
9	¿Para cada subsistema específico: declaro que hace, cual es el propósito del mismo y cuáles son sus entradas y que salidas produce?			
10	¿Para cada subsistema identificó los subsistemas subordinados al mismo?			
11	¿Se identificó para cada subsistema la dependencia con otros subsistemas; indicando las condiciones para que se realice la iteración y las características de la iteración como ser pasaje de parámetros, mensajes, datos compartidos etc.?			
12	¿Identificó todos los recursos externos al diseño, que necesita el subsistema para realizar su función (dispositivos físicos, servicios de software, recursos de procesamiento etc.)?			
13	¿Realizó un diseño detallado del comportamiento de la interfaz de usuario a partir de la especificación de requerimientos y el mismo esta de acuerdo con el entorno tecnológico?			
14	¿Definió la estructura de datos que utilizará el sistema, a partir de los requisitos funcionales y no funcionales establecidos para el sistema y las particularidades del entorno tecnológico, que consiga una mayor eficiencia en el tratamiento de los datos?			
15	¿Para la creación del Modelo de datos identificó las principales entidades de datos y los datos por los que están compuestas a partir de la especificación de requerimientos y las dependencias entre dichas entidades?			
16	¿Especificó el modelo de distribución de datos indicando la ubicación de los manejadores de bases de datos o sistemas de archivos, así como los distintos elementos de la estructura física de datos (base de datos, tablas, índices)?			

11.1.8. Revisión para Diseño Orientado a Objetos

Nº	Pregunta	Si	No	NA
1	¿Se han definido subsistemas como parte de la representación del diseño?			
2	¿Hay relativamente pocos subsistemas?			
3	¿Los límites de los subsistemas están bien definidos?			
4	¿Se ha establecido una bien definida interfaz para cada subsistema?			
5	¿Las clases que pueblan un subsistema tienen sentido dentro del contexto de los requerimientos funcionales/datos definidos para el subsistema?			
6	¿Se han establecido clases de comunicación para la comunicación entre subsistemas?			
7	¿Los subsistemas están diseñados para tener una interacción mínima con los demás?			
8	¿Los subsistemas están diseñados de tal forma que puedan ser usados en otros sistemas?			

9	¿Se han definido clases de gestión de tareas?			
10	¿Se han definido clases de gestión de datos?			
11	¿Se han definido clases de interfaz de usuario?			
12	¿El diseño diferencia entre los componentes del dominio del problema, de la interfaz de usuario, de gestión de tareas, y de gestión de datos?			
13	¿ Toda la comunicación dentro del subsistema está claramente detallada en un diagrama de colaboración?			
14	¿Se utiliza un protocolo de definición para cada objeto?			
15	¿El protocolo de descripción describe claramente la estructura de los mensajes para cada clase/objeto?			
16	¿Se han utilizado patrones de diseño reconocidos como parte del diseño?			
17	¿El programa será fácil de mantener?			
18	¿El diseño tiene en cuenta futuras extensiones al programa?			
19	¿Todas las partes del diseño son estrictamente necesarias?			

11.1.9. Revisión de la Gestión de Configuración del Software

N°	Pregunta	Sí	No	NA
1	¿Todos los integrantes del grupo están siguiendo los procedimientos definidos en el plan SCM?			
2	¿Está definida la línea base de la fase / iteración?			
3	¿Todos los ítems que indica el documento de la línea base están en el directorio correspondiente a esa línea base?			
4	¿Coinciden las versiones de cada ítem?			
5	¿Todos los ítems dentro del directorio de la línea base se encuentran en el documento correspondiente?			

11.1.10. Revisión del Plan de Gestión de Configuración del Software

N°	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?			

11.1.11. Revisión de la Implementación

N°	Pregunta	Sí	No	NA
1	Conociendo la nueva reestructuración de grupos y funcionamiento dentro de ellos, ¿cree que le es más fácil a los implementadores lograr implementar habiendo participado en el diseño? ¿Han tenido algún problema de no entendimiento del diseño? ¿Se respeta el diseño a la hora de implementar, o se le hacen modificaciones sobre el avance?			
2	¿Realizan integración?			
3	¿Realizan algún tipo de verificación?			
4	¿La documentación técnica la hacen a medida que implementan o luego? ¿La realiza el implementador que construyó la clase o se encarga otra persona?			
5	¿Han tenido algún problema en seguir las normas de implementación?			
6	¿Realizan mediciones sobre la cantidad de líneas de código y tiempo de implementación?			
7	¿Implementan de a pares? ¿Realizan revisión por pares? ¿Han obtenido algún buen resultado de esto último?			
8	¿Hasta cuándo se tiene pensado implementar? ¿La implementación cubre el alcance previsto para la fase?			
9	¿Existe un estándar de implementación?			
10	¿El estándar de implementación fue aprobado por todos?			
11	¿Existe un estándar de documentación técnica?			
12	¿El estándar de documentación técnica fue aprobado por todos?			
13	¿Se identificaron todas las herramientas a utilizar?			
14	¿Todos los integrantes del grupo poseen las herramientas que deben utilizar?			
15	¿Todos los integrantes del grupo tienen un mínimo conocimiento acerca de las herramientas que deben utilizar?			
16	¿Se midió el esfuerzo y tamaño de lo implementado (en LOCs)?			

11.1.12. Revisión del Plan de Proyecto

N°	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?			

11.1.13. Revisión de la Estimaciones y Mediciones

N°	Pregunta	Sí	No	NA
1	¿Se usó algún mecanismo formal (ej: casos de uso) para obtener los requerimientos básicos antes de realizar la estimación?			
2	¿Se estimó el tamaño del producto final y sus alcances			

	intermedios?			
3	¿La estimación anterior se realizó utilizando varias técnicas?			
4	¿Las estimaciones de tamaño fueron generadas por más de una persona?			
5	¿Las personas que realizan la estimación tienen experiencia en el dominio de aplicación?			
6	¿La descomposición en funciones se utilizó para generar las estimaciones de tamaño?			
7	¿El alcance del proyecto y avance por fases es consistente con las estimaciones?			
8	¿Se realizó un análisis de factibilidad para las funciones más importantes?			
9	¿Se consideró el impacto de componentes reusables?			

11.1.14. Revisión de la Documento de Riesgos

Nº	Pregunta	Sí	No	NA
1	¿Se identificaron riesgos en el proyecto?			
2	¿Los riesgos están priorizados?			
3	¿Se identificaron medidas para prevenirlos?			
4	¿Se identificaron medidas para minimizar su impacto?			
5	¿Se tomaron medidas para prevenir los riesgos más importantes?			
6	¿Se tomaron medidas para minimizar el impacto de los riesgos que ocurrieron?			

11.1.15. Revisión del 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?			

11.1.16. Revisión de la Documentación

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?			
---	-----------------	--	--	--