

# GVA

## Grupo 2

## Plan de SQA

## Versión 1.1

### Historia de revisiones

Fecha	Versión	Descripción	Autor
29/08/14	1.0	Creación del documento	Alejandro Casco
08/09/2014	1.1	Modificación atributos de calidad	Alejandro Casco

# Contenido

<b>1.PROPÓSITO.....</b>	<b>4</b>
<b>2.REFERENCIAS.....</b>	<b>4</b>
<b>3.GESTIÓN.....</b>	<b>4</b>
3.1.ORGANIZACIÓN.....	4
3.2.ACTIVIDADES.....	6
3.2.1.Ciclo de vida del software cubierto por el Plan.....	6
3.2.2.Actividades de calidad a realizarse.....	6
3.2.3.Revisar cada producto.....	6
3.2.4.Revisar el ajuste al proceso.....	6
3.2.5.Realizar Revisión Técnica Formal (RTF).....	7
3.2.6.Asegurar que las desviaciones son documentadas.....	7
3.2.7.Relaciones entre las actividades de SQA y la planificación.....	7
3.3.RESPONSABLES.....	7
<b>4.DOCUMENTACIÓN.....</b>	<b>8</b>
4.1.PROPÓSITO.....	8
4.2.DOCUMENTACIÓN MÍNIMA REQUERIDA.....	8
4.2.1.Especificación de requisitos del software.....	8
4.2.2.Descripción del diseño del software.....	9
4.2.3.Plan de Verificación & Validación.....	9
4.2.4.Reportes de Verificación & Validación.....	9
4.2.5.Documentación de usuario.....	9
4.2.6.Plan de Gestión de configuración.....	10
4.3.OTROS DOCUMENTOS.....	10
<b>5.ESTÁNDARES, PRÁCTICAS, CONVENCIONES Y MÉTRICAS.....</b>	<b>10</b>
5.1.ESTÁNDAR DE DOCUMENTACIÓN.....	10
5.2.ESTÁNDAR DE VERIFICACIÓN Y PRÁCTICAS.....	10
5.3.ESTÁNDARES DE IMPLEMENTACIÓN.....	11
<b>6.REVISIONES Y AUDITORÍAS.....</b>	<b>11</b>
6.1.OBJETIVO.....	11
6.2.REQUERIMIENTOS MÍNIMOS.....	11
6.2.1.Revisión de requisitos.....	11
6.2.2.Revisión de diseño preliminar.....	11
6.2.3.Revisión de diseño crítico.....	11
6.2.4.Revisión del Plan de Verificación & Validación.....	11
6.2.5.Auditoría funcional.....	11
6.2.6.Auditoría física.....	11
6.2.7.Auditorías internas al proceso.....	11
6.2.8.Revisiones de gestión.....	11
6.2.9.Revisión del Plan de gestión de configuración.....	11
6.2.10.Revisión Post Mortem.....	12
6.2.11.Agenda.....	12
6.3.OTRAS REVISIONES.....	12
6.3.1.Revisión de documentación de usuario.....	12
<b>7.VERIFICACIÓN.....</b>	<b>12</b>
<b>8.REPORTE DE PROBLEMAS Y ACCIONES CORRECTIVAS.....</b>	<b>12</b>
<b>9.HERRAMIENTAS, TÉCNICAS Y METODOLOGÍAS.....</b>	<b>12</b>
9.1.1.Revisión del Modelo de Casos de Uso.....	13

9.1.2.Revisión de las Pautas para la Interfaz de Usuario.....	14
9.1.3.Revisión de la Interfaz de Usuario.....	14
9.1.4.Revisión de la Arquitectura.....	15
9.1.5. Revisión de Descripción la Arquitectura.....	16
9.1.6.Revisión del Modelo de Diseño de los Casos de Uso.....	17
9.1.7.Revisión para Diseño Orientado a Objetos.....	18
9.1.8.Revisión de la Gestión de Configuración del Software.....	19
9.1.9.Revisión del Plan de Gestión de Configuración del Software.....	19
9.1.10.Revisión de la Implementación.....	20
9.1.1.Revisión del Plan de Proyecto.....	21
9.1.2.Revisión de la Estimaciones y Mediciones.....	21
9.1.3.Revisión de la Documento de Riesgos.....	21
9.1.4.Revisión del Plan de Verificación y Validación.....	22
9.1.5.Revisión de la Documentación.....	22
<b>10.GESTIÓN DE RIESGOS.....</b>	<b>22</b>

## 1. Propósito

El propósito del Plan de Calidad es definir y describir las pautas a seguir para asegurar la calidad del proyecto a desarrollar, así como la planificación del seguimiento de las mismas. Para ello se definirán mecanismos para monitorear la calidad tanto del proceso como del producto.

El proyecto a desarrollar consiste en una aplicación web para la empresa SONDA Uruguay S.A. cuyo objetivo es manejar el registro de componentes tanto de hardware como de software, los cuales de aquí en más serán denominados "Activos".

El objetivo de la aplicación es mantener un registro de toda la actividad de cada activo, para poder conocer datos de interés del mismo (ubicación, estado, etc), así como poder gestionar su trazabilidad.

El alcance de este plan solo cubre la parte del ciclo de vida correspondiente al desarrollo del software (evaluación, elaboración, construcción, etc) pero no cubre la parte del ciclo de vida correspondiente al mantenimiento.

Para que el plan sea efectivamente llevado a cabo, se requerirá del compromiso de todos los integrantes del grupo, para lo cual se intentará que las pautas aquí descritas sean adquiridas en la práctica diaria por parte de cada uno.

## 2. Referencias

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

[1] Plan de Verificación (VRPVVG2v2.0(Plan de VyV))

[1] Plan de SCM (SCMPLAG2V1.4\_(Plan de Config))

## 3. Gestión

### 3.1.Organización

Dentro de la organización del grupo, se establecen líneas de trabajo clasificadas en disciplinas básicas y disciplinas de gestión.

Las disciplinas básicas incluyen las tareas relacionadas directamente con el desarrollo del software, las cuales se pueden dividir en:

- Análisis
- Requisitos
- Diseño
- Implementación
- Verificación

Las disciplinas de gestión son las que ofrecen soporte a las disciplinas básicas y ayudan a que las mismas sean llevadas a cabo de forma

organizada, siguiendo los atributos de calidad especificados para el desarrollo del software. Dentro de estas disciplinas encontramos:

- Gestión del proyecto
- Comunicación
- SCM
- SQA

Las actividades de gestión de la calidad son llevadas adelante por el responsable de SQA y sus asistentes.

La división de roles en el grupo es la siguiente:

Integrante	Rol
Marina Acosta	Administrador – Asist. Verificación – Resp. Comunicación
Linnete Grill	Analista – Documentador de Usuario – Asist. Verificación
Malvina Betarte	Analista – Diseño UI – Implementador
Facundo Agüero	Analista – Implementador
Martin Santagata	Analista – Implementador
Alejandro Casco	Responsable SQA – Asist. Verificación
Christopher Quincke	Responsable Verificación – Asistente SQA
Diego Dastugue	Arquitecto – Asist. Verificación – Coordinador de Desarrollo
Martin Tambucho	Especialista Técnico – Implementador – Responsable de Integración
Emiliano González	Especialista Técnico – Implementador – Responsable de Integración
Nicolás Fiumarelli	Especialista Técnico – Implementador – Responsable de Integración
Nicolás Greising	Responsable de SCM

La disciplina Gestión de Calidad se relaciona 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.

Con el resto de las disciplinas tendrá relación en la medida que evalúe los procedimientos seguidos para la realización de actividades y la calidad de los elementos o productos generados por cada disciplina.

## **3.2.Actividades**

### **3.2.1. Ciclo de vida del software cubierto por el Plan**

Las etapas del ciclo de vida que cubre el Plan son:

- Etapa de requisitos y análisis.
- Etapa de diseño.
- Etapa de implementación.
- Etapa de verificación.

Todos los entregables de este proyecto deberán ser verificados 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.

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 concientes 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 concientes 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**

<b>Actividad</b>	<b>Semana cuando se realiza</b>
Identificar atributos de calidad	2,3
Plan de calidad	2,4
Evaluar y ajustar el plan de calidad	9,11
Revisión técnica formal	5,7,9,11
Revisar las entregas	Todas las semanas
Revisar el ajuste al proceso	3,5,7,9,11,13,14
Evaluar la calidad de los productos	3,5,7,9,11,13,14
Realizar el informe final de calidad	14

## **3.3.Responsables**

Los responsables de las actividades de seguimiento y aseguramiento del plan de calidad son responsabilidad del Responsable de SQA y del Asistente 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 requisitos.

#### 4.2.1. Especificación de requisitos del software

El documento de especificación de requisitos deberá describir, de forma clara y precisa, cada uno de los requisitos 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 requisito.
- Estar acompañada de un detalle de los procedimientos adecuados para verificar si el producto cumple o no con los requisitos.
- Incluir requisitos de calidad del producto a construir.

Los requisitos 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. Seguridad de los datos

#### **Confiabilidad**

- a. Tolerancia a fallas
- b. Recuperabilidad (Ambiente Transaccional)
- c. Integridad de datos (consistencia en BD)

#### **Usabilidad**

- a. Comprensible
- b. Aprendible
- c. Operable



d. Atractivo

### **Portabilidad**

a. Adaptable (Base de datos intercambiable)

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

#### **4.2.2. Descripción del diseño del software**

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

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 requisitos a incorporar:
  - a. Todo elemento del diseño debe contribuir a algún requisito
  - b. La implementación de todo requisito 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 requisitos descritos en el documento de requisitos 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 requisitos descritos en el documento de requisitos 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 requisitos expresados en el documento de requisitos.

#### **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**

La documentación de usuario debe especificar y describir los datos y entradas de control requeridos, así como la secuencia de entradas, opciones,

limitaciones de programa y otros elementos necesarios para la ejecución exitosa del software.

Todos los errores deben ser identificados y las acciones correctivas descritas.

Como resultado del proyecto el cliente obtendrá una documentación para el usuario de acuerdo a los requisitos específicos del proyecto.

#### **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, y registro y reporte del estado de los cambios implementados.

### **4.3.Otros documentos**

#### **4.3.1. Plan de proyecto**

Se debe revisar este plan para tomar acciones correctivas en caso que el trabajo real tenga una desviación con respecto al plan.

## **5. Estándares, prácticas, convenciones y métricas**

### **5.1.Estándar de documentación**

Como estándares de documentación se definirán dos documentos:

- Estándar de documentación técnica
- 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:

- Pie de página (Nombre del documento – Número y cantidad de páginas, Verdana 10)
- Estilo normal (Verdana 10)
- Títulos (Verdana 14 negrita)
- Subtítulos primer nivel (Verdana 13 negrita)
- Subtítulos segundo nivel Verdana 10 negrita)
- Caratula con nombre de proyecto, número de semana, número de grupo, nombre del documento, tabla de historial de revisiones.

### **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.Estándares de implementación**

Se utilizan las prácticas definidas en el documento "Estándar de Implementación" (IMEIG2V1.0.pdf). Las mismas son basadas en las convenciones de código de Microsoft.

## **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 requisitos**

Esta revisión se realiza para asegurar que se cumplió con los requisitos 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 requisitos.

#### **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 requisitos especificados en el documento de requisitos 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 requisitos funcionales, requisitos 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**

Se pasa a detallar una agenda tentativa, en la cual se especifican las semanas en que se realizarán las revisiones y auditorias (sujeta a disponibilidad de recursos del equipo y prioridades del equipo)

Revisión	Semanas
Revisión de Requerimientos	3, 4, 5, 6, 7
Revisión de Diseño preliminar	4, 5, 6
Revisión de Diseño Crítico	7, 8, 9, 10
Revisión de Plan de V&V	5, 7, 10
Auditoria funcional	12
Auditoria física	12
Auditorías internas al proceso	4, 6, 8, 10, 12
Revisión de gestión	4, 6, 8, 10, 12
Revisión del plan de gestión	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**

Ver Plan de Verificación y Validación (VRPVVG2v2.0(Plan de VyV).pdf)

## **8. Reporte de problemas y acciones correctivas**

Cada vez que se finalice un producto, el responsable de SQA realizará la revisión del mismo, procediendo a informar al responsable de dicho producto en caso de encontrar errores mayores. Según el momento de la semana en que se entregue el producto, se podrá corregir el mismo o incluir la corrección en la semana siguiente, realizando el seguimiento pertinente.

## **9. Herramientas, técnicas y metodologías**

Para la revisión del código, se analizará el mismo para chequear que cumple con los estándares definidos en el documento IMEIG2V1.0.pdf

Para la documentación de cada entrega semanal, se validará que cumpla con los estándares definidos en este documento, revisando correctitud, completitud, y que cumpla con el formato definido.

Para revisar los atributos de calidad de Funcionalidad, Confiabilidad y Portabilidad, se tomarán como válidos los resultados de las pruebas realizadas por el Responsable de Verificación.

Para las revisiones del producto, se utilizarán las siguientes checklists, definidas en MUM:

#### 9.1.1. Revisión del Modelo de Casos de Uso

Nº	Pregunta
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?

### 9.1.2. Revisión de las Pautas para la Interfaz de Usuario

Nº	Pregunta
1	¿Definió los formatos de los distintos tipos de pantallas?
2	¿Definió los controles y elementos de diseño de interfaz de pantalla sus aracterísticas, 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?

### 9.1.3. Revisión de la Interfaz de Usuario

Nº	Pregunta
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?

#### 9.1.4. Revisión de la Arquitectura

Nº	Pregunta
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 descriptas 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 descripto 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?

### 9.1.5. Revisión de Descripción la Arquitectura

Nº	Pregunta
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?



### 9.1.6. Revisión del Modelo de Diseño de los Casos de Uso

Nº	Pregunta
1	¿Especifiqué los casos de uso, mediante subsistemas u objetos de diseño que interactúan, y determiné 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	¿Identifiqué 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 identifiqué 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	¿Identifiqué 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 identifiqué 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	¿Especifiqué 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)?

### 9.1.7. Revisión para Diseño Orientado a Objetos

Nº	Pregunta
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?

### 9.1.8. Revisión de la Gestión de Configuración del Software

Nº	Pregunta
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?

### 9.1.9. Revisión del Plan de Gestión de Configuración del Software

Nº	Pregunta
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?

### 9.1.10. Revisión de la Implementación

Nº	Pregunta
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)?

### 9.1.1. Revisión del Plan de Proyecto

N°	Pregunta
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?

### 9.1.2. Revisión de la Estimaciones y Mediciones

N°	Pregunta
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?

### 9.1.3. Revisión de la Documento de Riesgos

N°	Pregunta
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?

#### 9.1.4. Revisión del Plan de Verificación y Validación

Nº	Pregunta
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 testado, 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?

#### 9.1.5. Revisión de la Documentación

Nº	Pregunta
1	¿Se verificó que el documento siga el formato estandar?
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?

## 10. Gestión de riesgos

Ver documento de Gestión de Riesgos (GPDRIG2V1.1.pdf)