

# **Sistema de Administración de Farmacias**

## **Plan de SQA**

### **Versión 2.0**

#### **Historia de revisiones**

Fecha	Versión	Descripción	Autor
29/08/2014	1.0	Realización del documento	Resp. SQA
14/09/2014	1.1	Actualización.	Resp. SQA
23/10/2014	1.2	Actualización.	Resp. SQA
20/11/2014	2.0	Cierre del Documento.	Resp. SQA

# ÍNDICE

<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. <i>Ciclo de vida del software cubierto por el Plan</i> .....	6
3.2.2. <i>Actividades de calidad a realizarse</i> .....	6
3.2.3. <i>Revisar cada producto</i> .....	6
3.2.4. <i>Revisar el ajuste al proceso</i> .....	6
3.2.5. <i>Realizar Revisión Técnica Formal (RTF)</i> .....	7
3.2.6. <i>Asegurar que las desviaciones son documentadas</i> .....	7
3.2.7. <i>Relaciones entre las actividades de SQA y la planificación</i> .....	7
3.3. RESPONSABLES .....	8
<b>4. DOCUMENTACIÓN.....</b>	<b>9</b>
4.1. PROPÓSITO .....	9
4.2. DOCUMENTACIÓN MÍNIMA REQUERIDA .....	9
4.2.1. <i>Especificación de requerimientos del software</i> .....	9
4.2.2. <i>Descripción del diseño del software</i> .....	10
4.2.3. <i>Plan de Verificación &amp; Validación</i> .....	11
4.2.4. <i>Reportes de Verificación &amp; Validación</i> .....	11
4.2.5. <i>Documentación de usuario</i> .....	11
4.2.6. <i>Plan de Gestión de configuración</i> .....	12
4.2.7. <i>Modelo de casos de uso</i> .....	12
4.2.8. <i>Pautas para la interfaz de usuario</i> .....	12
4.2.9. <i>Descripción de la arquitectura</i> .....	12
4.2.10. <i>Modelo de diseño</i> .....	12
4.2.11. <i>Criterios de Aceptación</i> .....	12
4.2.12. <i>Riesgos Encontrados</i> .....	12
4.3. OTROS DOCUMENTOS .....	13
4.4. ESTÁNDAR DE DOCUMENTACIÓN .....	14
4.5. ESTÁNDAR DE VERIFICACIÓN Y PRÁCTICAS .....	14
4.6. ESTÁNDAR DE USO DEL TECLADO .....	14
<b>5. REVISIONES Y AUDITORÍAS.....</b>	<b>15</b>
5.1. OBJETIVO .....	15
5.2. REQUERIMIENTOS MÍNIMOS .....	15
5.2.1. <i>Revisión de requerimientos</i> .....	15
5.2.2. <i>Revisión de diseño preliminar</i> .....	15
5.2.3. <i>Revisión de diseño crítico</i> .....	15
5.2.4. <i>Revisión del Plan de Verificación &amp; Validación</i> .....	15
5.2.5. <i>Auditoría funcional</i> .....	15
5.2.6. <i>Auditoría física</i> .....	15
5.2.7. <i>Auditorías internas al proceso</i> .....	15
5.2.8. <i>Revisiones de gestión</i> .....	15

5.2.9. <i>Revisión del Plan de gestión de configuración</i> .....	15
5.2.10. <i>Revisión Post Mortem</i> .....	16
5.2.11. <i>Agenda</i> .....	16
5.3. OTRAS REVISIONES .....	16
5.3.1. <i>Revisión de documentación de usuario</i> .....	16
<b>6. VERIFICACIÓN.....</b>	<b>17</b>
<b>7. REPORTE DE PROBLEMAS Y ACCIONES CORRECTIVAS.....</b>	<b>17</b>
<b>8. HERRAMIENTAS, TÉCNICAS Y METODOLOGÍAS.....</b>	<b>17</b>
<b>9. GESTIÓN DE RIESGOS .....</b>	<b>25</b>

# 1. Propósito

El objetivo del plan de calidad es definir la estrategia a seguir para intentar asegurar la calidad en el proceso de construcción del "Sistema de Administración de Farmacias".

Enfocándose tanto en el producto final como en la aplicación del proceso MUM para llegar a este.

Durante las 14 semanas de duración de este proyecto, se deberá desarrollar un sistema que permita la administración de farmacias, contando con módulos de facturación, manejo de stock y de clientes entre otros. Facilitando así la interacción y comunicación entre DUSA y estas.

Se definirán mecanismos para poder realizar seguimiento a la calidad, así como para detectar desviaciones y poder tomar medidas al respecto. De esta manera podremos corregir en forma temprana errores que serían costosos de corregir en etapas avanzadas del proyecto.

Este plan sólo cubre la parte del ciclo de vida correspondiente al desarrollo del software pero no cubre la parte del ciclo de vida correspondiente al mantenimiento.

# 2. Referencias

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

[2]Modelo de Proceso MUM - Proyecto de Ingeniería de Software.

[3]Memoria Organizacional

# 3. Gestión

## 3.1. Organización

El proceso MUM distingue las siguientes disciplinas:

- **Básicas:** Actividades "tradicionales" de desarrollo de software las cuales se realizan en cascada en cada iteración.
- **Gestión:** Conformadas por las actividades que dan soporte a las básicas. Estas se realizan en forma paralela a cada iteración.

Luego lineamientos de trabajo para estas disciplinas: **Requerimientos, Diseño, Gestión de Proyecto, Implementación, Verificación, Gestión de Configuración y Control de Cambios.**

Todos estos tienen plenamente identificadas las actividades a realizar y los roles involucrados.

	Líneas de trabajo	Roles
Básicas	Requerimientos	Analistas – Arquitecto
	Diseño	Arquitecto – Analistas
	Implementación	Especialistas Técnicos Implementadores

<b>Gestión</b>	Verificación	Resp. Verificación
	Implantación	Resp. SCM
	Gestión de Configuración (SCM)	Resp. SCM
	Gestión del proyecto	Administrador
	Gestión de la calidad	Resp. SQA

Todas estas actividades repercuten en mayor o menor medida en la calidad tanto del proceso como del producto por lo que es fundamental la participación de la Gestión de Calidad en todas estas. Ya sea la actividad misma como en la documentación generada.

ROL	INTEGRANTES
Administrador - Asistente de Verificación - Responsable de Comunicación	Gonzalo Capote
Analista – Documentador de Usuario – Asistente de Verificación	Santiago Ramírez
Analista - Implementador	Gonzalo Mercadante, Daniela Fagundez, Manuela Viola
Responsable de SQA – Asistente de Verificación	Alfonso Methol
Analista - Diseñador de Interfaz de Usuario - Implementador	Fernanda Toledo
Responsable de Verificación – Asistente de SQA	Juan Pablo Sierra
Arquitecto - Asistente de Verificación - Coordinador de Desarrollo	Guillermo López
Especialista Técnico - Implementador - Responsable de Integración	María José Rabaza.
Responsable de SCM - Especialista Técnico - Implementador	Santiago Nieves
Verificador	Estera Nocon

El objetivo principal es obtener un producto final de calidad. Esto no es posible si no se sigue esa línea durante todo el proceso, ya que los productos de cada etapa determinan la calidad de los de la siguiente, y así, hasta llegar al producto final.

## **3.2. Actividades**

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

El ciclo de vida del proyecto cubre desde la etapa de Relevamiento de Requisitos hasta la etapa de Liberación del producto. Es de mayor importancia la ejecución de revisiones de calidad para cada producto del proyecto en sus diferentes disciplinas, poniendo mayor hincapié en productos críticos en las diferentes fases ya que un pequeño error en una fase temprana puede tener grandes repercusiones en las fases siguientes.

### **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 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**

La siguiente tabla muestra la lista con las actividades de calidad a realizarse durante el proyecto, especificando en que semana del proyecto se realizan. Las fechas pueden ser modificadas según como se vaya avanzando.

<b>Actividad</b>	<b>Semana cuando se realiza</b>
Identificar propiedades de la calidad	Semana 1 , 2, 3 y 4
Planificar la Calidad	Semana 1, 2, 3 y 4
Revisar Calidad de Entregas	Todas
Evaluar y Modificar Plan de SQA	Semana 6, 8 y 11
Revisar el Ajuste al Proceso	Semana 3 a Semana 12
Evaluar Calidad Productos	Semana 3 a Semana 14
Realizar el informe Final de SQA	Semana 14
Revisión Técnica Formal	Semana 5, 6, 7, 8, 10, 11, 12

### 3.3. Responsables

Rol	Actividad
Responsable SQA	Revisar entregables. Auditorias (RTF). Plan de SQA. Informe Final de Calidad.
Analista	Especificación de Requerimientos. Modelo de Casos de Uso. Alcance del Sistema.
Implementadores – Esp. Tec.	Estudio de las tecnologías. Prototipos.
Administrador	Gestión de Riesgos. Estimaciones y Mediciones. Alcance. Plan de Proyecto.
Arquitecto	Modelo de Diseño, Descripción de Arquitectura.
Resp. De Verificación	Plan de Validación y Verificación
Resp. De SCM	Gestión del Cambio. Línea Base.
Diseñador IU	Especificación de la interfaz de usuario.



## 4. Documentación

### 4.1. Propósito

La documentación tiene 2 objetivos claves:

Permite la comunicación de la información relevante tanto al producto como al proceso entre los distintos integrantes y también para con el cliente.

Brinda un respaldo de todo lo sucedido en el proyecto, pudiendo ser analizado tanto durante el desarrollo del proyecto como en el futuro para aprender del mismo.

### 4.2. Documentación mínima requerida

La documentación mínima es la requerida para asegurar que la implementación logrará satisfacer los requerimientos. Es aquella que sin la cual no sería posible desarrollar el producto ni seguir el proceso.

Con lo cual abarca todos los planes (calidad, verificación, configuración), riesgos, requerimientos, situación de proyecto, planes de iteración, descripción de la arquitectura, modelo de implementación, modelo de datos, criterios de aceptación, pautas para interfaz de usuario, modelo de casos de uso y alcance del sistema.

De todas maneras cada documento sugerido por el proceso tiene su utilidad por lo que es de suma importancia tratar de cumplir con la mayor parte de los mismos. Gracias a estos uno puede llegar a notar/evitar/resolver problemas con más facilidad.

#### 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 (Ver si aplica)

*Usabilidad*

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

*Eficiencia*

- a. comportamiento respecto al tiempo (Ver si aplica)
- b. utilización de recursos

*Mantenibilidad*

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

*Portabilidad*

- a. adaptable (Ver si aplica)
- b. instalable
- c. co-existencia
- d. reemplazante (Ver si aplica)

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 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**

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 requerimientos 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.

#### **4.2.7. Modelo de casos de uso**

El documento contiene una lista de todos los Actores y Casos de Uso del sistema con una descripción de los mismos. El mismo se valida con los involucrados y se priorizan los Casos de Uso haciendo una selección de un subconjunto de Casos de Uso a detallar en cada iteración.

#### **4.2.8. Pautas para la interfaz de usuario**

El documento establece a nivel general las interfaces de usuarios en función de los requerimientos relevados y dar un pantallazo general del aspecto visual de la interfaz gráfica.

#### **4.2.9. Descripción de la arquitectura**

Este documento proporciona una apreciación global y comprensible de la arquitectura del sistema usando diferentes puntos de vista para mostrar distintos aspectos del sistema. Intenta capturar y llegar a las decisiones de arquitectura críticas que han sido hechas en el sistema.

#### **4.2.10. Modelo de diseño**

En el documento se presenta el diseño de los casos de uso del sistema a construir. A medida que avanza el proyecto se van agregando casos de uso en el mismo.

#### **4.2.11. Criterios de Aceptación**

Establecen los criterios que considerara el cliente para determinar la aceptación de producto. Es un documento clave para la calidad ya que permite determinar los aspectos claves para el cliente.

#### **4.2.12. Riesgos Encontrados**

Un punto clave para cualquier proyecto es la detección y mitigación de riesgos. Esta actividad es clave poder llevar adelante un proyecto exitoso ya permite evitar o prepararse para posibles problemas que puedan surgir.

### **4.3. Otros documentos**

Otros documentos que inciden en la calidad del producto a desarrollar son el estándar de implementación y el estándar de usabilidad.

Se detectó en nuestro proyecto la necesidad de distintas maneras de usar el producto dependiendo de la experiencia del usuario. Por lo tanto fue clave determinar y definir de la forma más completa posible estos "tipos de usos". EL estándar de implementación permite una mejora en la calidad del código fuente gracias a que unifica criterios, de esta forma todos los implementados codifican de la misma forma.

## **4.4. 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.

## **4.5. 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.

## **4.6. Estándar de uso del Teclado**

Debido a la necesidad de poder acceder a toda la aplicación mediante el uso exclusivo del teclado se debió definir un estándar para unificar el comportamiento de las distintas teclas y combinaciones en toda la aplicación.

## 5. Revisiones y auditorías

### 5.1. 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.

### 5.2. Requerimientos mínimos

[Se especifican las revisiones y auditorías que deben realizarse como mínimo, así como la agenda para la realización de las mismas.]

#### 5.2.1. Revisión de requerimientos

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

#### 5.2.2. Revisión de diseño preliminar

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

#### 5.2.3. Revisión de diseño crítico

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

#### 5.2.4. 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.

#### 5.2.5. 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.

#### 5.2.6. Auditoría física

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

#### 5.2.7. Auditorías internas al proceso

Estas auditorías son para verificar la consistencia: del código versus el documento de diseño, especificaciones de interfase, implementaciones de diseño versus requerimientos funcionales, requerimientos funcionales versus descripciones de testeo.

#### 5.2.8. 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).

#### 5.2.9. 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.

### **5.2.10. 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.

### **5.2.11. Agenda**

Revisión de Gestión	Todas las semanas.
Revisión de Documentos	Todas las semanas.
Revisión del plan de gestión de configuración.	Semana 6
Revisión de Requerimientos.	Semana 3, 4, 5, 6 y 7
Revisión de Diseño preliminar.	Semana 3, 4, 5 y 6
Revisión de Diseño Crítico.	Semana 7 y 8
Revisión Post Mortem.	Semana 15
Revisión del plan de V&V.	Semana 6, 8 y 10
Auditoria funcional.	Semana 13.
Auditoria física.	Semana 14.
Auditorías internas al proceso	Semana 12,13.

## **5.3. Otras revisiones**

### **5.3.1. Revisión de documentación de usuario**

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



## 6. Verificación

Se incluirán en este punto las verificaciones omitidas en el Plan de V&V.

## 7. Reporte de problemas y acciones correctivas

Para cada actividad se elige un responsable el cual tiene la tarea adicional de comunicar la situación de dicha actividad con los roles que correspondan. De esta manera ante cualquier problema hay una correcta vía de comunicación. Cada actividad tiene como salida un documento el cual al final de cada iteración debe cumplir con los requisitos de calidad. Esta tarea es responsabilidad del encargado de SQA y ante cualquier problema se comunicara con el responsable de dicha actividad.

## 8. Herramientas, técnicas y metodologías

En este punto se muestran algunas de las CheckLists que se utilizaran a lo largo del proyecto para realizar las revisiones de SQA.

### Requerimientos:

- ✓ ¿Los requerimientos están escrito en un lenguaje no técnico para que pueda ser comprensible para el usuario y/o el cliente?
- ✓ ¿Existen requerimientos que puedan tener más de una interpretación?
- ✓ ¿Cada característica del producto final es descripta?
- ✓ ¿Hay un glosario en el cual el significado de cada término está definido?
- ✓ ¿Los requerimientos pueden ser entendidos, implementados y verificados por diferentes grupos de personas?
- ✓ ¿Hay un índice de requerimientos?
- ✓ ¿Se tienen todas las imágenes, tablas y diagramas necesarios?
- ✓ ¿Todas las imágenes, tablas y diagramas están rotuladas?
- ✓ ¿Existen requerimientos que necesiten ser especificados con más detalle?
- ✓ ¿Existen requerimientos que necesiten ser especificados con menos detalle?
- ✓ ¿Todos los requerimientos están definidos?
- ✓ ¿Se ha definido información faltante si es que existe?
- ✓ ¿Están incluidos todos los requerimientos funcionales?
- ✓ ¿Hay algún requerimiento no satisfactorio?
- ✓ ¿Están incluidos todos los requerimientos de rendimiento?
- ✓ ¿Están incluidos todos los requerimientos de interfaces externas?
- ✓ ¿Están incluidos todos los requerimientos de persistencia de datos?
- ✓ ¿Están incluidos todos los requerimientos de software a utilizar?
- ✓ ¿Están incluidos todos los requerimientos de comunicaciones?
- ✓ ¿Están incluidos todos los requerimientos de hardware?
- ✓ ¿Están incluidos todos los requerimientos de entradas?
- ✓ ¿Están incluidos todos los requerimientos de salidas?

- ✓ ¿Están incluidos todos los requerimientos de seguridad?
- ✓ ¿Están incluidos todos los requerimientos de mantenibilidad?
- ✓ ¿Están incluidos todos los requerimientos de instalación?
- ✓ ¿Están incluidos todos los requerimientos de disponibilidad?
- ✓ ¿Están incluidos todos los requerimientos de recuperación?
- ✓ ¿Están especificados los cambios posibles a los requerimientos?
- ✓ ¿Existen requerimientos que describen el mismo objeto y están en conflicto en lo referente a las características del mismo?
- ✓ ¿Todos los requerimientos son trazables desde necesidades específicas del usuario?
- ✓ ¿Todos los requerimientos son trazables desde fuentes específicas (personas, documentos)?
- ✓ ¿Todos los requerimientos son trazables hacia documentos de diseño específicos?
- ✓ ¿Todos los requerimientos son trazables hacia módulos de software específicos?
- ✓ ¿El documento de requerimientos está organizado clara y lógicamente?
- ✓ ¿La estructura del documento se adhiere al estándar de documentación?
- ✓ ¿Hay redundancia en los requerimientos?
- ✓ ¿Cada requerimiento es relevante al problema y a su solución?
- ✓ ¿Hay requerimientos que contengan detalles de diseño?
- ✓ ¿Hay requerimientos que contengan detalles de gestión de proyectos?
- ✓ ¿Todas las fuentes de entrada están especificadas?
- ✓ ¿Todos los requerimientos de precisión de las entradas están especificados?
- ✓ ¿Todos los rangos de las entradas están especificados?
- ✓ ¿Todas las frecuencias de las entradas están especificadas?
- ✓ ¿Todos los formatos de las entradas están especificados?
- ✓ ¿Todos los requerimientos de precisión de las salidas están especificados?
- ✓ ¿Todos los rangos de las salidas están especificados?
- ✓ ¿Todas las frecuencias de las salidas están especificadas?
- ✓ ¿Todas las funciones del software están especificadas?
- ✓ ¿Todas las entradas para cada función están especificadas?
- ✓ ¿Todos los aspectos de procesamiento exitoso para cada función están especificados?
- ✓ ¿Todos los aspectos de procesamiento no exitoso para cada función están especificados?
- ✓ ¿Todas las tasas de transferencia de datos están especificadas?
- ✓ ¿Todas las tasas de rendimiento están especificadas?
- ✓ ¿Las consecuencias de las fallas del software para cada requerimiento están especificadas?
- ✓ ¿Está detallada la información a proteger de las fallas?
- ✓ ¿Está especificada la memoria principal mínima?

- ✓ ¿Está especificado el almacenamiento mínimo?
- ✓ ¿Está especificada la memoria principal recomendable?
- ✓ ¿Está especificado el almacenamiento recomendable?
- ✓ ¿Está definida la plataforma de software requerida?
- ✓ ¿Están definidas las herramientas de software requeridas?
- ✓ ¿Todos los productos de software adquiridos que se usarán con el sistema están especificados?
- ✓ ¿El número estimado de conexiones de red está especificado?
- ✓ ¿Los requerimientos de desempeño mínimos de la red están especificados?

### **Planificación del Proyecto:**

- ✓ ¿Está especificada la estrategia a seguir para realizar cambios al Plan de Proyecto? Tomando en cuenta lo siguiente:
  - Responsable de monitorear el Plan de Proyecto
  - Frecuencia que se realizarán modificaciones al Plan
  - ¿Cómo serán evaluados y aprobados los cambios al Plan?
  - ¿Cómo se comunicarán los cambios al Plan?
- ✓ ¿Especificó la estructura organizacional del proyecto?
- ✓ ¿Efectuó la planificación con un diagrama de Gantt? ¿Incluyó en la misma:
  - fases con sus objetivos,
  - iteraciones con sus objetivos,
  - hitos,
  - actividades críticas,
  - entregables,
  - responsables,
  - duraciones y
  - ¿productos más significativos?
- ✓ ¿Especificó los procedimientos administrativos y de gestión entre el proyecto y el cliente, gestión de configuración, gestión de calidad y verificación?
- ✓ ¿Especificó las condiciones que se asumen para el proyecto, eventos externos de los que depende y restricciones del proyecto?
- ✓ ¿Identificó los factores de riesgo asociados al proyecto y los procedimientos de monitoreo y contingencia?
- ✓ ¿Especificó los mecanismos de monitoreo y control de las actividades de Gestión de Calidad?
- ✓ ¿Especificó los mecanismos de monitoreo y control de las actividades de Gestión de Configuración?
- ✓ ¿Especificó los mecanismos de monitoreo y control de las actividades de Verificación y Validación?
- ✓ ¿Especificó los mecanismos de monitoreo y control de las actividades de Gestión de Proyecto?
- ✓ ¿Especificó la cantidad de personal en el proyecto, los roles asignados, los responsables por Línea de Trabajo y los métodos de entrenamiento a seguirse?
- ✓ ¿Realizó un inventario de los recursos tecnológicos del proyecto?

- ✓ ¿Definió los procedimientos técnicos, herramientas y tecnologías que se utilizarán en el proyecto?
- ✓ ¿Especificó los sistemas operativos, metodologías de desarrollo, lenguajes de programación, métodos para la especificación de diseño, construcción, verificación, integración, documentación, entregas al cliente, modificaciones y mantenimiento de entregables?
- ✓ ¿Especificó estándares técnicos, políticas y procedimientos a seguir en el desarrollo del producto?
- ✓ ¿Especificó las líneas de trabajo para las distintas actividades que se deben realizar (análisis, diseño, implementación, etc.)?
- ✓ ¿Estableció las dependencias entre las actividades de las distintas líneas?
- ✓ ¿Realizó un cronograma para las actividades del proyecto?

#### **Seguimiento del Proyecto:**

- ✓ ¿Detalló las mediciones realizadas en las horas de trabajo de los integrantes del equipo y en los productos obtenidos?
- ✓ ¿Especificó la cantidad de horas estimadas en el Plan de Iteración y la cantidad de horas realizada por Línea de Trabajo, por rol y por persona?
- ✓ Al final de cada iteración detalló:
  - mediciones de avance por Línea de Trabajo, como porcentaje de Casos de Uso detallados sobre Casos de Uso totales definidos en el Alcance del Sistema y cantidad de objetos correspondientes a los componentes implementados,
  - indicadores de Esfuerzo y Tiempo.
- ✓ ¿Describió el estado del proyecto comparando con lo establecido en el Plan del Proyecto y en el Plan de la Iteración?
- ✓ ¿Describió las desviaciones ocurridas a la planificación realizada y las causas de las mismas, detallando la Línea de Trabajo donde sucedió?
- ✓ ¿Describió las incidencias (eventos no planificados) ocurridas y las acciones correctivas tomadas o las acciones a tomar al respecto?
- ✓ ¿Especificó los riesgos que han ocurrido y cómo se han mitigado?
- ✓ ¿Realizó una evaluación subjetiva del avance del proyecto y perspectivas del mismo, basada en la información obtenida previamente?

#### **Evaluación y Ajuste al Plan de Calidad:**

- ✓ ¿Enumeró las actividades previstas en el Plan de Calidad para la Fase e Iteración a la que corresponde el documento que se está realizando y las que se hubieran re-planificado en documentos de Evaluación y Ajustes anteriores?
- ✓ ¿Enumeró las desviaciones encontradas respecto a las actividades planificadas, esto es las actividades que no se realizaron como y/o cuando estaban planificadas o que no se realizaron en absoluto?
- ✓ ¿Describió cada una de las desviaciones encontradas estableciendo su naturaleza y gravedad en cuanto al impacto de la misma en la calidad del proyecto y productos afectados?
- ✓ ¿Documentó las mediciones correspondientes a las actividades realizadas contra las actividades planificadas, cantidad de productos revisados contra cantidad de productos previstos para revisar en el Plan de Calidad?

- ✓ ¿Efectuó la evaluación y el ajuste de las actividades de Calidad tomando en cuenta si quedaron pendientes verificaciones de productos y/o pruebas por realizar que tengan que planificarse nuevamente?
- ✓ ¿Planificó las revisiones que quedaron pendientes?

#### **Revisión Técnica Formal:**

- ✓ Especificó para cada producto revisado:
  - nombre, versión y Línea de Trabajo a la que corresponde el producto,
  - nombre y rol de los participantes de la revisión,
  - forma en que se hizo la revisión,
  - aspectos del producto que serán revisados,
  - propiedades de Calidad que se busca se cumplan,
  - principios y estándares de Calidad aplican al producto y
  - problemas detectados tanto en el producto como en el proceso de elaboración del mismo.
- ✓ ¿Para cada problema detectado se especificó sus características, gravedad y su ubicación?
- ✓ ¿Para cada problema detectado se establecieron sugerencias a las correcciones a realizar para que el producto cumpla con los principios y estándares de calidad establecidos, y los procedimientos definidos?
- ✓ ¿Realizó una evaluación global del Producto revisado de acuerdo a los objetivos planteados y los problemas detectados?
- ✓ ¿Se establecieron las acciones a tomar para que el producto sea corregido y revisado en una nueva versión?
- ✓ ¿Estableció la próxima revisión de acuerdo a los puntos detallados anteriormente, en términos de fechas o Fase e iteración?

#### **Preguntas de ejemplo para la Revisión Técnica Formal:**

- Arquitectura
  - ✓ ¿La organización total del programa está clara e incluye una buena vista de la arquitectura y su justificación?
  - ✓ ¿La arquitectura está compuesta de capas?
  - ✓ ¿Los paquetes están bien definidos, incluyen su funcionalidad e interfaz para otros módulos?
  - ✓ ¿Todas las estructuras de datos importantes están descritas y justificadas?
  - ✓ ¿Todas las estructuras de datos importantes están ocultas mediante funciones
  - ✓ ¿Todas las estructuras de datos importantes están ocultas mediante funciones de acceso?
  - ✓ ¿Todos los algoritmos principales están descritos y justificados?
  - ✓ ¿Todos los objetos importantes están descritos y justificados?
  - ✓ ¿La interfaz con el usuario está modularizada tal que los cambios en ella no afecten el resto del programa?
  - ✓ ¿Los aspectos claves de la interfaz de usuario están definidos?
  - ✓ ¿Se ha estimado el uso de memoria, se ha descrito y justificado una estrategia para la gestión de la memoria?
  - ✓ ¿Se ha incluido una estrategia coherente de manejo de errores?

- ✓ ¿Se ha definido un nivel de robustez?
  - ✓ ¿La arquitectura está diseñada para acomodar cambios?
  - ✓ ¿Los objetivos principales del sistema están claramente establecidos?
  - ✓ ¿El diseño de alto nivel es independiente de la máquina y lenguaje que se usará para implementarlo?
  - ✓ ¿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)?
  - ✓ ¿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?
  - ✓ ¿Se especifican en el documento las correspondencias entre los requerimientos y el Modelo de Diseño
  - ✓ ¿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?
  - ✓ ¿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?
  - ✓ ¿Identificó las Clases de Diseño claves del Sistema?
  - ✓ ¿Describió las clases del Diseño y los subsistemas que corresponden a la Arquitectura del Diseño de cada Caso de Uso?
  - ✓ ¿Realizó la trazabilidad entre las entidades de diseño del Modelo de Diseño y los componentes del Modelo de Implementación?
  - ✓ ¿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?
  - ✓ ¿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?
  - ✓ ¿Muestra la correspondencia entre la Arquitectura del Software y la Arquitectura del Hardware?
  - ✓ ¿Se tienen en cuenta todas las propiedades de calidad que debe tener el sistema?
  - ✓ ¿Se ha alcanzado un grado adecuado de modularidad?
  - ✓ ¿Se ha diseñado para el cambio?
- Diseño Orientado a Objetos
    - ✓ ¿Se han definido subsistemas como parte de la representación del diseño?
    - ✓ ¿Los límites de los subsistemas están bien definidos?
    - ✓ ¿Se ha establecido una bien definida interfaz para cada subsistema?
    - ✓ ¿Se han establecido clases de comunicación para la comunicación entre subsistemas?

- ✓ ¿Las clases que pueblan un subsistema tienen sentido dentro del contexto de los requerimientos funcionales/datos definidos para el subsistema?
  - ✓ ¿Los subsistemas están diseñados para tener una interacción mínima con los demás?
  - ✓ ¿Los subsistemas están diseñados de tal forma que puedan ser usados en otros sistemas?
  - ✓ ¿Se han definido clases de gestión de tareas?
  - ✓ ¿Se han definido clases de gestión de datos?
  - ✓ ¿Se han definido clases de interfaz de usuario?
  - ✓ ¿El diseño diferencia entre los componentes del dominio del problema, de la interfaz de usuario, de gestión de tareas, y de gestión de datos?
  - ✓ ¿Toda la comunicación dentro del subsistema está claramente detallada en un diagrama de colaboración?
  - ✓ ¿Se utiliza un protocolo de definición para cada objeto?
  - ✓ ¿El protocolo de descripción describe claramente la estructura de los mensajes para cada clase?
  - ✓ ¿Se han utilizado patrones de diseño reconocidos como parte del diseño?
  - ✓ ¿El diseño es intelectualmente tratable?
  - ✓ ¿El programa será fácil de mantener?
  - ✓ ¿El diseño tiene en cuenta futuras extensiones al programa?
  - ✓ ¿Todas las partes del diseño son estrictamente necesarias?
- Modelo de Casos de Uso
    - ✓ ¿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?
    - ✓ ¿El modelo de diseño permite cumplir con todos los requerimientos?
    - ✓ ¿Para cada objeto se definió parámetros, reglas, operaciones que realiza, requerimientos de implementación, interfaz de usuario y relaciones con otros objetos?
    - ✓ ¿Esta especificación se realiza mediante Diagrama de Paquetes, Diseño de flujo de eventos, Diagramas de interacción y Requerimientos especiales o de implementación?
    - ✓ ¿Identifiqué los objetos y subsistemas de diseño que intervienen en el caso de uso y sus relaciones?
    - ✓ ¿Realicé un diagrama de interacción para representar la realización de cada Caso de Uso?
    - ✓ ¿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?
    - ✓ ¿Describí los requerimientos no funcionales en la realización de un caso de uso?
    - ✓ ¿Para cada subsistema específico: declaro que hace, cual es el propósito del mismo y cuáles son sus entradas y que salidas produce?
    - ✓ ¿Para cada subsistema identifiqué los subsistemas subordinados al mismo?

- ✓ ¿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.?
- ✓ ¿Identificó todos los recursos externos al diseño, que necesita el subsistema para realizar su función (dispositivos físicos, servicios de software, recursos de procesamiento etc.)?
- ✓ ¿Realizó un diseño detallado del comportamiento de la interfaz de usuario a partir de la especificación de requerimientos y el mismo está de acuerdo con el entorno tecnológico?
- ✓ ¿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?
- ✓ ¿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?
- ✓ ¿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)?
- Gestión de la Configuración
  - ✓ ¿Todos los integrantes del grupo están siguiendo los procedimientos definidos en el plan de SCM?
  - ✓ ¿Está definida la línea base de la fase?
  - ✓ ¿Todos los ítems que indica el documento de la línea base están en el directorio correspondiente a esa línea base?
  - ✓ ¿Todos los ítems dentro del directorio de la línea base se encuentran en el documento correspondiente?
- Implementación
  - ✓ ¿Han tenido problemas de no entendimiento del diseño?
  - ✓ ¿Se ha respetado el diseño a la hora de implementar o se han hecho modificaciones sobre el avance?
  - ✓ ¿Realizan integración? ¿Cuándo y cómo?
  - ✓ ¿Realizan algún tipo de verificación?
  - ✓ ¿La documentación técnica la hacen a medida que implementan o luego? ¿La realiza el implementador que construyó la clase o se encarga otra persona?
  - ✓ ¿Han tenido algún problema en seguir las normas de implementación?
  - ✓ ¿Realizan mediciones sobre la cantidad de líneas de código y tiempo de implementación?
  - ✓ ¿Implementan de a pares? ¿Realizan revisión por pares? ¿Cuáles han sido los resultados?
  - ✓ ¿Hasta cuándo se tiene pensado implementar?
- Verificación
  - ✓ ¿Cómo se está llevando a cabo la verificación?
  - ✓ ¿Realizan una integración progresiva para verificar o integran todo junto?



- ✓ ¿Las pruebas y resultados esperados están pensadas antes de comenzar o se documentan a medida que se realizan?
- ✓ ¿Piensa que tiene una buena cobertura sus casos de pruebas?
- ✓ ¿Hay un plan de pruebas del sistema?
- ✓ ¿Se valió de los casos de uso para realizar los casos de pruebas del sistema?
- ✓ ¿Hay una medida de la cantidad de bugs encontrados por líneas de código?
- ✓ ¿Cuándo se comenzará la prueba del sistema?
- ✓ ¿Cuántas personas están realizando la verificación? ¿Qué recursos posee?
- ✓ ¿Tiene pensado realizar alguna prueba del sistema en un ambiente "real"?

## 9. 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.