

# **Sistema de registros – Diálisis Crónico**

## **Plan de SQA**

### **Versión 2.0**

#### **Historia de revisiones**

Fecha	Versión	Descripción	Autor
24/08/2007	1.0	Versión inicial de este documento	Javier Lazo
27/08/2007	1.1	Revisión del documento y modificaciones al mismo	Javier Lazo
16/09/2007	2.0	Ajustes y modificaciones de acuerdo al avance del proyecto.	Javier Lazo

# Contenido

<b>1. PROPÓSITO .....</b>	<b>3</b>
<b>2. REFERENCIAS .....</b>	<b>3</b>
<b>3. GESTIÓN.....</b>	<b>3</b>
3.1. ORGANIZACIÓN .....	3
3.2. ACTIVIDADES .....	4
3.2.1. <i>Ciclo de vida del software cubierto por el Plan</i> .....	4
3.2.2. <i>Actividades de calidad a realizarse</i> .....	4
3.2.3. <i>Revisar cada producto</i> .....	4
3.2.4. <i>Revisar el ajuste al proceso</i> .....	5
3.2.5. <i>Realizar Revisión Técnica Formal (RTF)</i> .....	5
3.2.6. <i>Asegurar que las desviaciones son documentadas</i> .....	5
3.2.7. <i>Relaciones entre las actividades de SQA y la planificación</i> .....	6
3.3. RESPONSABLES .....	6
<b>4. DOCUMENTACIÓN .....</b>	<b>6</b>
4.1. PROPÓSITO .....	6
4.2. DOCUMENTACIÓN MÍNIMA REQUERIDA .....	6
4.2.1. <i>Especificación de requerimientos del software</i> .....	6
4.2.2. <i>Descripción del diseño del software</i> .....	8
4.2.3. <i>Plan de Verificación &amp; Validación</i> .....	8
4.2.4. <i>Reportes de Verificación &amp; Validación</i> .....	8
4.2.5. <i>Documentación de usuario</i> .....	8
4.2.6. <i>Plan de Gestión de configuración</i> .....	8
4.3. OTROS DOCUMENTOS .....	9
<b>5. ESTÁNDARES, PRÁCTICAS, CONVENCIONES Y MÉTRICAS .....</b>	<b>9</b>
5.1. ESTÁNDAR DE DOCUMENTACIÓN.....	9
5.2. ESTÁNDAR DE VERIFICACIÓN Y PRÁCTICAS .....	9
5.3. OTROS ESTÁNDARES .....	9
<b>6. REVISIONES Y AUDITORÍAS .....</b>	<b>9</b>
6.1. OBJETIVO .....	9
6.2. REQUERIMIENTOS MÍNIMOS.....	10
6.2.1. <i>Revisión de requerimientos</i> .....	10
6.2.2. <i>Revisión de diseño preliminar</i> .....	10
6.2.3. <i>Revisión de diseño crítico</i> .....	10
6.2.4. <i>Revisión del Plan de Verificación &amp; Validación</i> .....	10
6.2.5. <i>Auditoría funcional</i> .....	10
6.2.6. <i>Auditoría física</i> .....	10
6.2.7. <i>Auditorías internas al proceso</i> .....	10
6.2.8. <i>Revisiones de gestión</i> .....	10
6.2.9. <i>Revisión del Plan de gestión de configuración</i> .....	10
6.2.10. <i>Revisión Post Mortem</i> .....	10
6.2.11. <i>Agenda</i> .....	10
6.3. OTRAS REVISIONES.....	11
6.3.1. <i>Revisión de documentación de usuario</i> .....	11
<b>7. VERIFICACIÓN.....</b>	<b>11</b>
<b>8. REPORTE DE PROBLEMAS Y ACCIONES CORRECTIVAS.....</b>	<b>11</b>
<b>9. HERRAMIENTAS, TÉCNICAS Y METODOLOGÍAS .....</b>	<b>11</b>
<b>10. GESTIÓN DE RIESGOS.....</b>	<b>15</b>

## 1. Propósito

El propósito de este plan es controlar la calidad del producto y del proceso a través de la planificación de las actividades que deben llevarse a cabo para dicho propósito.

El alcance de este plan incluye las tareas que desempeñan los distintos roles así como los entregables que se producirán a lo largo del proyecto.

El software a desarrollar es un Sistema de Registros para Pacientes en Diálisis con insuficiencia renal crónica. Los elementos que serán cubiertos abarcarán requerimientos, arquitectura, diseño, implementación e implantación.

Solo será cubierta una porción del ciclo de vida del software, esta porción corresponde al desarrollo del software implicando desde la relevación de requerimientos hasta la liberación del producto quedando fuera 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.

## 3. Gestión

### 3.1. Organización

Existen varias líneas de trabajo que tienen influencia y controlan la calidad:

- Requerimientos
- Diseño (basado en una Arquitectura particular)
- Gestión del Proyecto
- Verificación
- Gestión de Configuración y Control de Cambios
- Gestión de la Calidad

El equipo de trabajo se encuentra organizado de la siguiente manera:

Línea de trabajo	Integrantes
Requerimientos	Walmar Laiolo, Gabriela Medina, Paola Amarillo, Pablo Pazos, Elisa Aguerre.
Verificación	Karina Rapetti
Arquitectura	Samanta de Barros
Gestión del Proyecto	Leonardo Méndez
Gestión de Configuración	Julio Stirling
Gestión de Calidad	Javier Lazo

Existen algunas dependencias entre las líneas de trabajo antes mencionadas.

- Los requerimientos (en particular los no funcionales) son necesarios para el diseño de la arquitectura.
- Tanto los requerimientos como el diseño son necesarios para la verificación.

- La gestión del proyecto debe entre otras cosas controlar el avance del proyecto que implica que se cumplan las tareas planificadas y asignadas en las demás líneas.
- La gestión de configuración debe entre otras cosas tener un control de los elementos generados por el equipo lo que implica una colaboración de las líneas básicas.
- La gestión de calidad debe tener todo lo anterior en cuenta para poder tener un control de la calidad del proyecto en su totalidad.

### **3.2. Actividades**

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

El alcance del Plan cubre desde la etapa de Requerimientos hasta la etapa de liberación del producto.

Los productos que tendrán revisiones son:

- 1) Especificación de Requerimientos
- 2) Descripción de la Arquitectura
- 3) Estándar de Documentación de Usuario
- 4) Plan de Verificación y Validación
- 5) Plan del Proyecto
- 6) Plan de la Configuración
- 7) Plan de Desarrollo
- 8) Plan de Implantación
- 9) Informe final del proyecto

#### **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 y Validación.*

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

Actividad	Semana cuando se realiza
Identificación de Propiedades de Calidad	1 a 5
Revisar la entrega semanal	1 a 14
Elaboración del Plan de Calidad	2 y 4
Realizar Revisión Técnica Formal	5 , 7 , 9 y 11
Revisar el ajuste al proceso	3, 5, 7, 9, 11, 13, 14
Evaluación y ajuste del Plan de Calidad	9, 11
Evaluación final de Calidad	14

### 3.3. Responsables

Cada una de estas actividades tiene como responsable al SQA. De todas formas estas actividades no pueden ser llevadas a cabo únicamente por él, para esto es necesario el apoyo de otros integrantes del grupo. En particular se cuenta con los responsables de cada área de trabajo los cuales pueden aportar el punto de vista de su rol sobre determinado tema (los responsables se indican en la sección 8).

## 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.
- No ser 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 en un elemento del diseño.
  - c. 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.



#### **4.3. Otros documentos**

Otros documentos que se identifican como incidentes sobre la calidad del producto a desarrollar son:

- Plan de desarrollo
- Plan de Implantación
- Informe de Situación del Proyecto
- Plan de la Integración de la Iteración

### **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.

Se utiliza como estándar de documentación las plantillas de documentación que son parte del material entregado para este curso.

#### **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**

Se cuenta con estándares de implementación.

### **6. Revisiones y auditorías**

#### **6.1. Objetivo**

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

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

## **6.2. Requerimientos mínimos**

### **6.2.1. Revisión de requerimientos**

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

### **6.2.2. Revisión de diseño preliminar**

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

### **6.2.3. Revisión de diseño crítico**

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

### **6.2.4. Revisión del Plan de Verificación & Validación**

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

### **6.2.5. Auditoría funcional**

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

### **6.2.6. Auditoría física**

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

### **6.2.7. Auditorías internas al proceso**

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

### **6.2.8. Revisiones de gestión**

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

### **6.2.9. Revisión del Plan de gestión de configuración**

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

### **6.2.10. Revisión Post Mortem**

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

### **6.2.11. Agenda**

A razón de no tener algunos elementos claves es que todavía no se cuenta con una agenda de revisiones y auditoria.

### 6.3. Otras revisiones

#### 6.3.1. Revisión de documentación de usuario

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

### 7. Verificación

[Se debe identificar todas las verificaciones que no fueron identificadas en el Plan de V & V para el software y debe especificar los métodos a ser usados.]

### 8. Reporte de problemas y acciones correctivas

Existe un responsable por cada disciplina:

Disciplina	Responsable
Gestión del Proyecto	Leonardo Méndez
Arquitectura	Samanta de Barros
Análisis	Elisa Aguerre
Gestión de Configuración	Julio Stirling
Verificación	Karina Rapetti
Gestión de Calidad	Javier Lazo
Implementación	Samanta de Barros

Para las líneas de trabajo Gestión de Proyecto, Verificación y Gestión de Configuración el reporte de problemas y acciones correctivas corresponden exclusivamente a su área de trabajo.

El responsable de Gestión de Calidad debe enfocarse en el correcto seguimiento del proceso de desarrollo teniendo para ello que controlar y evaluar a las restantes disciplinas.

Para todos estos casos el reporte del problema será implementado por medio de informes y/o documentos entregados a los involucrados directamente en el problema detectado. En caso de ser necesario como medida para las acciones correctivas se podría llegar a concretar reuniones que ayuden a dilucidar el problema.

### 9. Herramientas, técnicas y metodologías

#### Requerimientos

##### Generales

- ¿Se utiliza un lenguaje conocido por el cliente?
- ¿Se utiliza un lenguaje conocido por los desarrolladores?
- ¿Cada requerimiento está definido una única vez? (no hay requerimientos redundantes)
- ¿Los requerimientos están especificados con un nivel apropiado de detalle? (ni más, ni menos)
- ¿Los requerimientos funcionales cubren todas las situaciones anormales? (ejemplo: caída del sistema)
- ¿El documento contiene sólo detalles de implementación necesarios?
- ¿Se corresponde con las necesidades explicitadas en los documentos referentes a las reuniones de requerimientos que se mantuvieron con el cliente?

- ¿Este documento ha sido validado por el cliente?
- ¿Los requerimientos están priorizados?
- ¿Esta definido perfectamente el alcance del sistema?
- ¿Están descriptas claramente las principales interfaces externas e internas del producto? (si las hay)

#### Compleitud

- ¿Están incluidos los requerimientos de calidad? (como ser robustez, confiabilidad)
- ¿Están incluidos todos los requerimientos relacionados con las restricciones de Diseño?
- ¿Están incluidos todos los requerimientos relacionados con el hardware?
- ¿Están incluidos todos los requerimientos relacionados con entradas?
- ¿Están incluidos todos los requerimientos relacionados con salidas?

#### Consistencia

- ¿El documento de requerimientos está libre de contradicciones?
- ¿Hay algún requerimiento que pueda tener más de una interpretación?
- ¿Existe un glosario completo?

#### Verificabilidad

- ¿Hay algún requerimiento que es imposible de verificar?

#### Trazabilidad

- ¿Cada requerimiento puede ser trazado hacia el documento donde se describe el problema o hacia una necesidad específica del cliente?
- ¿Todos los requerimientos son trazables hacia un documento específico de diseño?
- ¿Todos los requerimientos son trazables hacia un módulo específico?

### Diseño

#### Generales

- ¿Todos los requerimientos de la Especificación están satisfechos en el diseño?
- ¿El estilo de presentación y detalle son consistentes durante todo el documento?
- ¿El diseño toma en cuenta todas las situaciones y condiciones esperadas?
- ¿El diseño es de baja complejidad?

### Arquitectura

- ¿La organización total del programa está clara e incluye una buena vista de la arquitectura y justificación?
- ¿La arquitectura esta compuesta de capas?
- ¿Los paquetes están bien definidos, incluyen su funcionalidad e interfaz para otros módulos?
- ¿Todas las estructuras de datos importantes están descriptas y justificadas?
- ¿Todas las estructuras de datos importantes están ocultas mediante funciones de acceso?
- ¿La organización y contenido de la base de datos están especificados?

- ¿Todos los algoritmos principales están descriptos y justificados?
- ¿Todos los objetos importantes están descriptos 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 esta 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?
- ¿Se dan motivaciones para todas las decisiones importantes?

## Diseño Orientado a Objetos

- ¿Se han definido subsistemas como parte de la representación del diseño?
- ¿Hay relativamente pocos subsistemas?
- ¿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/objeto?
- ¿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

### Implementadores:

- 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?
- ¿Realizan integración? ¿Cómo? ¿Cuándo?
- ¿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? ¿Han obtenido algún buen resultado de esto último?
- ¿Hasta cuándo se tiene pensado implementar?

### Verificación

- ¿Cómo se está llevando a cabo la verificación? ¿Realizan una integración progresiva para verificar o todo de una? ¿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" (con servidor accediendo a él por la línea telefónica utilizando distintos browsers)?

- ¿Quién corrige los errores? Si la respuesta es los implementadores: ¿Cómo se comunican los errores a los implementadores? ¿Realizan las correcciones debidas? ¿Se verifica que los errores encontrados fueron corregidos?

## **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, con lo cual se podrá hacer referencia a él.