

**Plan de Aseguramiento de Calidad**

**Versión 1.0**

**Proyecto: Impresión Web**

Información del Documento

|  |  |
| --- | --- |
| **Título del Documento:** | Plan de Calidad del Proyecto Impresión Web |
| **Nombre del Archivo del Documento:** | IMP2009\_Plan\_QA.docx |
| **Número de Versión:** | 1.0 |
| **Autor** | Matías Carrizo |
| **Fecha de Creación:** | 11/11/2009 |

Historia de Cambios

| Fecha | Versión | Descripción | Autor |
| --- | --- | --- | --- |
| 11/11/2009 | 1.0 | Versión Inicial | Carrizo, Matias |
| 11/11/2009 | 1.1 | Correcciones de redacción. Correcciones en establecimiento de actividades | Molina, Leandro Pisciolari, Antonela  Carrizo, Matias |
| 12/11/2009 | 1.2 | Correcciones de redacción.  Se completaron preguntas de inspección  Se agrega referencia al plan de gestión de configuración  Se agrega documentación de plan de métricas | Alberione, Gabriel  Merdine, Victoria  Molina, Leandro  Pisciolari, Antonela  Prez, Pablo |

Tabla de contenido

[1 Propósito 5](#_Toc245753368)

[2 Documentos de referencia 5](#_Toc245753369)

[3 Gestión 6](#_Toc245753370)

[3.1 Organización 6](#_Toc245753371)

[3.2 Actividades 7](#_Toc245753372)

[3.2.1 Ciclo de vida del software cubierto por el Plan 7](#_Toc245753373)

[3.2.2 Revisar cada producto 7](#_Toc245753374)

[3.2.3 Revisar el ajuste al proceso 8](#_Toc245753375)

[3.2.4 Realizar Revisión Técnica Formal (RTF) 8](#_Toc245753376)

[3.2.5 Asegurar que las desviaciones son documentadas 8](#_Toc245753377)

[3.2.6 Revisar las entregas 9](#_Toc245753378)

[3.3 Responsables 9](#_Toc245753379)

[4 Documentación 9](#_Toc245753380)

[4.1 Requerimientos mínimos de documentación 9](#_Toc245753381)

[4.1.1 Especificación de requerimientos de software (ERS) 9](#_Toc245753382)

[4.1.2 Descripción del diseño de software (DDS) 11](#_Toc245753383)

[4.1.3 Plan de verificación y validación de software (PVVS) 11](#_Toc245753384)

[4.1.4 Reporte de verificación y validación de software (RVVS) 11](#_Toc245753385)

[4.1.5 Documentación de usuario 11](#_Toc245753386)

[4.1.6 Plan de gestión de configuración de software (PGCS) 12](#_Toc245753387)

[4.2 Otra documentación 12](#_Toc245753388)

[4.2.1 Plan de desarrollo de software (PDS) 12](#_Toc245753389)

[4.2.2 Manual de estándares y procedimientos (MEP) 12](#_Toc245753390)

[4.2.3 Plan de gestión de proyecto de software (PGPS) 12](#_Toc245753391)

[4.2.4 Manual de mantenimiento de software 13](#_Toc245753392)

[4.2.5 Plan de métricas de software 13](#_Toc245753393)

[5 Estándares, prácticas, convenciones y métricas 13](#_Toc245753394)

[5.1 Estándar de documentación 13](#_Toc245753395)

[5.1.1 Estándar de documentación técnica e implementación 14](#_Toc245753396)

[5.1.2 Estándar de documentación de usuario 14](#_Toc245753397)

[5.1.3 Estándar de verificación y prácticas 14](#_Toc245753398)

[5.2 Otros estándares 14](#_Toc245753399)

[5.2.1 Estándares de programación 14](#_Toc245753400)

[6 Revisiones y auditorias 15](#_Toc245753401)

[6.1 Propósito 15](#_Toc245753402)

[6.2 Requerimientos mínimos 15](#_Toc245753403)

[6.2.1 Revisión de requerimientos de software (RRS) 15](#_Toc245753404)

[6.2.2 Revisión de diseño preliminar (RDP) 15](#_Toc245753405)

[6.2.3 Revisión de diseño crítico (RDC) 15](#_Toc245753406)

[6.2.4 Revisión del Plan de Verificación & Validación (RPVVS) 15](#_Toc245753407)

[6.2.5 Auditoría funcional 15](#_Toc245753408)

[6.2.6 Auditoría física 16](#_Toc245753409)

[6.2.7 Auditorías internas al proceso 16](#_Toc245753410)

[6.3 Otros 16](#_Toc245753411)

[6.3.1 Revisiones de gestión 16](#_Toc245753412)

[6.3.2 Revisión del Plan de gestión de configuración (RPGCS) 16](#_Toc245753413)

[6.3.3 Revisión Post Mortem 16](#_Toc245753414)

[6.3.4 Revisión de documentación de usuario 16](#_Toc245753415)

[7 Prueba 16](#_Toc245753416)

[7.1 Pruebas de Unidad 16](#_Toc245753417)

[7.2 Pruebas de Integración 17](#_Toc245753418)

[7.3 Pruebas del Sistema 17](#_Toc245753419)

[7.4 Pruebas de Aceptación 18](#_Toc245753420)

[8 Reporte de problemas y acciones correctivas 18](#_Toc245753421)

[9 Herramientas, técnicas, y metodologías 19](#_Toc245753422)

[9.1 Acciones y Comentarios 21](#_Toc245753423)

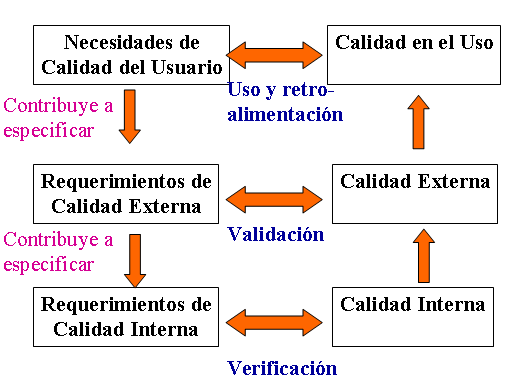
[10 Capacitación 24](#_Toc245753424)

# Propósito

El propósito de Plan de Calidad (PAQS) consiste en especificar las pautas a seguir durante el proceso de desarrollo para poder asegurar la calidad del mismo y del producto a construir, detallando todo lo referente a la planificación del seguimiento de la calidad en el proyecto, indicando para cada actividad, los atributos de calidad relevantes, los métodos de evaluación de calidad, en qué momento se realizarán dichos métodos y los responsables, centrándose en el producto más allá del proceso.

Este Plan solo cubre la parte del ciclo de vida correspondiente a las fases de evaluación, elaboración, construcción y transición del software, pero no cubre la parte del ciclo de vida correspondiente al mantenimiento, aunque pondrá énfasis en el futuro del producto.

El PAQS cubre todas las disciplinas involucradas en el proceso y se utilizara el modelo de calidad CMMI v1.4, es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. Para esto el plan debe asegurar la calidad interna del producto, no solo para poder obtener buenos atributos de calidad externa del mismo, sino también para poder construir un producto evolutivo, ya que el cliente requiere que el mismo pueda evolucionar en futuras iteraciones. Esto implica también que el plan deberá asegurar la calidad de los documentos que siguen el proceso, los cuales serán de gran utilidad para la próxima iteración.



Además el plan pretende hacer un seguimiento del proyecto para asegurar y medir la adecuación al proceso, y ayudar a la gestión del proyecto con métodos de verificación que permitan detectar defectos en forma temprana, en momentos donde las medidas correctivas no son tan costosas.

Las guías de calidad presentadas deben interpretarse como hilo conductor en las actividades y procesos involucrados en el proyecto. Debe especificar las pautas necesarias para la tarea de cuantificar la calidad de los entregables, el proceso, prototipos y producto final.

# Documentos de referencia

| ID Archivo de Documento | Título del Documento | Fecha de Publicación | Autor |
| --- | --- | --- | --- |
| IMP2009\_ERS | Documento de Especificación de Requerimientos de Software | 08/10/2009 | ISW |
| IMP2009\_ProjectCharter | Project Charter | 08/10/2009 | ISW |
| IMP2009\_Cronograma | Cronograma del Proyecto | 08/10/2009 | ISW |
| IMP2009\_PlanDesarrolloSoftware | Plan de Desarrollo de Software | 08/10/2009 | ISW |
| IMP2009\_Plan\_SCM | Plan de Gestión de Configuración | 27/10/2009 | ISW |

# Gestión

La organización de las actividades y los responsables asociados (para la primera fase), se encuentran en el correspondiente documento de “Plan de la iteración”. Allí se detallan todas las responsabilidades de cada integrante.

Cada integrante es responsable de la calidad de su trabajo. Se tendrá un líder o Gestor de Calidad que tendrá el mando de los aspectos de calidad de todo el proyecto. Para las actividades finales de pruebas (Pruebas de integración) se deberá asignar un equipo de ingenieros que incluya al líder o Gestor de Calidad.

## Organización

Las líneas de trabajo básicas que controlan la calidad del software son las que involucran las actividades tradicionales y están divididas en: Requerimientos, Análisis, Diseño, Implementación y Verificación. La línea de trabajo encargada de asegurar la calidad es parte del área de gestión en la estructura de la organización. Las áreas de trabajo relacionadas que ofrecen soporte a todas las demás son las áreas de gestión del proyecto y gestión de configuración. Toda actividad relacionada con el SQA se realizará bajo la supervisión del responsable SQA y posteriormente verificada por el equipo de Verificación y Validación.

Las disciplinas de gestión brindan soporte a las disciplinas básicas y se realizan en forma paralela a ellas. El Responsable de Calidad revisa tareas y productos elaborados por integrantes incluidos en las restantes líneas de trabajo, y es necesario que dicha elaboración se haga con el tiempo suficiente para una correcta revisión, y de ser necesario para realizar modificaciones. Para esto se llegó un acuerdo con el grupo de adoptar una actitud de responsabilidad para que esto suceda, y la planificación se hizo teniendo en cuenta esto y no las entregas semanales. Luego brindará soporte al administrador en las actividades relacionadas con la “Gestión del Proyecto”, en las tareas de métricas de calidad, de forma de tener una visión cuantificada del alcance del proyecto desde el punto de vista de la calidad pautada con el cliente. Este último enfoque pone a la vista la dependencia de las tareas de Planificación de Proyecto respecto a la cuantificación de calidad.

El encargado de SQA es el responsable de asegurar la calidad sobre los productos, pero es de importancia que cada integrante del grupo adopte un comportamiento responsable, y consiente acerca de decisiones tomadas que puedan en un futuro estropear la calidad del producto.

|  |  |
| --- | --- |
| Líneas de Trabajo | Trabajador |
| Requerimientos | Merdine, Victoria  Molina, Leandro |
| Análisis | Alberione, Gabriel  Carrizo, Matías  Pisciolari, Antonela |
| Diseño | Merdine, Victoria  Pres, Pablo  Pisciolari, Antonela |
| Implementación | Alberione, Gabriel  Molina, Leandro |
| Verificación | Carrizo, Matías  Pres, Pablo |
| Despliegue | Alberione, Gabriel  Molina, Leandro  Pres, Pablo |
| Gestión de Configuración | Pisciolari, Antonela |
| Gestión de Proyecto | Merdine, Victoria |
| Gestión de Calidad | Alberione, Gabriel  Molina, Leandro  Pres, Pablo |

## Actividades

### Ciclo de vida del software cubierto por el Plan

Como se indicó anteriormente, el plan cubre, del ciclo de vida del software, las fases de evaluación, elaboración, construcción y transición. 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 las entregas
* Revisar el ajuste al proceso
* Realizar Revisión Técnica Formal (RTF)
* Asegurar que las desviaciones son documentadas.
* Evaluación final de calidad.

### 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.

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 consientes de desviaciones o discrepancias encontradas. Éste documento a su vez sirve para el director del proyecto para tener una idea general del trabajo semanal del grupo en cuanto a documentación, y hace referencia a todos los documentos entregados, por lo que se debe de poder navegar a través de él fácilmente.

### 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 consientes de desviaciones o discrepancias encontradas.

### 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.

### 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.

### Revisar las entregas

Ésta actividad consiste en revisar semanalmente la totalidad de los entregables del proceso, se documentan las desviaciones y se corrigen si son menores, de lo contrario, el documento sigue el ciclo definido. Éste documento a su vez sirve para el director del proyecto para tener una idea general del trabajo semanal del grupo en cuanto a documentación, y hace referencia a todos los documentos entregados, por lo que se debe de poder navegar a través de él fácilmente.

## Responsables

*Líder o Gestor de Calidad:* Es responsable de ver que se realicen las tareas de la sección 3.2 y que se siga lo prescrito en este documento, incluido el calendario de las revisiones especificadas.

*Líder del Proyecto:* será responsable de asegurar que la administración de la calidad se lleve a cabo.

*Líder de Requerimientos:* será responsable de ver que se realicen todas las inspecciones y revisiones al diseño preliminar y crítico.

# Documentación

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

## Requerimientos mínimos de documentación

Es la mínima para asegurar que la implementación logrará satisfacer los requerimientos.

### Especificación de requerimientos de software (ERS)

Para lograr satisfacer los requerimientos debemos de tener un documento de ERS que describa 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. Si es necesario, se pueden construir 2 documentos para la especificación, uno técnico y otro más orientado al cliente, para esto, se debe aplicar SCM a cada versión.

La especificación debe ser:

* Ser completa :

a. Externa, respecto al alcance acordado.

b. Internamente, no deben existir elementos sin especificar.

* Ser consistente, no pueden haber elementos contradictorios.
* Ser no ambigua, todo término referido al área de aplicación debe estar definido en un glosario.
* Ser verificable, debe ser posible verificar siguiendo un método definido, si el producto final cumple o no con cada requerimiento.
* Estar acompañada de un detalle de los procedimientos adecuados para verificar si el producto cumple o no con los requerimientos.
* Incluir requerimientos de calidad del producto a construir.

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

*Funcionalidad*

a. adecuación a las necesidades

b. precisión de los resultados

c. interoperabilidad

d. seguridad de los datos

*Confiabilidad*

a. madurez

b. tolerancia a faltas

c. recuperabilidad

*Usabilidad*

a. comprensible

b. aprendible

c. operable

d. atractivo

*Eficiencia*

a. comportamiento respecto al tiempo

b. utilización de recursos

*Mantenibilidad*

a. analizable

b. modificable

c. estable, no se producen efectos inesperados luego de modificaciones

d. verificable

*Portabilidad*

b. instalable

c. co-existencia

Cada uno de los anteriores debe cumplir con las normas y regulaciones aplicables a cada atributo en su contexto.

### Descripción del diseño de software (DDS)

El documento de diseño especifica como el software será construido para satisfacer los requerimientos. Deberá describir los componentes y sub-componentes 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, y/o tener un valor agregado para una solución que agregue atributos de calidad al software.

1. 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

Para la revisión del diseño, se presentan las checklist en el inciso 9.

### Plan de verificación y validación de software (PVVS)

El Plan de Verificación y Validación deberá identificar y describir los métodos a ser utilizados en:

* La verificación de que:
  1. los requerimientos descritos en el ERS han sido aprobados por una autoridad apropiada. En este caso sería que cumplan con el acuerdo logrado entre el cliente y el equipo.

1. los requerimientos descritos en el ERS son implementados en el diseño expresado en el documento de diseño.
2. el diseño expresado en el DDS está implementado en código.

* Validar que el código, cuando es ejecutado, se adecua a los requerimientos expresados en ERS.

### Reporte de verificación y validación de software (RVVS)

Estos documentos deben especificar los resultados de la ejecución de los procesos descritos en el PVVS.

### 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.

### Plan de gestión de configuración de software (PGCS)

El PGCS debe contener métodos para identificar componentes de software, control e implementación de cambios, y registro y reporte del estado de los cambios implementados. Referirse a “IMP2009\_Plan\_SCM”.

## Otra documentación

### Plan de desarrollo de software (PDS)

El plan de desarrollo de software debe contener la planificación de las actividades a realizarse durante el proceso de desarrollo. Debe estimar el esfuerzo necesario para la realización del producto de forma de asegurar el nivel de calidad esperado. Debe especificar las responsabilidades de los integrantes del equipo de desarrollo así como el tiempo de dedicación requerido en cada una de las actividades asignadas. Se debe definir en este documento el alcance del proyecto y las limitaciones del mismo, especificando que tareas se realizarán y cuáles no. Debe brindar una descripción general del proceso de desarrollo y como serán implementadas las etapas del mismo.

Incluirá una estimación inicial de la agenda y una explicación del procedimiento a seguir para ajustarla. El Plan debe especificar los entregables requeridos en cada etapa y la prioridad de los mismos, de forma de asignar más esfuerzo a hitos más importantes para la calidad del producto.

Debe especificarse en el mismo las pautas a seguir en la actividad de ajuste del Plan, que variables serán analizadas y en qué medida se cuantificará cada una. Es responsabilidad de los integrantes del equipo de SQA brindar apoyo en las cuantificaciones de calidad y esfuerzo.

### Manual de estándares y procedimientos (MEP)

Los nombres de los documentos deben ser los nombres originales de las plantillas, salvo las siguientes excepciones: Los documentos que se entreguen más de uno por semana, o sea, los cuales el mismo nombre de entregable aplica a mas de una vez por semana (ej. Reunión de requerimiento), deberán llamarse: NOriginal\_DDMMAAAAvX.Y donde NOriginal es el nombre original del documento (SQAPLAG1), DD, MM, AAAA son día mes y año en cual se realizó la actividad respectivamente y X.Y es número de versión. Luego, los documentos que no evolucionen (cambien de versión) y se entreguen semanalmente o en una determinada semana, serán nombrados: NOriginal\_SemNvX.Y donde N es la semana en la que se realizó la actividad.

### Plan de gestión de proyecto de software (PGPS)

El plan de gestión de proyectos sw (spmp) es el documento controlador para un proyecto sw.

Este especifica los enfoques técnicos y administrativos a ser usados en el desarrollo de un producto sw. o de un componente sw. de un producto extenso. Los planes relacionados para el proyecto (por ejemplo: gestión de la configuración, aseguramiento de la calidad, verificación y validación) son considerados parte del SPMP y ellos deben ser incorporados directamente o por referencia en el SPMP. El plan debe especificar las funciones, actividades y tareas técnicas y administrativas, en suficiente detalle para asegurar que el resultado del proyecto satisfaga las necesidades, requerimientos y acuerdos establecidos del proyecto.

### Manual de mantenimiento de software

Otorgar servicios de soporte técnico, así como mantener el buen funcionamiento de los sistemas, con el fin de que todo el personal del órgano interno de control cuente con las herramientas informáticas necesarias para llevar a cabo sus labores.

### Plan de métricas de software

El propósito de este documento es detallar el plan de Métricas a utilizar en el proyecto.

Abarca la definición de las métricas a utilizar, un resumen sobre los resultados que se esperan obtener de estas métricas, incluyendo una descripción sobre como se utilizarán para poder controlar el avance del proyecto.

Referirse a “IMP2009\_PlanMetricas”.

# Estándares, prácticas, convenciones y métricas

El rol de la Gerencia de SQA es de asistir al jefe de proyecto a monitorear la correcta interpretación de los estándares, prácticas y convenciones. Es importante definir las métricas para la medición de calidad. Estas métricas deben ser cuantitativas y deben ser capaces de medir ciertos atributos como calidad, fiabilidad y mantenimiento. Ellas deben tener cierto significado para los usuarios y son parte del proyecto más que parte de un estándar. Estas métricas deben ser especificadas en el Plan de métricas para la ayuda de toma de decisiones. Por ejemplo si un software exhibe tres o más fallas en la fase de prueba, éste debe ser inspeccionado.

La adherencia a todos los estándares, prácticas y convenciones debieran monitorearse. En caso de desviaciones y disconformidades debiera notificarse al gerente de proyectos

## Estándar de documentación

Como estándares de documentación se definirán dos documentos, se debe unificar el uso de herramientas para documentar con uniformidad todos los documentos. La documentación debe reflejar con los siguientes atributos de calidad:

* + Legibilidad
    1. Estructura
    2. Tamaño
    3. Ilustraciones
    4. Facilidad para ubicar información relevante
  + Completo
  + Correcto

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.
* Historia de revisiones
* Índice de contenido

La responsabilidad de apegarse a los estándares definidos es del encargado del entregable, definido en el plan de iteración, el cual es resultado en primera instancia por la reunión quincenal del grupo, ajustado luego de la reunión de responsables por área, y finalmente definida por el administrador del equipo. El equipo de SQA revisará el cumplimento de éstos estándares y reportará eventuales defectos a los correspondientes responsables, auditando y formando el ciclo de defectos.

### Estándar de documentación técnica e implementación

La misma debe:

* Ser adecuada para que un grupo independiente del de desarrollo pueda encarar el mantenimiento del producto. Esto es importantísimo del punto de vista del actual cliente.
* Incluir fuentes, Modelos de Casos de Uso, Objetos

### Estándar de documentación de usuario

Existe un documento para éste propósito, referirse a [ISW\_Estándar\_Doc\_Usuario.doc](http://www.fing.edu.uy/~pgcalpro/Plant/implantacion/IPEDUGXvY.doc)x.

### 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. Referirse al PVVS (Plan de Verificación y Validación, ISW\_Plan\_Verificación\_Validación.docx).

## Otros estándares

### Estándares de programación

Los estándares de programación fomentan buenas prácticas en las herramientas en las cuales se utilizan, ayudan a terceros a entender las soluciones y evitan malas prácticas y/o enfoques. Se definieron nomenclaturas para el estándar, basados en el estándar de Sun <http://java.sun.com/docs/codeconv/> .

# Revisiones y auditorias

Propósito

* Definición de las revisiones y auditorias técnicas y de gestión que se realizarán.
* Especificación de cómo serán llevadas a cabo dichas revisiones y auditorias.
* Revisar y auditar objetivamente los productos y las actividades para verificar que están conformes con los procedimientos y estándares aplicables.
* Proporcionar los resultados de estas revisiones o auditorias informando a la directora del proyecto cuando sea necesaria su mediación.

Requerimientos mínimos

Todos los entregables son revisados semanalmente por el equipo SQA, los mismos se basan en los estándares y prácticas definidas, lo primero que se audita es quién hace las revisiones, lo cual queda en el historial de revisiones y además se formula el documento “Entrega Semanal de SQA”.

### Revisión de requerimientos de software (RRS)

Esta revisión se realiza para asegurar que se cumplió con los requerimientos especificados por el Cliente. El responsable de SQA no es a menudo el mejor recurso para hacer esta revisión, pues, el mismo no conoce tan a fondo los requerimientos como otros integrantes del staff, por lo que puede ser apropiado que participe un asistente en el mismo. El mismo debe de estar organizado según los diferentes conceptos (como lo pide el cliente), por ejemplo:

 X -   Actividad

        Explicar que es la actividad

        X.1 - Requerimiento 1 sobre actividad

### Revisión de diseño preliminar (RDP)

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

### Revisión de diseño crítico (RDC)

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

### Revisión del Plan de Verificación & Validación (RPVVS)

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

### 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.

### Auditoría física

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

### 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.

## Otros

### Revisiones de gestión

Estas revisiones se realizan periódicamente para asegurar la ejecución de todas las actividades identificadas en este Plan. Si es posible, se procurará realizar por una persona ajena al grupo de trabajo, por ejemplo el director del proyecto.

### Revisión del Plan de gestión de configuración (RPGCS)

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

### 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, y aunque no hay un mejor momento de realizarla, se fijará una agenda para la misma.

### Revisión de documentación de usuario

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

# Prueba

Hacer pruebas de unidad, integración, sistema y aceptación de ejecutables es esencial para asegurar la calidad del software. Planes, diseños, casos, procedimientos y reportes de pruebas son descritos en el [PVVS](#SVVP). Estos deberán ser revisados por personal de [SQA](#SQA). Se deberá monitorear las actividades de prueba llevadas a cabo por el personal de desarrollo, incluyendo la ejecución de la prueba. Adicionalmente, otras pruebas pueden ser propuestas en el [SQAP](#SQAP), las cuales deben llevarse a cabo por el personal de [SQA](#SQA)

Las pruebas se encuentran en su totalidad en el PVVS, lo único que se le agrega es utilizar los métodos y herramientas definidas en la Sección 9.

## Pruebas de Unidad

Las pruebas de unidad sólo se llevarán a cabo cuando se le haya hecho un seguimiento al código fuente.

**Propósito –** Las pruebas de unidad verifican la lógica, los cálculos/funcionalidad, y manejo de errores de la unidad.

**Plan –** Es usualmente escrito por el desarrollador de la unidad, y son informales. Los procedimientos para probar algoritmos muy complejos en el ámbito de la unidad estarán contenidos en el plan de pruebas general.

**Participantes –** El desarrollador de la unidad.

**Actividades –** El probador prepara los datos de prueba y cualquier dato ficticio. Luego él/ella ejecuta el plan de prueba, verificando todos los caminos lógicos, las condiciones de error, y las condiciones límites. Los resultados de las pruebas serán revisados para asegurar su corrección por el líder del equipo o la autoridad técnica que se designe. (Si la prueba es extensa se utilizará la hoja de plan de pruebas que aparece en los anexos).

**Monitoreo –** Se utilizará el número de unidades planeadas versus el número de unidades probadas; también se hará seguimiento al número de errores encontrados para medir la confiabilidad del software.

## Pruebas de Integración

**Propósito –** Las pruebas de integración verifican la integridad interna de una colección de unidades (llamadas módulos) relacionadas lógicamente y chequean las interfaces externas con otros módulos, archivos de datos, datos de entrada/salida externos, etc.

**Plan –** Aunque un plan formal de pruebas generalmente no es requerido, las pruebas de integración si se controlarán más cuidadosamente que las pruebas de unidad.

**Participantes –** Será realizada por los miembros del equipo de desarrollo responsables del módulo que esté siendo integrado.

**Actividades –** Durante las pruebas de integración, el software se irá agregando lentamente al adicionar unas pocas unidades al conjunto de módulos que ya hayan sido integrados. La integración seguirá un enfoque de arriba hacia abajo.

**Monitoreo --** Se utilizará el número de unidades planeadas versus el número de unidades integradas; también se hará seguimiento al número de errores encontrados para medir la confiabilidad del software.

## Pruebas del Sistema

**Propósito –** Las pruebas del sistema verifican que el sistema completo satisfaga sus requerimientos funcionales y operacionales. El sistema tiene que demostrar ser tanto funcionalmente correcto como robusto antes de que sea entregado para las pruebas de aceptación.

**Plan –** El plan de pruebas del sistema será con base en las pruebas de aceptación y retomará todos aquellos planes de pruebas previos (de unidad o integración) que hayan sido diseñados para mostrar cómo se puede verificar la corrección computacional del sistema.

**Participantes –** El equipo de pruebas del sistema está compuesto por los desarrolladores y guiados por una persona especialista en la aplicación desarrollada.

**Actividades –** La ejecución de las pruebas seguirá el patrón que se haya prescrito en el plan de pruebas del sistema. Las discrepancias entre los requerimientos del sistema y los resultados de las pruebas serán identificadas y asignadas a los miembros del equipo de desarrollo para su corrección. Cada cambio en el sistema será manipulado en concordancia con lo dispuesto en el procedimiento de control de la configuración. Se harán pruebas de regresión después que las correcciones hayan sido hechas para asegurar que los cambios no tienen efectos colaterales no intencionados. Las pruebas del sistema se harán hasta que no se identifiquen más errores.

**Monitoreo –** Se utilizará el número de casos de pruebas identificados versus el número probados exitosamente; se hará un seguimiento formal de los defectos hallados para medir la confiabilidad; se hará un seguimiento al esfuerzo requerido para hacer las correcciones para predecir la mantenibilidad del sistema.

## Pruebas de Aceptación

**Propósito –** Verifican que el sistema satisface sus requerimientos

**Plan –** Todas las pruebas de aceptación serán con base en el plan de pruebas de aceptación escrito por los analistas de manera previa al inicio de la fase.

**Participantes –** Las pruebas son ejecutadas por el equipo de aceptación soportados por los miembros del equipo de desarrollo. El equipo de aceptación incluirá representantes del equipo de aseguramiento de la calidad y del usuario.

**Actividades –** Después de que las preparaciones para las pruebas se hayan terminado, el equipo de aceptación intentará ejecutar todas las pruebas en el plan, tratando de bordear cualquier error descubierto. Esto asegura que los problemas mayores sean descubiertos de forma temprana en el proceso. Solamente cuando la ejecución de un módulo obstruya totalmente el proceso de pruebas, se suspenderá el proceso hasta que se disponga de un nuevo módulo. Cada módulo corregido será sometido a pruebas de regresión para asegurar que las capacidades previamente demostradas no hayan sido afectadas adversamente por las correcciones de errores. (Se utilizará la hoja de pruebas de aceptación que aparece en los anexos).

**Monitoreo –** Se utilizará el número de casos de pruebas identificados versus el número probados exitosamente; se hará un seguimiento formal de los defectos hallados para medir la confiabilidad; se hará un seguimiento al esfuerzo requerido para hacer las correcciones para predecir la mantenibilidad del sistema.

# Reporte de problemas y acciones correctivas

Para realizar el seguimiento de los problemas identificados en el producto de software en desarrollo se realizarán Reuniones de Inspección y Auditorias a la empresa Impresiones 2009 (Auditorias de Proyecto, Auditorias de Configuración Funcional y Auditorias de Configuración Física).

En las reuniones de Inspección se llevan a cabo las siguientes actividades:

* Elegir Equipo, preparar Material y calendario.
* Presentar proceso y producto.
* Análisis individual para encontrar potenciales defectos.
* Análisis del equipo para recolectar potenciales defectos previos, filtrar falsos positivos y encontrar más.
* Corregir Defectos.
* Verificar correcciones, recolectar datos.

Las acciones correctivas especificarán responsable, descripción de las mismas y seguimiento. Deben especificarse por escrito, de modo que quede constancia de las mismas.

El responsable de planificar las reuniones y auditorias será el Moderador (que puede ser también el Líder de Proyecto).

El responsable de registrar los hallazgos encontrados será el Anotador.

El Inspector será el responsable de examinar el producto antes de la reunión para encontrar defectos y registrar los tiempos de preparación.

Habrá también lectores que leen los productos a ser inspeccionados.

# Herramientas, técnicas, y metodologías

QStudio for Java® es una herramienta para analizar la “calidad” de un código Java, muestra errores y posibles errores incurridos y soluciones sugeridas para él. El responsable SQA se encargará de utilizar la herramienta y capacitar a quien sea necesario, según se planifique posteriormente.

La herramienta que se utilizará para realizar las actividades de aseguramiento de calidad del software será el check list.

Aquí los formatos de a tener en cuenta para hacer los seguimientos de calidad del Software:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NOMBRE DE LA COMPAÑÍA | | | | | | | | | | | | CONFIDENCIAL | | | | | | | | Reporte de Defecto # | | | | | | | | | | | |  |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PROGRAMA | |  | | | | | | | | | | | | | | | | | ENTREGA | | |  | | | | | VERSIÓN | | | |  | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TIPO DE REPORTE (1-6)\_\_\_ | | | | | | | | | | | | | | | | SEVERIDAD(1-3)\_\_\_ | | | | | | | ANEXOS (S/N)\_\_\_ | | | | | | | | | |
| 1 – Error Codificación | | | | | | | | 4 – Documentación | | | | | | | | 1 - Fatal | | | | | | | *En caso afirmativo, describa:* | | | | | | | | | |
| 2 – Asunto de Diseño | | | | | | | | 5 – Hardware | | | | | | | | 2 – Serio | | | | | | |  | | | | | | | | | |
| 3 – Sugerencia | | | | | | | | 6 – Consulta | | | | | | | | 3 - Menor | | | | | | |  | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SUMARIO PROBLEMA | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PUDO REPRODUCIR EL PROBLEMA (S/N)\_\_\_ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PROBLEMA Y COMO REPRODUCIRLO | | | | | | | | | | | | |  | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ARREGLO SUGERIDO (opcional) | | | | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| REPORTADO POR | | | | |  | | | | | | | | | | | | | | | | | | | | FECHA | | | | \_\_/\_\_ /\_\_ | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LOS ITEMES QUE APARECEN ENSEGUIDA SON PARA USO EXCLUSIVO DEL EQUIPO DE DESARROLLO | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AREA FUNCIONAL | | | | | |  | | | | | | | | | | | | ASIGNADO A | | |  | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| COMENTARIOS | | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ESTADO (1-2)\_\_\_ | | | | | | | | | | | | | | | | | | | | | | | | PRIORIDAD(1-5)\_\_\_ | | | | | | | | |
| 1 – Abierto | 2 – Cerrado | | | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RESOLUCION (1-9)\_\_\_ | | | | | | | | | | | | | | | | | VERSION CON RESOLUCION | | | | | | | | | | |  | | | | |
| 1 – Pendiente | | | | 4 – Diferido | | | | | | | | | | | 7 – Retirado por el que lo reportó | | | | | | | | | |  | | | | | | | |
| 2 – Arreglado | | | | 5 – Diseñado Así | | | | | | | | | | | 8 – Necesito mas información | | | | | | | | | |  | | | | | | | |
| 3 - Irreproducible | | | | 6 – No puede ser arreglado | | | | | | | | | | | 9 – En desacuerdo con la sugerencia | | | | | | | | | |  | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RESUELTO POR | | | |  | | | | | | | | | | | | | | | | | | | | | | FECHA | | | | \_\_/\_\_ /\_\_ | | |
| ESFUERZO REQUERIDO | | | | | | | | |  | | | | | | | |  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RESOLUCION PROBADA POR | | | | | | | | | |  | | | | | | | | | | | | | | | | FECHA | | | | \_\_/\_\_ /\_\_ | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TRATARLO COMO DIFERIDO (S/N)\_\_\_ | | | | | | | | | | | | | |  | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Hoja de Inspecciones

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: |  | | Fecha Inspección Inicial: | | | |  | |
| Nombre de la Unidad: |  | | | | | | | |
| Moderador Inspección |  | | | | | | | |
|  | | | | | | | | |
| **Preguntas Claves de la Inspección** | | | | | **Si** | **No** | | **Corregido** |
| 1. Algún argumento de entrada no se usa? Algún argumento de salida no se produce | | | | | [ ] | [ ] | | [ ] |
| 1. Algún tipo de dato es incorrecto o inconsistente? | | | | | [ ] | [ ] | | [ ] |
| 1. Algún algoritmo es inconsistente con el especificado en el Seudocódigo o en los requerimientos/especificaciones? | | | | | [ ] | [ ] | | [ ] |
| 1. Alguna variable local es utilizada antes de ser inicializada? | | | | | [ ] | [ ] | | [ ] |
| 1. Alguna interface externa está codificada incorrectamente? Es decir, una sentencia de llamado o acceso al archivo/base de datos es incorrecta? | | | | | [ ] | [ ] | | [ ] |
| 1. Algún camino lógico es incorrecto? | | | | | [ ] | [ ] | | [ ] |
| 1. La unidad tiene múltiples puntos de salida? | | | | | [ ] | [ ] | | [ ] |
|  | | | | | | | | |
| **Preguntas Adicionales de la Inspección** | | | | |  |  | |  |
| 1. Alguna parte del código es inconsistente con el diseño especificado en seudocódigo? | | | | | [ ] | [ ] | | [ ] |
| 1. El código o plan de pruebas tiene alguna desviación no autorizada de los estándares del proyecto? | | | | | [ ] | [ ] | | [ ] |
| 1. El código contiene algún mensaje de error que podría no ser claro para el usuario? | | | | | [ ] | [ ] | | [ ] |
| 1. Si la unidad fue diseñada para ser reutilizada, tiene algún impedimento, para ser reutilizada, introducido en el código? | | | | | [ ] | [ ] | | [ ] |
|  | | | | | | | | |
| Acciones y Comentarios | | | | | | | | |
| (Enumérelos en una hoja aparte. Refiérase a las preguntas anteriores por su número correspondiente.) | | | | | | | | |
|  | | | | | | | | |
| **Resultados de la Inspección** | | | | |  |  | |  |
| 1. Si todas las respuestas de la 1 a la 11 fueron “No”, la unidad es aceptada. Chequee aquí y firme abajo | | | | | [ ] |  | |  |
| 1. Si hay una deficiencia seria en el código (e.g. si más de una de las preguntas clave se respondió “Si”) el autor tiene que corregir el diseño de la unidad y el moderador tiene que programar una re-inspección. | | | | |  |  | |  |
| Fecha programada para la re-inspección: | | | | |  | | | |
| 1. Si hay deficiencias menores en el código, el autor tiene que corregir el diseño de la unidad y tener una reunión de seguimiento con el moderador | | | | |  |  | |  |
| Fecha programada para la reunión de seguimiento: | | | | |  | | | |
|  | | | | | | | | |
| La firma del moderador certifica que esta unidad cumple con todos los estándares aplicables y satisface sus requerimientos, y que cualquiera deficiencia identificada ha sido resuelta (aplicable a la inspección inicial, reunión de seguimiento o re-inspección). | | | | | | | | |
|  | | | | | | | | |
| Firma del Moderador | |  | | Fecha: | | |  | |

Plan de Pruebas – Nro. \_\_\_\_\_\_\_\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Módulo: |  | | Fecha: |  |
| Proceso: |  | | | |
|  | | | | |
| Antecedentes | | | | |
| **Objetivos:** |  | | | |
|  | | | | |
|  | | | | |
|  | | | | |
| Procedimiento **(Enumere la Secuencia General de Pasos a Seguir para Ejecutar la Prueba)** | | | | |
| 1. | |  | | |
|  | | 7. | | |
| 2. | |  | | |
|  | | 8. | | |
| 3. | |  | | |
|  | | 9. | | |
| 4. | |  | | |
|  | | 10. | | |
| 5. | |  | | |
|  | | 11. | | |
| 6. | |  | | |

Lista de Chequeo **(Cosas Específicas a Verificar)**

|  |  |  |
| --- | --- | --- |
| Evento de Entrada o Salida | Clases Equivalentes Válidas | Clases Equivalentes Inválidas |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Nota: Si necesita espacio adicional utilice otra hoja o el reverso de esta**

Prueba de Aceptación – Nro. \_\_\_\_\_\_\_\_

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Módulo: | |  | | | | | | | Fecha: | |  | | |
| Proceso: | |  | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
| Antecedentes | | | | | | | | | | | | | |
| **Objetivos:** | |  | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
| **Suposiciones:** | | |  | | | | | | | | | | |
|  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
| **Datos:** | **Preparados** | | | | **Reales** | | | **Improvisados** | | | | | |
|  | | | | | | | | | | | | | |
| Procedimiento | | | | | | Comentarios | | | | | | | |
|  | | | | | |  | | | | | | | |
|  | | | | | |  | | | | | | | |
|  | | | | | |  | | | | | | | |
|  | | | | | |  | | | | | | | |
|  | | | | | |  | | | | | | | |
|  | | | | | |  | | | | | | | |
|  | | | | | |  | | | | | | | |
|  | | | | | |  | | | | | | | |
|  | | | | | |  | | | | | | | |
|  | | | | | |  | | | | | | | |
|  | | | | | |  | | | | | | | |
| Lista de Chequeo | | | | | | | | | | | | **Si** | **No** |
|  | | | | | | | | | | | |  |  |
|  | | | | | | | | | | | |  |  |
|  | | | | | | | | | | | |  |  |
|  | | | | | | | | | | | |  |  |
|  | | | | | | | | | | | |  |  |
|  | | | | | | | | | | | |  |  |
|  | | | | | | | | | | | |  |  |
|  | | | | | | | | | | | |  |  |
|  | | | | | | | | | | | |  |  |
| Problemas Encontrados **(Descripción del problema, y solución recomendada si es posible)** | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | |
| **Aprobaciones** | | | | | | | | | | | | | |
| Nombre Usuario: | | | |  | | | Firma | | |  | | | |
| Conductor de la Prueba: | | | |  | | | Firma | | |  | | | |

# Capacitación

Se deberá capacitar a los profesionales en CMMI Nivel 2 para poder llevar a cabo la mejora de procesos por etapas.