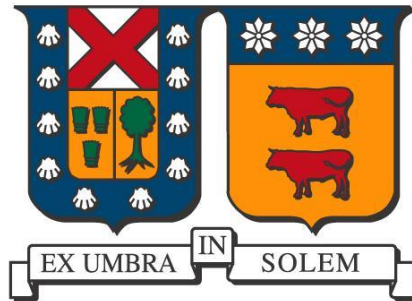


Universidad Técnica Federico Santa María

Departamento de Informática

Valparaíso, Chile



PLAN DE CALIDAD PARA “POST EMERGENCY SUPPORT SYSTEM”

**MEMORIA PARA OPTAR AL TÍTULO DE:
INGENIERO DE EJECUCION EN INFORMATICA**

Carlos Ramírez Espinoza

PROFESOR GUIA:

Marcello Visconti

PROFESOR CORREFERENTE:

Luis Hevia

FEBRERO 2017

RESUMEN

PESS (Post Emergency Support System) es una aplicación web con la cual se puede llevar registro de los trabajadores voluntarios y los recursos disponibles, para distribuirlos en distintos puntos de un mapa de una zona afectada por una catástrofe. Su objetivo es entregar a los dirigentes de las obras de reconstrucción una herramienta que registre que voluntarios son aptos de participar y guiarlos hacia los puntos en que sea necesario de sus habilidades.

El objetivo de esta memoria es confeccionar el plan de calidad que debe seguir esta aplicación software para garantizar el cumplimiento de los estándares internacionales de calidad, con el fin de asegurar esta supere las expectativas de los usuarios.

ABSTRACT

PESS (Post Emergency Support System) is a web application in which you can keep track of the volunteer workers and the resources available, to distribute them in different points of a map of an area affected by a catastrophe. Its objective is to give the leaders of the reconstruction works a tool to register which volunteers are able to participate in the works and lead them to the zones in which their abilities could be required.

The objective of this work is to prepare the quality plan that this software application must follow to ensure compliance with international quality standards, in order to ensure that it exceeds the expectations of the users.

INDICE

1. INTRODUCCION 4

2. REQUERIMIENTOS NO FUNCIONALES 6

 2.1 Definición de Atributos de Calidad 6

 2.2 Objetivos Cuantificables 6

 2.2.1 Funcionalidad 6

 2.2.2 Eficiencia 7

 2.2.3 Fiabilidad.....7

 2.2.4 Usabilidad 8

 2.2.5 Mantenibilidad 8

 2.2.6 Portabilidad 9

3. PROCESO DE DESARROLLO 10

 3.1 Elección de modelo de desarrollo 10

 3.1.1 Descripción de la metodología 10

 3.1.2 Beneficios de la metodología 11

 3.1.3 Pila de requerimientos 11

 3.2 Definición de Actividades.....12

 3.2.1 Especificación de Requerimientos.....12

 3.2.2 Planificación.....12

 3.2.3 Diseño 12

 3.2.4 Implementación por medio de Sprint 13

 3.2.5 Instalación.....14

 3.2.6 Operación 15

 3.3 Definición de Productos de Trabajo.....15

 3.3.1 Project Backlog 15

 3.3.2 Plan de Proyecto 16

 3.3.2 Plan de Gestión de riesgos 17

 3.3.4 Sprint Backlog 17

 3.3.5 Burn Down 18

 3.3.6 Plan de Aseguramiento de Calidad (SQA).....18

 3.3.7 Plan de Pruebas 18

 3.3.8 Plan de Gestión de Configuración (SCM).....19

 3.3.9 Manual de Usuario 19

 3.4 Atributos de Calidad 20

 3.4.1 Definición de los Atributos de Calidad 20

 3.4.2 Atributos de calidad por actividades del proceso de desarrollo 21

 3.4.3 Atributos de calidad por productos de trabajo 23

 3.5 Puntos de revisión (hitos) 26

4. GESTION DE CALIDAD 27

 4.1 Equipo de trabajo.....27

4.2 Recursos 27

4.2.1 Recursos Humanos 27

4.2.2 Infraestructura.....29

4.3 Actividades de SQA 29

4.4 Responsabilidades.....31

4.5 Análisis de Riesgos 33

4.5.1 Identificación de Riesgos 33

4.5.2 Clasificación de Riesgos 34

4.5.4 Riesgos Priorizados 36

4.5.4 Control de Riesgos 38

5. HERRAMIENTAS, TECNICAS Y METODOLOGÍAS.....48

5.1 Aplicación de métodos técnicos formales 48

5.2 Registro y generación de informes 48

5.3 Revisiones 48

5.3.1 Roles y responsabilidades.....48

5.3.2 Etapas de Revisión 50

5.4 Auditoria 56

5.4.1 Aplicación 56

5.4.2 Roles y responsabilidades.....57

5.4.3 Etapas 59

5.5 Checklist 63

6. PRUEBAS 64

6.1 Estructura de las pruebas 64

6.1.1 Planificación.....64

6.1.2 Especificación 65

6.1.3 Ejecución.....65

6.1.4 Análisis de resultados 65

6.1.5 Finalización 66

7. CONCLUSIONES.....69

8. GLOSARIO 71

9. BIBLIOGRAFIA 72

10. ANEXOS.....73

5.3.3 Pauta de Informe de Revisión.....73

5.4.4 Pauta de Informe de Auditoria 75

5.5 Pauta de Checklist.....78

5.5.1 Checklist por actividades del proceso de desarrollo evaluados por QA.....78

5.5.2 Checklist por productos de trabajo evaluados por QA.....81

5.5.3 Checklist de Revisión 91

5.5.4 Checklist de Auditoria.....95

1. INTRODUCCION

A través del siguiente documento se busca establecer el plan de calidad para el desarrollo de una versión funcional de la aplicación PESS (Post Emergency Support System). Actualmente PESS se encuentra en su versión inicial, la cual consiste en un prototipo funcional, pero que no cumple con los estándares necesarios para considerarse un producto comercializable en la sociedad actual.

A continuación, se presenta en qué contexto surge la aplicación PESS, y las problemáticas que busca solucionar.

Chile es un país que se enfrenta constantemente a desastres naturales. Terremotos, aluviones y erupciones volcánicas son parte del acontecer cotidiano en distintos puntos de todo el territorio nacional y tienen un impacto geográfico y psicosocial, pero también económico.

Uno de los desastres naturales que se da con mayor frecuencia en Chile, y sobre el cual este proyecto toma directa inspiración, son los incendios.

Cada año, entre los meses de Octubre y Mayo se constatan cerca de mil denuncias por incendios forestales, siendo la mayor parte de las denuncias realizadas en el sector sur de Chile.

Los incendios forestales, además de ser extremadamente difíciles de contener debido a los recursos y mano de obra requeridos para su control, dejan a su paso tremendos daños estructurales, ya sea en grandes ciudades como en hectáreas y hectáreas de bosques.

Con un promedio de 128 incendios cada 5 años que conllevan a una pérdida anual de aproximadamente 50 mil hectáreas al año tienen como causa la poca cultura que se tiene con el cuidado del medioambiente así como la falta de estudios e investigaciones en el área de incendios forestales.

Cuando se dan situaciones en que una ciudad sufre daños considerables producto de un incendio, luego de que se logra controlar su esparcimiento y se comienzan a normalizar los servicios primarios dentro de la ciudad, distintas entidades del gobierno organizan grupos de ayuda para las víctimas, siendo estos conformados principalmente por las fuerzas armadas con la misión de mantener el orden público. En el caso de Chile, en momentos de tragedia es un país muy solidario y ante estas situaciones surgen iniciativas de voluntarios, de vecinos, de pequeños empresarios y de organizaciones locales, haciéndose parte de los trabajos de respuesta ante una emergencia.

Sin embargo, al momento de la organización de grupos de ayuda voluntarios, surgen ciertos problemas, ya sea que algunos de estos grupos de voluntarios actúan por su cuenta, o no cuentan con comunicación con los otros grupos, o alguno de los grupos lo componen personas que no cuentan con las herramientas o con las condiciones físicas o médicas para participar de esta actividad.

Tomando en cuenta estas consideraciones, y concretamente los eventos acontecidos en Valparaíso durante el mes de Abril del 2014, fecha en la cual se registró un incendio de grandes proporciones que terminó con daños considerables y muchas casas totalmente destruidas, la aplicación PESS se desarrolla con el objetivo de coordinar la ayuda posterior a esta clase de eventos, tanto ayuda material como los trabajos realizados por voluntarios y las herramientas necesarias para realizar estos trabajos.

Utilizando la aplicación PESS, se podrá coordinar en un área fija (previamente seleccionada), funciones como registrar los puntos de trabajo, inscribir voluntarios, asignarles materiales o herramientas, y también asignar voluntarios a las distintas tareas de ayuda que requieran de ellos para restablecer el sector afectado por la catástrofe.

El propósito de este documento es definir un Plan de Calidad para la aplicación web PESS para posteriormente desarrollar una nueva versión.

La aplicación PESS, al momento de redactar este documento, se encuentra en su primera versión, la cual consiste en un prototipo funcional que cuenta con todas las funcionalidades establecidas en el plan de proyecto. Sin embargo, esta no se encuentra en un estado óptimo para ser comercializada, esto debido a que no cuenta con los estándares de calidad apropiados.

El alcance de este documento será definir las metodologías de trabajos en las distintas etapas de la creación del software, desde la planificación, pasando por el diseño y la metodología de desarrollo, hasta la puesta en marcha y futura mantención de la plataforma.

Se busca el poder establecer los requerimientos no funcionales, así como el definir los responsables, las actividades a realizar, los recursos y los riesgos a tomar en consideración al momento de realizar una versión funcional, más completa y abierta a nuevos mercados.

Cabe destacar que este Plan de Calidad dará principal enfoque a los requerimientos no funcionales que deben estar presentes en la aplicación y cómo estos requerimientos van a ser parte continúa del desarrollo del mismo.

2. REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales (RNF) consisten en propiedades, cualidades y restricciones que el sistema en desarrollo debe obedecer.

Los requerimientos no funcionales incluyen:

- **Restricciones:** Características que no pueden ser negociadas y que son impuestas por el cliente como guía para el sistema.
- **Atributos de Calidad:** Atributos que engloban la forma como se logran los requerimientos funcionales.

2.1 Definición de Atributos de Calidad

Siguiendo el estándar ISO 9126, estándar internacional para la evaluación de la calidad del software, en un intento de identificar los atributos clave de calidad para el software, clasifica la calidad de software en las siguientes categorías:

1. **Funcionalidad:** Conjunto de atributos sobre la existencia de un conjunto de funciones y sus propiedades. Las funciones satisfacen necesidades expresas o implícitas de los usuarios.
2. **Eficiencia:** Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesarios bajo condiciones establecidas.
3. **Usabilidad:** Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.
4. **Fiabilidad:** Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.
5. **Mantenibilidad:** Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.
6. **Portabilidad:** Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra.

2.2 Objetivos Cuantificables

Todos los requerimientos no funcionales deben poder medirse según su grado de cumplimiento de forma que pueda asegurarse su presencia en el producto final. Es por esto que se deben establecer explícitamente objetivos cuantificables, que muestren de forma clara y objetiva, tanto para el equipo como para el cliente, que se están desarrollando una aplicación con estándares razonables de calidad que le permitan competir en el mercado. Para esto se necesita establecer los criterios de validación con los cuales estos objetivos se aprobarán para cada uno de los requerimientos antes descritos.

2.2.1 Funcionalidad

Para PESS, sus funcionalidades manejan coordenadas para distintos puntos dentro del área afectada. Es por esto que es fundamental tanto que sean datos precisos, como que sean datos recientemente actualizados.

Es de gran importancia que estas funcionalidades estén solo al alcance de usuarios que tengan la autoridad necesaria para manipularlas, por lo que se requiere tomar ciertas medidas de seguridad.

2.2.1.1 Objetivos

- Servidor con clave de acceso.
- Contraseñas para cuentas de usuarios deben cumplir requisitos de complejidad.
- Limitar accesos a funciones según tipo de usuario
- Funciones deben dar resultados correctos.

2.2.1.2- Criterios de evaluación

- Debe contener una mayúscula
- Debe contener un número.
- Funciones con acceso según nivel de acceso de usuario.
- Funciones deben dar resultados esperados utilizando los datos más recientes del sistema.

2.2.2 Eficiencia

En el caso de PESS, siendo una aplicación web, su eficiencia debe ser calculada a través de factores como la velocidad en que se muestra el mapa de la zona, la velocidad de respuesta de las peticiones y/o los ingresos de datos, la manera en que responde peticiones simultáneas, que el diseño sea responsivo a distintas plataformas, que funcione correctamente con los requerimientos mínimos establecidos y la cantidad de usuarios con que la aplicación comienza a fallar.

2.2.2.1 Objetivos

- Visualizar el mapa a una velocidad prudente.
- Tener una buena velocidad de respuesta ante peticiones.
- Manejo adecuado de estrés.

2.2.2.2 Criterios de evaluación

- La aplicación debe ser capaz de soportar hasta 50 usuarios conectados simultáneamente.
- Generar mapa de zona no debe tomar más de 30 segundos de carga.
- Conexión al servidor no debe superar los 20 segundos.
- Tiempo de procesamiento de peticiones no debe superar los 2 minutos.

2.2.3 Fiabilidad

Para el caso de PESS, como el de cualquier aplicación web, las peticiones y envíos de datos no toman un tiempo muy considerable, por ello se requiere medir el tiempo medio de fallos y reparación en un periodo de tiempo.

2.2.3.1 Objetivos

- Mantener nivel de funcionamiento frente a errores
- Evitar que posibles errores conlleven a fallas de sistema
- Capacidad, tiempo y costo para reestablecer un nivel de servicio y recuperar datos después de una falla (Recuperabilidad)

La medición de fiabilidad se realiza por medio de monitoreo de un sistema en un tiempo determinado. En este se determina el tiempo medio entre los fallos detectados y el tiempo medio de reparación de estos. La cantidad de fallos no es una medida útil porque el usuario se enfrenta a la aparición de fallos y no a su cantidad total. No todos los fallos serán detectados, ya sea porque pasen desapercibidos o que no se registren hasta después del lapso de tiempo establecido, por ello que se miden a través de la media.

2.2.3.2 Criterios de evaluación

- Funcionamiento con 95% de efectividad.
- La aplicación debe contar con funciones de recuperabilidad en caso de fallas.

2.2.4 Usabilidad

El principal problema de la usabilidad es que es demasiado abstracta para ser medida directamente. Esto debido a que su valor se determina por medio de los usuarios finales, por lo que sus valores dependen directamente del tipo de usuario que lo evalúe.

Para el caso de PESS, siendo una aplicación web para usuarios de distintas labores, la plataforma debe contar con una interfaz intuitiva, consistente y responsiva; tiene que ser fácil de usar, fácil de aprender a utilizar y que cuente con documentación clara para cualquier instrucción que pudiera ser poco clara.

2.2.4.1 Objetivos

- Interfaz intuitiva y fácil de dominar por usuarios regulares.
- Formato responsivo para distintos dispositivos.
- Documentación General.
- Baja cantidad de acciones necesarias para hacer una función.
- En caso de errores, mostrar mensajes que den detalles de lo que se hizo mal.

2.2.4.2- Criterios de evaluación

- Interfaz intuitiva, simple y de fácil aprendizaje.
- Resolución de pantalla 16:9 con dimensiones mínimas de 1024 × 768
- Formato responsivo para dispositivos móviles como smartphones y tablets.
- Debe contar documentación general para todas sus funcionalidades.

2.2.5 Mantenibilidad

Para el caso de PESS, como el de cualquier aplicación web, se modifica constantemente para adecuarse a su entorno. Esto ya que se trata de una aplicación que se utiliza en ambientes variados, en muchos casos inhóspitos.

Para toda aplicación web es importante el contar con una buena mantenibilidad, o sea ser capaz de adaptar la aplicación a diferentes tipos de entornos y que funcione correctamente con poco esfuerzo. Esto se debe a que este tipo de aplicaciones tienen un alto tráfico de información y peticiones de usuarios, además de que es fundamental que esta sea capaz de modificarse fácilmente durante su funcionamiento.

Es por ello que la aplicación debe contar con un plan de mantención que le permita recibir fácilmente mejoras o modificaciones, y también que esté preparada para cualquier riesgo que estos cambios puedan provocar sobre la aplicación.

2.2.5.1 Objetivos

- Ser capaz de detectar fallas o modificaciones.
- Se capaz de mejorarse o modificarse.
- Evitar efectos inesperados de fallas o modificaciones.
- Ser capaz de realizar pruebas a las modificaciones.

2.2.5.2 Criterios de evaluación

Para medir la mantenibilidad de la aplicación, hay que medir el tiempo medio que toma una modificación de la aplicación. Una vez estimado esto, se compara con los datos históricos de la aplicación para determinar el estado en que se encuentran.

- Tiempo medio de cambio < 24 horas.

2.2.6 Portabilidad

2.2.6.1 Objetivos

- La aplicación debe ser capaz de adaptarse a diferentes entornos.
- Esta debe ser fácil de instalar.
- Debe ser capaz de ser reemplazada por una versión nueva u otra aplicación.

2.2.6.2 Criterios de evaluación

Sea Portabilidad = $1 - (ET/ER)$

ER: Recursos necesarios (\$) para mover la aplicación al nuevo entorno.

ET: Recursos necesarios (\$) para crear al sistema para el entorno residente.

Para medir la portabilidad de la aplicación, hay que medir que comparar los valores de ER y ET con los que se dispone y ver como es la razón que hay entre estos.

- $ER < ET$

3. PROCESO DE DESARROLLO

En esta sección se detallan las técnicas y herramientas que se utilizan en el desarrollo del sistema.

3.1 Elección de modelo de desarrollo

Para el desarrollo de este proyecto se ha optado por una metodología inspirada en Scrum, debido a que toma elementos de esta pero también toma elementos de evaluación por etapas que siguen el modelo cascada. Esto debido a que se considera es la mejor opción para el tipo de proyecto en cuestión, siendo más específicamente por las siguientes razones:

- **Cultura empresarial:** Para una pre-empresa, esta metodología permite una rápida y mejor auto-organización en el equipo. Esto se logra mediante la asignación de roles fijos para los miembros, la asignación de tareas según lo que pueden aportar al desarrollo, el contar con constante comunicación verbal con las otras disciplinas durante las reuniones y un fácil conocimiento de que tareas están realizando los otros miembros del equipo.
- **Compromiso con el cliente:** Los requerimientos y avances del sistema se discuten con el cliente constantemente. Los requerimientos que tenga el cliente puede que cambien constantemente según la situación que se presente. Esta metodología está diseñada para ser flexible y modificar sus requerimientos conforme el cliente estime necesario.
- **Control del desarrollo:** Ya que se harán entregas del producto frecuentemente, luego de estas se realizarán reuniones de evaluación de proyecto. En estas reuniones, se controla si se cumplieron con las expectativas y metas propuestas en el ciclo anterior. Posteriormente se evalúan los requerimientos y se discuten posibles cambios y mejoras al entregable. Una vez decidido el camino a seguir, se asignan a los distintos miembros del equipo las tareas a realizar durante el siguiente ciclo, y se discute acerca las distintas disciplinas acerca de problemáticas que hayan surgido durante el ciclo y de posibles colaboraciones necesarias para realizar algunas de las tareas.
- **Rápida resolución de problemas:** Por medio de reuniones semanales el equipo puede identificar problemas que puedan surgir a tiempo. El resto de los miembros del equipo pueden ofrecer ayuda a otros en la realización de tareas o para resolver problemas que ya tuvieron anteriormente. El Scrum Master, se encargará de solucionar los impedimentos que el equipo no puede solucionar por sí solo o que le quitan tiempo para cumplir con su compromiso fundamental de desarrollo de requisitos.

3.1.1 Descripción de la metodología

Para este proyecto se opta por una metodología inspirada en Scrum.

Scrum es en una metodología de trabajo ágil y de forma iterativa que consiste en asignar roles fijo a los miembros del equipo de trabajo, y elaborar constantes entregables en plazos cortos. Estas iteraciones tomarán como prioridad los requerimientos críticos, y no se basan en el seguimiento de un plan. Esto permite una mayor flexibilidad ante las situaciones que se generen durante su desarrollo que requieran cambios en los requerimientos.

Dentro de este proyecto se realizarán Sprint enfocados en la construcción del sistema.

3.1.2 Beneficios de la metodología

Scrum es un modelo de desarrollo ágil y flexible que ha sido escogido no solo por su estructura sino también por proveer los siguientes beneficios:

- **Cumplimiento de expectativas:** El cliente establece sus expectativas indicando el valor que le aporta cada requisito / historia del proyecto, el equipo los estima y con esta información el Product Owner establece su prioridad. De manera regular, en las demos de Sprint el Product Owner comprueba que efectivamente los requisitos se han cumplido y transmite se feedback al equipo.
- **Flexibilidad a cambios:** Alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado. La metodología está diseñada para adaptarse a los cambios de requerimientos que conllevan los proyectos complejos.
- **Reducción del Time to Market:** El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo.
- **Mayor calidad del software:** La metódica de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.
- **Mayor productividad:** Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse.
- **Maximiza el retorno de la inversión (ROI):** Producción de software únicamente con las prestaciones que aportan mayor valor de negocio gracias a la priorización por retorno de inversión.
- **Predicciones de tiempos:** Mediante esta metodología se conoce la velocidad media del equipo por sprint (los llamados puntos historia), con lo que consecuentemente, es posible estimar fácilmente para cuando se dispondrá de una determinada funcionalidad que todavía está en el Backlog.
- **Reducción de riesgos:** El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada.

3.1.3 Pila de requerimientos

Es el documento de registro de los requisitos detallados o tareas que va a desarrollar el equipo técnico en la iteración (actual o que está preparándose para comenzar)

● Creación de Vistas	● Peticiones de cuadrillas
● Creación de BD	● Filtrar puntos de trabajo
● Registro de Coordinador	● Creación de Vistas
● Registro Post-Emergencia	● Creación de BD
● Registro de Recursos	● Registro de Coordinador
● Registro de Usuarios	● Registro Post-Emergencia
● Sesiones de Usuario	● Registro de Recursos
● Incrustar Mapa	● Registro de Usuarios
● Registrar punto de trabajo	● Sesiones de Usuario
● Mantenedor de Recursos	● Incrustar Mapa
● Listar Recursos disponibles	● Registrar punto de trabajo
● Mantenedor de Usuarios	● Trazar Rutas
● Listar Usuarios disponibles	● Remover o Liberar Puntos de
● Mantenedor y Gestor de Cuadrillas	Trabajo
● Asignación de puntos de trabajo en	

el mapa

- Señalizaciones en el mapa
- Trazar Rutas

● Señalizaciones en el mapa

3.2 Definición de Actividades

Las actividades que se realizarán durante el desarrollo de la aplicación se agrupan en las siguientes 6 etapas de desarrollo.

- Especificación de Requerimientos
- Planificación
- Diseño
- Implementación por medio de Sprint
- Instalación
- Operación

3.2.1 Especificación de Requerimientos

La metodología Scrum comienza con un Product Backlog, el cual consiste en la documentación de los requerimientos del cliente. A diferencia de otras metodologías este proceso se encuentra siempre en crecimiento y evolución, a fin de adaptarlo en los avances del proyecto. Una vez alcanzados todos los requerimientos se dice que el proyecto ha finalizado.

SQA debe corroborar que estén expresados todos los requerimientos de tipo funcionales y no funcionales. Es decir, definir todas las funcionalidades que tendrán el sistema y las características y restricciones que poseerán. Es importante que sean específicos para facilitar su verificación en el producto final.

3.2.2 Planificación

Es esta etapa se elabora el Plan de Proyecto, el cual consiste en una visión general del proyecto y el producto a desarrollar y una lista de los entregables del proyecto.

En este plan se debe definir la especificación del modelo de proceso del Proyecto, una descripción de la estructura organizacional del proyecto, identificación de interfaces e interacciones y definición de responsables.

Durante la planificación también se elabora el Plan de proyecto SQA, el cual permite que todo lo planificado durante esta etapa pueda ser medido según su grado de cumplimiento. Es imperante que este plan de proyecto cuente con herramientas claras para la evaluación de los requerimientos.

Al final de cada sprint, luego de las reuniones de retrospectiva, se evaluará la validez, correctitud y eficiencia del plan con la intención de realizar los cambios que se consideren necesarios para mejorar el proceso en lo que reste del proyecto. Esa evaluación será realizada por el Scrum Master en colaboración con el Responsable de SQA.

3.2.3 Diseño

En la fase de Diseño se debe definir el cómo se implementarán las soluciones previamente definidas en la fase de planificación. En esta fase se manejarán los recursos y el tiempo para cumplir los objetivos establecidos y con ello desarrollar el producto de manera correcta.

SQA se encarga en esta fase de:

- Verificar que los entregables realizados en la fase de diseño cumplan con las especificaciones definidas en el plan de proyecto durante la fase de planificación.
- Garantizar que los entregables de la fase de diseño fueron evaluados correctos.

De aprobar los puntos anteriores, el diseño se añade a la configuración.

SCM por otro lado informará periódicamente sobre los cambios de diseño realizados durante esta fase.

Cuando la fase haya finalizado, se establecerá el diseño resultante como Baseline, o diseño preliminar, el cual servirá de base para el desarrollo posterior y que solo puede cambiarse a través de procedimientos formales de control de cambios.

Luego se divide cada uno de estos requerimientos en segmentos, los que se conoce como Sprint. Cada Sprint es un pequeño lanzamiento que puede ser entre 2 semanas y 1 mes. Dentro de cada sprint se seleccionan requerimientos en un Sprint Backlog, se realizan reuniones semanales de trabajo, aproximadamente de 15 a 30 min. Donde se responderán 3 preguntas:

- 1.¿Qué hiciste la semana pasada?
- 2.¿Qué harás esta semana?
- 3.¿Qué impedimento tienes para realizar correctamente tu trabajo?

3.2.4 Implementación por medio de Sprint

La fase de los Sprint consiste en la etapa de implementación y construcción del producto. Dependiendo del tamaño del proyecto, se dividirá su desarrollo por componentes.

El orden en que se desarrollen los componentes se decide según la prioridad de los requerimientos asociados a este.

En esta etapa SQA debe:

- Garantizar que cada proceso y las revisiones asociadas a estos sean conducidas de acuerdo a los estándares y procedimientos establecidos en el plan de proyecto.
- Asegurar la incorporación de los resultados de las revisiones en los entregables de esta fase.
- Verificar la implementación de las acciones correctivas derivadas de las pruebas
- Comprobar la utilización de la especificación de procedimientos y casos de prueba durante las pruebas.
- Corroborar la documentación del código y de los resultados de la prueba de unidad.

SCM en esta fase se asegurará de controlar los cambios e informar cuando haya alguno. Una vez se haya aprobado el código, se establece la *baseline* del código.

Dentro de un Sprint se realizan las siguientes actividades.

3.2.4.1 Planificación de Sprint.

El Sprint Planning consiste en la primera parte de cada Sprint. En este evento se definen los requerimientos a cumplir durante este ciclo, se asignan las tareas a cada uno de los integrantes y se conversan todas las posibles interrogantes que puedan responder el Product Owner, o los compañeros de trabajo. Sus 2 factores más relevantes son:

- **Comunicación con el Product Owner:** La comunicación que se genera entre el Cliente y el Jefe del proyecto debe ser fluida para la posible adquisición de nuevos requerimientos y comentarios sobre el avance del proyecto.
- **Comunicación con el Equipo:** La comunicación es el componente principal dentro de esta metodología para el buen desarrollo del proyecto, donde todos los cambios deben ser informados e incluso los posibles errores y problemas no controlados que se van presentando en el avance de este.

3.2.4.2 Reunión Semanal

Dentro de cada sprint se realizan Scrum Meeting, reuniones semanales de trabajo, de 15 a 30 min, en las cuales se reportan los avances y se da contexto de la semana de trabajo. En estas reuniones se responderán 3 preguntas:

1. ¿Qué hiciste la semana pasada?
2. ¿Qué harás esta semana?
3. ¿Qué impedimento tienes para realizar correctamente tu trabajo?

Esto se hace con el propósito de evaluar el progreso actual para llegar a la meta del Sprint y evaluar la tendencia del progreso en finalizar el trabajo en el Sprint Backlog.

3.2.4.3 Trabajo y Desarrollo

Es el desarrollo de los objetivos establecidos en el Sprint Planning. Durante su curso, debemos asegurarnos que:

- No se realizan cambios que afectan al objetivo del Sprint;
- No disminuyen los objetivos de calidad
- El Alcance podrá aclararse y re-negociarse entre el propietario del producto y el equipo de Desarrollo a medida que se va aprendiendo.

3.2.4.4 Especificación y revisión de Sprint

Se lleva a cabo al final del Sprint, para inspeccionar el incremento y adaptar, si es necesario, el Product Backlog. El Equipo Scrum y las partes interesadas colaboran durante la revisión de lo que se hizo en el Sprint. Basado en ese y cualquier cambio en el Product Backlog durante el Sprint, los asistentes trabajan en las próximas cosas que se podrían hacer. Esta es una reunión informal, y está destinada a obtener retroalimentación y fomentar la colaboración.

3.2.4.5 Retrospectiva de Sprint (Revisión de Sprint Backlog)

Después de terminar un Sprint se realizará una reunión de 2 horas para analizar y compartir la experiencia de desarrollo y compararla con lo redactado en el Backlog. Donde se responderán 3 preguntas:

1. ¿Hay incremento demostrable del producto?
2. ¿Lo acepta el cliente?
3. ¿Se ha cumplido con los criterios de aceptación?

Realizados los Sprint necesarios para construir todos los componentes del producto, se pasa a la etapa siguiente.

3.2.5 Instalación

La etapa de instalación o integración es en la cual se entrega el producto resultante de lo que se planificó. En esta fase todos los componentes se juntan. Cada sección es enlazada con otra y, entonces, probada. Este proceso se repite hasta que se han agregado todos los módulos y el sistema se prueba como un todo.

En esta fase SQA es responsable de verificar que el proceso de integración y las actividades de prueba sean realizados conforme al plan de proyecto, el diseño, el plan de prueba y los estándares y procedimientos establecidos. SQA debe además monitorear las actividades de prueba y certificar sus resultados.

SCM controla los cambios e informa periódicamente sobre el estado de la configuración.

Integrado y aprobado el software se establece la *baseline* de integración y prueba. Luego, se realiza la auditoría física y funcional.

3.2.6 Operación

Realizado el último entregable, y siendo este aceptado satisfactoriamente por el cliente, se entra en la última fase de desarrollo, la cual consiste en configurar y dejar operativo el producto en el ambiente de producción.

En esta fase SQA debe efectuar todas las evaluaciones de preparación para el producto final y su respectiva documentación para la entrega al cliente. Para ello el cliente debe participar de la auditoría funcional y física.

Aprobado el software y su configuración se establece la *baseline* del producto. La versión del software liberada junto a su documentación es almacenada en el repositorio del software.

3.3 Definición de Productos de Trabajo

Los productos de trabajo consisten en la documentación que debieran entregarse durante las fases de desarrollo del proyecto. En esta sección se definen los productos y la estructura que deben poseer:

- Project Backlog
- Plan de Proyecto
 - Plan de Gestión de Riesgos
- Sprint Backlog
- Burn Down
- Plan de Aseguramiento de Calidad (SQA)
 - Plan de Pruebas
- Plan de Gestión de Configuración (SCM)
- Manual de Usuario

3.3.1 Project Backlog

Es el primer documento que se genera para el proyecto, y vendría siendo el equivalente de Scrum para informe de especificación de requerimientos. Este contiene registro actualizado acerca de cada uno de los requerimientos, estando ordenados según su grado de prioridad. Estos reflejan las necesidades detectadas tanto por los usuarios, como por los desarrolladores. Abarcan problemáticas propias del proyecto en cuestión, así como también requerimientos generales que conlleva el desarrollo del sistema.

El documento servirá como base para las etapas de desarrollo del proyecto, por lo que debe ser claro y conciso, escrito utilizando lenguaje de alto nivel y no contener detalles demasiados específicos. Esto último debido a que los requerimientos se verán con más detalle cuando planeamos el Sprint en el que vamos a abordar esos requisitos en concreto.

- 1. INTRODUCCIÓN
 - 1.1. Propósito del documento de requerimientos
 - 1.2. Alcance del producto
 - 1.3. Definiciones, acrónimos y abreviaturas
 - 1.4. Referencias
 - 1.5. Descripción del resto del documento
- 2. DESCRIPCIÓN GENERAL
 - 2.1. Perspectiva del producto
 - 2.2. Funciones del producto
 - 2.3. Características del usuario
 - 2.4. Restricciones generales
 - 2.5. Suposiciones y dependencias
- 3. REQUERIMIENTOS ESPECÍFICOS

Incluyen los requerimientos funcionales, no funcionales y de interfaz. Obviamente ésta es la parte más sustancial del documento, pero debido a la amplia variabilidad en la práctica organizacional, no es apropiado definir una estructura estándar para esta sección. Los requerimientos pueden documentar las interfaces externas, describir la funcionalidad y el rendimiento del sistema, especificar los requerimientos lógicos de la base de datos, las restricciones del diseño, las propiedades emergentes del sistema y las características de calidad.
- 4. APÉNDICES
- 5. ÍNDICE

Estructura 1: Project Backlog

3.3.2 Plan de Proyecto

Es un documento que será leído por clientes y/o por otros profesionales que actúen a nombre de los clientes. El propósito del documento será indicar los objetivos del proyecto, las problemáticas que se enfrentan y las soluciones con las que se busca el poder resolverlas. Con esto se espera demostrar al cliente que esto es un estudio serio del problema, se conocen los riesgos presentes, se sabe cómo controlarlos y, como resultado, convencerlo de que se tienen soluciones razonables y una buena idea de cómo construirlas. No se espera un detalle exhaustivo más allá de lo necesario para convencer, pero por cierto no se espera menos detalle tampoco.

Cabe destacar que este planeamiento no es algo que se limite a hacerse solamente una vez al comienzo del proyecto. Durante el desarrollo es posible que se actualicen los requerimientos en base a las historias de usuario. Actualizar adecuadamente el plan de proyecto es una tarea constante del jefe del proyecto.

El Plan de Proyecto debe contar con la siguiente estructura:

- 1 **Introducción**
- 2 **Solución conceptual**
 - 2.1 Diagnóstico de la situación actual
 - 2.2 Caracterización del cambio
 - 2.3 Análisis de las alternativas de solución
 - 2.4 Solución recomendada
- 3 **Técnicas y herramientas de desarrollo**
 - 3.1 Modelo de desarrollo
 - 3.2 Herramientas y técnicas de soporte para el desarrollo
 - 3.3 Personal y capacitación del equipo de desarrollo
- 4 **Gestión de riesgos**
 - 4.1 Análisis de riesgos
 - 4.2 Preparación para control de riesgos
- 5 **Entrega: Operación y Mantenimiento**
 - 5.1 Plan de operación del sistema
 - 5.2 Plan de mantenimiento
- 6 **Planificación de actividades**

- 6.1 Work Breakdown Structure (WBS)
- 6.2 Carta Gantt
- 6.3 Resumen de Compromisos
- 7 Planificación de la Comunicación
- 8 Seguimiento

Estructura 2: Plan de Proyecto

3.3.2 Plan de Gestión de riesgos

Se registran los posibles riesgos que pueden surgir durante el desarrollo de productos. Cada riesgo posee un índice de riesgo, factor según el cual se les dará prioridad por sobre otros. El constante ajuste en los requerimientos hace que surjan nuevas definiciones de riesgos, lo que exige un mayor control de los riesgos posibles.

- 1. INTRODUCCIÓN
 - 1.1. Objetivo
 - 1.2. Alcance
- 2. IDENTIFICACIÓN DEL RIESGO
 - Clasificación del riesgo
- 3. PROYECCIÓN DEL RIESGO
 - Determinación de responsabilidades. Establecer gestión necesaria.
- 4. GESTIÓN DEL RIESGO
 - Tabla de riesgo del proyecto. Estrategia General. Plan de contingencia

Estructura 3: Plan de Gestión de Riesgos

3.3.4 Sprint Backlog

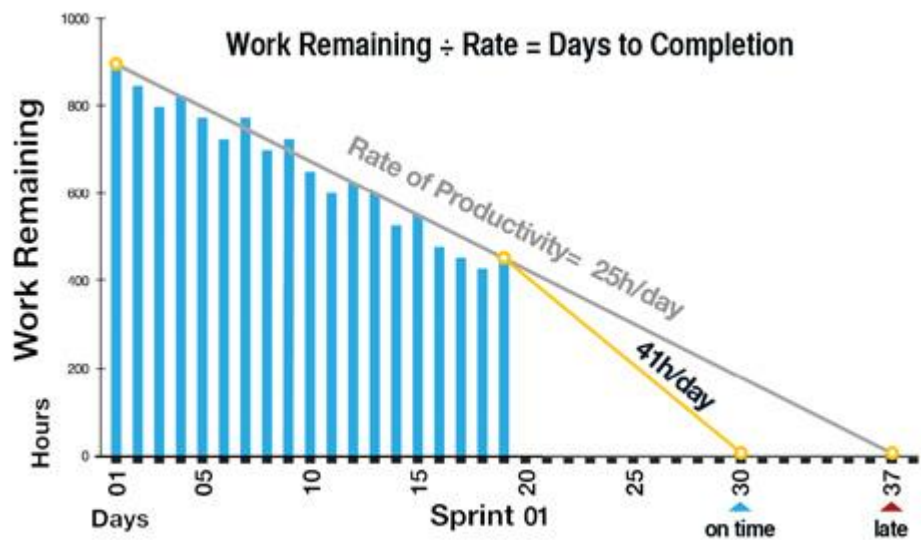
El sprint Backlog es un documento detallado donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint. Las tareas se dividen en periodos cortos, que puedan ser completadas durante un ciclo de sprint. Las tareas en el sprint Backlog nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.

User Story	Tasks	Day 1	Day 2	Day 3	Day 4	Day 5
As a member, I can read profiles of other members so that I can find someone to date.	Code the ...	8	4	8	0	
	Design the ...	16	12	10	4	
	Meet with Mary about ...	8	16	16	11	
	Design the UI	12	6	0	0	
	Automate tests ...	4	4	1	0	
	Code the other ...	8	8	8	8	
As a member, I can update my billing information.	Update security tests	6	6	4	0	
	Design a solution to ...	12	6	0	0	
	Write test plan	8	8	4	0	
	Automate tests ...	12	12	10	6	
	Code the ...	8	8	8	4	

Estructura 4: Sprint Backlog

3.3.5 Burn Down

Una tabla Burn Down es una gráfica que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los Sprint completados, podremos ver el progreso del proyecto. Lo normal es que esta línea sea descendente, hasta llegar al eje horizontal, momento en el cual el proyecto se ha terminado.



Estructura 5: Burn Down

3.3.6 Plan de Aseguramiento de Calidad (SQA)

El propósito de este plan es especificar las actividades que se realizarán para asegurar la calidad del software a construir. En él se detallan los productos que se van a revisar y los estándares, normas o métodos a aplicar; los métodos y procedimientos que se utilizarán para revisar que la elaboración de los productos se realice como lo establece el modelo de ciclo de vida del proyecto; y procedimientos para informar a los responsables de los productos los defectos encontrados y realizar un seguimiento de dichos defectos hasta su corrección.

- 1. INTRODUCCIÓN
 - 1.1. Objetivo
 - 1.2. Alcance
 - 1.3. Definiciones
 - 1.4. Resumen
- 2. GESTIÓN SQA
 - 2.1. Actividades
 - 2.2. Resultado obtenido en las Actividades
 - 2.3. Revisiones y auditoría

Estructura 6: Plan SQA

3.3.7 Plan de Pruebas

Este plan de Pruebas está contemplado dentro del plan SQA para el aseguramiento de la calidad del sistema. Con este plan se seleccionan y se coordinan las actividades para asegurar la calidad del software durante el ciclo de vida del proyecto y aún después al ser entregado al cliente. Los objetivos que se pretenden alcanzar con la aplicación del plan de pruebas son las siguientes:

- Encontrar la mayor cantidad de errores de diseño como sea posible.
- Supervisar si se cumple las especificaciones de diseño establecidas por el cliente.
- Realizar pruebas las pruebas necesarias de rendimiento y capacidad del sistema.
- Encontrar los problemas importantes y determinar los riesgos percibidos en cuanto a la calidad del producto.

- 1. INTRODUCCIÓN
 - 1.1. Objetivo
 - 1.2. Alcance
- 2. DEFINICIÓN DE ELEMENTOS A PROBAR
- 3. DEFINICIÓN ESTRATEGIA A UTILIZAR
- 4. Recursos a emplear. Especificación de la técnica que se usará. Herramientas utilizadas. Criterio indicador de éxito de las pruebas.
- 5. EQUIPO DE PRUEBAS

Estructura 7: Plan de Pruebas

3.3.8 Plan de Gestión de Configuración (SCM)

El propósito del plan es definir la organización, las actividades y responsabilidades asociadas al proceso de SCM durante el proyecto.

El plan de SCM está dirigido al jefe de proyectos, los desarrolladores y al responsable de SCM, responsable de la elaboración, actualización y monitoreo del plan.

El presente documento establece, de acuerdo a la política organizacional, las responsabilidades de SCM durante el ciclo de vida del software definido para el proyecto. Estas responsabilidades consisten en controlar los cambios en la configuración de la aplicación, y asegurarse que estos cambios se implementen adecuadamente para así notificarlos a todos los involucrados e interesados.

- 1. INTRODUCCIÓN
 - 1.1. Objetivo
 - 1.2. Alcance
 - 1.3. Definiciones
 - 1.4. Resumen
- 2. IDENTIFICACIÓN DE LOS ÍTEMS DE CONFIGURACIÓN
 - 2.1. Líneas de Base
- 3. IDENTIFICACIÓN DE LA CONFIGURACIÓN
 - 3.1. Control de Cambios
 - 3.2. Control de Documentos
- 4. ESTADO DE LA CONFIGURACIÓN
- 5. HERRAMIENTAS SCM
- 6. CONJUNTO DE REGISTROS

Estructura 8: Plan SCM

3.3.9 Manual de Usuario

Un manual de usuario es un documento de comunicación técnica que busca brindar asistencia a los sujetos que usan un sistema. Más allá de su especificidad, los autores de los manuales intentan apelar a un lenguaje ameno y simple para llegar a la mayor cantidad posible de receptores.

Dada su complejidad, todos los productos electrónicos o informáticos suelen contar con su propio manual de usuario. Los artículos más simples (como una pelota o una mesa) no requieren de explicaciones para que los consumidores sepan cómo utilizarlos.

Los manuales de usuarios suelen estar escritos en diversos idiomas y contar tanto con textos como con imágenes. De esta forma se facilita la comprensión de los conceptos. Los diagramas y esquemas también son habituales.

Una estructura frecuente de los manuales de usuario incluye una introducción al producto en cuestión, un índice con los contenidos del manual, la guía en sí misma, una sección de problemas frecuentes y su forma de solucionarlos, los datos de contacto y un glosario.

1. PREFACIO

1.1. Resumen

1.2. Cómo usar el manual

2. ÍNDICE

3. MODELO DEL SISTEMA

4. FUNCIONES PRINCIPALES DEL SITEMA

5. SECCIÓN DE PREGUNTAS

5.1. Preguntas frecuentes

6. CONTÁCTENOS

Estructura 9: Manual de Usuario

3.4 Atributos de Calidad

Para cada uno de los elementos generados durante este proyecto se definen los atributos medibles que se considerarán y sus objetivos cuantificables. Esto se hace con el objetivo de tener una medición concreta de sus atributos que compruebe si en realidad los poseen, y con esto verificar que en realidad es de alta calidad.

3.4.1 Definición de los Atributos de Calidad

Atributo de Calidad	Descripción	Objetivo cuantificable
Mantenibilidad	Esfuerzo requerido para localizar y corregir un error en un programa en funcionamiento.	¿Se pueden corregir los errores?
Flexibilidad	Esfuerzo requerido para modificar un programa en funcionamiento.	¿Pueden cambiarlo?
Facilidad de Prueba	Esfuerzo requerido para probar un programa (para garantizar que realiza la función deseada).	¿Puede ser probado?
Corrección	Grado en el que un programa satisface las especificaciones y cumple los objetivos del usuario.	¿Hace lo que se le pidió?
Fiabilidad	Grado en el que un programa se espera que realice su función con una precisión requerida.	¿Es confiable todo el tiempo?
Usabilidad	Esfuerzo necesario para aprender, operar, preparar entradas e interpretar la salida de un programa.	¿Está diseñado para ser usado?
Integridad	Grado en el que se controla el acceso al programa o los datos por usuarios no autorizados.	¿Es seguro (inviolable)?

Eficiencia	Cantidad de recursos y código requeridos por un programa para realizar una función.	¿Podrá ejecutarse lo mejor que pueda?
Portabilidad	Esfuerzo requerido para transferir un programa de una configuración hardware o entorno software a otro.	¿Podrá usarse en otra máquina?
Reusabilidad	Grado en el que un programa se puede utilizar en otras aplicaciones.	¿Podrá rehusar algo del software?
Interoperabilidad	Esfuerzo requerido para acoplar un sistema con otro.	¿Podrá interactuar con otro sistema?
Completitud	Grado de contenido que tienen los documentos.	¿Se encuentra todo lo que se requería?
Claridad	Grado de claridad y no-ambigüedad de los documentos.	¿Son suficientemente claros los conceptos e ideas que se intentan expresar?

Tabla 1 Atributos de Calidad [SEPA-1]

3.4.2 Atributos de calidad por actividades del proceso de desarrollo

Actividad	Planificación
Objetivo Cuantificable	La Planificación del proyecto debe respetar los plazos establecidos. Se requiere la participación del cliente.
Atributos de Calidad	Claridad, Mantenimiento, Flexibilidad, Fiabilidad, Corrección, Facilidad de uso, Eficiencia, Integridad.

Tabla 2 Planificación

Actividad	Product Backlog
Objetivo Cuantificable	Deben estar identificados los problemas o necesidades del en un 95%.El cliente debe estar de acuerdo con un 100% de los requerimientos descritos.

Atributos de Calidad	Claridad, Mantenibilidad, Flexibilidad, Corrección, Fiabilidad, Facilidad de uso, Integridad, Eficiencia
----------------------	--

Tabla 3 Product Backlog

Actividad	Diseño
Objetivo Cuantificable	Se debe definir la Funcionalidad y Solución Física que va a satisfacer los Requerimientos en un 90%. Se debe planificar cómo se va a implementar y aceptar la Solución Propuesta en un 90%. Se debe planificar el soporte de la Solución Propuesta en un 90%.
Atributos de Calidad	Claridad, Completitud, Mantenibilidad, Flexibilidad, Corrección, Fiabilidad, Facilidad de uso, Integridad, Eficiencia

Tabla 4 Diseño

Producto	Sprint
Objetivo Cuantificable	Los componentes de la Solución deben ser contruidos en un 100%. Las Pruebas deben cubrir el 100% de los Componentes contruidos.
Atributos de Calidad	Corrección, Claridad, Mantenimiento, Integridad, Completitud, Facilidad de uso, Integridad, Eficiencia

Tabla 5 Implementación mediante Sprint

Producto	Instalación
Objetivo Cuantificable	Requiere la participación del cliente. El producto se prueba en un ambiente operacional hasta que esté lista en un 100% para la prueba de aceptación por parte del cliente. El cliente debe estar de acuerdo con que la

	solución cumpla las expectativas, y que la organización acepta el producto entregado, en un 100%.
Atributos de Calidad	Mantenimiento, Corrección, Claridad, Fiabilidad, Integridad, Completitud, Facilidad de uso, Eficiencia

Tabla 6 Instalación

Producto	Operación
Objetivo Cuantificable	Se requiere la participación del cliente. Se debe archivar el contenido del Proyecto y considerar el proyecto como terminado, en un 100%. Debe documentarse todo lo realizado durante el desarrollo del proyecto. Cualquier falla detectada debe ser atendida y corregida en menos de 48 horas.
Atributos de Calidad	Corrección, Flexibilidad, Facilidad de uso, Fiabilidad, Claridad

Tabla 7 Operación

3.4.3 Atributos de calidad por productos de trabajo

Producto	Plan de Proyecto
Objetivo Cuantificable	El documento no debe superar las 120 hojas Debe ser comprendido por todo el Equipo de trabajo. La Planificación del Proyecto debe ser capaz de ser modificada en caso de cambios en los requerimientos. Se requiere la participación del cliente. El Plan del Proyecto debe contemplar la estrategia, organización, riesgos, contingencias, recursos y tareas, para la fase de diseño.
Atributos de Calidad	Claridad, Mantenimiento, Flexibilidad, Confiabilidad, Facilidad de uso, Integridad,

	Eficiencia
--	------------

Tabla 8 Plan de Proyecto

Producto	Plan de Gestión de Riesgos
Objetivo Cuantificable	El documento no debe superar las 50 hojas. El documento como mínimo debe contener 10 riesgos. Debe ser posible de modificar en caso de detectar más riesgos.
Atributos de Calidad	Claridad, Mantenimiento, Flexibilidad, Corrección, Confiabilidad, Facilidad de uso, Integridad, Eficiencia.

Tabla 9 Plan de Gestión de Riesgos

Producto	Project Backlog
Objetivo Cuantificable	Requiere la participación del cliente y de todo el equipo de trabajo. Debe estar redactado en lenguaje de tercer nivel, sin especificaciones técnicas, y que sea fácilmente entendible por el cliente.
Atributos de Calidad	Completitud, Mantenimiento, Claridad, Flexibilidad, Corrección, Fiabilidad, Facilidad de uso, Integridad, Eficiencia

Tabla 10 Project Backlog

Producto	Sprint Backlog
Objetivo Cuantificable	Debe realizarse en un plazo máximo de dos días y un mínimo de una hora. Las tareas publicadas no deben superar duración del ciclo de sprint.
Atributos de Calidad	Flexibilidad, Corrección, Claridad, Fiabilidad, Facilidad de uso.

Tabla 11 Sprint Backlog

Producto	Burn Down
Objetivo Cuantificable	Debe actualizarse según se vayan completando tareas. Las tareas publicadas no deben superar duración del ciclo de sprint. El gráfico debe mostrar claramente cuántos ítems quedan pendientes y cuánto tiempo queda para que acabe el ciclo.
Atributos de Calidad	Facilidad de uso, Mantenimiento, Claridad, Corrección.

Tabla 12 Burn Down

Producto	Plan de Aseguramiento de Calidad (SQA)
Objetivo Cuantificable	Se requiere la participación del Jefe de SQA, y el Jefe de Proyecto. Se debe asegurar que la calidad requerida para la solución y el proceso de desarrollo esté definido, incorporado y verificado en todas las fases del proyecto. Se deben definir todas las actividades de aseguramiento de calidad que se harán durante el proyecto.
Atributos de Calidad	Mantenimiento, Claridad, Flexibilidad, Corrección, Confiabilidad, Facilidad de uso, Integridad, Eficiencia

Tabla 13 Plan SQA

Producto	Plan de Pruebas
Objetivo Cuantificable	El cliente debe comprender en un 90% el documento. Se deben detectar a lo menos un 20% de errores en los Casos de Prueba aplicados.
Atributos de Calidad	Compleitud, Mantenimiento, Claridad, Flexibilidad, Corrección, Confiabilidad, Facilidad de uso, Integridad, Eficiencia

Tabla 14 Plan de Pruebas

Producto	Manual de Usuario
Objetivo Cuantificable	El manual no debe manejar un lenguaje técnico, debe ser entendido en un 95% por los usuarios finales No debe superar las 50 hojas
Atributos de Calidad	Compleitud, Mantenimiento, Claridad, Flexibilidad, Corrección, Confiabilidad, Facilidad de uso, Integridad, Eficiencia

Tabla 15 Manual de Usuario

3.5 Puntos de revisión (hitos)

- **Especificación de requerimientos**
 - Revisión y aprobación de la especificación de requerimientos.
 - Reporte del estado y los resultados de las actividades de SQA.
- **Planificación**
 - Revisión y aprobación del plan de SQA.
 - Revisión y aprobación del plan de proyecto.
 - Revisión y aprobación del plan de gestión de riesgos.
 - Reporte de estado y los resultados de las actividades de SQA.
- **Diseño**
 - Revisión y aprobación de la Especificación de Diseño de Sistema.
 - Revisión y aprobación del diseño del modelo de datos.
 - Revisión y aprobación del Plan de Pruebas del sistema
 - Reporte del estado y los resultados de las actividades de SQA.
- **Sprint**
 - Revisión y aprobación de los casos de prueba.
 - Revisión y aprobación de la especificación de los procedimientos de prueba.
 - Revisión y aprobación del código y su documentación.
 - Revisión y aprobación del Manual de Usuario.
 - Reporte del estado y los resultados de las actividades de SQA.
- **Instalación (Aceptación y entrega)**
 - Revisión y aprobación el software y su documentación.
 - Reporte del estado y los resultados de las actividades de SQA.
- **Operación**
 - Revisión y aprobación de cada cambio producido durante la mantención en su especificación, diseño, implementación y prueba.
 - Revisión y aprobación de la documentación asociada a los cambios.
 - Revisión y aprobación de la nueva versión del software y de su documentación.
 - Reporte del estado y los resultados de las actividades de SQA.

4. GESTION DE CALIDAD

4.1 Equipo de trabajo



Tabla 16 Equipo de Trabajo [WEB-1]

La Unidad de SQA es responsable de monitorear el cumplimiento de las actividades planificadas en el plan de SQA y de garantizar la calidad de los entregables, la documentación y de los procesos utilizados para producir software.

La unidad de SQA debe organizarse para desempeñar sus labores y otorgar soporte al desarrollo de software. Para ello cuenta con tres posibilidades básicas:

- Cada miembro de la unidad de SQA se especializa en todas sus prácticas y se hace responsable de las actividades por realizar en un proyecto.
- Cada miembro de la unidad se especializa en un subconjunto de prácticas y provee el soporte sobre ellas
- La unidad actúa como un equipo en el cual sus miembros cooperan para llevar a cabo las tareas de SQA.

4.2 Recursos

4.2.1 Recursos Humanos

El esfuerzo requerido para las actividades de SQA asciende a aproximadamente el 25% del esfuerzo total requerido en el proyecto.

Para llevar a cabo el plan de aseguramiento de calidad, se hace necesario contar con un grupo especializado de SQA, quienes desempeñarán distintas funciones para el cumplimiento de los objetivos. Se debe guiar al equipo de desarrollo, de tal manera de lograr un producto de alta calidad. Hay que dejar en claro que el grupo de SQA no tiene la total responsabilidad de lograr una aplicación de calidad, SQA solo planea las estrategias para lograr esa calidad. El aseguramiento de calidad es responsabilidad de todo su equipo.

Es el jefe de SQA quien estará a cargo de interactuar con todos los miembros del equipo de trabajo.

Las actividades de SQA requieren de aproximadamente el 25% del tiempo total requerido en el proyecto.

Por todo lo dicho anteriormente se adoptará la siguiente distribución de cargos:

Scrum Master

En los proyectos Scrum, debido a que los equipos auto-organizados son un principio central de Scrum, el papel de gerente del proyecto se hace insostenible por lo que se elimina. Esto no significa que ese trabajo y sus responsabilidades desaparezcan. Gran parte de la responsabilidad previamente asumidas por el gerente del proyecto es transferido al equipo Scrum. Sin un gerente de proyecto que asigne tareas a los individuos, los miembros del equipo asumen la responsabilidad de la selección de las propias tareas. Otras responsabilidades se han movido al Scrum Master.

No hay que confundirse, un Scrum Master no es un gerente, este no da órdenes, este ayuda a que las tareas se hagan correctamente [WEB 2]

El Scrum Master debe de:

- Revisar y aprobar el plan de SQA para cada proyecto.
- Facilitar al equipo las actividades de SQA.
- Resolver cualquier conflicto relacionado con las actividades de SQA.

Jefe de Proyecto

El Jefe de Proyecto debe:

- Establecer un programa de calidad para el proyecto de desarrollo de software de acuerdo a las políticas organizacionales.
- Identificar las actividades de SQA requeridas para el proyecto.
- Revisar y aprobar el plan de SQA para el proyecto.
- Identificar los participantes de las actividades de SQA.
- Implementar las actividades de SQA de acuerdo al plan.
- Monitorear las actividades de SQA planificadas en el plan.
- Identificar los factores de calidad para la implementación del software.
- Identificar, desarrollar y mantener la documentación del proyecto.

SQA

Responsable de monitorear el cumplimiento de las actividades planificadas en el plan de SQA y de garantizar la calidad de los entregables, la documentación y de los procesos utilizados para producir software.

Ingenieros y Desarrolladores

Entre sus responsabilidades se cuentan:

- Revisar y entregar sus observaciones sobre el plan de SQA para el proyecto.
- Implementar las actividades de SQA de acuerdo al plan.
- Participar de la solución de los problemas detectados por las actividades de SQA que sean de su competencia.
- Implementar las prácticas, procesos y procedimientos definidos en el plan de proyecto y en otros planes o documentos complementarios.

SCM

Responsable de:

- Revisar y comentar el plan de SQA para el proyecto.
- Implementar las actividades de calidad de acuerdo al plan de SQA.
- Resolver los problemas detectados por SQA relacionados con SCM.
- Implementar las prácticas, procesos y procedimientos definidos en el plan de SCM y en otros planes o documentos complementarios.

4.2.2 Infraestructura

SQA necesita tener acceso a equipos de trabajo que le permitan realizar sus labores de evaluación y auditoría eficazmente. Por ello dispondrá de la infraestructura y equipamiento destinados en el plan de proyecto:

Se necesitarán 5 Computadores personales con las siguientes especificaciones:

- Procesador: Intel® Core™ i7-6500U 2.5 GHz hasta 3.1 GHz
- Número de núcleos: Dual Core
- Memoria: 4GB de SDRAM DDR4 2133 MHz
- Almacenamiento: 1 TB SATA (de 5400 rpm)
- Pantalla: 15.6"
- Al menos un puerto de salidas USB 3.0
- Wi-Fi: 802.11b/g/n

Cada uno de los equipos debe contar con un segundo monitor de a lo menos 17 pulgadas.

También se necesitará una tablet para pruebas con las siguientes características:

- Sistema operativo Android 5.x o superior.
- RAM: 2 GB
- Almacenamiento: 16 GB
- Wi-Fi: 802.11 a/b/g/n 2.4+5GHz, HT40
- Lector de tarjetas Micro SD

El lugar de trabajo debe contar con las comodidades mínimas de una oficina de trabajo, más los servicios básicos y un servicio de conexión a internet con velocidad de descarga de 10MB o superior.

4.3 Actividades de SQA

1. Evaluación de la selección los productos de trabajo

El plan de proyecto identifica los productos de trabajo que deben ser desarrollados y evaluados, incluyendo los estándares y guías aplicables a su desarrollo. SQA debe asistir el jefe de proyectos en la selección de los estándares y guías aplicables a cada entregable.

2. Evaluación de las herramientas

SQA debe evaluar la selección de las herramientas existentes y adquiridas para el desarrollo. Éstas deben ser evaluadas según su funcionalidad, disponibilidad y facilidad de operación.

3. Evaluación de la planificación y el monitoreo del proyecto

Durante la planificación, SQA debe apoyar la identificación apropiada de guías y estándares aplicables a los entregables del proyecto y responsabilizarse de la elaboración del plan de SQA. Posteriormente, debe supervisar el cumplimiento del plan de proyecto

4. Evaluación de la especificación de requerimientos

SQA debe:

- Comprobar la adherencia de los entregables a los estándares definidos en el plan de proyecto.
- Verificar la adherencia del proceso de especificación de requerimientos a los procedimientos definidos en el plan de proyecto.
- Garantizar que se revisaron adecuadamente los entregables (especificación del sistema y de requerimientos) de la fase de especificación de requerimientos.
- Asegurar la incorporación de los resultados de las revisiones en los entregables de esta fase.
- Corroborar que estén expresados y documentados los requerimientos funcionales, técnicos, operacionales y de interfaz, de manera tal que puedan ser verificados en el producto final.

5. Evaluación del diseño

SQA es responsable de:

- Comprobar la adherencia de los entregables a los estándares definidos en el plan de proyecto.
- Verificar la adherencia del proceso de diseño a los procedimientos definidos en el plan de proyecto.
- Garantizar que se revisaron adecuadamente los entregables (diseño preliminar, diseño detallado, plan de pruebas, especificación de casos y procedimientos de prueba) de la fase de diseño.
- Asegurar la incorporación de los resultados de las revisiones en los entregables de esta fase.

6. Evaluación de la implementación y de la prueba de unidad

SQA debe:

- Garantizar que el proceso de codificación, las revisiones asociadas y la prueba de unidad sean conducidos de acuerdo a los estándares y procedimientos establecidos en el plan de proyecto.
- Asegurar la incorporación de los resultados de las revisiones en los entregables de esta fase.
- Verificar la implementación de las acciones correctivas derivadas de la prueba de unidad.
- Comprobar la utilización de la especificación de procedimientos y casos de prueba durante la prueba de unidad.
- Corroborar la documentación del código y de los resultados de la prueba de unidad.

7. Evaluación de la integración y prueba

SQA es responsable de:

- Verificar que el proceso de integración y las actividades de prueba sean realizadas conforme al plan de proyecto, el diseño, el plan de prueba y los estándares y procedimientos establecidos.
- Asegurar que la prueba de integración fue completada satisfactoriamente, que sus resultados fueron registrados y divulgados y que las acciones correctivas derivadas de ella fueron implementadas.
- Corroborar el desarrollo adecuado de las pruebas de aceptación y del sistema.
- Monitorear las actividades de prueba y certificar sus resultados.

- Revisar las pruebas.

8. Evaluación del producto antes de su liberación

SQA debe evaluar las actividades de preparación del producto final y sus documentación para la entrega al cliente, para lo cual debe participar de la auditoría funcional y física.

9. Evaluación del proceso de revisión

SQA debe verificar que todo producto que se encuentre listo para revisión sea revisado y que las acciones correctivas identificadas durante la revisión sean implementadas

10. Evaluación de las acciones correctivas

SQA debe analizar los problemas detectados para determinar sus causas, impactos y frecuencia de ocurrencia, y para establecer acciones preventivas.

Además, SQA es responsable de monitorear la adecuada implementación de las acciones correctivas derivadas de estos problemas.

11. Evaluación del proceso de SCM

SQA debe:

- Revisar el plan de SCM.
- Asegurar la correcta identificación de los ítems de configuración.
- Garantizar un adecuado control de cambios.
- Corroborar que la contabilidad del estado de la configuración sea preparada oportunamente y que refleje la situación real de los ítems de configuración en relación con el proyecto.
- Comprobar la adherencia de las actividades de SCM al plan de SCM.
- Verificar el correcto funcionamiento de la librería del software.

12. Verificar la implementación de los procesos

SQA debe corroborar la adherencia de todos los procesos a los estándares y procedimientos definidos en el plan de proyecto.

13. Establecer las auditorías

SQA es responsable en la institución por el desarrollo de las auditorías internas. Por lo tanto debe gestionarlas de ser preciso.

Además, es su responsabilidad participar en la auditoría física y funcional.

4.4 Responsabilidades

La autoridad de SQA deriva del gerente técnico del grupo StarLampStudio y su principal obligación es monitorear las actividades del proceso de desarrollo y revisar la adherencia de los productos de trabajo a los estándares, procedimientos y al plan del proyecto. Los resultados de este seguimiento deben ser informados al jefe de proyectos y, según sea aplicable, al Gerente Técnico.

En la siguiente tabla se adjunta una matriz de responsabilidades sobre las actividades de SQA.

Actividad	Scrum Master	Jefe proyecto	SQA	SCM	Analís ta	Diseña dor	Programa dor	Repr. Prueba
Evaluación de la selección los productos de trabajo	X	X	X	X	X			
Evaluación de las herramientas	X	X	X					
Evaluación de la planificación y el monitoreo del proyecto	X	X	X	X				
Evaluación de la especificación de requerimientos	X	X	X		X	X	O	X
Evaluación del diseño	X	X	X		X	X	O	X
Evaluación de la implementación y de la prueba de unidad			X			X	X	X
Evaluación de la integración y prueba			X			X	O	X
Evaluación del producto antes de su liberación		X	X		X	X	X	X
Evaluación del proceso de revisión		X	X					
Evaluación de las acciones correctivas	X	X	X	X				
Evaluación del proceso de SCM	X	X	X	X				
Verificar la implementación de		X	X					

los procesos								
Establecer las auditorías		X	X	X				

Tabla 17 Responsabilidades

4.5 Análisis de Riesgos

4.5.1 Identificación de Riesgos

ID	Riesgo
1	Aparición de productos similar en el mercado
2	El servidor no soporta una gran cantidad de usuarios conectados al mismo tiempo
3	Que no haya conexión a internet en el momento de la post-catástrofe:
4	Los requerimientos no estén del todo claros
5	Mala estimación de los recursos necesarios para desarrollo
6	No tener una visión clara de las metas del proyecto
7	Dificultades siguiendo ritmo de trabajo de Scrum
8	Desconocimiento o falta de dominio al trabajar con Ruby on Rails
9	Interfaz no satisface a cliente
10	Desconocimiento o falta de dominio al trabajar con APIs
11	Al ingresar un punto de trabajo, coordenadas no coinciden con las del mapa
12	El mapa no muestre información actualizada
13	Los caminos trazados en el mapa sean erróneos o no se muestren adecuadamente
14	La base de datos no puede almacenar los diferentes tipos de recursos
15	Las funcionalidades de la plataforma no se pasan correctamente en dispositivos móviles.

Tabla 18 Riesgos identificados

4.5.2 Clasificación de Riesgos

Los riesgos se clasifican según las 3 siguientes categorías:

- **Riesgos Técnicos:** Potenciales problemas que comprenden a los factores asociados con el desarrollo de productos, ya sean de diseño, implantación del sistema, la interfaz, entre otros.
- **Riesgos de Negocios:** Posible riesgo de que el producto o el servicio desarrollado falle en el mercado.
- **Riesgos Humanos:** Riesgos causados por problemas en relaciones de los participantes en los proyectos.

4.5.2.1 Riesgos de Negocios

ID	Riesgo
1	Aparición de productos similar en el mercado

Tabla 19 Riesgos de Negocios

4.5.2.2 Riesgos Técnicos

ID	Riesgo
2	El servidor no soporta una gran cantidad de usuarios conectados al mismo tiempo.
3	Que no haya conexión a internet en el momento de la post-catástrofe:
11	Al ingresar un punto de trabajo, coordenadas no coinciden con las del mapa
12	El mapa no muestre información actualizada
13	Los caminos trazados en el mapa sean erróneos o no se muestren adecuadamente
14	La base de datos no puede almacenar los diferentes tipos de recursos
15	Las funcionalidades de la plataforma no se pasan correctamente en dispositivos móviles.

Tabla 20 Riesgos Técnicos

4.5.2.3 Riesgos Humanos

ID	Riesgo
4	Los requerimientos no estén del todo claros

5	Mala estimación de los recursos necesarios para desarrollo
6	No tener una visión clara de las metas del proyecto
7	Dificultades siguiendo ritmo de trabajo de Scrum
8	Desconocimiento o falta de dominio al trabajar con Ruby on Rails
9	Interfaz no satisface a cliente
10	Desconocimiento o falta de dominio al trabajar con APIs

Tabla 21 Riesgos Humanos

4.5.3 Estimación de los Riesgos

- Probabilidad
- Impacto

ATRIBUTO	VALOR	DESCRIPCIÓN
Impacto	Bajo	No merecen ser tenidos en cuenta.
	Medio	Están dentro de un margen de aceptación, por lo cual no comprometen ni el proyecto, ni el producto, ni la organización.
	Alto	Comprometen gravemente el proyecto o el producto o la organización.
	Catastrófica	Amenazan la supervivencia del proyecto o del producto o de la organización.
Probabilidad	Muy baja	<10%
	Baja	del 10 al 25%
	Moderada	del 25 al 50%
	Alta	del 50 al 75%
	Muy alta	>75%

Tabla 22 Estimación de Riesgos

Tomando en consideración estos factores, estos riesgos se enlistan por nivel de prioridad, en orden descendente.

4.5.4 Riesgos Priorizados

ID	Riesgo	Probabilidad	Impacto	Nivel de Prioridad
1	Aparición de productos similar en el mercado	Moderada	Medio	4
2	El servidor no soporta una gran cantidad de usuarios conectados al mismo tiempo	Baja	Alto	5
3	Que no haya conexión a internet en el momento de la post-catástrofe:	Moderada	Alto	3
4	Los requerimientos no estén del todo claros	Baja	Alto	5
5	Mala estimación de los recursos necesarios para desarrollo	Moderada	Medio	5
6	No tener una visión clara de las metas del proyecto	Baja	Alto	5
7	Dificultades siguiendo ritmo de trabajo de Scrum	Alta	Alto	2
8	Desconocimiento o falta de dominio al trabajar con Ruby on Rails	Muy Alta	Alto	1
9	Interfaz no satisface a cliente	Muy Alta	Alto	1
10	Desconocimiento o falta de dominio al trabajar con APIs	Muy Alta	Medio	3
11	Al ingresar un punto de trabajo, coordenadas no coinciden con las del mapa	Muy Baja	Medio	6
12	El mapa no muestre información actualizada	Muy Baja	Medio	6
13	Los caminos trazados en el mapa sean erróneos o no se muestren adecuadamente	Moderada	Medio	4
14	La base de datos no puede almacenar los diferentes tipos de recursos	Muy Baja	Alto	5
15	Las funcionalidades de la plataforma no se pasan correctamente en dispositivos	Baja	Bajo	6

	móviles.			
--	----------	--	--	--

Tabla 23 Riesgos con prioridad

Para disminuir el efecto negativo que pudieran producir estos riesgos durante el desarrollo del proyecto, se desarrolla para los 10 riesgos más prioritarios sus “Hojas de Control de Riesgos”. Estas definen sus planes de mitigación y contingencia.

ID	Riesgo	Nivel de Prioridad
8	Desconocimiento o falta de dominio al trabajar con Ruby on Rails	1
9	Interfaz no satisface a cliente	1
7	Dificultades siguiendo ritmo de trabajo de Scrum	2
3	Que no haya conexión a internet en el momento de la post-catástrofe:	3
10	Desconocimiento o falta de dominio al trabajar con APIs	3
1	Aparición de productos similar en el mercado	4
13	Los caminos trazados en el mapa sean erróneos o no se muestren adecuadamente	4
4	Los requerimientos no estén del todo claros	5
5	Mala estimación de los recursos necesarios para desarrollo	5
14	La base de datos no puede almacenar los diferentes tipos de recursos	5

Tabla 24 Riesgos ordenados según prioridad.

En las hojas de control de riesgos de los riesgos seleccionados se generarán los planes de mitigación (para evitar que sucedan) y planes de contingencia (para actuar en caso que sucedan) correspondientes. Esto con el fin de poder controlar la ocurrencia de los riesgos que podrían retrasar el proyecto, o ponerlo en peligro de no completarse exitosamente.

4.5.4 Control de Riesgos

Hoja de Control de Riesgo		
Nombre Proyecto: Post Emergency Support System		
ID: 8	Declaración del riesgo: Desconocimiento o falta de dominio al trabajar con Ruby on Rails.	
Prioridad: 1		
Probabilidad: Muy Alta		
Impacto: Alto	Origen: Equipo	Clasificación: Humano
Periodo: Todo el Proyecto		
Contexto: Ruby on Rails es una herramienta de desarrollo muy poderosa pero que requiere esfuerzo para aprender a utilizarla. Esto puede llevar a retrasos en el desarrollo y hasta incluso a optar por cambiarla por una herramienta nueva.		
Plan de Mitigación: Investigar y desarrollar un plan de capacitación de Ruby on Rails que cubra dudas que puedan surgir durante el proceso.		
Plan de Contingencia: Cambiar de herramienta por una que cumpla los plazos, volver a planificar el proyecto y asignar nuevas responsabilidades.		

Hoja de Control de Riesgo		
Nombre Proyecto: Post Emergency Support System		
ID: 9	Declaración del riesgo: Interfaz no satisface al cliente	
Prioridad: 1		
Probabilidad: Muy Alta		
Impacto: Alto	Origen: Cliente	Clasificación: Humano
Periodo: Etapa de Evaluación		
Contexto: El diseño de la interfaz no satisface las expectativas presentadas por el cliente.		
Plan de Mitigación: Se conversa con el cliente que busca en la interfaz del producto. Se diseñan múltiples prototipos para presentarlos al cliente.		
Plan de Contingencia: Se conversa el asunto con el cliente en reuniones posteriores y diseña una nueva interfaz.		

Hoja de Control de Riesgo		
Nombre Proyecto: Post Emergency Support System		
ID: 7	Declaración del riesgo: Dificultades siguiendo el ritmo de trabajo de Scrum.	
Prioridad: 2		
Probabilidad: Alta		
Impacto: Alto	Origen: Equipo	Clasificación: Humano
Periodo: Fase de desarrollo		
Contexto: Scrum es un modelo de desarrollo ágil que permite trabajo ininterrumpido en una parte de la funcionalidad y con ello el desarrollo del producto completo. De no cumplir los plazos de los Sprint, este se considera fallido.		
Plan de Mitigación: Se establecen metas simples que incrementan su complejidad posteriormente. Al quinto sprint el equipo ya se habrá familiarizado y con ello tendrá la confianza para seguir el ritmo de trabajo.		
Plan de Contingencia: Analizar las causas de que fallara, establecer futuras metas más acordes al equipo		

Hoja de Control de Riesgo		
Nombre Proyecto: Post Emergency Support System		
ID: 3	Declaración del riesgo: No hay conexión a internet al momento de la post catástrofe.	
Prioridad: 3		
Probabilidad: Moderada		
Impacto: Alto	Origen: Falla eléctrica.	Clasificación: Medioambiente del Trabajo
Periodo: Instalación		
Contexto: La función está diseñada para operar luego de la post catástrofe, cuando los servicios primarios estén funcionando. Puede darse que siga con fallas y que deje de funcionar el internet, impidiendo su correcta operación.		
Plan de Mitigación: Ninguno. No se puede mitigar.		
Plan de Contingencia: Establecer funcionalidades que permitan operar la aplicación, por lo menos parcialmente, sin conexión a internet.		

Hoja de Control de Riesgo		
Nombre Proyecto: Post Emergency Support System		
ID: 10	Declaración del riesgo: Desconocimiento o falta de dominio al trabajar con APIs	
Prioridad: 3		
Probabilidad: Muy Alta		
Impacto: Medio	Origen: Equipo	Clasificación: Humano
Periodo: Fases de desarrollo		
Contexto: Las APIs se renuevan constantemente. Cambian sus términos de uso o cambian en términos de compatibilidad. Adaptar la aplicación a esos cambios puede llevar a retrasos o cambios en la planificación.		
Plan de Mitigación: Investigar y documentar información acerca de las APIs, registrando cambios que hayan tenido hasta el momento de la fase de planificación.		
Plan de Contingencia: Conseguir asesoramiento en las APIs o considerar posibles APIs que puedan sustituir a las APIs actuales.		

Hoja de Control de Riesgo		
Nombre Proyecto: Post Emergency Support System		
ID: 1	Declaración del riesgo: Aparición de productos similares en el mercado	
Prioridad: 4		
Probabilidad: Moderada		
Impacto: Medio	Origen: Mercado	Clasificación: Negocio
Periodo: Todo el proyecto		
Contexto: Al momento de planificar la aplicación o durante pleno proceso de desarrollo se pueden avistar productos recién lanzados que compartan las funcionalidades de este producto. Esto trae como consecuencia mayor dificultad para hacerse un espacio en el mercado internacional.		
Plan de Mitigación: Análisis de la competencia en estudio de mercado. Buscar valor único que tiene este producto por sobre los otros.		
Plan de Contingencia: Análisis de la competencia en estudio de mercado. Potenciar valor único del producto por sobre los otros.		

Hoja de Control de Riesgo		
Nombre Proyecto: Post Emergency Support System		
ID: 13	Declaración del riesgo: Los caminos trazados en el mapa sean erróneos o no se muestren adecuadamente	
Prioridad: 3		
Probabilidad: Moderada		
Impacto: Medio	Origen: Base de Datos	Clasificación: Técnico
Periodo: Operación		
Contexto: La funcionalidad principal del sistema es la visualización del área afectada y las acciones que se realizan para reconstrucción. Para ello se requiere que cada parte del proceso de obtención de información, procesamiento y visualización funcione perfectamente.		
Plan de Mitigación: Estandarización de los datos recolectado mediante un formato fijo, de manera que se mantengan valores fijo al momento de almacenarlos en la base de datos.		
Plan de Contingencia: Verificar la parte de la aplicación encargada de procesar la información, verificar y restringir formato de los datos ingresados en los campos fijos.		

Hoja de Control de Riesgo		
Nombre Proyecto: Post Emergency Support System		
ID: 4	Declaración del riesgo: Los requerimientos no estén del todo claros	
Prioridad: 5		
Probabilidad: Baja		
Impacto: Alto	Origen: Equipo	Clasificación: Humano
Periodo: Especificación de requerimientos.		
Contexto: Para que el proyecto cumpla con las expectativas del cliente, es necesario tanto el cliente como el equipo de desarrollo tengan la misma interpretación de los requerimientos. Si hay una mala comprensión de los requerimientos del proyecto, eso puede causar demoras en las entregas debido a ajustes en las funcionalidades producto del rechazo de parte del cliente consumiéndose recursos de más y eventualmente llevar el proyecto al fracaso.		
Plan de Mitigación: Durante las reuniones con el cliente, conversar respecto a los requerimientos, y conversar cualquier posible cambio que pueda surgir en estos.		
Plan de Contingencia: Planificar el cómo adaptarse intensivamente a la característica conflictiva y resolverla dentro de los márgenes de tiempo asignado a problemas imprevistos.		

Hoja de Control de Riesgo		
Nombre Proyecto: Post Emergency Support System		
ID: 5	Declaración del riesgo: Mala estimación de los recursos necesarios para desarrollo.	
Prioridad: 5		
Probabilidad: Moderada		
Impacto: Media	Origen: Planificación	Clasificación: Humana
Periodo: Especificación de Requerimientos.		
Contexto: La estimación de recursos y plazos en proyectos es una tarea crítica, que puede conducir al fracaso del proyecto, debido al alargamiento de la duración o a una mala estimación del esfuerzo necesario para ejecutarlo.		
Plan de Mitigación: Reuniones al comienzo de cada Sprint en los que se establezcan o redefinen los requerimientos para el desarrollo del producto.		
Plan de Contingencia: Planificar el cómo adaptarse intensivamente a la característica conflictiva y resolverla dentro de los márgenes de tiempo asignado a problemas imprevistos.		

Hoja de Control de Riesgo		
Nombre Proyecto: Post Emergency Support System		
ID: 14	Declaración del riesgo: La base de datos no puede almacenar los diferentes tipos de recursos.	
Prioridad: 5		
Probabilidad: Muy Baja		
Impacto: Alto	Origen:	Clasificación: Técnica
Periodo: Instalación		
Contexto: La aplicación toma datos de distintas APIs, además de los datos ingresados por los usuarios. La errónea recolección y almacenamiento de estos dentro de la base de datos puede provocar incoherencias al momento de recibir consultas.		
Plan de Mitigación: Estandarización de los datos recolectados mediante establecimiento de un formato fijo, de manera que se mantengan valores fijos al momento de almacenarlos en la base de datos.		
Plan de Contingencia: Verificar la parte de la aplicación encargada de procesar la información, verificar y restringir formato de los datos ingresados en los campos fijos.		

5. HERRAMIENTAS, TECNICAS Y METODOLOGÍAS

Se identifican herramientas, técnicas y metodologías especiales para el soporte de las actividades de SQA. Además se indica el propósito de su utilización en el proyecto.

5.1 Aplicación de métodos técnicos formales

Son metodologías que ayudan al analista a conseguir una especificación y un diseño de alta calidad. Para el desarrollo del software se requiere documentar a través del desarrollo, los siguientes procesos:

- Product Backlog
- Análisis
- Diseño
- Implementación mediante Sprint
- Instalación (aceptación y entrega)
- Operación(mantenición)

5.2 Registro y generación de informes

Toda información respecto al proyecto debe ser debidamente registrada. Los resultados de las revisiones, auditorías, control de cambios, prueba y otras actividades de SQA deben convertirse en parte de la historia del proyecto. La documentación debe ser una de las tareas principales, dejándose constancia en ella de cualquier avance, cambio o retraso. Cada miembro del equipo de proyecto debe velar por que la información referente a sus tareas llegue oportunamente al encargado de documentación con el fin de ser registrada.

5.3 Revisiones

Las revisiones constituyen la primera forma de monitorear y evaluar la calidad de los productos de trabajo, y además, proveen mayor visibilidad al desarrollo.

El proceso de revisión es una metodología definida, estructurada y disciplinada para la detección e identificación de defectos en los productos de trabajo durante el ciclo de vida del software. Cuenta con seis etapas: planificación, orientación, preparación, inspección, rework y seguimiento, las cuales son llevadas a cabo por un equipo con tareas y responsabilidades definidas, con documentación específica y por un período de tiempo determinado.

El propósito de las revisiones es remover, temprana y eficientemente, los defectos de los productos de trabajo. Esto, fuera de facilitar el monitoreo y evaluación de la calidad, tiene por resultado una mejor comprensión de los productos de trabajo por parte de los desarrolladores y fomenta la prevención de defectos como una práctica de real importancia.

Las revisiones involucran un examen metódico de los productos de trabajo para identificar defectos y posibles cambios. Los productos a ser evaluados son seleccionados durante la definición del proyecto, donde además se planifican las actividades requeridas para ello.

5.3.1 Roles y responsabilidades

El rol de SQA es observar, participar de ser necesario y verificar que las revisiones se realicen y documenten correctamente. También debe supervisar que cualquier acción requerida, fruto de estas actividades, sea asignada, documentada, programada y actualizada.

StarLampStudio hace responsable a su jefe de proyecto de la correcta implementación de las revisiones.

El equipo de inspección contempla los siguientes roles:

Moderador	Es el facilitador de todo el proceso de revisión, responsable de su dirección y, por lo tanto, su pilar fundamental. El moderador participa en todas las etapas del proceso de revisión.
Autor	Es el desarrollador del producto de trabajo. Es responsable de preparar el producto de trabajo para la revisión, de acoger y de realizar las correcciones de los defectos detectados durante la revisión. El autor participa en todas las etapas del proceso de revisión con excepción de la preparación.
Presentador	Es el relator del producto durante la inspección. Su responsabilidad es apoyar al moderador en el control de la reunión de inspección, para lo cual debe mantener la atención de los participantes en las secciones más relevantes del producto. El presentador participa en la etapa de preparación e inspección. Además, puede participar como inspector en las etapas que así lo ameriten.
Inspector	Es un examinador del producto de trabajo bajo revisión. Cada uno es responsable de identificar y detectar errores. En conjunto, son responsables de hacer recomendaciones que puedan ser utilizadas para definir acciones correctivas y para introducir mejoras en el proceso de revisión. Participa hasta la etapa de inspección.
Secretario	Es quien lleva el registro de la reunión de inspección. Es responsable de completar la documentación sobre los defectos y su clasificación, la asignación de acciones y las lecciones aprendidas. El secretario participa en la etapa de inspección. Además, puede participar como inspector en las etapas que así lo ameriten.
Observador (opcional)	Es responsable de observar y apuntar datos sobre los participantes, sus roles e interacción con el propósito de obtener información que permita el mejoramiento del proceso de revisión. La persona asignada a este rol debe ser experta en el proceso de revisión y estar interesada en su mejoramiento. El observador participa sólo durante la etapa de inspección.

Tabla 25 Equipo de Inspección.

5.3.2 Etapas de Revisión

Se describen las etapas del proceso de inspección en términos de sus objetivos, criterios de entrada/salida y de sus actividades. Además, se enuncian los participantes de cada etapa y sus responsabilidades durante ella.

Planificación	Responsable(s): Moderador y el autor.
Objetivo Verificar que el producto se encuentre en condiciones de ser inspeccionado y planificar las actividades para garantizar el éxito de la revisión.	
Criterios de Entrada - El autor señala que el producto de trabajo está listo para la revisión. - El autor cree que el producto de trabajo cumple con los criterios de calidad requeridos para aprobar la revisión. - Existen participantes apropiados y con disponibilidad de tiempo. - El moderador y los participantes propuestos son apropiados y cuentan con la disponibilidad de tiempo requerida para participar en el proceso de revisión.	
Actividades - Selección de los participantes y asignación de roles. - Determinar el tamaño del producto de trabajo por ser revisado. - Determinar los criterios que el producto debe satisfacer. - Establecer la necesidad de una reunión orientación. - Planificar la reunión de orientación y/o la inspección. - Preparación y distribución del paquete de revisión a los participantes. - Registro del tiempo usado durante la fase de planificación.	
Criterios de Salida Cada participante completa satisfactoriamente la checklist de las tareas asignadas a la etapa de planificación.	

Orientación (opcional)	Responsable(s): Moderador, autor y cualquier otro participante que requiera información sobre el producto o el proceso de revisión.
Objetivo Instruir a los participantes sobre el producto de trabajo y reconocer el proceso de revisión que se aplicará a dicho producto.	
Criterios de Entrada Los criterios de salida de la etapa de planificación han sido completados satisfactoriamente.	
Actividades 1.Descripción general del producto de trabajo sujeto a inspección. 2.Revisar la asignación de roles y el plan para la inspección. 3.Responder cualquier consulta.	
Criterios de salida • Todos los participantes están preparados para proceder. • Cada participante completa satisfactoriamente la checklist de las tareas asignadas a la etapa de orientación.	

Preparación	Responsable(s): Moderador, presentador y cualquier participante bajo el rol de inspector.
Objetivo El objetivo de esta etapa es que cada miembro evalúe el producto de trabajo para detectar defectos, clasifique estos defectos y los documente.	
Criterios de Entrada Los criterios de salida de la etapa de planificación y de la orientación han sido completados satisfactoriamente.	
Actividades 1. Los inspectores revisan el producto de trabajo y registran los defectos detectados. 2. Clasificar todos los defectos. <i>(Los participantes clasifican los defectos detectados en el producto de trabajo según su impacto en: Menor/Mayor/No Resuelto)</i> 3. El presentador se prepara para exponer el producto durante la reunión de inspección. 4. Entrega del informe de revisión.	
Criterios de salida • Los defectos han sido correctamente documentados. • Todos los participantes están preparados para proceder en la inspección. • Cada participante completa satisfactoriamente la checklist de las tareas asignadas a la etapa de preparación.	

Inspección	Responsable(s): Moderador, presentador, autor, inspector(es), secretario y, opcionalmente, un observador.
Objetivo: Los objetivos de esta etapa son que los participantes lleguen a un consenso respecto de los defectos que afectan al producto de trabajo, decidan si requiere de una revisión adicional, y que discutan y documenten sobre las lecciones aprendidas durante el proceso.	
Criterios de Entrada Todos los participantes pueden participar de la inspección.	
Actividades Se enuncian los roles, el enfoque y las guías para llevar a cabo la inspección. 1. El presentador describe el producto de trabajo. 2. Se nombran los defectos globales que afectan la completitud del producto de trabajo. 3. Se nombran los defectos específicos por ser revisados y registrados. 4. Los errores de formatos son entregados al autor para acciones correctivas. 5. Asignación de los defectos no resueltos. 6. Determinación sobre la necesidad de tiempo adicional. 7. Emplear la hora adicional para terminar las actividades restantes. 8. Establecer la necesidad de una revisión adicional. 9. Solicitar retroalimentación a los participantes en relación con la inspección.	
Criterios de salida - Los defectos han sido registrados. - Los defectos no resueltos han sido asignados.	

Rework	Responsable(s): Autor
Objetivo El objetivo de esta etapa es corregir los defectos y resolver aquellos clasificados como no resueltos.	
Criterios de Entrada Los criterios de salida de la etapa de inspección han sido completados satisfactoriamente.	
Actividades 1. Estimar el tiempo requeridos para resolver los defectos no resueltos. 2. Corregir los defectos identificados en el producto de trabajo. 3. Resolver los defectos no resueltos del producto de trabajo 4. Corregir los errores de forma identificados en el producto de trabajo.	
Criterios de salida Todas las correcciones han sido completadas.	

Seguimiento	Responsable(s): Moderador y autor
Objetivo Los objetivos de esta etapa son asegurar que todos los defectos y los errores de forma han sido corregidos. Además, que se haya dado solución a los defectos no resueltos.	
Criterios de Entrada Los criterios de salida de la etapa de rework han sido completados satisfactoriamente.	
Actividades 1. El moderador confirma que todos los defectos y errores de forma hayan sido corregidos, y que a los defectos no resueltos se les haya dado solución. 2. Completar la documentación de la revisión.	
Criterios de salida - Los templates de revisión han sido completados y distribuidos. - Todos los defectos no resueltos han sido incorporados al monitoreo del proceso.	

5.3.3 Pauta de Informe de Revisión

- Pauta adjunta en ANEXOS

5.4 Auditoria

Una auditoría consiste en una evaluación independiente sobre la adherencia de los productos y procesos de software a las especificaciones, los estándares, procedimientos y otros acuerdos existentes. Además compara el estado del proceso y de los productos versus el estado reportado, y evalúa la efectividad real de los procedimientos.

El propósito general de una auditoría es realizar una evaluación independiente sobre la adherencia de los productos y procesos de software a las normas, estándares, guías, planes y procedimientos existentes. Y, paralelamente, comparar el estado del proceso y de los productos versus el estado reportado, y evaluar la efectividad real de los estándares y procedimientos.

Una auditoría comprende cuatro fases principales:

- Planificación
- Reunión de orientación
- Evaluación
- Seguimiento.

Y contempla los siguientes roles:

- Iniciador
- Moderador
- Auditores
- Institución auditada
- Secretario.

El proceso de auditoría comienza cuando el iniciador identifica la necesidad de una auditoría. Esto da lugar a que un auditor elabore un plan de auditoría, el cual es expuesto a los demás auditores y a la institución auditada durante la reunión de orientación. Ya definido el curso de la auditoría los auditores pueden comenzar con la evaluación. Para ello se presentan en la institución auditada para entrevistar a los desarrolladores, revisar la documentación asociada a los procesos examinados e inspeccionar los productos. Con la información recopilada, el auditor entrega las observaciones y conclusiones preliminares a la institución auditada en la reunión de cierre. Después de la discusión de estos resultados, el auditor desarrolla un informe de auditoría. Con este último la institución está en condiciones de definir las acciones correctivas y monitorear su implantación.

5.4.1 Aplicación

El proceso de auditoría es aplicable durante cualquier punto del desarrollo de software. Sin embargo, su aplicación varía según el tipo de auditoría y la fase del desarrollo.

La auditoría puede ser externa o interna. La única diferencia teórica entre estas categorías radica en el origen del equipo de auditoría. Como resulta natural, en una auditoría externa el equipo auditor no pertenece a la institución. En la interna, la situación es la contraria. En lo práctico, el mayor grado de independencia de los miembros del equipo auditor en relación con la institución auditada en el caso de una auditoría externa deriva en una opinión y en resultados más objetivos. Por otra parte, la etapa de comprensión sobre el proyecto y la forma en que se trabaja es menor e inclusive puede resultar redundante durante una auditoría interna

En el caso de la auditoría interna es importante planificarlas frecuentemente para evitar potenciales problemas y así disminuir la probabilidad de sorpresas indeseables. Lo ideal es llevarlas a cabo en la transición de las etapas del desarrollo.

Por su parte, el gran esfuerzo invertido en la planificación limita la aplicación de auditorías externas. Por ello, se recomienda realizarlas al inicio de la etapa de implementación

5.4.2 Roles y responsabilidades

El responsable de la actualización y soporte del proceso de auditoría al interior de la institución es la Unidad de SQA. Por lo tanto son de su competencia las siguientes obligaciones:

- Difundir la importancia de las auditorías.
- Ser el punto de información del proceso de auditoría.
- Capacitar a los miembros de y a sus subcontratistas sobre el proceso de auditoría.
- Proveer entrenamiento y orientación sobre los tipos de auditorías aplicables a un proyecto dado.
- Apoyar al jefe de proyectos y al nivel de gestión en la planificación de las auditorías y en el seguimiento de las acciones correctivas
- Mantener actualizado e incorporar las mejoras necesarias al proceso de auditoría

En el proceso de auditoría se incluyen cinco roles: iniciador, moderador, auditores, institución auditada y secretario. Sin embargo, algunos de estos roles no son incompatibles.

Por ejemplo:

- El iniciador puede actuar como el moderador de tratarse de una auditoría interna.
- Y, el moderador puede cumplir el rol de secretario.

También es importante resaltar que se pueden llevar a cabo auditorías con un único auditor. En tal caso el iniciador no puede actuar como moderador.

Iniciador

El iniciador es quien debe iniciar y aprobar el proceso de auditoría. Entre sus obligaciones se cuentan:

- Identificar la necesidad de una auditoría.
- Decidir el propósito y el objetivo de la auditoría.
- Decidir cuáles serán los productos/procesos auditados.
- Decidir sobre los criterios de evaluación, incluyendo las normas, guías, estándares, planes y procedimientos aplicables durante la evaluación.
- Decidir sobre el curso de la auditoría.
- Revisar y aprobar el informe de auditoría.
- Dentro del marco del proyecto, establecer y dirigir la etapa de seguimiento de las acciones correctivas.

Distribuir el informe de auditoría al interior de la institución.

Moderador (Líder del Equipo auditor)

El moderador es el facilitador de gran parte del proceso de auditoría. Como facilitador y representante máximo del equipo de auditoría es responsable de la gestión del proceso y de asegurar el logro de los objetivos trazados. Son parte de sus obligaciones:

- Preparar el plan de auditoría.
- Conformar el equipo de auditoría.
- Dirigir el equipo de auditoría.
- Explicar el plan de auditoría durante la reunión de orientación.
- Tomar decisiones sobre el curso de la auditoría.

- Actuar como auditor durante la recopilación de información de ser preciso.
- Informar sobre la evaluación durante la reunión de cierre.
- Preparar el informe de auditoría.
- Informar sobre la inhabilidad aparente de cualquiera de los participantes en el proceso de auditoría.
- Aclarar y llegar a acuerdo sobre las discrepancias e inconsistencias con el iniciador.
- Recomendar acciones correctivas.

Observación: El moderador debe ser una persona libre de influencias que puedan reducir su capacidad de toma de decisiones.

Auditores

Los auditores son responsables de examinar los productos/procesos según lo estipulado en el plan de auditoría. Además, deben registrar todas las observaciones recopiladas durante la evaluación y transmitírselas al moderador para su incorporación en el informe de auditoría.

Sus responsabilidades incluyen:

- Prepararse para la auditoría. Es decir, comprender los objetivos de trabajo y el plan de auditoría.
- Participar de la reunión de orientación, si ésta tiene lugar.
- Recopilar datos sobre los productos/procesos examinados objetivamente centrándose en detectar disconformidades y absteniéndose de emitir juicios sobre los resultados de la auditoría o sobre posibles recomendaciones.
- Registrar todas las observaciones coleccionadas durante la evaluación.
- Participar en la reunión de cierre y apoyar al moderador de ser necesario.
- Entregar todas las observaciones registradas al moderador para su inclusión en el informe de auditoría.

Institución auditada

La institución auditada debe comprometerse con el proceso de auditoría en curso nombrando a sus representantes en el proceso, facilitando a los auditores toda la información y los recursos solicitados por ellos, y, finalmente, implementando acciones correctivas.

Secretario

El secretario es quien registra y documenta todas las anomalías, decisiones, recomendaciones y conclusiones realizadas durante la reunión de cierre.

Nivel de Gestión

Los directivos del nivel de gestión (incluyendo al jefe de proyectos), tienen responsabilidades importantes asociadas a él. Éstas incluyen:

- Programar las auditorías en consideración con el proceso y las normas definidas para ello.
- Entregar las bases y las facilidades requeridas para cada una de las etapas del proceso de auditoría.
- Proveer entrenamiento y orientación sobre los tipos de auditorías aplicables a un proyecto dado.
- Asegurar una apropiada comprensión y conocimiento sobre los productos de trabajo auditados.
- Asegurar que el plan de auditoría sea llevado a cabo.

Tomar acciones pertinentes según las recomendaciones incluidas en el informe de auditoría.

5.4.3 Etapas

A continuación se describen las etapas del proceso de auditoría en términos de sus objetivos, criterios de entrada/salida y de sus actividades. Además, se enuncian los participantes de cada etapa y sus responsabilidades durante ella.

I. Planificación	Participantes(s): Iniciador, moderador (líder del equipo auditor).
Objetivo El objetivo principal de esta etapa es establecer el ámbito y los recursos para la auditoría, como también, planificar sus actividades.	
Criterios de Entrada 1. Una autoridad competente ha autorizado la auditoría. 2. Se encuentra disponible la información requerida para esta etapa de la auditoría.	
Actividades 1. El iniciador establece la necesidad de una auditoría y contacta al organismo competente. 2. El moderador debe comprender el objetivo del proyecto de desarrollo de software y los productos producidos. 3. El moderador debe informarse sobre el estado de avance del proyecto. y los problemas detectados 4. Definir, según reportes de estado, qué áreas son difusas y/o requieren de una mayor atención. 5. Desarrollar un <i>checklist</i> para la auditoría. 6. El moderador debe presentar el plan de auditoría al iniciador para su corrección y aprobación. 7. El iniciador notifica a la institución auditada sobre el desarrollo de una auditoría. 8. El auditor selecciona los miembros del equipo de auditoría.	
Criterios de Salida 1. El plan de auditoría (presentado por el moderador) ha sido aprobado. 2. La institución auditada ha sido notificada sobre la futura auditoría.	

Tabla 26 Tabla de Planificación

II. Reunión de Orientación	Participantes(s): Iniciador, moderador, auditores, secretario, institución auditada.
Objetivo El propósito de esta etapa es clarificar el contenido del plan de auditoría a los miembros de la institución auditada y corroborar que el equipo de auditoría comprende	

los objetivos del proyecto.
Criterios de Entrada 1. El plan de auditoría ha sido aprobado.
Actividades 1. El moderador explica a los participantes de la reunión el contenido del plan de auditoría. 2. Un representante de la institución auditada presenta el proyecto a los auditores. 3. Moderador e Iniciador resuelven las dudas planteadas por las partes.
Criterios de Salida 1. El equipo de auditoría comprende los objetivos del proyecto. 2. La institución auditada comprende el plan de auditoría.

Tabla 27 Reunión de Orientación

III Evaluación

Esta etapa consiste en tres actividades: Site Visit, Reunión de Cierre, e Informe de Resultados

III-1. Site Visit	Participantes(s): Auditores, institución auditada.
Objetivo El propósito esta etapa es comprobar que los productos requeridos están siendo desarrollados de acuerdo a los estándares aplicables, que el proceso se ajusta a los procedimientos definidos y que los reportes del estado del proyecto reflejan su situación actual.	
Criterios de Entrada 1. El equipo de auditoría comprende los objetivos del proyecto. 2. La institución auditada comprende el plan de auditoría. 3. Los recursos solicitados en el plan de auditoría se encuentran disponibles.	
Actividades 1. Los auditores entrevistan al equipo desarrollador. <i>Observación: Durante esta etapa la checklist desarrollada durante la planificación debe servir de guía a los auditores.</i> 2. Los auditores examinan los registros del proyecto con el objetivo de establecer si los procedimientos son seguidos adecuadamente. 3. Los auditores examinan los productos de trabajo con la finalidad de establecer si concuerdan con los definidos en los requerimientos.	
Criterios de Salida	

1. Los auditores han recopilado información suficiente sobre los productos/procesos auditados. 2. Todas las observaciones han sido debidamente registradas.
--

Tabla 28 Site Visit

III-2. Reunión de cierre	Participantes(s): Iniciador, moderador, auditores, institución auditada, secretario.
Objetivo El propósito de esta reunión es crear una instancia en que los auditores presenten los resultados (al nivel de observaciones) de la evaluación a la institución auditada para permitir a esta última manifestar su opinión frente a ellas, clarificar cualquier mal interpretación en la que los auditores hayan incurrido e indicar posibles omisiones importantes dentro de la etapa previa.	
Criterios de Entrada 1. Los auditores han recopilado información suficiente sobre los productos/procesos auditados. Como mínimo información suficiente como para responder el checklist de auditoría. 2. Todas las observaciones han sido debidamente registradas.	
Actividades 1. Los auditores informan sobre el estado de avance del proceso de auditoría. 2. Los auditores exponen las observaciones preliminares, junto con las conclusiones y recomendaciones preliminares. 3. Los representantes de la institución auditada se manifiestan ante las observaciones.	
Criterios de Salida 1. Los auditores han expuesto las conclusiones y recomendaciones preliminares. 2. Se han resuelto todas las observaciones hechas por la institución auditada. 3. Se ha llegado a acuerdo en relación con los resultados de la auditoría.	

Tabla 29 Reunión de Cierre

III-3. Informe de Resultados	Participantes(s): Moderador, auditores.
Objetivo El objetivo de la presente etapa es desarrollar y entregar un informe sobre los resultados de la auditoría.	

Criterios de Entrada - La reunión de término ha finalizado con éxito. Es decir: –Se han resuelto las discrepancias sobre los resultados de la auditoría entre los auditores y la institución auditada. –Se ha llegado a acuerdo sobre los resultados de la auditoría.
Actividades - El moderador prepara un informe de auditoría. <i>Observación: Es recomendable que el informe de auditoría sea desarrollado con posterioridad a la reunión de término y lo más tempranamente posible. Pauta en los anexos.</i> - El moderador entrega el informe de auditoría al iniciador. - El iniciador recepciona y distribuye el informe de auditoría.
Criterios de salida El informe de auditoría fue entregado al iniciador.

Tabla 30 Informe de Resultados

IV Seguimiento	Participante(s): Iniciador, institución auditada.
Objetivo El propósito del seguimiento es que el iniciador junto a la institución auditada identifique las acciones correctivas necesarias para eliminar o prevenir las disconformidades para su posterior implantación.	
Criterios de entrada El informe de auditoría fue entregado al iniciador.	
Actividades - El iniciador junto a la institución auditada identifican acciones correctivas para eliminar o prevenir las disconformidades. - Implementación de las acciones correctivas definidas. - Verificar la implantación de las acciones correctivas.	
Criterios de salida La institución auditada está comprometida con el seguimiento de las acciones correctivas iniciadas en pro de resolver las disconformidades detectadas durante la auditoría.	

Tabla 31 Seguimiento

5.4.4 Pauta de Informe de Auditoria

- Pauta adjunta en ANEXOS

5.5 Checklist

Un checklist (o lista de comprobación) es una herramienta de ayuda en el trabajo diseñada para reducir los errores provocados por los potenciales límites de la memoria y la atención en el ser humano. Ayuda a asegurar la consistencia y exhaustividad en la realización de una tarea.

Para este caso se utilizan checklist para evaluar el cumplimiento de las prácticas y actividades de SQA durante el proceso de desarrollo de software. La información resultante debe servir como base para informar sobre el estado de avance de las actividades de SQA en el proyecto.

Cada participante debe completar la checklist de los roles que le fueron asignados al finalizar cada etapa del proceso de revisión. Al finalizar su participación, el participante debe entregar la checklist de cada rol desempeñado al moderador.

- Pauta de los checklist adjunta en ANEXOS

6. PRUEBAS

La prueba es una actividad que evalúa los atributos y la capacidad de un programa o sistema para determinar si se cumple con los resultados definidos. Esto implica verificar la satisfacción de los requerimientos e identificar diferencias entre los resultados esperados y los actuales. Es decir, la prueba consiste en ejecutar un programa o sistema con la intención de encontrar defectos.

6.1 Estructura de las pruebas

El proceso de prueba incluye cinco etapas: planificación, especificación, ejecución, análisis de resultados y finalización. Durante la planificación se identifican los tipos de pruebas, se estiman los recursos y el esfuerzo requerido para ellas y se programan sus actividades. Posteriormente, para cada tipo de prueba se especifica un conjunto de casos de pruebas y los procedimientos respectivos. Sobre la base de esta especificación, se ejecutan las pruebas y se registran sus resultados, los cuales servirán para identificar defectos y sus posibles causas, así como también para establecer acciones correctivas durante el análisis. Por último se preparara el equipamiento y los casos de pruebas para uso posteriores y se terminará la documentación de todo el proceso.

A continuación se describen las diferentes etapas del proceso de prueba, con el propósito de entregar una visión global del mismo.

6.1.1 Planificación

La planificación de las actividades de la prueba comienza junto a la planificación del proyecto. Durante ese período se identifican a grandes rasgos los esfuerzos (actividades) y recursos necesarios en relación con la prueba.

Ahora bien, debido a la naturaleza de las pruebas, es necesario que los requerimientos del producto estén bien definidos para poder someter bajo evaluación dichas funcionalidades, por lo que se hace necesario esperar que los requerimientos estén definidos antes de llevar a cabo esta etapa.

Sobre la base de la especificación de requerimientos y del plan de proyectos; el jefe de proyectos, la unidad responsable de prueba y su líder serán responsables de desarrollar el plan de pruebas.

Para ello deberán realizar las siguientes actividades:

- Sobre la base de los requerimientos, definir los niveles, tipos y metodologías de prueba que serán utilizadas.
 - Especificar los estándares y guías involucradas.
 - Identificar el ambiente en que se llevarán a cabo las pruebas.
 - Establecer los criterios de aprobación/reprobación para el sistema de software.
 - Estimar el esfuerzo y los recursos necesarios.
 - Asignar un responsable del proceso de prueba para el proyecto.
 - Trazar los requerimientos con los tipos de pruebas para verificar su completitud.
 - Programar los diferentes tipos de prueba, incluyendo tiempo para las correcciones y las pruebas de estas últimas.
- Desarrollar el plan según el template correspondiente.
- Terminado el plan de pruebas, someterlo a una revisión.
- Corregir las discrepancias e incorporar las recomendaciones.
- Aprobar, distribuir y poner bajo control de configuración el del software el plan de pruebas.

6.1.2 Especificación

Sobre la base del plan de pruebas, se debe describir para los diferentes tipos de prueba los casos de prueba a nivel funcional (propósito) y a nivel detallado (procedimiento).

La descripción de los casos de prueba, involucra las siguientes actividades:

- Identificar el conjunto de casos de pruebas por cada tipo de prueba, especificando el propósito de cada caso.
 - Definir por cada caso de prueba, el conjunto de entradas requeridas para satisfacer el propósito de la prueba.
 - Definir por cada caso de prueba, las salidas esperadas.
 - Especificar los métodos de inserción/extracción para el test.
 - Identificar los tipos y los volúmenes de datos de entrada/salida.
 - Determinar criterios de evaluación para el análisis de resultados de las pruebas.
 - Definir rangos, capacidades y tiempos que permitan establecer su aprobación/reprobación.
 - Trazar los requerimientos con los casos de prueba para verificar su completitud.
 - Documentar la descripción funcional de los casos de prueba por tipo de prueba.
- Someter a revisión este documento, corregirlo, aprobarlo y ponerlo bajo control de configuración del software.

La descripción detallada abarca:

- Seleccionar los métodos y las herramientas para la ejecución de los casos de pruebas.
- Definir el procedimiento para proveer las entradas a cada caso de prueba.
- Definir criterios de aprobación/reprobación para cada etapa de los casos de prueba.
- Coleccionar los datos de entrada.
- Definir procedimientos para el análisis de los resultados de las pruebas y para comparar los resultados actuales con los esperados.
- Documentar la descripción detallada de los casos de prueba. Someter a revisión este documento, corregirlo, aprobarlo y ponerlo bajo control de configuración del software.

6.1.3 Ejecución

Una vez que se encuentra lista la especificación de las pruebas y que la codificación ha finalizado, comienza la ejecución de las pruebas.

En términos generales, para cada tipo de prueba se ejecutarán los casos de pruebas predefinidos, se registran sus resultados y se establecerá el éxito/fracaso de la prueba sobre la base de los criterios establecidos durante la especificación. Luego, se indicarán las correcciones necesarias, las cuales serán llevadas a cabo por el equipo desarrollador, para posteriormente ser nuevamente probadas con el objeto de verificar su correctitud y para prevenir la introducción de nuevos defectos. Por último, una autoridad competente dará por finalizadas las actividades correspondientes a dicho tipo de pruebas, abriéndose paso al siguiente.

Los niveles de pruebas serán ejecutados secuencialmente y en el siguiente orden:

- Pruebas de unidad al finalizar la codificación.
- Prueba de integración en paralelo con la etapa de integración.
- Prueba de aceptación durante la etapa de aceptación y entrega.
- Prueba del sistema previo a la entrega y operación del sistema.

La primera de ellas es responsabilidad del equipo desarrollador y las demás, de la unidad responsable de la prueba.

6.1.4 Análisis de resultados

Sobre la base de los resultados de las pruebas, es importante realizar una identificación de los defectos y de sus posibles causas. Esto implica establecer responsabilidades asociadas con el proceso de desarrollo y no responsabilidades individuales.

Por ejemplo, debe determinarse la fase en la cual el defecto fue introducido y la causa (no-adherencia a los estándares y procedimientos, especificación imprecisa, etc.)

Luego sobre la base de esta información deben proponerse acciones correctivas a las unidades involucradas.

6.1.5 Finalización

Para cerrar el proceso de prueba es necesario que su unidad responsable prepare el equipamiento y los casos de prueba para su uso posterior y, además, complete la documentación del presente proceso.

6.2 Pruebas Ejemplo PESS

Para el caso de la aplicación PESS, con el objetivo de mejorar la calidad de los requerimientos, se busca poder detectar fallos lo antes posible.

En esta sección se describen ejemplos de casos de pruebas que deben establecerse para el Plan de Pruebas de PESS.

6.2.1 Prueba funcional

La prueba funcional es de tipo black box. Es decir, examina el comportamiento externo del software sin indagar en su implementación interna. Ve al software como una «caja negra» en la que se ingresa una entrada y de la que se extrae una salida, pero sin analizar el proceso intermedio.

Caso de Prueba	Registrar una Post-Emergencia en el sistema	Id	P01
Propósito	Verificar si es posible registrar una Post Emergencia en el sistema.		
Prerrequisito	Todos los campos obligatorios deben ser rellenados correctamente.		
Datos de Entrada	• Título: Prueba • Tipo: Incendio • Región: V • Descripción:		
Pasos	1. Ingresar al sistema mediante login. 2. Llamar a método "AGREGAR NUEVA POST EMERGENCIA" 3. Ingresar datos de entrada requeridos. 4. Sistema presenta resultado. 5. Llama a método " LISTAR POST EMERGENCIAS" 6. Sistema presenta resultado.		
Resultados Esperados	Título de Post Emergencia aparece listado entre los registrados por el sistema.		
Componentes	• Crear Post Emergencia • Listar Post Emergencias		
Caso de Prueba	Eliminar Post Emergencia	Id	P02
Propósito	Verificar que la post emergencia que ha sido eliminada, no siga listada en el sistema y que los atributos que se relacionen con esta también hayan sido eliminados.		
Prerrequisito	Deben existir Post Emergencias definidas en la base de datos		

Datos de Entrada	Id o Título de la Post Emergencia que se desea eliminar
Pasos	1. Ingresar al sistema mediante login 2. Llamar a método "ELIMINAR POST EMERGENCIA" 3. Seleccionar una Post Emergencia para eliminarla 4. Sistema presenta resultado 5. Llamar a función "Listar Post Emergencias" para verificar si está la Post Emergencia Eliminada 6. Sistema presenta resultado
Resultados Esperados	Título de Post Emergencia no aparece listado entre los registrados por el sistema.
Componentes	• Crear Post Emergencia • Eliminar Post Emergencias

6.2.2 Prueba de estrés

Un tema importante a considerar en PESS es la capacidad de manejar muchas y constantes peticiones. Es por ello que se realiza la prueba de estrés. Esta evaluación pone a prueba la robustez y la confiabilidad de la aplicación sometiéndola a condiciones de uso extremas. Entre estas condiciones se incluyen el envío excesivo de peticiones y la ejecución en condiciones de hardware limitadas. El objetivo es saturar el programa hasta un punto de quiebre donde aparezcan defectos potencialmente peligrosos. Los efectos de la saturación pueden ser la pérdida o adulteración de datos, el uso excesivo de recursos incluso una vez finalizada la situación de stress, un mal funcionamiento de componentes de la aplicación o la aparición de errores inesperados.

En el caso de PESS, estas pruebas se realizan utilizando la herramienta Apache Bench.

Apache Bench está presente en todos los servidores apache, pero también se puede aplicar aunque no se esté usando Apache. Para medir el manejo de estrés de PESS se ejecutará un script que envíe a la aplicación 100 peticiones, siendo como máximo 10 de estas enviadas simultáneamente. La aplicación debe ser capaz de manejar todas las peticiones en un periodo de tiempo aceptable por petición. Si se encontraron problemas, es necesario revisar el diseño o el código de la aplicación para descubrir el origen del conflicto.

Caso de Prueba	Manejo de estrés del sistema	Id	P03
Propósito	Verificar el comportamiento del sistema en condiciones de sobrecarga debido a alto tráfico de peticiones.		
Prerrequisito	• Se debe contar con los privilegios para la configuración de la aplicación. • Debe realizarse desde equipo de la siguiente plataforma: ○ Arquitectura de 64 bits ○ Sistema Operativo GNU/Linux		
Datos de Entrada	ab -n 1000 -c 30 "url de PESS" Esto genera 1000 llamadas a la web de PESS, estas con 30 llamadas concurrentes.		
Pasos	1. Usuario ejecuta comando 2. Sistema entrega resultados 3. Observar los siguientes campos:		

	• Requests per second: peticiones atendidas por segundo durante la prueba. • Time per request (mean, across all concurrent requests): tiempo medio que el servidor ha tardado en atender una petición individual.
Resultados Esperados	El campo “Time per request(mean, across all concurrent requests)” entregará un valor inferior a 500.0[ms]

6.2.3 Prueba de aceptación de cliente

La prueba de aceptación es la última de las pruebas por las que debe atravesar una aplicación. En esta el cliente pone a prueba la aplicación y verifica si es que cumple con sus expectativas. Estas pruebas no se evalúan el comportamiento interno de la aplicación ya que esto no es de importancia para el cliente. El verdadero objetivo es evaluar las entradas que se realizan sobre el software y las salidas que produce, no se indaga en su comportamiento interno.

La prueba de aceptación termina de definir el nivel de calidad y le permite conocer al equipo de desarrollo qué tan bien se interpretaron los requerimientos del cliente durante la gestación del proyecto. Al final es el cliente quien tiene la decisión final de aprobar el producto, aprobarlo bajo ciertas modificaciones o invalidarlo por completo.

Caso de Prueba	Iniciar sesión como voluntario	Id	P04
Propósito	Comprobar que sistema toma correctamente los datos necesarios para iniciar sesión, que estos estén dentro de la base de datos y que, el usuario al ingresar al sistema con sus datos, solo cuente con las funciones que le corresponden a su nivel de usuario como voluntario.		
Prerrequisito	Deben existir usuarios definidos en la base de datos		
Datos de Entrada	Contraseña = 123456		
Pasos	1. Usuario ingresa a la pantalla principal del sistema 2. Ingresa rut y además password que se le fue entregado. 3. Usuario presiona Login		
Resultados Esperados	Inicio de sesión exitoso como usuario nivel voluntario.		

7. CONCLUSIONES

En esta memoria se elaboró el plan de calidad para la aplicación Post Emergency Support System.

PESS fue desarrollado con el propósito de ofrecer una plataforma de apoyo para los trabajos de los grupos de voluntarios, y que les permitiese a los dirigentes mayor control al momento de desplegar ayuda y recursos a una zona afectada por una catástrofe.

La calidad de un software no se puede medir directamente, pero si se puede medir a través de sus atributos. Para este proyecto se propone el modelo ISO 9126 como estándar para evaluar la calidad del sistema.

Usando el modelo ISO 9126 se analizan los requerimientos no funcionales y se identifican los que pertenezcan a uno de sus 6 grupos de características de calidad, funcionalidad, eficiencia, usabilidad, fiabilidad, mantenibilidad y portabilidad. Los atributos identificados deben tener un objetivo que pueda ser medible y un criterio de aceptación. El cumplimiento de estos atributos determinará qué tan satisfecho se sentirá el cliente con el producto final.

Para asegurar el correcto desarrollo del software y que este cumpla con los estándares de calidad, se elabora el plan de calidad. En este se especifican las actividades que se realizarán, los riesgos que surgen durante su desarrollo.

En pocas palabras, el plan de calidad contará con todo lo necesario para que el software PESS cumpla con estándares internacionales de calidad.

Para el caso de PESS, debido posee un modelo de desarrollo poco convencional que toma elementos inspirados en Scrum, se realizará un plan de calidad que sea capaz de coincidir con el modelo de desarrollo definido en su plan de proyecto de PESS.

Por último, se analizó la aplicación del modelo de madurez para el proceso de SQA, el cual consiste en una selección de métodos, pautas y herramientas disponibles para facilitar la implementación de SQA en una organización. Además el modelo cuenta con una introducción a la implementación de un proceso de prueba con el propósito de dar un apoyo a los desarrolladores y al responsable de la definición, planificación, e implementación del proceso de SQA.

Para este proyecto se optó por implementar el nivel 1 de madurez de SQA, el cual consiste de los métodos:

- Revisiones
- Auditorias

Y la herramienta Checklist.

Las revisiones son exámenes metódicos que se aplican a los productos de trabajo con el propósito de hallar defectos y determinar posibles cambios a su construcción.

Las auditorías son procesos de prueba que se realizan a los productos de trabajo y a los llevan a cabo su construcción con el propósito de evaluar si estos se están adhiriendo a los estándares y procedimientos definidos en el plan de calidad. Son realizados por un auditor capacitado que haya sido designado por una autoridad competente, siendo en este caso el equipo de SQA.

La herramienta de apoyo sugerida por el modelo de madurez es uso de las Checklist. Utilizando las checklist se puede monitorear el cumplimiento de las prácticas y actividades de aseguramiento de calidad de una forma que reduzca los errores provocados por los potenciales límites de la memoria y atención del ser humano.

Para medir los atributos de calidad se cuenta con las pruebas.

Mediante las pruebas se evaluará si se alcanzaron los resultados que se esperaban en las diferentes actividades. Durante la fase de planificación se identifican los tipos de pruebas, se estiman los recursos y el esfuerzo requerido para ellas y se programan sus actividades. Con esta base, se ejecutan las pruebas y se registran sus resultados, los cuales servirán para identificar defectos y sus posibles causas, así como también para establecer acciones correctivas.

En resumen, el plan de calidad es uno de los documentos más importantes a elaborar para poder desarrollar un software competente en el mercado actual. En este se establecen las acciones a llevar a cabo, las herramientas de apoyo para realizar estas acciones, los equipos responsables de estas, los riesgos que pueden surgir durante su desarrollo y los costos a tomar en cuenta para que se pueda desarrollar un producto de alta calidad. Además en este se establecerá la documentación de cada acción durante el desarrollo y se registrará cualquier riesgo que pueda surgir para posteriormente establecer medidas para prevenir y/o solucionar estos riesgos. Por último establecerá pruebas e informes de cierre a realizar cuando el proceso de desarrollo estará completado con los que se verificará que el resultado de proceso cumple con los objetivos que se definieron en un principio. Y con estos objetivos aprobados, se pueda por fin, concluir que se desarrolló un producto satisfactorio y de alta calidad.

8. GLOSARIO

Ambiente de producción: Ambiente para el cual el cliente busca sea la aplicación.

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.

Baseline: Una imagen, es decir, una posición o situación que ha sido registrada. La “baseline” servirá como referencia del estado original antes de comenzar actividades de desarrollo, y se usa como comparación con la posición actual. Los productos que hayan pasado la inspección de calidad y que estén aprobados se denominan productos baseline.

Historias de usuario: son descripciones cortas de una necesidad de un cliente del software en desarrollo. Su utilización es común cuando se aplican marcos de trabajo ágiles. No son requerimientos ni casos de uso, son más cortas, no abarcan más de 5 palabras y no deben describir la interfaz con el usuario, los pasos de navegación o flujo de procesos de la aplicación.

PESS: Post Emergency Support System.

SQA (Software Quality Assurance): Son las actividades sistémicas que ayudan a asegurar, en el desarrollo de un software, los requerimientos, estándares y procedimientos de éste.

SCM (Software Configuration Management): Gestión de configuración de software, es una especialización de la Gestión de configuración a todas las actividades en el sector del desarrollo de software.

Scrum: Modo de desarrollo de tipo ágil para gestionar el desarrollo de software, consistente en desarrollar de forma incremental y auto organizada, sin una planificación completa del producto, priorizando primero los requerimientos de mayor prioridad y posteriormente cuando se tengan todos juntarlos en el producto final.

9. BIBLIOGRAFIA

9.1. Enlaces Web

[Wiki - IEC_9126] - https://es.wikipedia.org/wiki/ISO/IEC_9126

[WEB-1] Herramientas específicas para SQA y SCM -
<https://www.inf.utfsm.cl/~visconti/herramientas/>

[WEB-2] <http://www.martinalaimo.com/es/blog/gerentes-de-proyecto-agiles>

[Wiki - Pruebas de Carga] - https://es.wikipedia.org/wiki/Pruebas_de_rendimiento_del_software

[Testing como Práctica para Evaluar la Eficiencia en Aplicaciones Web] - Delvis Echeverría Pérez y Ariannis Abella Paumier -

<http://sistemas.unla.edu.ar/sistemas/redisla/ReLAIS/relais-v2-n5-307-309.pdf>

[Modelos de Fiabilidad de Software] - Jesús Plaza Rubio -
<https://uvadoc.uva.es/bitstream/10324/4095/1/PFC-B.3.pdf>

9.2. Libros

[SEPA-1] Ingeniería del software. UN ENFOQUE PRÁCTICO. SÉPTIMA EDICIÓN. - Roger S. Pressman,

9.3. Memorias

PLAN DE CALIDAD PARA PRODUCTO DE SOFTWARE "X-PRO-L". - Gonzalo Tomas Perez Lara

PLAN DE CALIDAD PARA PRODUCTO DE SOFTWARE "Huellas Universitarias". - Joaquín Alfonso Nuñez Garcia

PLAN DE CALIDAD PARA PRODUCTO DE SOFTWARE "CL.I.K". - Andrés David Marchant Lopez

10. ANEXOS

5.3.3 Pauta de Informe de Revisión

1 Información del Proyecto y del Producto		
Proyecto:		
Producto	Tipo	

2 Inspector			
Nombre			
e-mail	Fono		

3 Resumen				
Recibido: __/__/__	Entregado: __/__/__	Resumen de Defectos		
Preparación			Preparación	Inspección
Fecha	Horas	Mayores		
		Menores		
		No resueltos		
Total Horas: ____		Total		

4 Observaciones	
Tiempo de rework estimado: ____	
Comentarios:	

5 Defectos							
Nro	Inicial	Ubicación	Descripción	Mayor	Menor	No Resuelto	Excluido

Informe de Auditoría

1 Identificación de la auditoría			
Proyecto			
Proyecto:			
Jefe de Proyecto:			
Fase del ciclo de vida			
☐ Planificación ☐ Esp. de Requerimientos ☐ Diseño ☐ Implementación ☐ Integración y pruebas ☐ Aceptación y entrega ☐ Mantención			
Auditoría			
Propósito y objetivos:			
Tipo de auditoría: ☐ Interna ☐ Externa		Fecha inicio: __/__/__	Fecha término: __/__/__
Participantes			
Institución auditada: _____		Institución auditora: _____	
Representantes		Representantes	
Iniciador		Moderador	
Otros representantes		Auditor	
		Auditor	
		Auditor	
		Auditor	

2 Objetos de auditoría	
Procesos:	
1 _____	6 _____
2 _____	7 _____
3 _____	8 _____
4 _____	9 _____
5 _____	10 _____
Productos:	
1 _____	6 _____
2 _____	7 _____
3 _____	8 _____
4 _____	9 _____
5 _____	10 _____

3 Bases para la evaluación		
Normas, estándares, guías, planes y procedimientos		
1 _____	6 _____	
2 _____	7 _____	
3 _____	8 _____	
4 _____	9 _____	
5 _____	10 _____	
Criterios de evaluación		
Criterio	Descripción	Nivel de satisfacción

Informe de Auditoría

4 Actividades de auditoría				
Actividad	Fecha (Desde- Hasta)		Estado ¹	Observación

¹R: Realizada, I: Incompleta, NR: No realizada

5 Resumen de resultados				
Anomalías más relevantes				
Nro.	Descripción	Ubicación	Prioridad ¹	Recomendación
¹ A: Alta, M: Media; B: Baja				
Conclusiones				
Conclusiones:				
Estado de la auditoría ☐ Aprobada ☐ Condicionada ☐ Reprobada Fecha auditoría de verificación: __/__/__ Tipo de auditoría: ☐ Interna ☐ Externa				
Documentos anexados			Firma del Moderador Firma del Iniciador	

5.5 Pauta de Checklist

5.5.1 Checklist por actividades del proceso de desarrollo evaluados por QA

Planificación		
1. Identificación		
Proyecto:	Fecha de inicio: _/_/____ Fecha de término: _/_/____	
Nombre:		
E-mail:	Fono:	
2. Actividades		
	Sí	No
• ¿Se elaboró el plan de SQA?		
• ¿Se apoyó la elaboración del plan de proyecto?		
• ¿Se entregó el soporte solicitado por el personal de desarrollo?		
Auditoría (opcional)		
• ¿Se auditó el proceso de planificación?		
Reporte de problemas y acciones correctivas		
• ¿Se informó sobre el estado de avance de las actividades de SQA?		
• ¿Se informaron las desviaciones detectadas?		
Revisión		
• ¿Se evaluó la selección de los productos de trabajo??		
• ¿Se evaluó la selección de las herramientas?		
• ¿Se evaluó el plan de proyecto?		
• ¿Se evaluó el plan de SQA.?		
• ¿Se evaluó el plan de SCM?		

Especificación de Requerimientos		
1. Identificación		
Proyecto:	Fecha de inicio: _/_/____ Fecha de término: _/_/____	
Nombre:		
E-mail:	Fono:	
2. Actividades		
	Sí	No
• ¿Se entregó el soporte solicitado por el personal de desarrollo?		
Auditoría (opcional)		
• ¿Se auditó el proceso de especificación de requerimientos?		
Reporte de problemas y acciones correctivas		
• ¿Se informó sobre el estado de avance de las actividades de SQA?		
• ¿Se informaron las desviaciones detectadas?		
Revisión		
• ¿Se evaluó la especificación de requerimientos?		

Diseño		
1. Identificación		
Proyecto:	Fecha de inicio: _/_/____ Fecha de término: _/_/____	
Nombre:		
E-mail:	Fono:	
2. Actividades		
	Sí	No
• ¿Se entregó el soporte solicitado por el personal de desarrollo?		
Auditoría (opcional)		
• ¿Se auditó el proceso de diseño?		
Reporte de problemas y acciones correctivas		
• ¿Se informó sobre el estado de avance de las actividades de SQA?		
• ¿Se informaron las desviaciones detectadas?		
Revisión		
• ¿Se evaluó el diseño preliminar?		
• ¿Se evaluó el diseño detallado?		
• ¿Se evaluaron los planes de prueba (sistema, aceptación, integridad y unidad)?		
Prueba		
• Planificación de la prueba del sistema.		
• Planificación de la prueba de aceptación.		
• Planificación de la prueba de integración.		
• Planificación de la prueba de unidad.		

Implementación mediante Sprint		
1. Identificación		
Proyecto:	Fecha de inicio: _/_/____ Fecha de término: _/_/____	
Nombre:		
E-mail:	Fono:	
2. Actividades		
	Sí	No
• ¿Se entregó el soporte solicitado por el personal de desarrollo?		
Auditoría (opcional)		
• ¿Se auditó el proceso de codificación?		
• ¿Se auditó la prueba de unidad?		
Reporte de problemas y acciones correctivas		
• ¿Se informó sobre el estado de avance de las actividades de SQA?		
• ¿Se informaron las desviaciones detectadas?		
Revisión		
• ¿Se evaluó la especificación de los casos de prueba?		
• ¿Se evaluó la especificación de los procedimientos de prueba?		
• ¿Se evaluó el código y su documentación?		
• ¿Se evaluaron los resultados de la prueba de unidad?.		
Prueba		
• Especificación de casos de prueba.		
• Especificación de procedimientos de prueba.		
• Prueba de unidad		

Integración y pruebas		
1. Identificación		
Proyecto:	Fecha de inicio: _/_/____ Fecha de término: _/_/____	
Nombre:		
E-mail:	Fono:	
2. Actividades		
	Sí	No
• ¿Se entregó el soporte solicitado por el personal de desarrollo?		
Auditoría (opcional)		
• ¿Se auditó el proceso de integración?		
• ¿Se auditó el proceso de prueba o parte de él?		
Reporte de problemas y acciones correctivas		
• ¿Se informó sobre el estado de avance de las actividades de SQA?		
• ¿Se informaron las desviaciones detectadas?		
Revisión		
• ¿Se evaluó el proceso de integración?		
• ¿Se evaluó la prueba y sus resultados?		
Prueba		
• Prueba de integridad.		
• Prueba de aceptación.		
• Prueba del sistema.		
• Análisis y reporte de resultados		

Aceptación y entrega		
1. Identificación		
Proyecto:	Fecha de inicio: _/_/____ Fecha de término: _/_/____	
Nombre:		
E-mail:	Fono:	
2. Actividades		
	Sí	No
• ¿Se entregó el soporte solicitado por el personal de desarrollo?		
Auditoría		
• ¿Se realizaron las auditorías de la configuración funcional y física.		
Reporte de problemas y acciones correctivas		
• ¿Se informó sobre el estado de avance de las actividades de SQA?		
• ¿Se informaron las desviaciones detectadas?		
Revisión		
• ¿Se evaluó el software y su documentación		

Mantención		
1. Identificación		
Proyecto:	Fecha de inicio: __/__/____ Fecha de término: __/__/____	
Nombre:		
E-mail:	Fono:	
2. Actividades		
	Sí	No
• ¿Se planificaron las actividades de SQA asociadas a la mantención?		
• ¿Se entregó el soporte solicitado por el personal de desarrollo?		
Auditoría (opcional)		
• ¿Se auditó el proceso de mantención?		
Reporte de problemas y acciones correctivas		
• ¿Se informó sobre el estado de avance de las actividades de SQA?		
• ¿Se informaron las desviaciones detectadas?		
Revisión		
• ¿Se evaluó cada cambio introducido durante la mantención a nivel de análisis, diseño, e implementación?.		
• ¿Se evaluó la documentación asociada a los cambios?		
• ¿Se evaluó la nueva versión/edición del software? ¿Y su documentación?		
Prueba		
• Planificación, especificación, desarrollo de las pruebas, análisis y reporte de resultados para los cambios introducidos durante la mantención.		

5.5.2 Checklist por productos de trabajo evaluados por QA

Checklist: Planes del Proyecto			
1. Identificación del proyecto y del producto			
Proyecto:			
Producto:			
2. Inspector			
Nombre		Rol ☐ Moderador ☐ Presentador ☐ Inspector	☐ Secretario ☐ Observador
e-mail	Fono		
3. Checklist			
	Sí	No	N/A
Adherencia			
¿El documento se adhiere a los estándares establecidos?			
Claridad			
¿Se alcanza el propósito principal del plan y de sus secciones?			
¿Se encuentran claras y bien precisadas las asignaciones de recursos, la calendarización y los hitos?			
¿El plan es de fácil lectura?			
¿La terminología utilizada es consistente y comprensible por el jefe de proyectos y los desarrolladores?			
Compleitud			
¿Se cuenta con una especificación de requerimientos preliminar adecuada para el estudio de soluciones factibles?			
¿Se informan los resultados de los análisis de alternativas realizados? ¿Se justifican las decisiones tomadas?			
¿El proceso de desarrollo ha sido apropiadamente seleccionado? ¿Es explicado por procedimientos sobre su monitoreo y aplicación?			

¿Se especifican todas las funciones necesarias y suficientes para completar los objetivos del sistema?			
¿Se describen las entradas/proceso/salidas necesarias y suficientes para cada función?			
¿Se establecen los tiempos de respuesta esperados por el usuario?			
¿Se definen formalmente todas las interfaces internas/externas del sistema?			
¿Se incluyen los requerimientos de interfaz entre el hardware, software y el usuario?			
¿Se definen los niveles de seguridad requeridos?			
¿Se especifica la confiabilidad incluyendo las consecuencias de las fallas del software, la información que debe ser protegida de estas fallas, la detección de errores y la recuperación?			
¿Se definen los criterios de éxito? ¿Se definen atributos de calidad que permitan medir los requerimientos? ¿Se les asigna un valor objetivo?			
¿Se definen métodos de prueba para cada requerimiento de software?			
¿Se definen los requerimientos de mantenibilidad especificando la escalabilidad del software, interfaces con otros sistemas, precisión, rendimiento, etc.?			
¿Se especifica el impacto del incumplimiento de los requerimientos?			
Consistencia			
¿Los requerimientos son consistentes entre ellos y con requerimientos de sistemas relacionados?			
¿Los requerimientos son consistentes con la especificación de requerimientos preliminar de la planificación?			
Facilidad de pruebas			
¿Es factible probar, demostrar o analizar el cumplimiento de los requerimientos?			
¿Los requerimientos son lo suficientemente precisos para facilitar la especificación de las pruebas?			
Factibilidad			
¿Es posible implementar los requerimientos con las técnicas, herramientas, recursos y personal definidos y bajo los costos y la calendarización estipulada?			
¿Es posible satisfacer los atributos de calidad definidos?			
¿Son factibles el diseño, implementación, mantención y operación del software?			

Checklist: Diseño Preliminar			
1. Identificación del proyecto y del producto			
Proyecto:			
Producto:			
2. Inspector			
Nombre		Rol ☐ Moderador ☐ Presentador ☐ Inspector ☐ Secretario ☐ Observador	
e-mail	Fono		
3. Checklist			

	Sí	No	N/A
Adherencia			
¿El documento se adhiere a los estándares establecidos?			
¿El diseño fue desarrollado de acuerdo a las metodologías y técnicas predefinidas?			
Claridad			
¿El diseño representa claramente la arquitectura (flujos de datos, flujos de control e interfaces)?			
¿Se documentan todos los objetivos, suposiciones, restricciones, decisiones y dependencias de este diseño?			
¿La terminología utilizada es consistente con la empleada por los desarrolladores?			
Compleitud			
¿Se encuentran claros los objetivos del diseño preliminar?			
¿Se incluye una descripción del procedimiento que se utilizó para desarrollar el diseño preliminar (técnicas, representación del diseño, etc.)?			
¿Existe una lista de las funciones que deben ser provistas por el software?			
¿La especificación de módulos cubre completamente la funcionalidad de los requerimientos del software?			
¿La especificación de los módulos contempla su funcionalidad, entradas, salidas, los criterios de ejecución y la interfaz con otros módulos?			
¿Existe un modelo de la interfaz entre el sistema y el usuario final: (1) descripción de los conocimientos técnicos del usuario, (2) información sobre la flexibilidad y adaptabilidad de la interfaz usuaria, (3) información sobre tutoriales, asistencia y manuales para el usuario, (4) tareas que el usuario deberá desempeñar, (5) y la apreciación del usuario con respecto a las tecnologías de la información?			
¿Se modelan todas las interfaces?			
¿Se diseña la interfaz considerando al usuario final?			
¿Se describen y justifican las estructuras de datos?			
¿Se especifica la organización y los contenidos de la base de datos?			
¿Se describen y justifican los algoritmos más relevantes?			
¿Se han identificado y analizado las rutas de ejecución críticas?			
Confiabilidad			
¿El diseño prevé la detección y recuperación de errores?			
¿Son descritas completamente las condiciones de error?			
Consistencia			
¿Se utilizan consistentemente los nombres de los elementos de datos, procedimientos y funciones a lo largo de la descripción y representación del diseño?			
¿El diseño representa el hardware, el software y el entorno del sistema?			
¿El diseño es consistente con la especificación de requerimientos?			
Facilidad de pruebas			
¿Es factible probar, demostrar o analizar que el diseño preliminar satisface los requerimientos?			

¿Es posible integrar y probar el software generado a partir de este diseño?			
Factibilidad			
¿El diseño es factible según la calendarización, el presupuesto y la tecnología disponibles?			
Mantenibilidad			
¿El diseño es modular?			
¿Los módulos tienen alta cohesión y bajo acoplamiento?			
Trazabilidad			
¿Es posible trazar el diseño con los requerimientos?			

Checklist: Diseño Detallado						
1. Identificación del proyecto y del producto						
Proyecto:						
Producto:						
2. Inspector						
Nombre		Rol		☐ Moderador ☐ Presentador ☐ Inspector ☐ Secretario ☐ Observador		
e-mail	Fono					
3. Checklist						
				Sí	No	N/A
Adherencia						
¿El documento se adhiere a los estándares establecidos?						
¿El diseño fue desarrollado de acuerdo a las metodologías y técnicas predefinidas?						
Claridad						
¿El diseño representa claramente flujos de datos, flujos de control e interfaces?						
¿La terminología utilizada es consistente con la empleada por los desarrolladores?						
Completitud						
¿El diseño satisface completamente los requerimientos y su propósito?						
¿El diseño implementa los algoritmos previamente especificados?						
¿Se definen e inicializan todas las variables y constantes definidas?						
¿Se definen y verifican apropiadamente todas las entradas y salidas?						
¿Se especifican las unidades de medición, los rangos, la exactitud y la precisión de todos los parámetros?						
¿Se especifican las restricciones, como tiempo de procesamiento y tamaño, para cada unidad de diseño?						
¿Existe validación de las entradas, salidas, interfaces y resultados?						
¿Se utilizan mensajes de error significativos para los errores?						
Consistencia						

¿Se utilizan consistentemente los nombres de los elementos de datos a lo largo de la descripción y representación del diseño?			
¿El diseño de las interfaces es consistente entre sí y con el diseño preliminar?			
Facilidad de pruebas			
¿Se puede probar, demostrar y analizar que cada unidad de diseño cumple con los requerimientos establecidos?			
Factibilidad			
¿El diseño es factible según la calendarización, el presupuesto y la tecnología disponibles?			
Mantenibilidad			
¿Se ha reducido el nivel de complejidad del diseño?			
¿Las unidades de diseño son claras, comprensibles y modificables en relación con los requerimientos de mantención?			
Trazabilidad			
¿Es posible trazar el diseño con los requerimientos?			

Checklist: Código

1. Identificación del proyecto y del producto

Proyecto:

Producto:

2. Inspector

Nombre

e-mail

Fono

Rol

☐ Moderador

☐ Presentador

☐ Inspector

☐ Secretario

☐ Observador

3. Checklist

	Sí	No	N/A
Completitud			
¿El código es completo y preciso de acuerdo con la documentación del diseño?			
¿El código se integra y debugged para satisfacer la especificación del diseño?			
¿Se generan las bases de datos necesarias, incluyendo los datos iniciales?			
¿No existen variables, constantes y tipos de datos superfluos? ¿Están bien definidos y son correctamente referenciados?			
Consistencia			
¿El código es consistente con el diseño?			
¿Se utilizan siempre los mismos formatos, tipos de invocaciones y estructuras?			
Correctitud			
¿El código se adhiere a los estándares definidos?			
¿Se definen y utilizan correctamente todas las variables?			
¿Los comentarios son precisos?			

¿El número de parámetros de las invocaciones es el correcto?			
Facilidad de comprensión			
¿Los comentarios describen cada rutina en forma clara y completa?			
¿Existe código ambiguo o innecesario? Si es así, ¿está adecuadamente documentado?			
¿Se utilizan técnicas de formato para facilitar la comprensión (indentación, espacios en blancos, etc.?			
¿Se ocupan convenciones mnemotécnicas para la denominación de las variables? ¿Los nombres reflejan el tipo de variables			
¿Los rangos válidos para las variables están definidos?			
¿El código utiliza ecuaciones matemáticas concordantes con la descripción de los algoritmos contenida en el diseño?			
Facilidad de pruebas			
¿Se suprime el uso de técnicas y prácticas que dificulten las pruebas?			
Mantenibilidad			
¿El código referencia simbólicamente las constantes para facilitar los cambios?			
¿Se incluyen diccionarios de datos y cross-references para mostrar el acceso a las variables y constantes dentro del programa?			
¿Los “subprogramas” cuentan con un único punto de entrada y salida?			
Previsibilidad			
¿El código fue escrito en un lenguaje con sintaxis y semántica bien definidas?			
¿El código evita utilizar los parámetros predefinidos por el lenguaje?			
¿El código se encuentra libre de loops infinitos?			
¿Se evita la recursividad?			
Robustez			
¿Se prevén los errores de ejecución como divisiones por cero, rangos no aceptables para las variables, stack overflow, entradas inválidas, etc.?			
Trazabilidad			
¿El código identifica cada programa de una única forma?			
¿El código puede ser fácilmente trazado con el diseño?			
¿El código contiene un histórico con los cambios que se le han realizado junto con las razones asociadas?			

Checklist: Plan de pruebas			
1. Identificación del proyecto y del producto			
<u>Proyecto:</u>			
<u>Producto:</u>			
2. Inspector			
<u>Nombre</u>		<u>Rol</u> ☐ Moderador ☐ Presentador ☐ Inspector	☐ Secretario ☐ Observador
<u>e-mail</u>	<u>Fono</u>		

3. Checklist			
	Sí	No	N/A
Adherencia			
¿El documento se adhiere a los estándares establecidos?			
Claridad			
¿El plan es de fácil lectura?			
¿La terminología utilizada es consistente con la empleada por los desarrolladores?			
Complejidad			
¿Se describen adecuadamente las funciones que serán probadas?			
¿Para todos los requerimientos considerados no testeables, se entrega una explicación?			
¿Se definen los tipos de pruebas apropiadamente?			
¿Se definen los criterios de éxito para todas las pruebas?			
¿Se establecen las condiciones sobre las cuales las pruebas serán interrumpidas?			
¿Se especifica claramente el orden de los pasos a seguir para la integración de las pruebas?			
¿Se definen correctamente las pruebas de regresión?			
¿Se prevé recopilar suficientes datos para la estimación de la confiabilidad del software?			
¿Se ha calendarizado la obtención/utilización de los recursos, métodos y herramientas necesarias para realizar las pruebas?			
¿Se describe la programación de la etapa de pruebas con el suficiente nivel de detalle?			
¿Se han definido los roles y responsabilidades para todos los individuos involucrados en las pruebas?			
¿Se menciona la participación de personal de SQA para la verificación de las actividades de prueba?			
Correctitud			
¿Son realistas los criterios de entrada y salida de las pruebas?			
¿El conjunto de casos de pruebas incluye la cobertura de entradas ilegales y conflictivas?			
¿El conjunto de casos de pruebas contempla el uso adecuado de los valores de entrada por defecto?			
¿El conjunto de casos de pruebas contiene un número apropiado de rutas de error?			
¿Son suficientes y adecuadas las pautas para ejecutar el plan de pruebas?			
Factibilidad			
¿Es posible realizar las actividades descritas en el plan con la calendarización, el presupuesto y la tecnología disponibles??			
Mantenibilidad			
¿Se contemplan en el plan de pruebas el manejo de los cambios que podrían ocurrir en la especificación de requerimientos, diseño o código?			
Trazabilidad			
¿Son los criterios de aceptación de las pruebas trazables con los requerimientos?			

¿El conjunto de casos d prueba contempla las interfaces definidas?			
--	--	--	--

Checklist: Procedimientos y casos de prueba				
1. Identificación del proyecto y del producto				
<u>Proyecto:</u>				
<u>Producto:</u>				
2. Inspector				
<u>Nombre</u>		<u>Rol</u> ☐ Moderador ☐ Presentador ☐ Inspector	☐ Secretario ☐ Observador	
<u>e-mail</u>	<u>Fono</u>			
3. Checklist				
		Sí	No	N/A
Adherencia				
¿El documento se adhiere a los estándares y procesos definidos?				
Claridad				
¿Las explicaciones sobre la ejecución de los procedimientos de prueba son claras y explícitas?				
¿Las instrucciones son entregadas como un conjunto ordenados de pasos por seguir?				
¿Los pasos del inicio y de los procedimientos de las pruebas son precisos y están libres de ambigüedades?				
¿Los criterios de éxito y fracaso son claros y no ambiguos?				
Completitud				
¿La función probada se describe con exactitud?				
¿La función probada corresponde a la última versión de dicha función?				
¿Se asocia cada requerimiento y las funciones asociadas a él con un procedimiento de prueba?				
¿El propósito de los procedimientos y casos de prueba es claro y preciso?				
¿Los procedimientos de prueba enumeran la precedencia de los casos de prueba?				
¿Los procedimientos de prueba especifican los equipos, el software y el personal requerido para los casos de prueba?				
¿El procedimiento describe las respuestas esperadas para cada caso de prueba?				
¿Los procedimientos indican como evaluar los resultados de los casos de prueba (criterios de éxito/fracaso de la prueba)?				
¿El procedimiento indica si es o no posible seguir con las pruebas ante la caída del sistema?				
¿Los casos de prueba validan la respuesta del sistema a entradas ilegales o conflictivas?				
Confiabilidad				
¿Se ha validado el equipo para las pruebas?				
¿Se han validado las pruebas?				

¿Se han verificado todas las entradas de datos?			
¿ Se recopilan y documentan suficientes datos para la estimación de la confiabilidad del software?			
Consistencia			
¿Se identifican todas las dependencias entre los procedimientos?			
Correctitud			
¿Concuerdan los resultados de éxito definidos en los procedimientos con el comportamiento del sistema esperado?			
Facilidad de pruebas			
¿Es factible realizar las pruebas con el mínimo respaldo de los desarrolladores?			
Trazabilidad			
¿Los procedimientos de prueba indican todas las especificaciones, procedimientos, guías o manuales requeridos para su operación?			
¿Es visible la trazabilidad entre los requerimientos y la combinación de las pruebas?			

Checklist: Manual del usuario				
1. Identificación del proyecto y del producto				
<u>Proyecto:</u>				
<u>Producto:</u>				
2. Inspector				
<u>Nombre</u>		<u>Rol</u> ☐ Moderador ☐ Presentador ☐ Inspector	☐ Secretario ☐ Observador	
<u>e-mail</u>	<u>Fono</u>			
3. Checklist				
	Sí	No	N/A	
Adherencia				
¿El documento se adhiere a los estándares definidos?				
Claridad				
¿La terminología utilizada es consistente con los conocimientos del usuario?				
¿La documentación es de fácil lectura y comprensión para el usuario final?				
¿El manual del usuario se encuentra a nivel del usuario final del software?				
Compleitud				
¿El manual del usuario describe toda la funcionalidad del sistema?				
¿El manual del usuario entrega las referencias pertinentes al usuario?				
¿El manual del usuario detalla las interrelaciones entre las funciones del software?				
¿El manual del usuario describe la ayuda en línea que provee el sistema?				

¿El manual del usuario referencia los tutoriales disponibles para el sistema?			
¿El manual del usuario referencia las interfaces con otros sistemas?			
¿Se cita en el manual del usuario los estándares de performance acordados?			
Mantenibilidad			
¿El manual del usuario fue diseñado para facilitar su mantenibilidad?			

5.5.3 Checklist de Revisión

Checklist del Moderador		
1. Identificación		
Proyecto:	Fecha de recepción: _/_/_	
Producto:	Fecha de entrega: _/_/_	
Nombre:		
E-mail:	Fono:	
2. Checklist		
	Sí	No
• ¿Se registró el esfuerzo invertido en cada una de las etapas?		
Planificación		
• ¿Trabajó con el jefe de proyectos en la selección de los participantes y en la asignación de roles?		
• ¿Determinó el tamaño del producto y los criterios de aprobación?		
• ¿Determinó la necesidad de una reunión de orientación?		
• ¿Planificó la reunión de orientación y/o la de inspección?		
• ¿Creó y distribuyó el paquete de revisión?		
• ¿Confirmó la recepción del paquete de revisión con los participantes?		
Orientación		
• ¿Confirmó la comprensión del producto, explicó los roles asignados y respondió a cualquier consulta de los participantes?		
Preparación		
• ¿ Se preparó para la inspección utilizando la checklist de inspector?		
• ¿Revisó la completitud de los Informes de Revisión?		
• ¿Determinó si todos los participantes se han preparado adecuadamente para la inspección?		
Inspección		
• ¿Enunció los roles, el enfoque y entregar pautas sobre la inspección?		
• ¿Al finalizar la reunión, decidió si se requiere de tiempo adicional?		
• ¿Al finalizar la reunión, asignó los defectos no resueltos?		
• ¿Al finalizar la reunión, decidió si una revisión adicional es necesaria?		
• ¿Al finalizar la reunión, solicitó retroalimentación a los participantes y al observador?		
• ¿Al finalizar la reunión, consultó al autor sobre el tiempo estimado para llevar a cabo las correcciones?		
Seguimiento		
• ¿Verificó que todos los defectos y errores de forma hayan sido corregidos satisfactoriamente?		
• ¿Informó y distribuyó los resultados de la revisión?		

Checklist del Autor		
1. Identificación		
Proyecto:	Fecha de recepción: _/_/_	
Producto:	Fecha de entrega: _/_/_	
Nombre:		
E-mail:	Fono:	
2. Checklist		
	Sí	No
• ¿Registró el esfuerzo invertido en cada una de las etapas?		
Planificación		
• ¿Confirmó la completitud del producto de trabajo para la revisión?		
• ¿Recomendó posibles participantes al moderador?		
• ¿Apoyó al moderador en determinar el tamaño del producto y los criterios de aprobación?		
• ¿Apoyó al moderador en la decisión de la necesidad de una reunión de orientación?		
Orientación		
• ¿Describió globalmente el producto de trabajo?		
• ¿Respondió a las consultas de los participantes?		
Inspección		
• ¿Reconoció los defectos detectados en el producto de trabajo?		
• ¿Entregó breves explicaciones técnicas sobre el producto?		
Rework		
• ¿Corrigió todos los defectos identificados en el producto?		
• ¿Corrigió todos los problemas de forma detectados en el producto?		
Seguimiento		
• ¿Verificó que todos los defectos hayan sido corregidos durante el rework?		
• ¿Cooperó con el moderador en caso de necesitarse acciones correctivas adicionales?		
• ¿Cooperó con el moderador y los inspectores en la revisión adicional si esta tiene lugar?		

Checklist del Inspector		
1. Identificación		
Proyecto:	Fecha de recepción: _/_/_	
Producto:	Fecha de entrega: _/_/_	
Nombre:		
E-mail:	Fono:	
2. Checklist		
	Sí	No
• ¿Registró el esfuerzo invertido en cada una de las etapas?		
Planificación		
• ¿Confirmó la habilidad para desempeñar el rol asignado?.		
Orientación		
• Si se estimó necesario, asistió a la reunión de orientación, confirmó el rol y/o los múltiples roles asignados a la revisión.		
Preparación		
• ¿Confirmó la recepción del paquete de revisión y de cualquier información entregada por el moderador?		
• ¿Revisó los <i>templates</i> , <i>checklist</i> , guías y estándares pertenecientes al paquete de revisión?		
• ¿Examinó el producto de trabajo según las indicaciones impartidas por el moderador?		
• ¿Clasificó y documentó los defectos identificados en el Informe de la Revisión?		
• ¿Hizo una lista con los errores de forma (léxico, sintaxis, semántica, etc.)?		
• ¿Entregó el informe de inspección al moderador?		
Inspección		
• ¿Nombró los defectos pertinentes durante la presentación del producto?		
• ¿Trabajó con los demás inspectores para alcanzar consenso sobre los defectos y su clasificación?		
• ¿Entregó la lista de errores de forma al autor?		
• ¿Expuso las lecciones aprendidas y propuso mejoras al moderador?		

Checklist del Presentador		
1. Identificación		
Proyecto:	Fecha de recepción: _/_/_	
Producto:	Fecha de entrega: _/_/_	
Nombre:		
E-mail:	Fono:	
2. Checklist		
	Sí	No
• ¿Registró el esfuerzo invertido en cada una de las etapas?		
Planificación		
• ¿Coordinó con el moderador las prioridades de la inspección de forma tal que se logren los objetivos dentro del lapso de tiempo de las 2 horas establecido para esta reunión?		
Orientación		
• Si se estimó necesario, asistió a la reunión de orientación, confirmó el rol y/o los múltiples roles asignados a la revisión.		
Preparación		
• ¿Revisó los <i>templates</i> , <i>checklist</i> , guías y estándares pertenecientes al paquete de revisión?		
• Si además cumplió el rol de inspector, se preparó para la inspección con la <i>checklist</i> de inspector		
• ¿Preparó la presentación del producto de trabajo?		
Inspección		
• ¿Siguió las pautas entregadas por el moderador al inicio de la reunión?		
• ¿Presentó el producto de trabajo a los participantes centrándose en los aspectos principales?		
• ¿Aseguró la participación equitativa de los miembros de la inspección?		
• ¿Participó como un inspector en igualdad de condiciones que los otros participantes?		

Checklist del Secretario		
1. Identificación		
Proyecto:	Fecha de recepción: _/_/_	
Producto:	Fecha de entrega: _/_/_	
Nombre:		
E-mail:	Fono:	
2. Checklist		
	Sí	No
¿Registró el esfuerzo invertido en cada una de las etapas?		
Planificación		
¿Coordinó con el moderador la mejor forma de registrar los defectos, acciones y lecciones aprendidas?		
Orientación		
Si se estimó necesario, asistió a la reunión de orientación, confirmó el rol y/o los múltiples roles asignados a la revisión.		
Preparación		
¿Revisó los <i>templates</i>, <i>checklist</i>, guías y estándares pertenecientes al paquete de revisión?		
Si además se cumplió el rol de inspector, se preparó para la inspección con la checklist de inspector		
¿Se preparó para inspección revisando los planes y reuniendo los recursos necesarios para un registro más oportuno de la inspección?		
Inspección		
¿Cuando se llegó a consenso con relación a un defecto, registró la información oportuna?		
¿Advirtió a los otros participantes si su capacidad de registrar los acontecimientos se vio sobrepasada?		
Al final de la inspección, leyó la información registrada para su verificación.		
¿Participó como un inspector en igualdad de condiciones que los otros participantes?		
¿Registró las acciones, asignaciones y fechas acordadas durante el termino de la reunión?		
¿Registró las mejoras propuestas y las lecciones aprendidas expuestas según las indicaciones del moderador?		

Checklist del Observador		
1. Identificación		
Proyecto:	Fecha de recepción: _/_/_	
Producto:	Fecha de entrega: _/_/_	
Nombre:		
E-mail:	Fono:	
2. Checklist		
	Sí	No
¿Registró el esfuerzo invertido en cada una de las etapas?		
Planificación		
¿Coordinó con el moderador la mejor forma de capturar las lecciones aprendidas y de registrar las observaciones sin interrumpir el transcurso habitual de la inspección?		
Preparación		
¿Revisó la documentación y las bases de la revisión de acuerdo al tipo de producto de trabajo con el apoyo del moderador?		
Inspección		
¿Registró observaciones sobre las áreas que requieren de mejoramiento en relación a los roles individuales y sus interacciones?		
¿ Registró las mayores diferencias entre las guías entregadas y la práctica real?		
¿ Registró la adherencia al proceso definido?		
¿Preparó un resumen de las observaciones para entregarlo al moderador al finalizar la inspección?		

5.5.4 Checklist de Auditoria

Checklist: SQA		
1. Identificación de la auditoría		
<u>Institución auditada:</u>		
<u>Proyecto:</u>	<u>Fase del ciclo de vida</u>	
<u>Iniciador:</u>	☐ Planificación	☐ Integración y pruebas
<u>Tipo de auditoría:</u> ☐ Interna ☐ Externa	☐ Esp. de Requerimientos	☐ Aceptación y entrega
	☐ Diseño	☐ Mantención
☐ Implementación		
2. Auditor		
<u>Nombre</u>		
<u>e-mail</u>		<u>Fono</u>
3. Checklist		
	Sí	No
• ¿Se creó un plan de SQA como parte del plan del proyecto? ¿Se encuentra actualizado?		
• ¿El plan de SQA fue revisado y aprobado?		
• El plan de SQA incluye: (a) los requerimientos para SQA y las actividades que deben ser desarrolladas, (b) la calendarización de las actividades definidas, (c) los recursos requeridos, (d) la participación de SQA en el desarrollo de software, (e) la participación de SQA en el proceso de SCM, y (f) la participación de SQA en el proceso de pruebas.		
• ¿Existe evidencia sobre la implementación de las actividades de SQA?		

Checklist: Proceso de Documentación		
1. Identificación de la auditoría		
<u>Institución auditada:</u>		
<u>Proyecto:</u>	<u>Fase del ciclo de vida</u>	
<u>Iniciador:</u>	☐ Planificación	☐ Integración y pruebas
<u>Tipo de auditoría:</u> ☐ Interna ☐ Externa	☐ Esp. de Requerimientos	☐ Aceptación y entrega
	☐ Diseño	☐ Mantención
☐ Implementación		
2. Auditor		
<u>Nombre</u>		
<u>e-mail</u>		<u>Fono</u>
3. Checklist		
	Sí	No
• ¿Existen estándares definidos para preparar la documentación de los productos de trabajo?		
• ¿La documentación existente se ajusta a dichos estándares?		
• ¿Existen procedimientos documentados para asegurar la adherencia a estos estándares?		
• ¿Estos procedimientos distinguen los cambios a los documentos bajo control de configuración del software? ¿Este tipo de cambios es revisado?		
• ¿El contenido de la documentación de los productos de trabajo es clara, concisa, completa y comprensible?		
• ¿Los miembros de las revisiones de esta documentación se encuentran lo suficientemente familiarizados con ella como para detectar inconsistencias fácilmente?		
• ¿Existe una autoridad competente para la aprobación de la documentación de los entregables (productos de trabajo)? ¿Es visible para los desarrolladores?		
• ¿Se entrega oportunamente la documentación solicitada por el cliente?		
• ¿Existen suficientes copias de los documentos?		
• ¿La documentación es desarrollada paralelamente a las otras actividades del desarrollo de software? ¿Refleja el estado real del proyecto y de los productos de trabajo?		

Checklist: SCM		
1. Identificación de la auditoría		
Institución auditada:		
Proyecto:	Fase del ciclo de vida	
Iniciador:	☐ Planificación ☐ Esp. de Requerimientos ☐ Diseño ☐ Implementación	☐ Integración y pruebas ☐ Aceptación y entrega ☐ Mantención
Tipo de auditoría:	☐ Interna ☐ Externa	
2. Auditor		
Nombre		
e-mail		Fono
3. Checklist		
	Sí	No
• ¿Se preparó un plan de SCM? ¿Se encuentra actualizado?		
• ¿El plan de SCM fue revisado y aprobado?		
• ¿Se definen en el plan los mecanismos de selección e identificación de ítems de configuración? ¿Se define el esquema de versiones y revisiones?		
• ¿Los procedimientos de SCM son implementados adecuadamente? ¿Existen procedimientos para el acceso a la librería del software?		
• ¿Existe un grupo de SCM con responsabilidades bien definidas? ¿Cuenta con los recursos adecuados?		
• ¿Las líneas bases se ajustan a los requerimientos?		
• ¿Existen procedimientos para gestionar el control de cambios adecuadamente?		
• ¿Existe un CCB? ¿Quiénes pertenecen al él? ¿SQA forma parte del comité? ¿Existen procedimientos claros para sus actividades? ¿Sus actividades son monitoreadas?		
• ¿Se mantiene información sobre el estado de la configuración del software? ¿Actualizada?		
• ¿El plan de SCM contempla las auditorías FCA y PCA? ¿Se llevan a cabo?		

Checklist: Librería del software		
1. Identificación de la auditoría		
Institución auditada:		
Proyecto:	Fase del ciclo de vida	
Iniciador:	☐ Planificación ☐ Esp. de Requerimientos ☐ Diseño ☐ Implementación	☐ Integración y pruebas ☐ Aceptación y entrega ☐ Mantención
Tipo de auditoría:	☐ Interna ☐ Externa	
2. Auditor		
Nombre		
e-mail		Fono
3. Checklist		
	Sí	No
• ¿Se ha establecido una librería del software? ¿Se ha asignado un responsable?		
• ¿Existen procedimientos adecuados para el acceso y la gestión de la librería del software?		
• ¿Se documentan apropiadamente las versiones de los productos de trabajo?		
• ¿Existe un índice de los tópicos de la librería del software? ¿Actualizado?		
• ¿Existe un registro del ingreso/salida (check in/chek out) de los entregables de la librería del software?		
• ¿Se asigna a cada ítem un identificador que refleje la versión y el tipo de producto de trabajo?		
• ¿Se controla la gestión de la librería del software? ¿Cómo?		

Checklist: Identificación y seguimiento de problemas		
1. Identificación de la auditoría		
Institución auditada:		
Proyecto:	Fase del ciclo de vida	
Iniciador:	☐ Planificación ☐ Esp. de Requerimientos ☐ Diseño ☐ Implementación	☐ Integración y pruebas ☐ Aceptación y entrega ☐ Mantención
Tipo de auditoría:	☐ Interna ☐ Externa	
2. Auditor		
Nombre		
e-mail		Fono
3. Checklist		
	Sí	No
• ¿Existen procedimientos que aseguren la detección y corrección de los problemas y/o discrepancias detectadas?		
• ¿Se examinan los informes de problemas y de discrepancias para determinar las posibles causas?		
• ¿Se analiza la relación entre las diferentes actividades de desarrollo para prevenir disconformidades en los productos?		
• ¿Se definen y planifican acciones correctivas? ¿Se asignan los recursos adecuados?		
• ¿Las acciones correctivas son registradas y documentadas minuciosamente?		
• ¿Se revisan y monitorean las acciones correctivas para determinar su efectividad, completitud y complacencia respecto de los estándares?		
• ¿El nivel de gestión apoya las acciones correctivas?		
• ¿Los desarrolladores están de acuerdo en generar informes de problemas y de discrepancias? ¿Los utilizan?		

Checklist: Estado del proyecto		
4. Identificación de la auditoría		
Institución auditada:		
Proyecto:	Fase del ciclo de vida	
Iniciador:	☐ Planificación ☐ Esp. de Requerimientos ☐ Diseño ☐ Implementación	☐ Integración y pruebas ☐ Aceptación y entrega ☐ Mantención
Tipo de auditoría:	☐ Interna ☐ Externa	
5. Auditor		
Nombre		
e-mail		Fono
6. Checklist		
	Sí	No
• ¿El estado real del proyecto concuerda con la planificación? ¿Si no es así, que tan grande es la brecha?		
• De acuerdo con el plan de proyecto: ¿cuál es el estado de las actividades, recursos, productos de trabajo, hitos?		
• Determinar: (a) fase de desarrollo actual, (b) estado de avance de las actividades, (c) conformación y organización del equipo desarrollador, (d) productos de trabajo, (e) hitos, y (f) resultados de las revisiones.		