

0/H 39  
632p  
2da Etapa  
V

46470

Subsecretaría de Turismo de Santa Cruz  
Provincia de Santa Cruz

### Plan de Mantenimiento

Proyecto de Virtualización de Servicios Turísticos, Parte II

Informe Final

Ángel A. Guzmán

Buenos Aires

01/07/2007



**Subsecretaría de Turismo de Santa Cruz  
Provincia de Santa Cruz**

**Plan de Mantenimiento**

**Proyecto de Virtualización de Servicios Turísticos, Parte II**

**Informe Final**

**Ángel A. Guzmán**

**Buenos Aires  
01/07/2007**

## **Resumen Ejecutivo**

En esta fase del proyecto de Virtualización de Servicios Turísticos Parte II se ha desarrollado un plan de mantenimiento para el Sistema de Información desarrollado en etapas anteriores,

El desarrollo del presente proyecto en la subsecretaría de Turismo de Santa Cruz ha significado un avance en la sistematización de los procesos determinados por la organización. Durante el proceso de desarrollo e implantación fue posible realizar un trabajo integrado con el área de gestión de la calidad, aspecto determinante para lograr la sistematización de los principales procesos. El SIT en si se convirtió en un soporte para la norma ISO 9001 que se tiene como referencia para el sistema de gestión.

Por otra parte el uso que le dan hoy los usuarios a cada uno de los módulos evidencia que en la actualidad el SIT es una herramienta protagónica en el quehacer institucional. Es por lo tanto clave comprender que este desarrollo requiere de un mantenimiento que posibilite que no sólo el mismo pueda subsistir en el tiempo, sino evolucionar acompañando al crecimiento de la organización.

A lo largo de este informe se darán las directrices para poder llevar adelante el plan de mantenimiento. No sólo desde lo que dicen hoy los teóricos respecto a esta fase clave de cualquier sistema, sino sobre todo desde las necesidades y particularidades que se han detectado en la organización y la alineación que tiene esta con la gestión de la calidad incluyendo los procedimientos principales.

## **Resumen**

Teniendo en cuenta que el Sistema Información turística (SIT) está compuesto por diferentes módulos que interactúan entre sí y permiten sistematizar los procesos y procedimientos, esta segunda etapa del trabajo tuvo como objetivo completar la implementación de los módulos restantes, lo cual optimizó la calidad de los servicios prestados, beneficiando tanto a la organización que los lleva adelante, como así también a los usuarios de los mismos.

El desarrollo de las tareas implicadas en el presente proyecto contribuyeron: a la optimización del sitio web y su integración al SIT; a consolidar un proceso epistémico atendiendo las instancias de conocimiento compartido, conceptual, sistémico y operacional; consolidar y fomentar el uso de metodologías para el mejoramiento de la capacidad técnica; integración de las distintas áreas a través de la virtualización de los procedimientos sistematizados; propender al desarrollo de plataformas tecnológicas normalizadas; contribuir con las comunicaciones entre las áreas.

En el presente documento se presenta un compendio de las actividades realizadas durante este periodo de desarrollo, como así también un detallado plan de mantenimiento que posibilitará a la subsecretaría de turismo visualizar la necesidad de abordar con minuciosidad la organización necesaria entorno al sistema, imprescindible para el normal funcionamiento del mismo.

## **Compendio de las actividades realizadas en el presente proyecto**

*Módulo para Marketing:* El eje central de este módulo fue la sistematización de la información sobre las acciones de promoción, permitiendo procesar, controlar y registrar entre otros: participación en ferias, congresos, exposiciones, bolsas, encuentros de comercialización y otras muestras dirigidas a los distintos mercados, control de las pautas publicitarias y de promoción contratadas en cada ejercicio (períodos, costos, medios, etc.), control de stock de material promocional en las distintas dependencias de la Subsecretaría, integración con los demás módulos con pertinencia.

*Tablero de Monitoreo, Control y Gestión “Revisión por la Dirección”:* Debido al impacto producido por el SIT fue oportuno desarrollar un módulo de organización y gestión que contribuya con la alta dirección a llevar un monitoreo permanente de cada uno de los módulos implementados. Tiene entre otras las siguientes características: monitoreo de todos los módulos del SIT, staff con el personal de toda la organización, seguimiento de procesos, revisiones que permita a la alta dirección realizar revisiones directamente y que estas se manifiesten en los diferentes módulos para ser leídas por los actores organizacionales que participan de módulo revisado.

*Módulo para Inversores:* Se desarrolló un módulo para brindar información sistematizada y estratégica dirigida a potenciales inversores turísticos para la provincia. Este módulo permitió sistematizar los requerimientos aportados por el personal de la Subsecretaría y genera una base sistemática para agregar nuevos requerimientos que pudieran surgir en el futuro, tal como por ejemplo de estudios específicos. Entre las características principales del módulo se encuentran: estadísticas de oferta y demanda turística, visualización de indicadores prioritarios para estos destinatarios, como por ejemplo número de inversiones por comarca, localidad, etc., banco de proyectos de inversión, posibilidad de publicar en el módulo distintos estudios, tales como por ejemplo de oportunidad comercial, prefactibilidad, etc.

*Georeferenciamiento de la información:* El primer objetivo de este módulo es dar una plataforma eficaz y fácil para la puesta en práctica de métodos para referenciar geográficamente la información turística y el segundo, es hacer estos métodos accesibles de una manera fácil, lo cual se logra principalmente mediante su interfaz

gráfica de usuario. Entre sus características principales se pueden nombrar: asignación de mapas, asignación de notas en mapas, oferta turística (recursos naturales, recursos culturales, facilidades turísticas, etc.) y demanda turística.

*Implementación de un CMS (Content Management System) para el Sitio web:*

desarrollo de un sistema gestor de contenidos, que aumentó la facilidad e interactividad de administrar la información del sitio web de la Subsecretaría. Entre sus características principales se encuentran: creación de páginas ilimitadas configuración avanzada y múltiple (permite configurar toda opción del sitio); gestión de secciones y categorías, que permitirán organizar el sitio web; publicación de contenidos, los que podrán editarse para darles el formato deseado y agregarles imágenes; gestión de contenido, incluyendo validación y aprobación antes de publicar; gestión de recursos multimedia; estadísticas incorporadas; encuestas y votaciones dinámicas; gestión de usuarios (altas de usuarios y contraseñas, envío de mensajes masivos); navegación configurable; automatización en la publicación (publicación y baja de contenido programable); gestión de banners integrado; generación de noticias para ofrecerlas vía XML automáticamente en otros sitios web.; motor de búsqueda integrado; soporte para distintos lenguajes.

## Índice General

<b>1</b>	<b>Introducción</b>	<b>7</b>
<b>2</b>	<b>Plan de Mantenimiento</b>	<b>8</b>
2.1	Tipos de mantenimiento	9
2.2	Codificación	10
2.3	Software	14
2.4	Actividades	16
2.5	Perfiles requeridos	31
2.6	Hardware y Software de Base	32
2.7	Copias de Seguridad	32
2.8	Migración	33
2.9	Contingencias	33
<b>3</b>	<b>Glosario de Términos</b>	<b>35</b>
<b>4</b>	<b>Bibliografía</b>	<b>36</b>

## **1. Introducción**

En este hito del proyecto de virtualización de servicios parte II se han desarrollado un plan de mantenimiento para el sistema de información desarrollado en etapas anteriores (SIT).



## **2. Plan de Mantenimiento**

Esta tarea consistió en desarrollar el plan de mantenimiento del Sistema de Información desarrollado en las etapas anteriores.

No se incluyen en el plan de mantenimiento los servicios que no están directamente relacionados con el desarrollo de software tales como procedimientos para cambio de claves, alta de nuevos registros, etc.

El mantenimiento de software es “la modificación de un producto software después de su entrega al cliente o usuario para corregir defectos, mejorar el rendimiento u otras propiedades deseables, o para adaptarlo a un cambio del entorno” (IEEE, 1990). El proceso de mantenimiento se considera un proceso principal en el estándar internacional ISO/IEC 12207 y según este estándar las tareas relacionadas con el mantenimiento del software abarcan desde que el producto se encuentra instalado y en funcionamiento hasta que éste llega al fin de su vida útil y es retirado.

El mantenimiento enfoca simultáneamente cuatro aspectos principales de la evolución del sistema:

- A) Mantener el control sobre las funciones diarias del sistema
- B) Mantener el control sobre las modificaciones del sistema
- C) Perfeccionar las funciones aceptables existentes
- D) Impedir que el desempeño del sistema se degrade a niveles inaceptables.

El equipo asignado estará involucrado en varias actividades:

- Comprender el sistema
- Localizar la información en la documentación del sistema
- Mantener la documentación del sistema actualizada.
- Ampliar las funciones existentes para incorporar los requerimientos nuevos o cambiantes
- Agregar nuevas funciones al sistema

- Encontrar el origen de las fallas del sistema o de los problemas
- Localizar y corregir defectos
- Responder las preguntas acerca de la forma en que trabaja el sistema
- Reestructurar el diseño y los componentes del sistema
- Volver a escribir el diseño y los componentes del código
- Anular el diseño y los componentes del código que ya no resulten útiles.
- Administrar los cambios al sistema a medida que se realizan.

## **2.1 Tipos de mantenimiento**

Existen distintos tipos de mantenimientos del software.

### **2.1.1 Mantenimiento Correctivo**

El tratamiento de los fallos descubiertos se conoce como mantenimiento correctivo.

El Mantenimiento Correctivo tiene por objetivo localizar y eliminar los posibles defectos del software. A pesar de las pruebas y verificaciones que aparecen en etapas anteriores del ciclo de vida del software, los programas pueden tener defectos. Un defecto en un sistema es una característica del sistema que podría ser la causa de un fallo. El fallo se produce cuando el comportamiento del sistema es diferente al esperado por su especificación.

Estos fallos pueden ser de procesamiento, de rendimiento, de programación o de documentación.

En este tipo de mantenimiento es necesario especificar, entre otras cosas, el tiempo de respuesta en que el equipo de desarrollo debe acudir a verificar o determinar el problema ante un fallo eventual según el tipo de equipamiento, como, así mismo, los plazos para su resolución. Qué tipo de servicios abarcará, delimitar claramente qué será provisto por la empresa y que no.

### **2.1.2 Mantenimiento Preventivo**

Este tipo de mantenimiento consiste en la modificación del producto software sin modificar las especificaciones originales para mejorar las propiedades del sistema. Es similar al mantenimiento perfectivo. Involucra la modificación de algún aspecto del sistema a fin de prevenir fallas.

Puede incluir la incorporación de validación de tipos y datos, el perfeccionamiento del manejo de defectos, etc.

### **2.1.3 Mantenimiento Perfectivo**

El mantenimiento perfectivo implica hacer cambios para mejorar algún aspecto del sistema incluso cuando los cambios no son sugeridos por los defectos.

Los cambios de la documentación para clarificar métodos o funciones y las modificaciones del código y del diseño para reforzar la legibilidad son todos ejemplos de mantenimiento perfectivo. En el caso de cambios por modificación de los requisitos de usuario o de las especificaciones funcionales, se puede nombrar el aumento del número de usuarios simultáneos lo que podría repercutir en el rendimiento del sistema o el aumento de las funcionalidades.

### **2.1.4 Mantenimiento Adaptativo**

El mantenimiento adaptativo es la modificación del software después de su puesta en producción, para mantener operativo un programa mientras se realiza un cambio en el entorno de producción.

## **2.2 Codificación**

A continuación se describen algunas recomendaciones para la programación. La finalidad de estas recomendaciones son mantener la consistencia del código para que sea fácilmente legible y entendible. En algunos casos, según el lenguaje o las librerías/frameworks/toolkits utilizados se deben tener en cuenta sus especificaciones para evitar incompatibilidades.

Para mayor información:

Code Conventions for the Java Programming Language:

<http://java.sun.com/docs/codeconv/>

PEAR - Coding Standards: <http://pear.php.net/manual/en/standards.php>

2.2.1 Generalidades

- No se deben utilizar caracteres especiales en el código y nombres de archivos tales como la Ñ o palabras acentuadas. Reemplazar la “Ñ” con la “N” o “NI” y en las palabras acentuadas eliminar el acento.
- Para mantener coherencia con denominaciones comunes en el ámbito de la programación y evitar problemas con la traducción de ciertos términos, se recomienda utilizar en los nombres de métodos y clases palabras de uso común en inglés (ver puntos a continuación). Las demás palabras deberían estar en español. Por ejemplo: “obtenerNombre()” se debería escribir “getNombre()”. Esto puede variar según la tecnología a utilizar que en algunos casos obliga a utilizar determinados prefijos.
- Se deberían utilizar los términos en inglés como parte de nombres de métodos:

Término en inglés	Significado	Ejemplo
Get	Obtener	<b>getNombre()</b>
Set	Asignar	<b>setNombre()</b>
Find, search	Buscar	<b>findMesas()</b>
add	Agregar, alta	<b>addUsuario()</b>
delete	Eliminar, baja	<b>deleteUsuario()</b>
update	Actualizar, modificar	<b>updateUsuario()</b>
All	Todos	<b>getAll()</b>
By	Por, o también se puede utilizar “X”	<b>getUsuariosByNombre()</b> <b>getUsuariosXNombre()</b>

- Se deberían utilizar los términos en inglés como parte de nombres de clases e interfaces (estos en general deben situarse al final de la denominación - subfijos):

Término en inglés	Significado	Ejemplo
Manager	Gestor	Usuario <b>Manager</b>
Instance	Instancia	DB <b>Instance</b>
Factory	Factoria	Usuario <b>Factory</b>
Observer	Observador	Usuario <b>Observer</b>
Form	Formulario	Usuario <b>Form</b>
Input	Ingreso, entrada	Fecha <b>Input</b>
UI	Interfaz de usuario	Panel <b>UI</b>
DAO	Objeto de acceso a datos	Usuario <b>DAO</b>

### 2.2.2 Prefijos y Subfijos de archivos

	Significado	Ejemplo
<b>class.php</b>	Clase	usuario. <b>class.php</b>
<b>inc.php</b>	Archivo para ser incluido. Generalmente son librerías, funciones generales y rutinas. Es privado.	biblioteca. <b>inc.php</b>
<b>.tpl</b>	Plantilla. Es un archivo privado.	formulario. <b>.tpl</b>

### 2.2.3 Nombres de archivos comunes

index.php / index.jsp	Archivo de ingreso. Es público. Todos los accesos públicos deben ser gestionados por este archivo.
-----------------------	--

home.php / home.jsp	Archivo de ingreso para cada módulo o directorio. Es privado. Actúa como controlador. Todos los accesos al módulo/directorio deberían ser gestionados por este archivo
config.php	Archivo de configuración. Es privado.

#### 2.2.4 Declaración de clases e interfaces

- El nombre de la clase debe ser simple y descriptivo. Generalmente son sustantivos.
- Se recomienda codificar una clase/interface por archivo.
- Se debe seguir la convención UpperCamelCase  
<http://es.wikipedia.org/wiki/CamelCase>): se escriben las palabras eliminando espacios y la primera letra de cada palabra se escribe con mayúsculas (la primera letra de todas debe estar escrita con mayúsculas).  
Ejemplo: "pedido mejora" se debe escribir como "PedidoMejora"
- Se deben eliminar artículos y preposiciones si no agregan significado.
- Las clases e interfaces se deben escribir en singular, si corresponde.

#### 2.2.5 Declaración de métodos y funciones

- El nombre debe ser lo más descriptivo posible. Generalmente son verbos.
- Se debe seguir la convención lowerCamelCase  
<http://es.wikipedia.org/wiki/CamelCase>): se escriben las palabras eliminando espacios y la primera letra de cada palabra se escribe con mayúsculas (la primera letra de todas debe estar escrita en minúsculas).  
Ejemplo: "calcular precio" se debe escribir como "calcularPrecio"

#### 2.2.6 Declaración de variables y constantes

- Los nombres deben ser descriptivos.

- [Sólo se aplica en PHP] Las constantes y variables globales se deben escribir con mayúsculas y cada palabra separada con guiones bajos “\_”. En el caso de conflictos de nombres con constantes ya definidas, se debe anteponer el nombre del paquete o discriminador. Por ejemplo si se define una constante “formato de editor” en el paquete “UI” se debe escribir como “UI\_FORMATO\_EDITOR”.
- En general no se recomienda el uso de variables globales y constantes. En su lugar y si es posible utilizar clases con métodos estáticos o implementar algún patrón de diseño.
- Los nombres de las variables no deben ser muy extensos. Variables de un solo carácter no se deben permitir a excepción de variables temporales (bucles). No incluir preposiciones ni artículos.

#### 2.2.7 Comentarios

- En general se deberían utilizar los comentarios multilínea del tipo `/** ... */` para posteriormente generar la documentación de forma automatizada.
- Los archivos deben tener al comienzo un comentario de tipo `/* .... */` que incluya el nombre del archivo, la versión, la fecha y hora del último cambio y el autor del mismo. Esto se puede conseguir de forma automática utilizando la etiqueta `$Id` de SVN o CVS. Este tipo de comentarios no deberían ser incluidos en la documentación.
- Cada clase, método, interfaz, función y variable debería tener comentarios del tipo `/** ... */` inmediatamente antes de la declaración. Los comentarios deberían incluir mínimamente una descripción y el autor.
- En el caso de métodos y funciones se deberían incluir en los comentarios los parámetros incluyendo nombre, descripción, tipo y valor por defecto. También se debería comentar el valor retornado si corresponde.

### 2.3 Software

### 2.3.1 Herramientas de desarrollo

Funcionalidad	Software	Versión	Sitio Web
Entorno de Desarrollo Integrado (IDE) con plugin para PHP	Eclipse	3.2	<a href="http://www.eclipse.org/">http://www.eclipse.org/</a>
Gestión del código fuente	Subversión	1.1	<a href="http://subversion.tigris.org/">http://subversion.tigris.org/</a>
Compilador Flash DHTML	OpenLaszlo	4.x	<a href="http://www.openlaszlo.org/">http://www.openlaszlo.org/</a>
Generación de Diagramas UML	ArgoUML	0.24	<a href="http://argouml.tigris.org/">http://argouml.tigris.org/</a>
Administración de Base de Datos	PhpMyAdmin	2.9	<a href="http://www.phpmyadmin.net/">http://www.phpmyadmin.net/</a>
Generación de documentación (PHP)	PHPDocumentor	1.3	<a href="http://www.phpdoc.org/">http://www.phpdoc.org/</a>

### 2.3.2 Lenguajes

Nombre	Versión
PHP	5.1
JAVA	1.5
Javascript	1.x

### 2.3.3 Gestor de Base de Datos

Nombre	Versión
MySQL	4.1.x

### 2.3.4 Servidores

Nombre	Versión
Apache	1.x
Apache	2.x
Tomcat	5.x



<b>OpenLaszlo Server (Tomcat)</b>	<b>4.x</b>
-----------------------------------	------------

### 2.3.5 Frameworks, Librerías, Toolkits

<b>Nombre</b>	<b>Funcionalidad</b>	<b>Versión</b>	<b>Sitio Web</b>
<b>Prototype</b>	Javascript	1.5.1	<a href="http://www.prototypejs.org/">http://www.prototypejs.org/</a>
<b>Scriptaculous</b>	Javascript	1.7.1	<a href="http://script.aculo.us/">http://script.aculo.us/</a>
<b>OpenRico</b>	javascript	2.0	<a href="http://www.openrico.org/">http://www.openrico.org/</a>

### 2.3.6 Clientes - navegadores

<b>Nombre</b>	<b>Versión</b>	<b>Sitio Web</b>
<b>Mozilla Firefox</b>	1.5 – 2.0	<a href="http://www.mozilla.org/">http://www.mozilla.org/</a>
<b>Internet Explorer</b>	6.0	<a href="http://www.microsoft.com/">http://www.microsoft.com/</a>
<b>Opera</b>	9.x	<a href="http://www.opera.com/">http://www.opera.com/</a>
<b>Nokia Mobile Browser Simulator (Wap)</b>	Según celular	<a href="http://www.forum.nokia.com/">http://www.forum.nokia.com/</a>
<b>OpenWave Simulator (Wap)</b>	7	<a href="http://www.openwave.com/">http://www.openwave.com/</a>

### 2.3.7 Directrices Técnicas y de Gestión

Se utilizan las normas, estándares, guías, recomendaciones y especificaciones que se detallan a continuación:

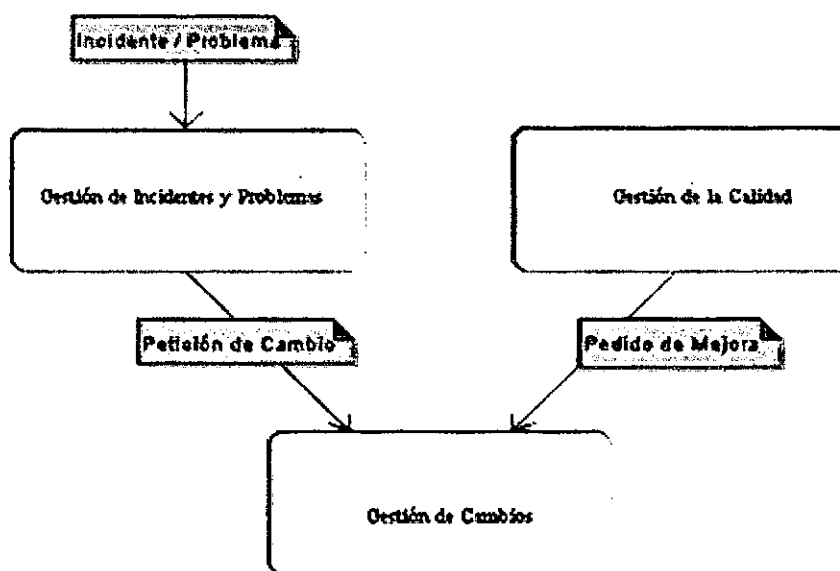
	<b>Versión</b>
<b>HTML</b>	4.x
<b>CSS</b>	2.x
<b>UML</b>	1.x
<b>ECMAScript</b>	1.x
<b>XML</b>	1.0

## 2.4 Actividades

Las actividades se encuentran distribuidas en dos procesos principales:

- Gestión de Incidentes y Problemas
- Gestión de Cambios

Los errores y problemas comunicados por los usuarios del SIT son gestionados por Gestión de Incidentes y Problemas y constituye el primer paso de soporte. En caso de que se deba realizar algún cambio en el SIT se deberá generar un pedido o petición de cambio. También Gestión de la Calidad podrá iniciar el proceso de cambio mediante las propuestas de mejora. El proceso de Gestión de Cambio también podrá generar pedidos de cambio para optimizar y mejorar el sistema.



Los responsable de incidentes y problemas deberán registrar la siguiente información:

- Número de referencia único
- Clasificación del incidente
- Fecha/hora
- Nombre de la persona que registró el incidente
- Medio utilizado para comunicar el incidente (TE,email,etc).
- Descripción de los síntomas
- Categoría
- Impacto / Urgencia / Prioridad

- Estado del incidente (nuevo, aceptado, planificado, asignado, en progreso, resuelto, cerrado).
- Persona o grupo de soporte asignado al tratamiento del incidente
- Error/Problema relacionado
- Fecha y Hora de resolución
- Fecha y hora de cierre
- Pedido de cambio asociado si corresponde

Para tener el control durante el ciclo de vida del incidente, para cada acción se debe guardar:

- Nombre de la persona o grupo que guarda la acción
- Tipo de acción (rutina, diagnóstico, recuperación, resolución, cierre).
- Fecha y Hora de la acción
- Descripción y resultados de la acción.

En el caso de un pedido de cambio se deberá registrar como mínimo:

- Fecha y Hora de inicio
- Incidente/problema/pedido de mejora asociado si corresponde
- Tipo de mantenimiento (correctivo, adaptativo, perfectivo, preventivo).
- Usuario que origina el pedido
- Descripción
- Mensajes de error o detalle del entorno si corresponde.
- Módulos afectados
- Prioridad e Impacto
- Estado
- Persona responsable de la implementación

Cada acción realizada se deberá registrar.

## **2.4.7 Gestión de Incidentes y Problemas**

### **2.4.7.1 Objetivo**

El principal objetivo es restaurar el normal funcionamiento del SIT tan rápido como sea posible y minimizar cualquier riesgo subyacente.

El proceso tiene dos aspectos: reactivo y proactivo. El aspecto reactivo está relacionado con resolver problemas en respuesta a uno o más incidentes. El aspecto proactivo está relacionado con identificar y resolver problemas antes de que estos ocurran.

Este proceso se podría implementar como un módulo del SIT mediante un sistema de ticket de soporte. De esta forma los usuarios podrían consultar el estado de las solicitudes y Gestión de Cambio podría consultar la información de registros históricos.

#### **2.4.7.2 Alcance**

El control de incidentes, problemas y de errores están dentro del alcance de este proceso. Un problema es una causa subyacente desconocida de incidentes y un error conocido es un problema que ha sido satisfactoriamente diagnosticado. Un incidente es un evento que puede causar la reducción o interrupción en la calidad del servicio.

En el presente plan de mantenimiento no se incluyen los servicios tales como cambio de claves, altas de localidades y en general cualquier gestión de la información. Sólo se consideran aquellos incidentes que afecten al normal funcionamiento del SIT.

#### **2.4.7.3 Actividades**

Las principales tareas son:

##### **2.4.7.3.1 Detección y Registro**

El proceso tiene las siguientes entradas:

- El detalle de un incidente incluyendo usuario que inicia la petición, fecha y hora y una descripción breve.
- Un evento generado por el SIT

Las salidas serán:

- Detalle del incidente actualizado
- El reconocimiento de cualquier error
- Aviso al usuario final cuando el incidente haya sido resuelto

#### 2.4.7.3.2 Soporte Inicial y Clasificación

Las entradas son:

- Detalles del incidente
- Respuesta de búsqueda de incidentes de problemas y errores conocidos

Acciones a realizar:

- Clasificar el incidente
- Búsqueda de problemas y errores comunes
- Informar a la Gestión de Cambios de la existencia de nuevos problemas si corresponde o incidente
- Asignar impacto, urgencia y prioridad
- Proveer soporte inicial (determinar detalles del incidente, buscar solución rápida)
- Cerrar incidente o delegar el incidente a otro grupo de soporte o a Gestión del Cambio.
- Informar al usuario.

Salidas:

- Petición de cambio para la resolución del incidente
- Detalles del incidente actualizados
- Historial de las acciones realizadas actualizado.

La clasificación es el proceso de identificar la razón del incidente/problema y la acción de resolución correspondiente. Los datos para clasificación necesarios son:

- Descripción del incidente

- Fecha y Hora
- Módulos implicados
- Detalle de los datos ingresados si corresponde
- Impacto

El impacto es una medición de la criticidad del negocio de un incidente o problema. Se podría establecer como el número de usuarios o módulos afectados.

La Urgencia es la velocidad necesaria para resolver un incidente de cierto impacto. En este caso se deben contemplar el horario de trabajo y los días laborales. Por ejemplo un incidente de alto impacto por defecto puede que no sea resuelto inmediatamente si el usuario comunica el incidente un día viernes a última hora.

La Prioridad es el esfuerzo esperado. Un incidente con impacto bajo y urgencia promedio que puede ser resuelto con el menor esfuerzo será resuelto inmediatamente (por ejemplo cambio de clave, de email).

El soporte inicial implica la resolución del incidente para la satisfacción del usuario por parte de la persona responsable del soporte del SIT. Si no puede ser resuelto pasa a la siguiente etapa "Investigación y desarrollo".

#### 2.4.7.3.3 Investigación y Diagnóstico

Entradas:

- Detalles del incidente/problema actualizados.
- Historial de acciones realizadas y de seguimiento.

Acciones:

- Actualización de los detalles del incidente
- Recopilación y análisis de toda la información relacionada y resolución.
- Ejecución de acción, solución temporal o delegación a otro grupo si corresponde

Salidas:

- Detalle de incidente actualizado y una especificación de la solución si corresponde.

Siempre que sea posible el usuario debe ser provisto del procedimiento para continuar con el servicio. El efecto de la solución temporal es proveer de más tiempo para investigar y obtener una resolución. Las soluciones temporales deben ser comunicadas a los usuarios directamente afectados. En algunos casos la solución temporal puede implicar deshabilitar una parte o todo el módulo si está gravemente comprometido.

Una vez que el incidente/problema ha sido asignado al grupo de soporte, este debería:

- Aceptar la asignación del incidente especificando la fecha y hora asegurándose que el estado del incidente se actualice.
- Revisar el incidente en la base de conocimiento de errores conocidos, problemas, soluciones y cambios planeados.
- Reasignar el incidente para su cierre.

Este puede ser un proceso iterativo que incluya diferentes grupos.

#### 2.4.7.3.4 Resolución y Recuperación

Entradas:

- Detalles del incidente/problema actualizados.
- Cualquier respuesta sobre una petición de cambio para implementar una solución.
- Cualquier solución temporal o resolución.

Acciones:

- Resolver el incidente utilizando la solución temporal o generar una petición de cambio.
- Tomar las acciones de recuperación necesarias.

Salidas:

- Petición de cambio para resolución.
- Incidente/problema resuelto incluyendo detalles de recuperación
- Detalles del incidente/problema actualizado.

#### 2.4.7.3.5 Cierre del Incidente / Problema

##### Entradas:

- Detalles del incidente/problema actualizados
- Incidente/problema resuelto

##### Acciones:

- Confirmar la resolución del incidente/problema al Usuario
- Cerrar el incidente/problema

##### Salidas:

- Detalles del incidente/problema actualizados

Al cerrar el incidente/problema se debe verificar que:

- Los detalles del incidente y la solución son claros
- La clasificación del incidente está completa
- La resolución del incidente fue aprobada por el usuario. Es conveniente que se utilice como medio el email para archivar toda comunicación que pueda servir de respaldo.
- Todos los detalles del proceso han sido guardados: tiempo de resolución, fecha y hora de cierre, persona que cierra el incidente.

Luego de que el incidente ha sido cerrado no podrá ser abierto nuevamente. Se debe iniciar uno nuevo y relacionarlo con el incidente cerrado.

#### 2.4.7.3.6 Supervisión, seguimiento y comunicación

##### Entradas:



- Registros de incidentes

Acciones:

- Supervisar incidentes
- Informar a los usuarios

Salidas:

- Reportes de gestión sobre progreso de incidentes.

#### **2.4.7.4 Roles**

##### **2.4.7.4.1 Responsable de incidentes y problemas**

La persona con este rol tiene la responsabilidad de:

- Dirigir y supervisar la eficiencia y efectividad del proceso
- Producir la información de gestión
- Gestionar el trabajo del grupo de soporte

##### **2.4.7.4.2 Soporte de Incidente/Problema**

La persona con este rol tiene la responsabilidad de:

- Registrar el incidente/problema.
- Resolución rápida y recuperación de incidentes asignados.
- Clasificación del incidente/problema.
- Detección de posibles problemas.
- Supervisar el estado del incidente.
- Mantener informado a los usuarios afectados.
- Delegar resolución a otro grupo o persona.
- Generar una petición de cambio.

#### **2.4.7.5 Indicadores**

- Número total de incidentes/problemas por período.
- Número de incidentes/problemas resueltos según el tiempo estimado.

- Número de incidentes/problemas nuevos no clasificados anteriormente.
- Número de incidentes/problemas por módulo.

## **2.4.8 Gestión de Cambios**

### **2.4.8.1 Objetivo**

Los cambios se presentan como resultados de Problemas y otros como mejoras del sistema. El objetivo es asegurar que los métodos y procedimientos son utilizados para eficiencia y gestión de todos los cambios manteniendo orden para minimizar los impactos.

### **2.4.8.2 Alcance**

La gestión del cambio es responsable por la gestión del proceso de cambio incluyendo:

- Generar y registrar cambios.
- Desarrollar la justificación y obtener la aprobación.
- Gestionar y coordinar la implementación del cambio.
- Supervisar la implementación.
- Cierre y revisión de peticiones de cambios.

Las entradas al proceso son:

- Pedidos de cambios
- Pedidos de mejora
- Historial de incidentes/problemas
- Historial de cambios implementados
- Cronograma de cambios aprobados

Las actividades principales son:

- Filtrar cambios
- Gestionar cambios y proceso de cambios

- Revisión y cierre de peticiones de cambios
- Gestionar reportes

Las salidas del proceso son:

- Peticiones de cambios
- Minutas de reunión
- Reportes del proceso de cambio

Un incidente no es un cambio y un problema puede no conducir a un cambio. Sólo se tienen en cuenta las peticiones de cambios, no de servicios (por ejemplo el cambio de clave).

### **2.4.8.3 Actividades**

#### **2.4.8.3.1 Registro y filtro de cambios**

Todas las peticiones de cambio deben ser registradas y se le deben asignar un número identificador que debe ser único y en orden cronológico. Cuando una petición de cambio fue realizada como solución a un problema, incidente o pedido de mejora se debe preservar la relación entre problema-cambio-solución. Todas las acciones deben ser registradas. Solamente el responsable del proceso puede cerrar una petición de cambio justificando dicha decisión.

Cualquier petición de cambio impráctica o no aplicable debe ser filtrada. Estas deben ser retornadas al iniciador de la petición junto con la justificación del rechazo. Esta acción de rechazo debe ser registrada.

#### **2.4.8.3.2 Determinación de prioridades**

Cada petición de cambio debe tener asignada una prioridad que se establece según el impacto del problema y de la urgencia de la solución. Las prioridades son utilizadas para determinar que cambios deberían ser discutidos y gestionados en primer lugar por el equipo de gestión de cambios. El equipo de gestión de cambios debe asignar esta prioridad en colaboración con el iniciador de la petición de cambio.

Se deberían utilizar las prioridades que se detallan a continuación:

Prioridad	Descripción
Baja	El cambio es necesario y está justificado, pero puede esperar hasta la próxima actualización del módulo / sistema
Media	No existen impactos severos, pero no se puede esperar hasta la próxima actualización del sistema.
Alta	Existen impactos severos que afectan a varios usuarios
Urgente	Causa pérdida del servicio a varios usuarios. Se compromete el sistema. Se requiere acción inmediata.

#### 2.4.8.3.3 Categorización de cambios

Los responsables de la Gestión del Cambio deben examinar las peticiones pendientes y decidir como proceder basándose en la categoría de la petición de cambio.

El proceso de categorización debe examinar el impacto de aprobar el cambio en términos de recursos necesarios para implementarlo.

Las categorías podrían ser:

Categoría	
Impacto Menor	Cualquier responsable del proceso puede planificarlo.
Impacto Medio	Puede requerir reuniones y acuerdos para implementar los cambios.
Gran Impacto	Deben ser aprobados por el máximo responsable del proyecto.

#### 2.4.8.3.4 Reuniones

En casos de cambios de alto impacto, muy complejos o de alto riesgo es necesario planificar una reunión con los responsables del proyecto para analizar y aprobar el cambio. También las reuniones deben ser planificadas en forma periódica para tratar los siguientes temas:

- Cambios fallidos o cambios implementados
- Revisiones de cambios
- Cambios propuestos pendientes

Todos los comentarios de la reunión deben ser volcados en una minuta que será archivada para posteriores consultas.

#### 2.4.8.3.5 Aprobación de cambios

La aprobación formal de cada cambio debe ser realizada por la autoridad específicamente determinada para dicha acción a excepción de los cambios de bajo riesgo que pueden ser directamente aprobados por algún responsable del proceso de gestión del cambio. Los niveles de aprobación de cada cambio deben ser analizados según el riesgo, complejidad y tamaño del cambio.

#### 2.4.8.3.6 Desarrollo, implementación y pruebas del cambio

Las peticiones de cambios aprobadas deben ser enviadas al grupo responsable del desarrollo del SIT. Estas peticiones pueden incluir:

- El desarrollo de un nuevo módulo
- Crear una nueva versión de uno o más módulos
- Producir nueva documentación
- Actualizar librerías del sistema

Se deben respetar las recomendaciones de codificación, estándares y demás procesos definidos por la Subsecretaría de Turismo de Santa Cruz. Antes de realizar cualquier cambio se debe realizar una copia de seguridad de los componentes

afectados y si corresponde de los datos y otros recursos asociados del ambiente de producción.

El programador asignado debe realizar un plan de pruebas para verificar la correcta implementación de la solución.

#### 2.4.8.3.7 Tratamiento de cambios urgentes

El número de cambios urgentes debe mantenerse al mínimo posible porque insumen recursos que no estaban planificados y pueden generar fácilmente fallos al intentar darle una rápida solución. El cambio debe ser aprobado para poder ser implementado.

#### 2.4.8.3.8 Revisión de cambios

Todos los cambios implementados deben tener una revisión en un período determinado. Las revisiones tienen como objetivo:

- Los usuarios están satisfechos con los cambios
- Los cambios fueron implementados según el tiempo asignado.
- Los cambios implementados funcionan correctamente según la especificación establecida.
- Los cambios implementados no tuvieron impactos negativos en otros módulos del SIT o componentes relacionados.

#### 2.4.8.3.9 Cierre de la Petición de Cambio

Luego de que el cambio ha sido implementado y que el usuario que inició el pedido ha verificado y aprobado el cambio se procederá al cierre del pedido de cambio incluyendo toda la información de las acciones ejecutadas. Si el usuario no aprueba el cambio pasados 15 días de la notificación de que la petición ha sido implementada, el pedido será cerrado indicando esta situación. El usuario no tendrá derecho a ningún reclamo. Si el usuario lo considera necesario podrá iniciar una nueva petición de cambio pero se debería tener en cuenta el historial del usuario al momento de asignar prioridades y definir el cronograma de implementación de cambios.

#### **2.4.8.4 Roles**

##### **2.4.8.4.1 Responsable de cambios**

La persona con este rol tiene la responsabilidad de:

- Registrar las peticiones de cambios y asignarles prioridad en forma conjunta con el iniciador.
- Rechazar peticiones.
- Planificar reuniones con el equipo.
- Autorizar cambios.
- Gestionar el trabajo del personal asignado.
- Mantener actualizado el registro de la petición incluyendo todas las acciones realizadas y el estado.
- Revisar todos los cambios implementados para verificar que se cumplieron los objetivos.
- Cerrar peticiones de cambios.
- Generar reportes.

##### **2.4.8.4.2 Implementador de cambios**

La persona con este rol tiene la responsabilidad de:

- Implementar la solución propuesta para el cambio asignado.
- Resolver peticiones de cambios.
- Actualizar el registro de la petición incluyendo acciones realizadas y el estado.
- Preparar y ejecutar el plan de pruebas para los cambios implementados.
- Realizar copias de seguridad del entorno de producción antes de efectuar cualquier cambio.
- Migración de los componentes afectados del entorno de desarrollo al entorno de producción.
- Gestión de la configuración de los componentes afectados incluyendo gestión de versiones.

#### **2.4.8.5 Indicadores**

- Número total de peticiones de cambios resueltas por período.
- Número total de peticiones de cambios aprobadas por período.
- Número total de peticiones de cambios rechazadas por período.
- Número total de peticiones de cambios por estado.
- Número total de peticiones de cambios resueltas por módulo.

## **2.5 Perfiles requeridos**

Para la realización de las actividades de mantenimiento se requieren los siguientes perfiles:

- **Analista Funcional:** persona con conocimientos en sistemas, procedimientos y flujos. Debe tener al menos 2 años de experiencia.
- **Programador Web:** persona con conocimientos en los lenguajes y estándares establecidos en el presente documento. Debe tener al menos 2 años de experiencia en el desarrollo de aplicaciones Web. Debe tener conocimientos y experiencia en la gestión de configuración y versiones. Debe tener conocimientos de base de datos y consultas sql. Debería tener experiencia en las herramientas comentadas en el presente documento.
- **Diseñador Web:** persona con conocimientos en lenguajes y estándares, diseño gráfico y diseño de interfaces Web. Debe tener al menos 1 año de experiencia.
- **Desarrollador Web:** persona con conocimientos de todas las etapas del desarrollo de software: especificación de requisitos, planificación, diseño, programación, configuración, pruebas. Debe tener al menos 2 años de experiencia en el desarrollo de aplicaciones Web.
- **Soporte de Usuarios:** persona con capacidad para comunicarse y comprender las solicitudes de los usuarios. Debe tener al menos 6 meses de experiencia.

Una persona puede tener todos estos perfiles y podría cumplir con todas las actividades si la planificación y las peticiones de cambios lo permite. Queda a criterio de la Subsecretaria de Turismo de Santa Cruz establecer el número de personas que conformarán el equipo de mantenimiento del SIT.



## **2.6 Hardware y Software de Base**

Este punto no es aplicable porque el SIT se encuentra instalado en servidores ajenos a la Subsecretaría de Turismo. El equipo de mantenimiento no puede realizar modificaciones a la configuración del servidor ni al software instalado.

## **2.7 Copias de Seguridad**

Antes de realizar cualquier cambio en el sistema se debe realizar una copia de seguridad de los archivos a modificar o reemplazar.

Se debe realizar una copia de seguridad por día a cada base de datos que componen el sistema. Esta copia de seguridad se deberá utilizar en el caso de algún problema o restauración de datos. Antes de realizar cualquier cambio en el esquema o datos cargados en el entorno de producción se deberá realizar una copia de seguridad de la base de datos completa o de la/s tablas afectadas siguiendo la siguiente nomenclatura:

FECHAHORA\_NOMBRE\_BASE\_DATOS.sql

FECHAHORA\_NOMBRE\_TABLA.sql

El término FECHAHORA se debe ingresar como YYYYMMDDHHSS:

- YYYY: año en cuatro dígitos
- MM: mes de dos dígitos
- DD: día de dos dígitos
- HH: hora de dos dígitos (formato 24 horas)
- MM: minutos de dos dígitos.

Ejemplo:

200706291435\_usuario\_permiso.sql

También se deberán realizar copias de seguridad de aquellos archivos que se gestionan con el SIT: imágenes, fotos, documentos, etc. Como estos pueden ocupar demasiado espacio se recomienda realizar la copia de seguridad de los archivos

nuevos o modificados, es decir, no se hace una copia completa de todos los archivos. La regularidad dependerá de la existencia de nuevos archivos o modificaciones en el período considerado.

## **2.8 Migración**

Al migrar el sistema a otro servidor se deberá tener en cuenta entre otras cuestiones:

- Gestores de bases de datos disponibles y versiones.
- Tráfico permitido (límite).
- Lenguajes disponibles y versiones (intérpretes, motores).
- Nombres de archivos y extensiones.
- Servidores disponibles y configuración de los mismos.
- Soporte
- Adaptaciones que se deberán realizar al código para que el SIT funcione correctamente y con la calidad de servicio esperada.
- Cantidad máxima de procesos permitidos.
- Cantidad máxima de conexiones a bases de datos permitidas.
- Espacio disponible descontando el espacio utilizado por otros servicios (email, ftp, archivo temporales, etc).
- Si el servicio incluye Panel de Administración y que incluye.
- Si tiene acceso shell.
- Software de soporte instalado en el servidor.

## **2.9 Contingencias**

La Subsecretaría de Turismo deberá elaborar un plan de contingencias para reanudar con la mayor brevedad posible el funcionamiento del SIT ante un grave problema de seguridad o un incidente ajeno al sistema, y deberá definir el procedimiento que deberán seguir los usuarios.

Como primer paso ante una problema de seguridad (un ataque que ha violado el sistema o el servidor por ejemplo) se deberá deshabilitar inmediatamente el sistema. Esto se puede hacer desde la configuración del SIT o cambiando los archivos

index.php5 con otro alternativa (por ejemplo index.htm). Todas las peticiones ingresan por el archivo nombrado. Luego se procederá al cambio de claves del administrador y usuarios de acceso al servidor (shell, bases de datos, ftp) si el servidor lo permite y teniendo precaución de que la conexión con el servidor no esté comprometida. Caso contrario solicitar al soporte del servidor deshabilitar el servidor hasta nuevo aviso.

Luego de un incidente se recomienda cambiar las claves de todos los usuarios del SIT y cuentas asociadas. Se deben guardar todos los archivos de registros del servidor.

### 3. Glosario de Términos

**CORREO ELECTRÓNICO** (email): mensajes enviados a través de Internet.

**ENLACE**: palabra o frase que contiene la dirección de otro archivo electrónico.

**HTML**: lenguaje de marcado de hipertexto.

**GIF**: Graphics Interchange Format. Es un formato gráfico.

**JPEG**: Joint Photographic Experts Group. Es un algoritmo y formato gráfico.

**PNG**: Portable Network Graphics. Es un formato gráfico.

**SVG**: Scalable Vector Graphics. Es un lenguaje para describir gráficos vectoriales bidimensionales.

**XML**: Extensible Markup Language. Es un metalenguaje extensible de etiquetas.

**URL**: localizador uniforme de recursos.

**WORKFLOW**: flujo de trabajo.

## **4. Bibliografía**

- [1] PIATTINI, MARIO. Calidad en el desarrollo y mantenimiento del software. Rama
- [2] PIATTINI, MARIO. Mantenimiento del software. Rama
- [3] OFFICE OF GOVERNMENT COMMERCE. ITIL - The IT Infrastructure Library. Crown
- [4] SHARI LAWRENCE PFLEEGER (2002), Ingeniería de software: Teoría y práctica. Pearson Educación.