## UNIDAD NACIONAL DE PROTECCIÓN - UNP

## INFORME GRUPO DE GESTIÓN DE LAS TECNOLOGÍAS DE LA INFORMACIÓN

## -GGTI-

## ECOSISTEMA UNP – VERSION 1.0

## perfil_sin_foto

## ANALISIS Y TESTEO

## CIBERSEGURIDAD

## Test. V.1 – Pruebas Estáticas y Dinámicas

## Formulario solicitud análisis de riesgo - Ruta Individual

## Bogotá, 12 de noviembre del 2024

## CONTENIDO

|  |  |  |
| --- | --- | --- |
| [Nomenclatura] | [Tema] | [Página] |
| I | Identificación parcial de Riesgos relacionados con los sistemas de información “Sistema de gestión de seguridad de la información (SGSI) – Análisis Ciberseguridad | 6 |
| II | (Análisis de Pruebas de Ciberseguridad – Prevención de Ciberataques) | 7 |
| III | Pruebas de Penetración o Intrusión | 14 |

## **PROPÓSITO**

## Análisis de Ciberseguridad para **FORMULARIO SOLICITUD ANÁLISIS DE RIESGO - RUTA INDIVIDUAL** e identificación parcial de Riesgos relacionados con los sistemas de información “Sistema de gestión de seguridad de la información (SGSI)”. Ecosistema UNP

## Aplicación de pruebas estáticas y dinámicas (Análisis de Riesgos de Ciberseguridad – Prevención de Ciberataques).

## **ALCANCE**

## Personal Involucrado Directo e Indirecto

## **DEFINICIONES**

Estas son algunas pruebas que se harán en esta entrega que es un PRE-ANALISIS; posteriormente evaluaremos algunas otras según el despliegue:

### Pruebas de Penetración (Penetration Testing)

Realizaremos pruebas de penetración, también conocidas como pruebas de intrusión, como si fueran un simulacro de ataque. Actúo como un hacker ético, llevando a cabo ataques controlados para descubrir cualquier punto débil en el sistema o red antes de que personal no autorizado lo hagan. Es como un ensayo general en el que analizamos mejoras en seguridad.

### Pruebas de Seguridad de Aplicaciones Web (Web Application Security Testing)

Me enfoco en encontrar fallos en las aplicaciones web. En los formularios de inicio de sesión, el procesamiento de datos y cualquier otro punto donde las personas interactúan con la aplicación. Mi objetivo es asegurarme de que no haya brechas que los hackers puedan explotar.

### Pruebas de Seguridad Estática (Static Application Security Testing - SAST)

Analizo el código fuente de la aplicación sin necesidad de ejecutarlo. para asegurarme de que no hay brechas por donde puedan infliltrar los intrusos. Identifico vulnerabilidades potenciales directamente desde el código, antes de que la aplicación esté en funcionamiento.

### Pruebas de Seguridad Dinámica (Dynamic Application Security Testing - DAST)

Aquí es donde la ejecución se ponen en movimiento. Examino la aplicación mientras está en funcionamiento, buscando comportamientos sospechosos o vulnerabilidades que solo se pueden detectar cuando el software está activo. Es como hacer un chequeo general o de paquetes completo mientras está corriendo el sistema.

### Pruebas de Seguridad de Código Abierto (Open Source Security Testing)

Muchas aplicaciones modernas utilizan componentes de código abierto. Verifico que esos componentes no tengan vulnerabilidades que puedan comprometer la aplicación. Es como asegurarme de que las librerías, complementos o versiones que se están usando no tengan defectos o se encuentre con alguna brecha.

### Pruebas de Ingeniería Social

Evalúo las vulnerabilidades de los usuarios frente a intentos de engaño para obtener información confidencial. Es decir, evalúo qué tan fácil sería para un atacante manipular a alguien de la unidad con correos fraudulento o métodos de troya para que revele contraseñas u otros datos sensibles. Es crucial tener en cuenta el factor humano en la seguridad.

## **ASPECTOS RELEVANTES**

## Desde Ecosistema-UNP, evaluaremos el formulario web desde la perspectiva de ciberseguridad, incluyendo aspectos clave como vulnerabilidades y posibles brechas o fugas de información, entre otros. Esto nos permitirá establecer un marco sólido basado en las normas legales e ISO existentes en la Unidad, incluyendo la Ley 1581 de 2012 y la Ley 1266 de 2008, que regulan la protección de datos personales en Colombia y demás Normas Concordantes.

## Además, desarrollaremos un plan de contingencia para posibles eventualidades, lo que resultará en mejores tiempos de respuesta a las solicitudes y una mejor estructuración en los parámetros de la triada (confidencialidad, integridad y disponibilidad).

## Para identificar posibles vulnerabilidades y asegurar que se cumplan las buenas prácticas de seguridad, primero realizo un análisis con herramientas de pruebas estáticas y dinámicas. Esto incluye la revisión ISO 27001 de 2022, que establece una serie de controles y procedimientos para la gestión de la seguridad de la información.

## Elementos que abordamos de la normativa ISO 27001:2022 en Este Formulario WEB:

## Elemento de la Norma ISO 27001:2022 Numeral

## Política de Seguridad de la Información 5.2

## Gestión de Riesgos 6.1

## Declaración de Aplicabilidad (SoA) 4.4

## Objetivos de Seguridad de la Información 6.2

## Plan de Gestión de Riesgos 6.2

## Controles del Anexo A 6.2

## **CONTENIDO.**

## **Gestión de Riesgos**

## Situación analizada y gestión realizada:

## **Validación del lado del cliente y del servidor (Aprobada)**

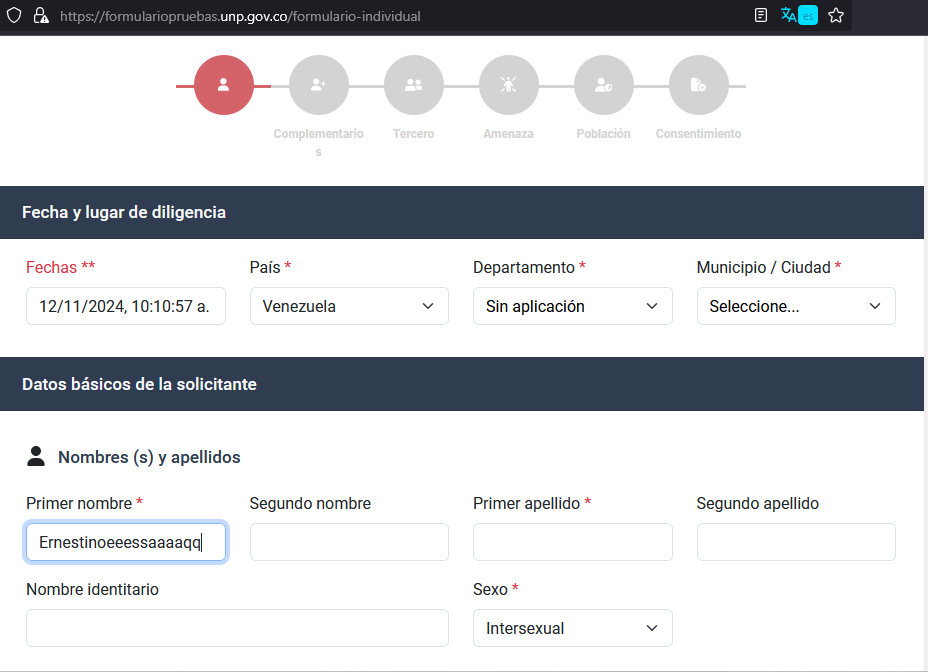
## Riesgos asociados:

## Cliente: Verificamos que el formulario incluya validación en el lado del cliente (React) para evitar entradas inválidas y mejorar la experiencia del usuario.

## Servidor: La validación del lado del cliente no es suficiente, por lo que debe haber validación también en el lado del servidor para prevenir manipulación de datos enviados. (cifrado y anonimizacion)

## Alternativas de solución planteadas: No es necesaria ninguna Acción

## Plan De Acción: Se Realizar pruebas unitarias de Funcionamiento y se evaluaran más Pruebas una vez hecho el desplazamiento de Software. Se realizar pruebas Dinámicas y Estáticas.



## Situación analizada y gestión realizada:

## **Protección contra ataques de inyección (Aprobada)**

## Riesgos asociados:

## Inyección SQL: Asegurar de que los campos que interactúan con una base de datos no sean vulnerables a inyecciones SQL. Utiliza consultas parametrizadas o declaraciones preparadas para evitar este riesgo.

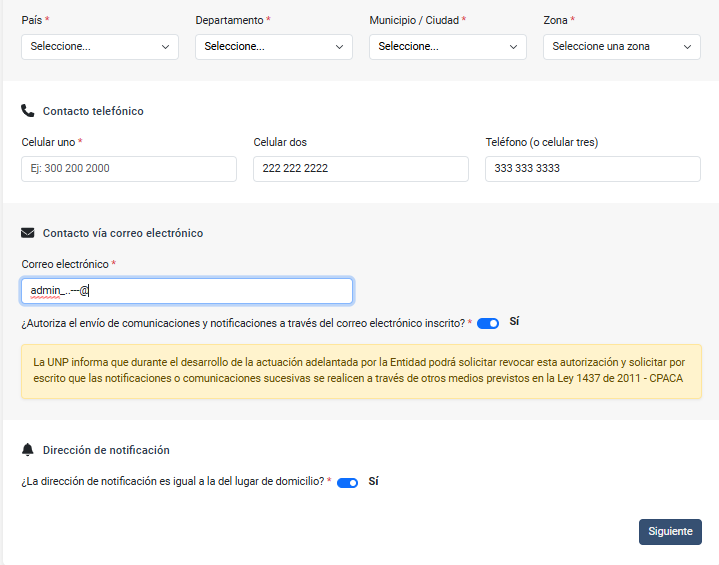
## Inyección de código: Valida y escapa cualquier entrada de usuario para evitar inyección de código que podría comprometer el sistema

## Alternativas de solución planteadas: [Opcional]

## [] No es necesario Ninguna Acción

## Plan De Acción: [Opcional]

## Se Realizar pruebas unitarias de Funcionamiento



## Situación analizada y gestión realizada:

## **Protección contra Cross-Site Scripting (XSS) (Validación Incompleta)**

## Riesgos asociados: [Opcional]

## Verificar que el formulario escape correctamente cualquier entrada de usuario antes de mostrarla en el navegador. Puedes usar bibliotecas de saneamiento de datos o asegurar que el servidor escape caracteres HTML peligrosos (<, >, ", ', etc.).

## Alternativas de solución planteadas: [Opcional]

## [] Realizar pruebas de Validación antes y después con el respectivo análisis una vez realizado el Despliegue en WEB.

## Plan De Acción: [Opcional]

## Situación analizada y gestión realizada:

## **Cross-Site Request Forgery (CSRF) (Token) (Validado)**

## Riesgos asociados: [Opcional]

## Asegurarse de que el formulario incluya un token CSRF. Esto previene que un atacante envíe solicitudes no autorizadas en nombre del usuario autenticado.

## Alternativas de solución planteadas: [Opcional]

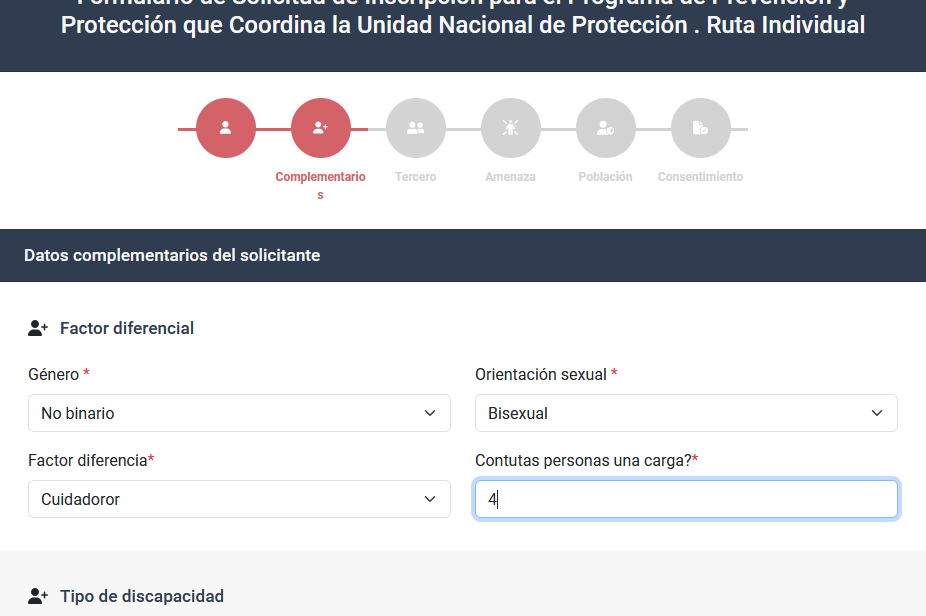
## [] Verificar el mismo en el código y su funcionamiento con react en función o similar en esta forma:

## const token=localStorage.getItem('token');

## Plan De Acción: [Ninguno]

## 

## Situación analizada y gestión realizada:

* **Autenticación y autorización (No Requerido)**
* Riesgos asociados: Si el formulario maneja información sensible, se debe confirmar que el usuario esté autenticado y que tenga los permisos adecuados para acceder o enviar el formulario.
* Alternativas de solución planteadas: [Opcional] A Futuro se buscara implementar una validación con la registraduría para garantizar que los usuarios registrados sean Reales y que sus datos sean Verídicos.
* Plan De Acción: [Opcional]

## Situación analizada y gestión realizada:

## **Protección contra File Upload vulnerabilidades (Validado)**

## Riesgos asociados: (Resultado Final)

## Si el formulario permite subir archivos, validamos el tipo de archivo permitido (por ejemplo, solo imágenes) y establece restricciones de tamaño.

## Almacena los archivos subidos en una ubicación segura y fuera del directorio raíz del servidor web para evitar accesos directos.

## Alternativas de solución planteadas:

## Plan De Acción:

## Realizar pruebas unitarias de Funcionamiento y una vez hecho el desplazamiento de Software realizar pruebas Dinámicas y Estáticas.

## 

## Situación analizada y gestión realizada:

## **Validación de tipos de entrada (Aprobada)**

## Riesgos asociados: [Opcional]

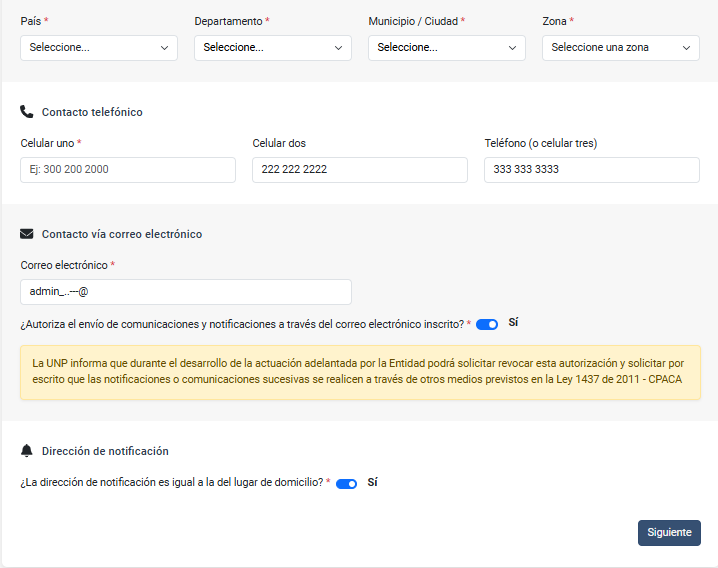
## Restringe los tipos de datos permitidos en cada campo (números, letras, fechas, etc.). Esto ayuda a prevenir inyecciones y mejora la precisión de los datos recopilados.

## Define restricciones de longitud en los campos para evitar ataques de desbordamiento de búfer.

## Alternativas de solución planteadas: [Opcional]

## []

## Plan De Acción: [Opcional]



## Situación analizada y gestión realizada:

## **Cifrado de datos sensibles (Validado)**

## Riesgos asociados:

## Si el formulario envía datos sensibles, como contraseñas o información personal, utiliza HTTPS para asegurar que los datos están cifrados en tránsito.

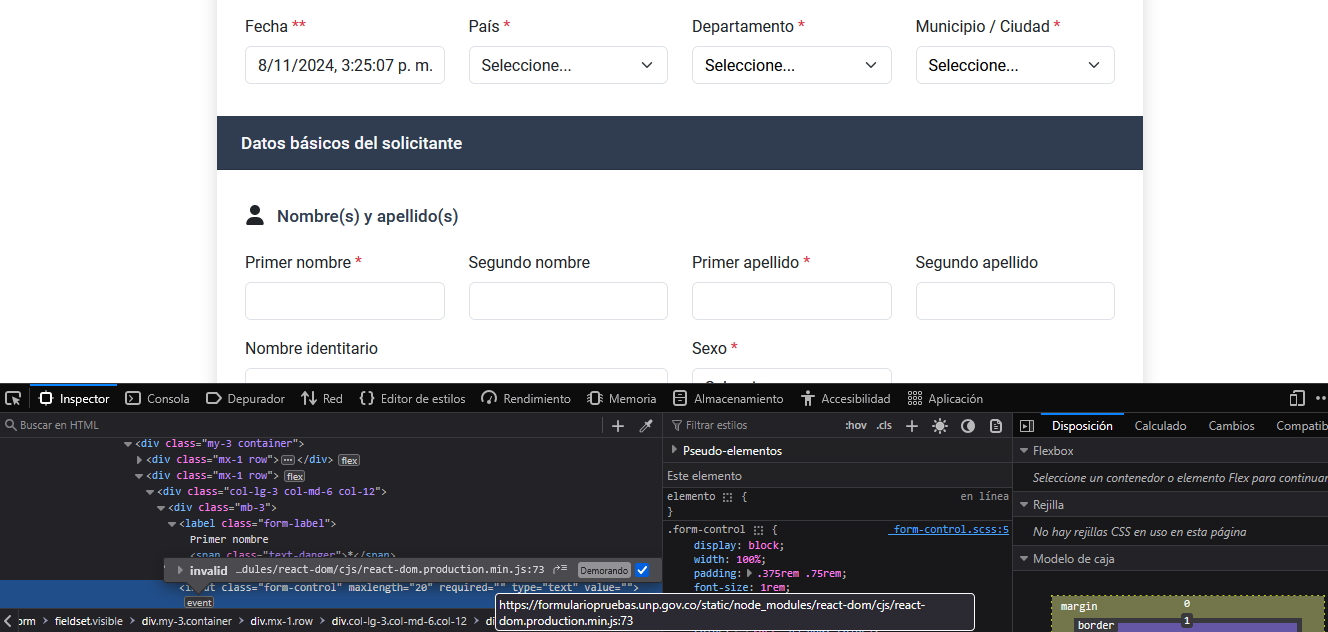
## Almacena cualquier dato sensible (como contraseñas) de manera cifrada en el servidor.

## Alternativas de solución planteadas:

## Plan De Acción:

## Realizar pruebas unitarias de Funcionamiento y una vez hecho el desplazamiento de Software realizar pruebas Dinámicas y Estáticas.

## Situación analizada y gestión realizada:



## **Configuración de encabezados de seguridad (Por Validar)**

## Riesgos asociados:

## Verifica que el servidor esté configurado con encabezados de seguridad HTTP (como Content Security Policy, X-Content-Type-Options, X-Frame-Options y Strict-Transport-Security) para reducir riesgos de XSS, clickjacking y otros ataques.

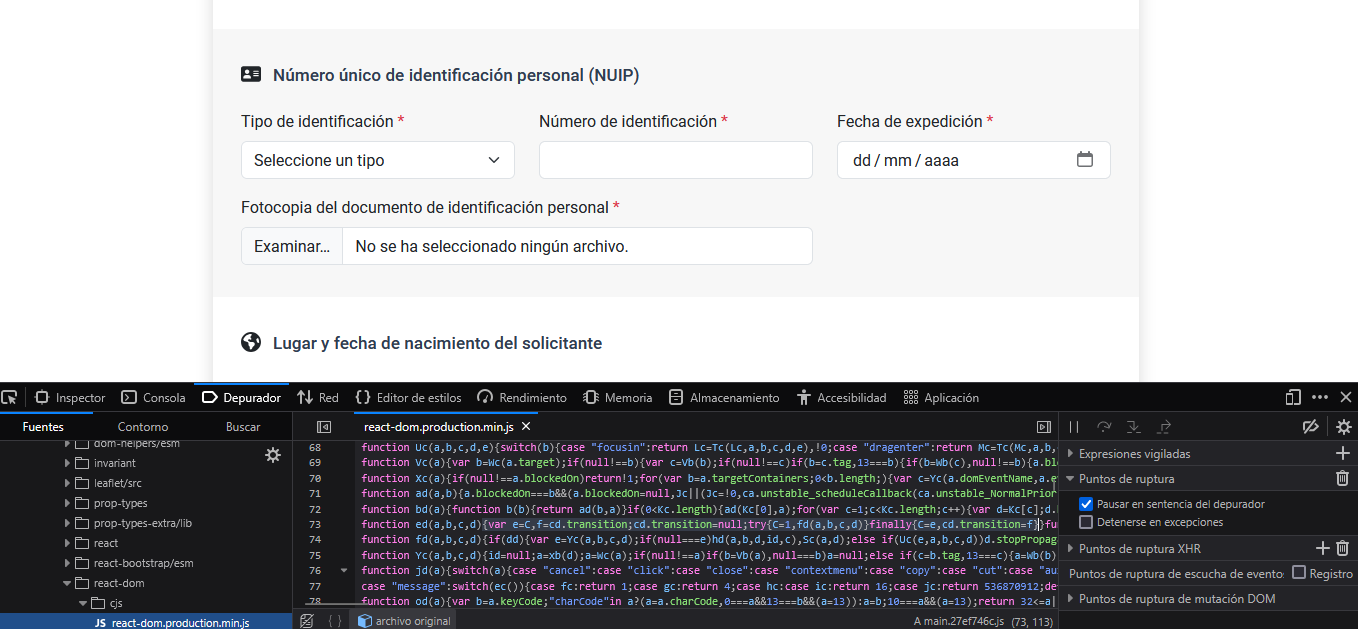
## Alternativas de solución planteadas:

## Sugerencia Esencial Incorporar El CAPTCHA de Google, especialmente la versión reCAPTCHA, para ayudar a prevenir varios tipos de ataques cibernéticos, como fuerza bruta, Ataques DDoS; entre otros.

## Plan De Acción:

## Realizar pruebas unitarias de Funcionamiento y una vez hecho el desplazamiento de Software realizar pruebas Dinámicas y Estáticas. (Incluir la validación de Certificados HTTP)

## Situación analizada y gestión realizada:



## **Pruebas de penetración y auditoría de seguridad (Por Validar)**

## Riesgos asociados:

## Realiza pruebas de penetración en el formulario para identificar vulnerabilidades de seguridad que podrían no ser evidentes.

## Revisa el código y realiza auditorías de seguridad para garantizar que el formulario cumple con las prácticas de seguridad recomendadas.

## Siguiendo estos pasos, podemos ayudar a garantizar que el formulario sea seguro y que no represente un riesgo para la integridad del sistema ni para la privacidad de los usuarios, dando los pilares para un plan de contingencia robusto.

## Alternativas de solución planteadas:

## Sugiero el uso de librerías como **Metasploit** es altamente recomendable para realizar pruebas de penetración y mejorar la ciberseguridad de las aplicaciones.

La instalación podría darse con GIT CLONE y la URL en cuestión.

Plan De Acción: Realizar pruebas unitarias de Funcionamiento y una vez hecho el desplazamiento de Software realizar pruebas Dinámicas y Estáticas.

| No. | Acción de mejora | Responsable y plazo |
| --- | --- | --- |
|  |  |  |
|  |  |  |

## ANEXOS.

## [] [NO APLICA]

## Cordialmente,

## [JEAVER ALEXANDER VERGARA FERNANDEZ].

## [CIBERSEGURIDAD - ECOSISTEMA]

## [Grupo GGT]

## [OAPI]

## Unidad Nacional de Protección

|  |  |  |
| --- | --- | --- |
| GET-FT-05 /V1 | Oficialización: 15/07/2015 | Página 1 de 5 |

## INSTRUCTIVO

## Pautas para la elaboración del Cuerpo del Formato:

## Contienen los diferentes campos de información que deben ser diligenciados. Incluir la firma o firmas del responsable de la información consignada en él y la fecha del diligenciamiento del Formato Informe de Estado y Gestión..

## El cuerpo del Informe debe incluir:

|  |  |  |
| --- | --- | --- |
| *CONTENIDO* | *FR* | *ASPECTOS A TENER EN CUENTA* |
| ENCABEZADO | X | Según las Pautas del numeral 5.4 de la Guía SGI-GU-01/V4 |
| PROPÓSITO | X | Según la estructura del numeral 5.1.2 de la Guía SGI-GU-01/V4 |
| ALCANCE | X | Según la estructura del numeral 5.1.2 de la Guía SGI-GU-01/V4 |
| DEFINICIONES | X | Según la estructura del numeral 5.1.2 de la Guía SGI-GU-01/V4 |
| ASPECTOS RELEVANTES | X | Relacionar los aspectos significativos que impactan el Estado y la Gestión Tecnológica |
| CONTENIDO: | X | El contenido del Informe debe describir cada uno de los temas y/o actividades desarrolladas, de manera concreta, incluyendo especificaciones técnicas de ser necesario.Para cada uno de los eventos y/o actividades incluidas en el Informe, es necesario que se detalle:Situación analizada y gestión realizada [X]Riesgos asociados [Opcional]Alternativas de solución planteadas [Opcional]Plan de Acción [Opcional]Diligenciar Formato:  | No. | Acción de mejora | Responsable y plazo | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  |  La información incluida por cada aspecto debe evidenciar las acciones específicas en materia de la Gestión Tecnológica que el personal del GGTI realiza como parte de las funciones asignadas, bajo los lineamientos o instrucciones impartidas. Para casos particulares es posible que requiera la gestión y participación de otras áreas. |
| Anexos | O |  |
| Firma | X |  |
| Créditos | X |  |

## Requerido = X Opcional= O