

## **PLAN DE RESPALDO**

## **ESCOART**

**SERVICIO NACIONAL DE APRENDIZAJE SENA**

**2024**

## **1. OBJETIVO**

El objetivo de este plan de respaldo es establecer con la creación, almacenamiento y recuperación de datos críticos relacionados con el observador digital

## **2. ALCANCE**

Este plan cubrirá todos los aspectos relacionados con la creación, almacenamiento y recuperación de datos críticos asociados con el observador digital, incluyendo frecuencia de respaldo, medios de almacenamiento, procedimientos de recuperación y responsabilidades del personal

## **3. RESPONSABILIDADES**

- **Diego Arias:** Responsable de configurar y mantener las copias de seguridad en los sistemas informáticos, así como de supervisar su correcto funcionamiento y realizar pruebas periódicas de restauración.
- **Yonatan Bonilla:** Responsable de supervisar el plan de respaldo en un nivel más estratégico, asegurándose de que esté alineado con los objetivos de la empresa, asignando los recursos necesarios y comunicando su importancia a la alta dirección.
- **Gabriel Quintero:** Encargaría de asegurar que las políticas de respaldo cumplen con los estándares de seguridad establecidos por la organización, además de gestionar el acceso a los datos de respaldo y garantizar su integridad.

## **4. CRONOGRAMA DE RESPALDO**

### **4.1. Respaldo después de Despliegue**

- **Frecuencia:** Mensualmente
- **Hora:** 02:00 AM (hora local)
- **Descripción:** Realizar un backup completo de todos los datos del sistema, incluyendo la base de datos de productos y la base de datos de clientes.

### **4.2. Respaldo mientras la codificación Local**

- **Frecuencia:** Diaria
- **Hora:** 08:00 PM (hora local)
- **Descripción:** Crear una copia de seguridad del repositorio de GitHub. Esto puede hacerse manualmente descargando un archivo ZIP del repositorio o utilizando herramientas de respaldo automatizadas como GitHub Actions o scripts personalizados.

## 5. PROCEDIMIENTO DE RESPALDO

### 5.1. Identificación de datos crítico

Para EscoArt la gran mayoría de datos son irremplazable, por ese motivo los datos que se procuran respaldar regularmente son:

- Usuarios
- Roles
- Productos
- Categoría

### 5.2. Configuración de programas de respaldo Desplegado

- **Inicia sesión en tu cuenta de 000webhost:** Ve al sitio web de 000webhost y accede a tu cuenta utilizando tus credenciales de inicio de sesión.
- **Accede al panel de control:** Una vez que hayas iniciado sesión, accede al panel de control de tu cuenta. Por lo general, encontrarás un enlace o botón para acceder al panel de control desde el área de usuario.
- **Busca la opción de backups:** Dentro del panel de control, busca una sección o una opción relacionada con los backups o copias de seguridad. Puede tener un nombre como "Backups", "Copias de seguridad" o "Backup Manager".
- **Configura la frecuencia de los backups:** Una vez que estés en la sección de backups, busca la opción para configurar la frecuencia de los backups. Es posible que haya una configuración predefinida para realizar backups automáticos mensuales. Si no la encuentras, verifica si hay una opción para personalizar la frecuencia de los backups y selecciona "Mensual".
- **Guarda la configuración:** Después de configurar la frecuencia de los backups, asegúrate de guardar los cambios. Esto puede ser a través de un botón de "Guardar" o "Aplicar cambios" en la interfaz.
- **Verifica la configuración:** Una vez que hayas guardado la configuración, verifica que se haya aplicado correctamente. Puedes hacerlo revisando la sección de backups para asegurarte de que esté configurada para realizar backups automáticos mensuales.
- **Realiza una prueba:** Después de configurar los backups automáticos mensuales, es recomendable realizar una prueba para asegurarte de que funcionen correctamente. Puedes hacer esto revisando si se creó una copia de seguridad en el momento programado.

### 5.3. Configuración de programas de respaldo Local

- Para realizar un backup diario de tu repositorio en GitHub, primero clona tu repositorio en tu sistema local si aún no lo has hecho, luego crea un script de respaldo que actualice tu repositorio local y empuje los cambios a un repositorio remoto de respaldo en GitHub. Configura GitHub Actions para ejecutar este script diariamente utilizando un archivo de flujo de trabajo YAML, que automáticamente realizará el respaldo diario según una programación específica. Asegúrate de que el repositorio de respaldo esté configurado correctamente como un nuevo repositorio remoto en tu sistema local y que tengas los permisos adecuados para hacer push a él desde GitHub. Este enfoque garantiza que tu código esté respaldado de manera regular y automática, ofreciendo una capa adicional de protección para tus datos y código fuente en GitHub.

## 6. PROCEDIMIENTOS DE RECUPERACIÓN

### 6.1. Identificación de incidentes

Identificar los incidentes que podrían requerir la ejecución de un plan de respaldo es crucial para una respuesta efectiva ante emergencias. Entre estos incidentes se incluyen la pérdida de datos debido a errores humanos, corrupción de archivos, ataques de malware o fallos del sistema. Asimismo, la detección de ataques de ransomware, fallos de hardware o desastres naturales, como incendios o inundaciones, son señales claras de que se necesita activar el plan de respaldo para restaurar la integridad y la disponibilidad de los datos críticos de la organización. Contar con un plan de respuesta a incidentes bien definido que identifique estos escenarios y establezca los procedimientos adecuados para su manejo es fundamental para minimizar el impacto de los incidentes y garantizar la continuidad del negocio.

### 6.2. Procedimiento de recuperación

- En caso de un borrado de repositorio tendremos que realizar lo siguiente:
  - ingresar a [github.com](https://github.com),
  - clic en la foto de perfil, luego, en settings.
  - En la seccion “Codigo, planificación y automatización” clic en repositorios
  - Clic en Repositorios borrados
  - junto al repositorio que se borro clic en Restore(restaurar)

- En caso de querer realizar un backup se hace mediante Git Bash:
  - con el comando:
    - `git log --oneline`
    - nos saldrá el id de los commits realizados y su comentario de la siguiente manera:
      - `(id commit)(comentario)`
    - ahora con esa información escribiremos el comando:
      - `git checkout (id commit)`

### **6.3. Pruebas de Recuperación**

Para las pruebas de recuperación se realizarán borrados intencionales, de datos específicos, como tablas y del repositorio completo y se restaurarán para probar la fiabilidad de la recuperación, se realizarán con una frecuencia de 3 meses.