

*GERO Y NATIS*

*Plan de restauración*

*Daniel Felipe Bermeo Torres*

*Deisy Caterine Gonzalez Bueno*

*Jennifer Vanessa Mateus Farfan*

*ADSO – 2900602*

Tabla de contenido

[2. INTRODUCCIÓN 2](#_Toc200297581)

[1.1 Objeto 2](#_Toc200297582)

[1.2 Alcance 2](#_Toc200297583)

[3. Información General 2](#_Toc200297584)

[3. DESCRIPCIÓN GENERAL DEL SISTEMA 2](#_Toc200297585)

[3.1 Características del aplicativo 2](#_Toc200297586)

[3.2 Arquitectura y tecnologías 3](#_Toc200297587)

[3.3 Roles del sistema 3](#_Toc200297588)

[4. OBJETIVOS DEL PLAN 3](#_Toc200297589)

[5. IDENTIFICACIÓN DE RIESGOS POTENCIALES 3](#_Toc200297590)

[6. ESTRATEGIA DE RESPALDO Y RECUPERACIÓN 4](#_Toc200297591)

[6.1 Elementos a respaldar 4](#_Toc200297592)

[6.2 Frecuencia de respaldo 4](#_Toc200297593)

[6.3 Almacenamiento de respaldos 4](#_Toc200297594)

[6.4 Validación de respaldos 4](#_Toc200297595)

[7. PROCEDIMIENTO DE RESTAURACIÓN 4](#_Toc200297596)

[7.1 Activación del plan 4](#_Toc200297597)

[7.2 Restauración de la base de datos 4](#_Toc200297598)

[7.3 Restauración de archivos del sistema 4](#_Toc200297599)

[7.4 Validación funcional 5](#_Toc200297600)

[7.5 Registro del incidente 5](#_Toc200297601)

[8.Roles y responsabilidades 5](#_Toc200297602)

[9. Medidas preventivas 5](#_Toc200297603)

[10. TIEMPOS DE RECUPERACIÓN ESTIMADOS 5](#_Toc200297604)

[11. ACTUALIZACIÓN DEL PLAN 6](#_Toc200297605)

[12. ANEXOS 6](#_Toc200297606)

# INTRODUCCIÓN

## Objeto

Establecer un plan claro y estructurado para la restauración del aplicativo web Gero y Natis, con el objetivo de garantizar la continuidad operativa ante fallos técnicos, errores humanos o amenazas externas.

## Alcance

Este plan abarca los procedimientos necesarios para la recuperación del sistema en caso de caídas parciales o totales. Incluye estrategias de respaldo, restauración, validación y documentación de incidentes.

# Información General

* Nombre del proyecto: Gero y Natis
* Tecnologías usadas: HTML, CSS, Bootstrap, PHP, MySQL
* Arquitectura: cliente servidor
* Control de versiones: GitHub
* Equipo de desarrollo:
* Deisy Caterine Gonzalez Bueno
* Daniel Felipe Bermeo Torres
* Jennifer Vanessa Mateus Farfán
* Roles del sistema:
* Administrador
* Vendedor

## 3. DESCRIPCIÓN GENERAL DEL SISTEMA

### 3.1 Características del aplicativo

### Gero y Natis es un sistema web diseñado para gestionar ventas e inventario. Sus principales funcionalidades son:

* Registro y edición de productos
* Control de versiones
* Registró de ventas diarias
* Generación de reportes
* Asignación de roles
* Seguimiento de movimientos de productos
* Gestión de proveedores

### 3.2 Arquitectura y tecnologías

El aplicativo se construyó usando una arquitectura cliente-servidor clásica. Las tecnologías empleadas incluyen:

* Frontend: HTML5, CSS3, Bootstrap
* Backend: PHP 7.x o superior
* Base de datos: MySQL
* Servidor web: Apache
* Sistema operativo: Windows/Linux
* Control de versiones: Git

### 3.3 Roles del sistema

* Administrador: Control total sobre el sistema, incluyendo alta y baja de usuarios, asignación de roles, gestión de productos, proveedores, reportes e inventario.
* Vendedor: Acceso restringido a funcionalidades de registro de ventas y consulta de productos.

## 4. OBJETIVOS DEL PLAN

* Minimizar el tiempo de inactividad
* Preservar la integridad y disponibilidad de los datos
* Establecer un procedimiento estándar y reproducible para la restauración del sistema
* Determinar responsabilidades y recursos necesarios ante una contingencia
* Documentar todos los incidentes y acciones realizadas

## 5. IDENTIFICACIÓN DE RIESGOS POTENCIALES

1. Fallas del servidor: Por sobrecarga, daño físico o mal funcionamiento
2. Errores humanos: Eliminación accidental de datos, cambios erróneos en la configuración.
3. Ataques informáticos: Inyecciones SQL, malware, ransomware, accesos no autorizados.
4. Fallas de red o conectividad: Pérdida de conectividad
5. Problemas con la base de datos: Corrupción de archivos, conflictos de integridad.
6. Desastres naturales: Incendios, inundaciones, etc

## 6. ESTRATEGIA DE RESPALDO Y RECUPERACIÓN

### 6.1 Elementos a respaldar

* Archivos del código fuente (HTML, CSS, PHP, JS)
* Base de datos MySQL
* Configuraciones del servidor Apache
* Archivos de medios (imágenes, logos)
* Archivos .env (configuración de entorno)

### 6.2 Frecuencia de respaldo

* Base de datos: Diaria (Manual hasta automatizar)
* Archivos del proyecto: Semanal y antes de cualquier actualización
* Configuraciones: Mensual o tras cambios importantes

### 6.3 Almacenamiento de respaldos

* Almacenamiento local (discos duros externos cifrados)
* Almacenamiento en la nube (Google Drive, Dropbox, Amazon S3)
* Repositorio Git (para el código fuente, sin datos sensibles)

### 6.4 Validación de respaldos

Se deben realizar pruebas mensuales para validar que los respaldos puedan restaurarse sin errores. Se recomienda mantener al menos 3 versiones anteriores de los respaldos.

## 7. PROCEDIMIENTO DE RESTAURACIÓN

### 7.1 Activación del plan

Ante la detección de una falla grave o caída del sistema:

* El incidente debe ser reportado al responsable del plan de restauración.
* Se debe evaluar el alcance del daño (base de datos, archivos, ambos).
* Se determina si la restauración debe realizarse completa o parcial.

### 7.2 Restauración de la base de datos

* Detener temporalmente el acceso al sistema para evitar conflictos.
* Acceder al servidor de base de datos y cargar el respaldo más reciente
* Verificar integridad de las tablas y datos críticos.

### 7.3 Restauración de archivos del sistema

* Subir los archivos desde el respaldo al servidor web
* Verificar permisos de ejecución y lectura/escritura.
* Restaurar configuraciones de Apache y reiniciar el servicio.

### 7.4 Validación funcional

* Acceder como administrador y vendedor
* Registrar una venta ficticia
* Confirmar correcto funcionamiento de todas las funciones

### 7.5 Registro del incidente

Se debe documentar:

* Fecha y hora del incidente
* Causa identificada
* Acciones tomadas y responsables
* Tiempo total de restauración

## 8.Roles y responsabilidades

|  |  |
| --- | --- |
| Funciones | Responsables |
| Realización y verificación de respaldos | Daniel Bermeo |
| Restauración de base de datos | Deisy Gonzalez |
| Restauración de archivos del sistema | Vanessa mateus |
| Comunicación y documentación | Todos los integrantes |

## 9. Medidas preventivas

* Uso de contraseñas seguras y actualización periódica
* Validaciones en formularios y consultas prepaeadas
* Supervisión de logs del sistema
* Configuración adecuada de permisos y roles
* Monitoreo de actividad sospechosa

## 10. TIEMPOS DE RECUPERACIÓN ESTIMADOS

|  |  |
| --- | --- |
| Tipo de falla | Tiempo de restauración estimado |
| Restauración de base de datos | 15 - 30 minutos |
| Restauración de archivos del sistema | 30 - 60 minutos |
| Caída total del servidor | 1 - 3 horas |
| Falla catastrófica (todo perdido) | 4 - 6 horas |

## 11. ACTUALIZACIÓN DEL PLAN

## El plan será revisando cada 3 meses o cuando ocurran cambios significativos en:

* Arquitectura del sistema
* Equipo de desarrollo
* Funcionalidades críticas
* Políticas de seguridad

Cada actualización debe incluir:

* Registro de cambios
* Nueva Versión del documento
* Notificación al equipo

## 12. ANEXOS

* Script de respaldo automático MySQL
* Bitácora de incidentes
* Checklists de validación post-restauración
* Políticas de contraseñas y accesos
* Contacto de proveedores
* Control de versiones del documento