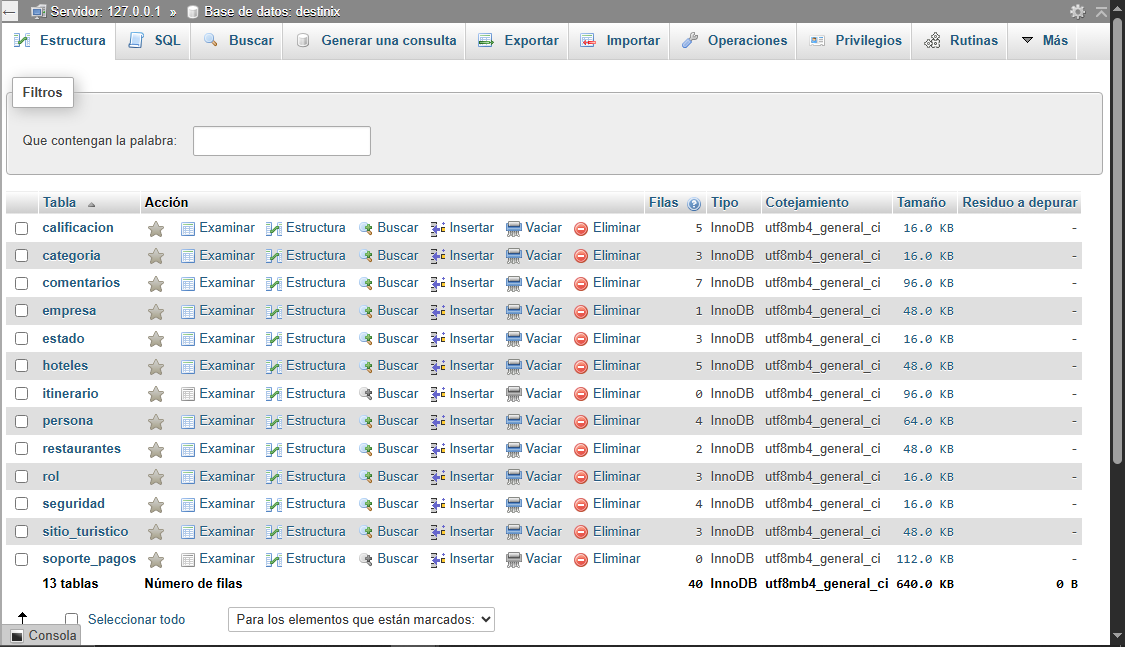
Plan de Respaldo del Proyecto React + MySQL

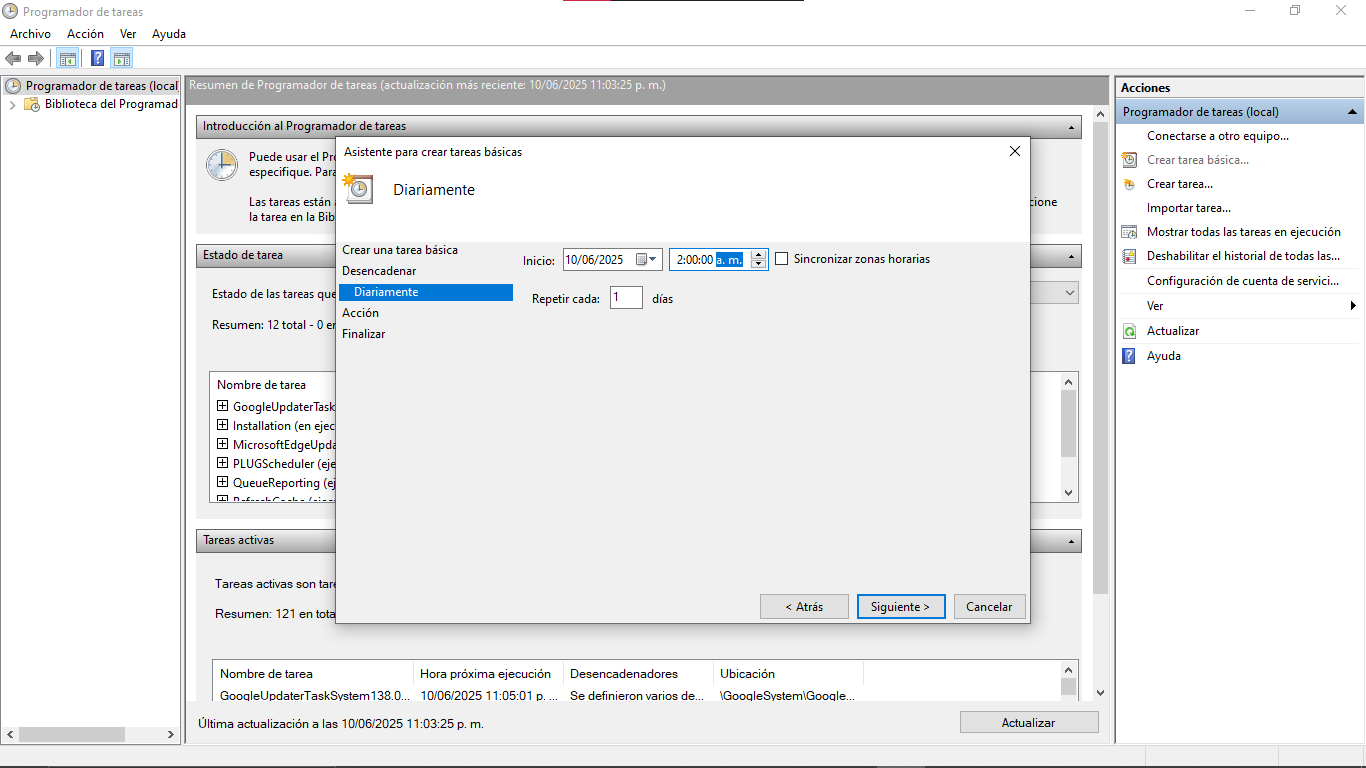
# 1. Introducción

Objetivo del plan: Proteger la integridad y disponibilidad de la información crítica del proyecto Destinix.  
  
Alcance: Este plan cubre el respaldo del código fuente del proyecto React ubicado en 'C:\Users\diego\destinix' y la base de datos 'destinix'.  
  
Importancia: Se mitigan riesgos como pérdida accidental de datos, fallos del sistema, errores humanos y posibles infecciones por malware o ransomware.

# 2. Inventario de activos

- Código fuente del proyecto Destinix (React).  
- Base de datos MySQL llamada 'destinix'.  
  
Clasificación por prioridad:  
- Alta: Base de datos 'destinix'.  
- Alta: Código fuente del proyecto.

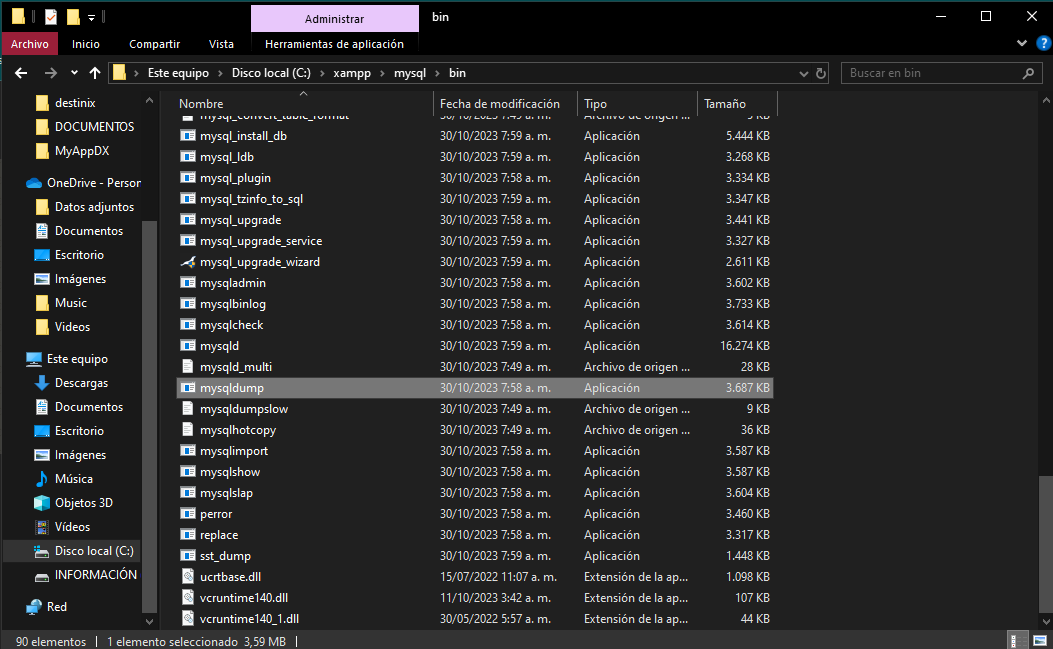
# 3. Política de respaldo

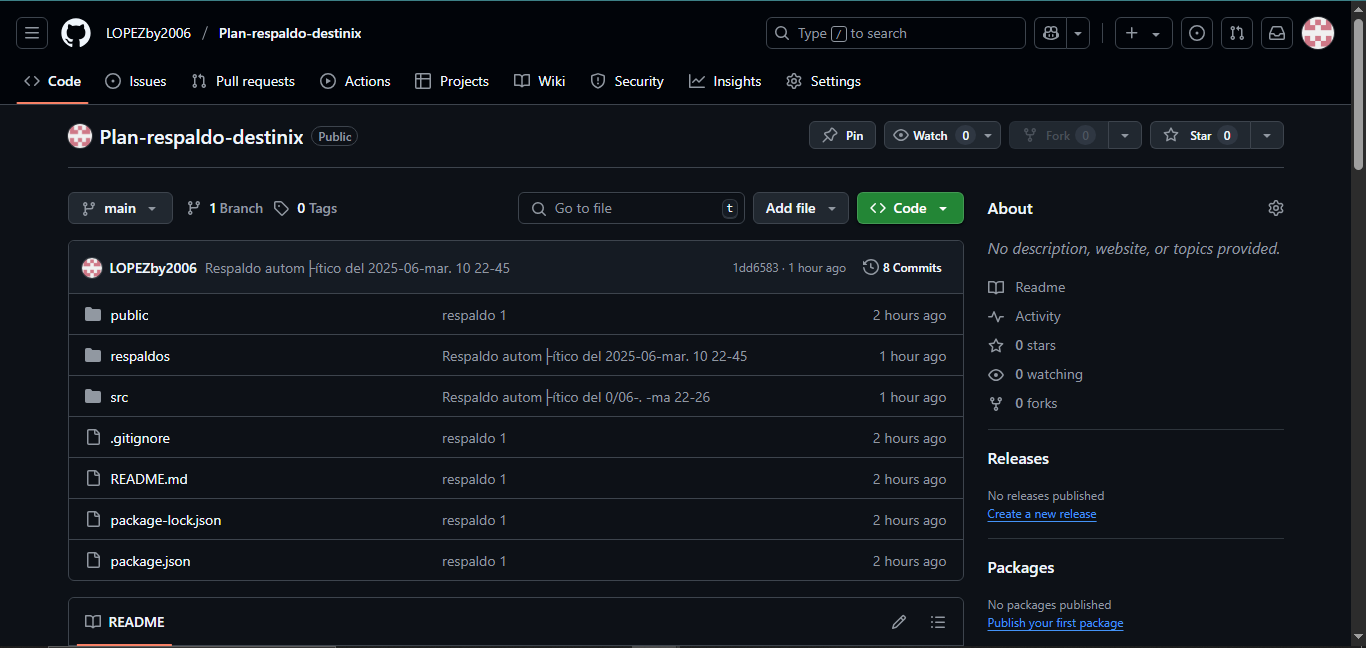
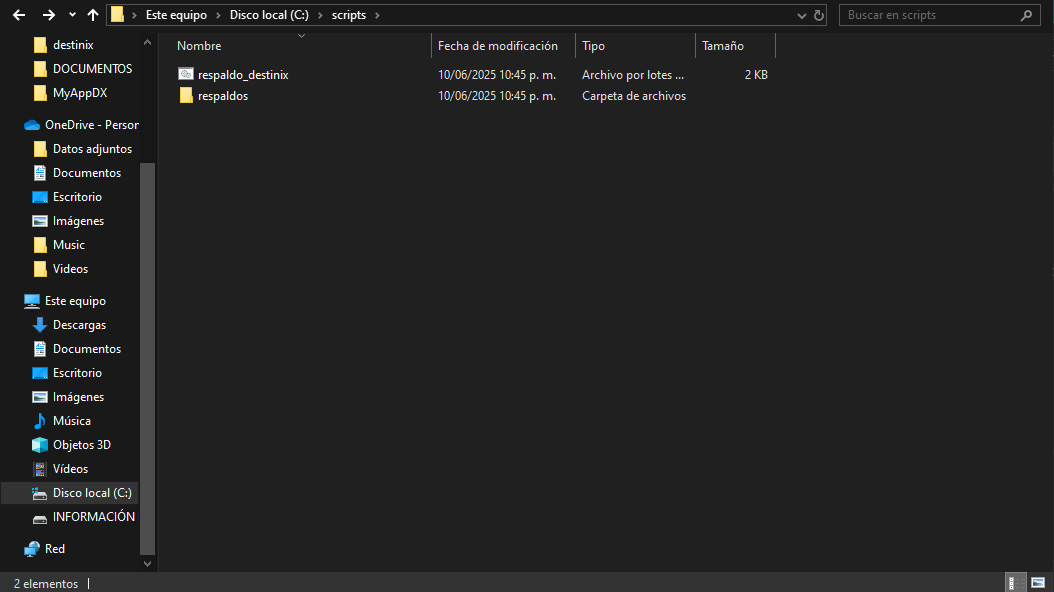
Frecuencia del respaldo:  
- Diario (código y base de datos).  
  
Tipo de respaldo:  
- Completo (copias completas del proyecto y la base de datos).  
  
Horarios programados:  
- Diariamente a las 2:00 a.m. mediante tareas programadas en Windows.  
 

# 4. Método y medio de respaldo

Software o herramientas utilizadas:  
- Git para respaldo del código fuente.

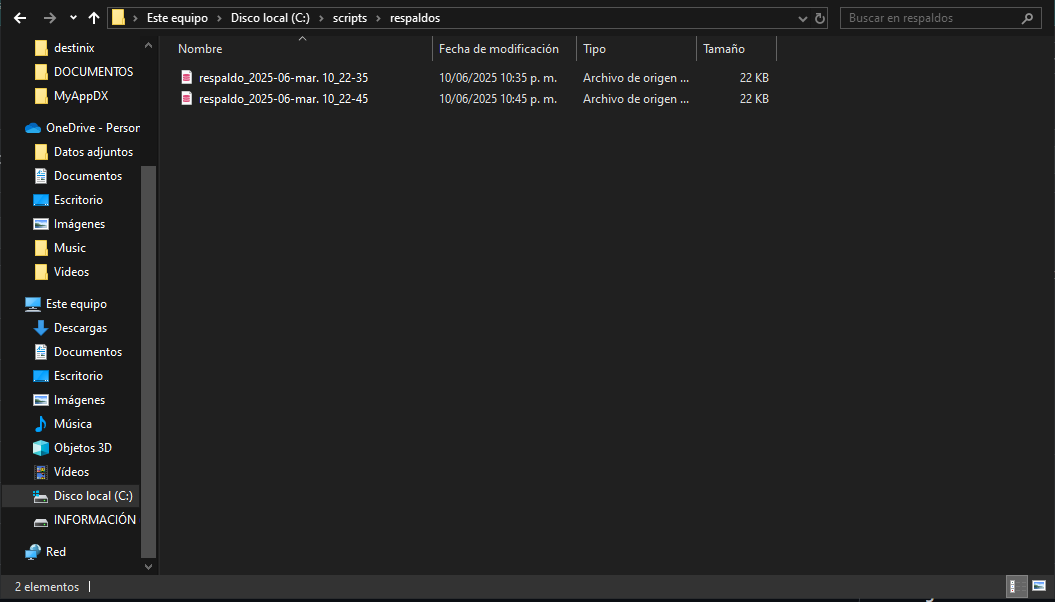
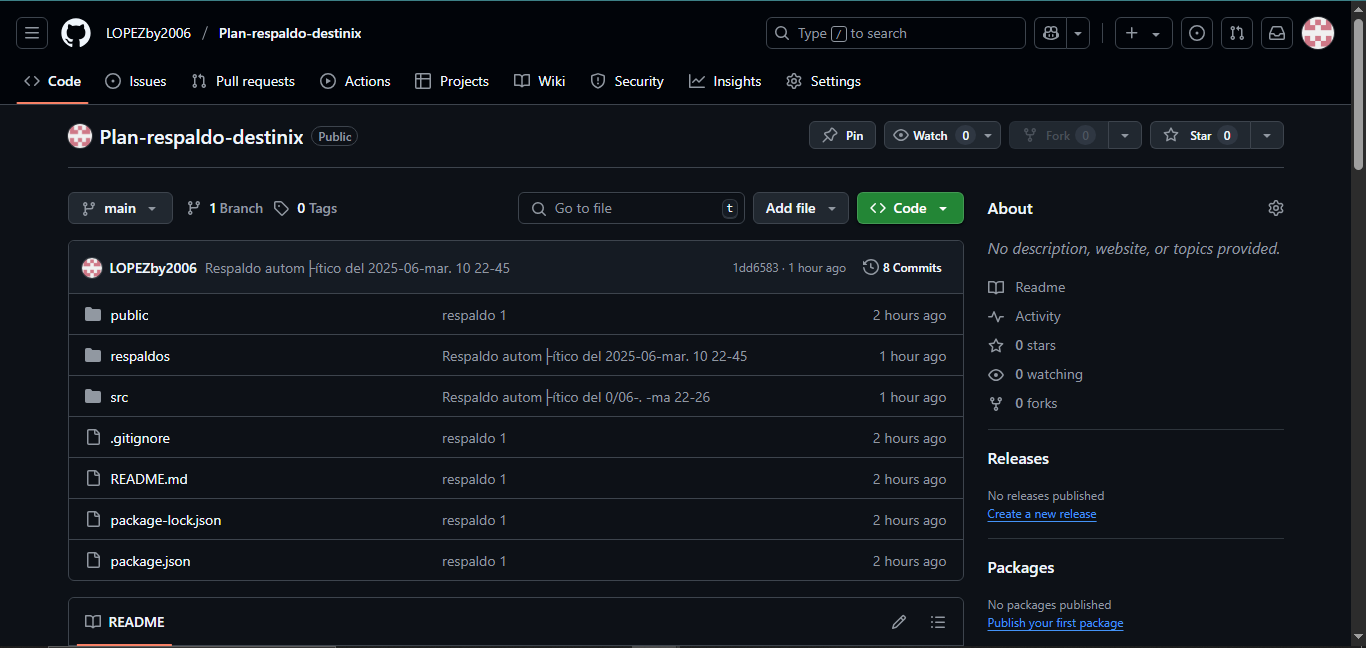
- mysqldump para respaldo de base de datos.

  
  
Medio de almacenamiento:  
- Repositório GitHub público.

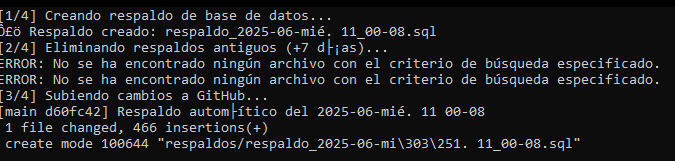
  
- Carpeta local: 'C:\scripts\respaldos'.  
 

# 5. Ubicación del respaldo

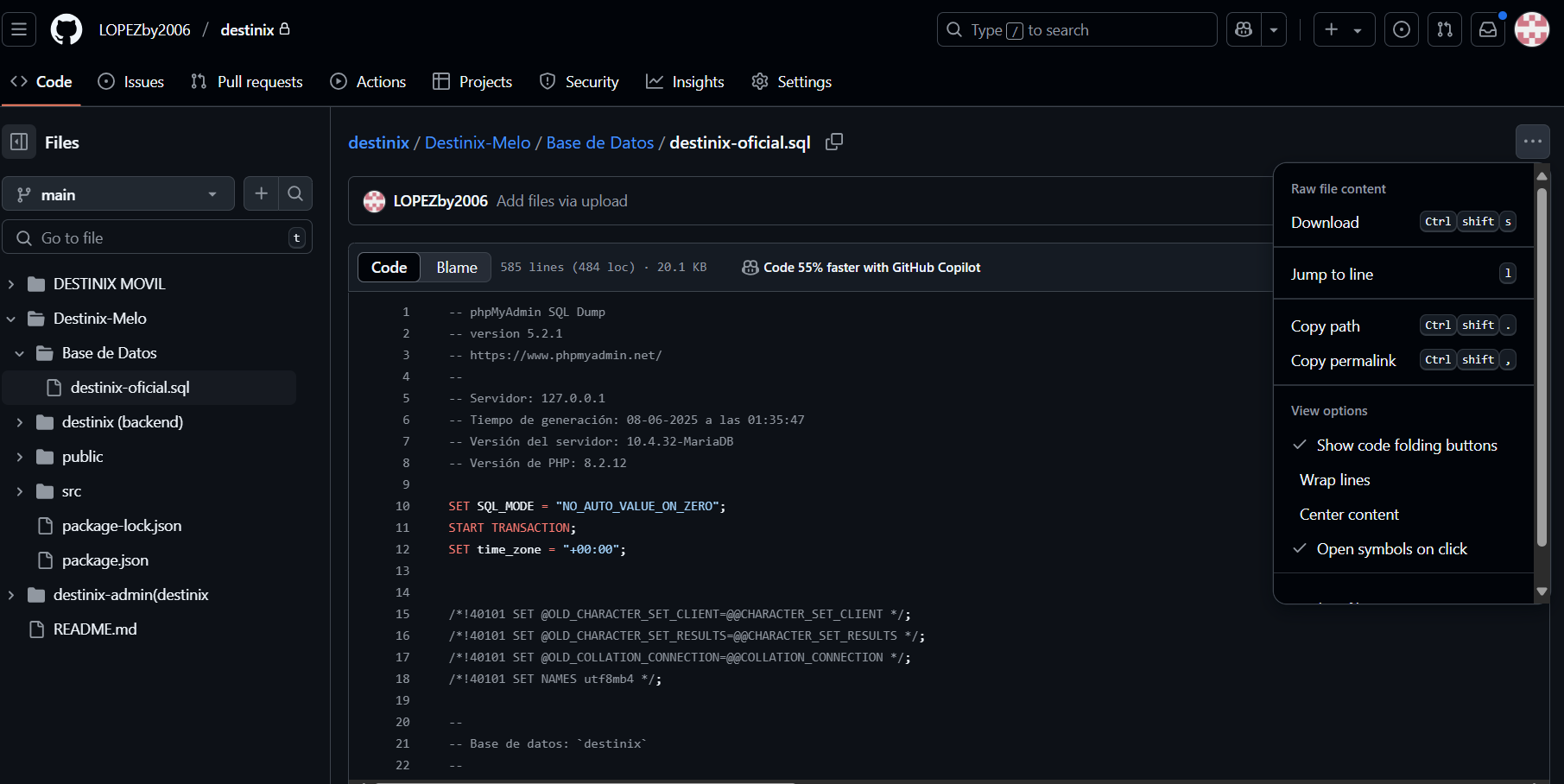
- Local: C:\scripts\respaldos (archivos .sql de la base de datos).

  
- GitHub: Repositorio remoto donde se sube todo el proyecto y los archivos .sql.  
 

# 6. Proceso de verificación

- Se verifica si el archivo .sql se crea correctamente con un script batch.  
- Se revisa el historial de commits en GitHub.  
- Validación manual del contenido del archivo de respaldo.  
  
Frecuencia:  
- Verificación con cada ejecución automática (diaria).

# 7. Procedimiento de restauración

1. Descargar el archivo .sql desde el repositorio.  
2. Abrir XAMPP y ejecutar el siguiente comando desde la terminal:  
 mysql -u root destinix < respaldo.sql  
3. Clonar el repositorio de GitHub en la máquina local.  
4. Ejecutar `npm install` y luego `npm start` para levantar el proyecto.  
  
Entramos al repositorio , y accedemos a la base de datos , y descargamos el archivo.sql   
  
  
RTO estimado: 10 a 15 minutos.  
Datos críticos: Base de datos y configuración del frontend.

# 8. Responsables

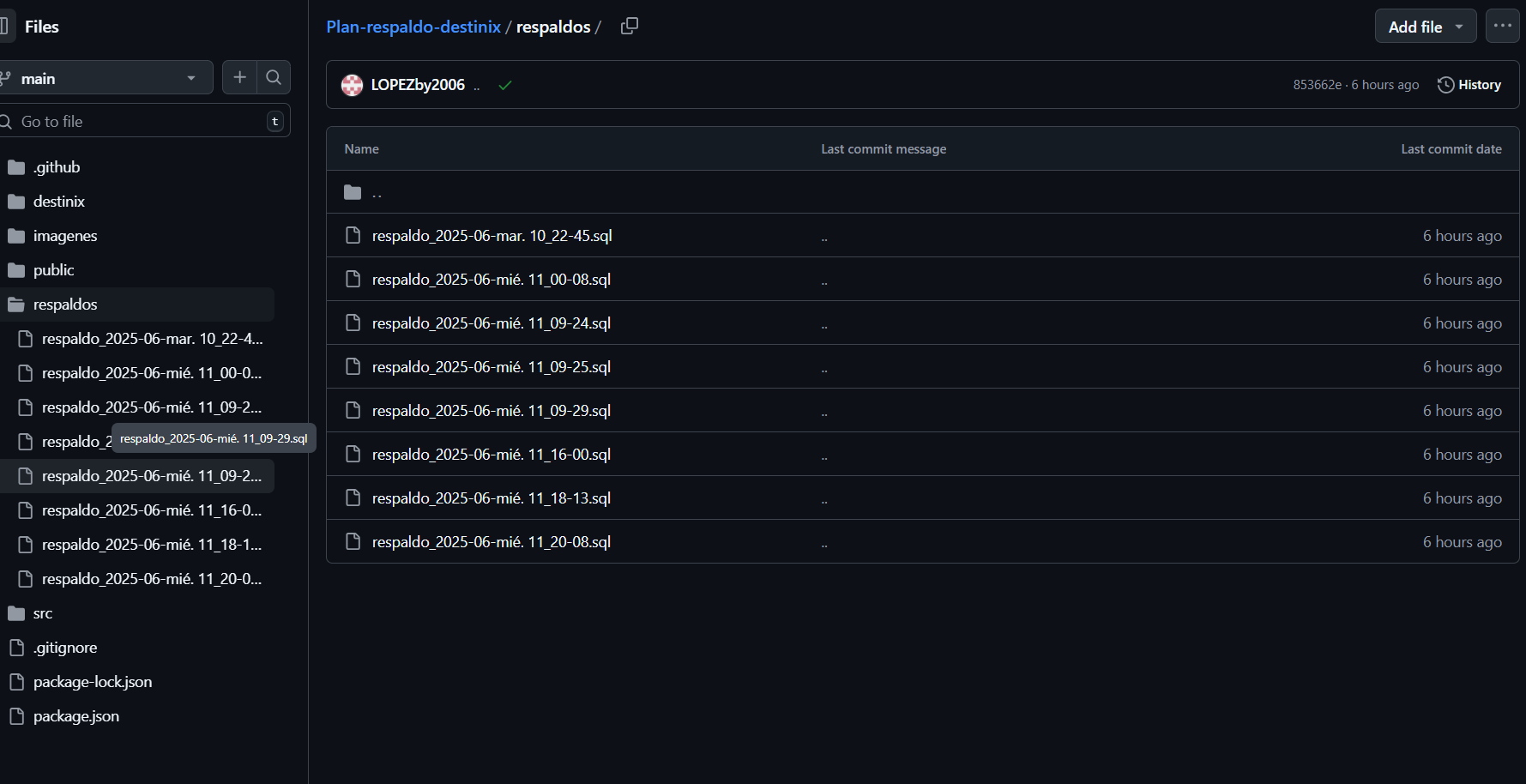
Encargado del respaldo: Usuario principal del proyecto (Diego)  
  
Responsables de verificar:  
- El mismo usuario o encargado designado.  
  
Soporte técnico externo: No aplica (respaldo gestionado localmente).

# 9. Cronograma

- Respaldo diario del proyecto y base de datos a las 2:00 a.m.  
- Verificación del respaldo cada mañana.  
  
Uso del Programador de tareas de Windows para ejecutar automáticamente el script .bat.

# 10. Historial y registros

- Bitácora de respaldos disponible en los commits de GitHub.  
- Cada respaldo lleva un nombre con fecha y hora (respaldo\_YYYY-MM-DD\_HHMM.sql).



# 11. Medidas de seguridad

- Control de acceso: Repositorio con autenticación.  
- Los respaldos .sql pueden protegerse con herramientas de cifrado si se requiere.  
- El equipo debe tener acceso restringido.

# 12. Revisión y mantenimiento del plan

- Revisión mensual del plan y funcionamiento de los scripts.  
- Actualización del script ante cambios de estructura en la base de datos o código fuente.