# 

Contenido

[**PLAN DE RESPALDO – SISTEMA MASTER BARBER 2**](#_heading=h.p6yabehw2i4r)

[**1. INTRODUCCIÓN 2**](#_heading=h.fp3vabu0smis)

[**2. OBJETIVOS DEL RESPALDO 3**](#_heading=h.j0o6l6aixol3)

[**3. ELEMENTOS A RESPALDAR 3**](#_heading=h.jt5xa4mu93nt)

[**3.1 Base de Datos – MySQL (AWS RDS) 3**](#_heading=h.stwdvotop68d)

[**3.2 API Backend – Node.js + Express 3**](#_heading=h.vxu8c2wuwaok)

[**3.3 Frontend Web – ReactJS 4**](#_heading=h.zgtuycwmugwp)

[**3.4 Aplicación Móvil – React Native (Expo) 4**](#_heading=h.tcp5jxvlom7u)

[**4. ESTRATEGIA DE RESPALDO 5**](#_heading=h.yt67u34np3ml)

[**4.1 Base de Datos (RDS MySQL) 5**](#_heading=h.lvv17rpp7muo)

[**4.2 Código Fuente (GitHub) 5**](#_heading=h.64hhyrhyh1i2)

[4.3 Configuraciones y Variables de Entorno 6](#_heading=)

[**4.4 Despliegue Web (Amplify) y API (Render) 6**](#_heading=h.1pmuhm9bi8d5)

[**5. RESPONSABILIDADES 7**](#_heading=h.5g2w1764g7kh)

[**6. PRUEBAS DE RESTAURACIÓN 7**](#_heading=h.fp5l9766dpfu)

[**7. RECOMENDACIONES ADICIONALES 8**](#_heading=h.bnvnq9wp2xhn)

[**9. ANEXOS 8**](#_heading=h.6qw3fb5ntw61)

# 

# **PLAN DE RESPALDO – SISTEMA MASTER BARBER**

## **1. INTRODUCCIÓN**

Este documento describe el plan de respaldo y recuperación para el sistema **Master Barber**, una solución integral que permite gestionar:

* Inventario de productos
* Reservas y turnos de clientes
* Cuentas y horarios de barberos

El sistema está compuesto por cuatro módulos principales:

1. Base de datos (MySQL en AWS RDS)
2. API Backend (Node.js + Express en Render)
3. Frontend Web (ReactJS en AWS Amplify)
4. Aplicación Móvil (React Native)

Repositorios de código fuente:

* Frontend: [master\_barber\_web](https://github.com/DavidV4iss/master_barber_web)
* Backend + Base de datos: [master\_barber\_api](https://github.com/DavidV4iss/master_barber_api)
* App móvil: [master\_barber\_movil](https://github.com/DavidV4iss/master_barber_movil)

## **2. OBJETIVOS DEL RESPALDO**

1. Asegurar la integridad de los datos del sistema Master Barber
2. Permitir recuperación rápida ante fallas de software, hardware o errores humanos
3. Minimizar el tiempo de inactividad
4. Cumplir con buenas prácticas de seguridad y disponibilidad

## **3. ELEMENTOS A RESPALDAR**

### **3.1 Base de Datos – MySQL (AWS RDS)**

* Base de datos: master\_barber
* Información crítica: reservas, usuarios, productos
* Respaldos automáticos en AWS activados

### **3.2 API Backend – Node.js + Express**

* Código fuente en GitHub: [master\_barber\_api](https://github.com/DavidV4iss/master_barber_api)
* Variables de entorno y archivos de configuración
* Dependencias (package.json, .env)
* Logs de actividad en Render

### **3.3 Frontend Web – ReactJS**

* Código fuente en GitHub: [master\_barber\_web](https://github.com/DavidV4iss/master_barber_web)
* Configuraciones de despliegue en AWS Amplify
* Variables de entorno del entorno de producción

### **3.4 Aplicación Móvil – React Native (Expo)**

* Código fuente en GitHub: [master\_barber\_movil](https://github.com/DavidV4iss/master_barber_movil)
* Configuraciones de conexión con la API

## 

## **4. ESTRATEGIA DE RESPALDO**

### **4.1 Base de Datos (RDS MySQL)**

* **Tipo de respaldo:** Automático y manual
* **Frecuencia automática:** Diaria (AWS RDS)
* **Retención automática:** 7 días
* **Respaldos manuales:**
  + Antes de actualizaciones mayores o migraciones
  + Retención: 30 días

### **4.2 Código Fuente (GitHub)**

* **Tipo de respaldo:** Control de versiones (Git) + Clonación en almacenamiento externo
* **Frecuencia:**
  + Push continuo con ramas protegidas
  + Clonación semanal a almacenamiento externo (Drive o local)

### 

### **4.3 Configuraciones y Variables de Entorno**

* **Tipo de respaldo:** Copias cifradas de archivos .env y configuraciones de despliegue
* **Frecuencia:** Semanal
* **Almacenamiento:**
  + Backup cifrado en servicio de almacenamiento en la nube (Google Drive privado)
  + Copia local en USB cifrado

### **4.4 Despliegue Web (Amplify) y API (Render)**

* **Tipo de respaldo:**
  + Código ya respaldado vía GitHub
  + Configuraciones exportadas manualmente (entorno, dominio, rutas)
* **Frecuencia:** Mensual o ante cambios importantes
* **Método:** Captura de configuración + exportación de configuración como documentación interna

## **5. RESPONSABILIDADES**

| **Rol** | **Tareas asignadas** |
| --- | --- |
| Administrador DevOps | Monitoreo y configuración de respaldos RDS, Render y Amplify |
| Desarrollador Líder | Verificación de integridad del código y configuración semanal |
| Soporte Técnico | Almacenamiento y verificación de archivos .env y respaldo físico |

## **6. PRUEBAS DE RESTAURACIÓN**

* **Frecuencia:** Trimestral
* **Componentes a restaurar:**
  + Clonar base de datos
  + Recuperar código desde backup externo
  + Validar funcionamiento del sistema con entorno de pruebas

## **7. RECOMENDACIONES ADICIONALES**

1. Automatizar exportación semanal de datos críticos
2. Configurar alertas de fallos de respaldo en RDS
3. Implementar sistema de versionado en base de datos si se realizan cambios frecuentes en la estructura

## **9. ANEXOS**

* Accesos a plataformas utilizadas (AWS, Render, GitHub)
* Instrucciones paso a paso para restaurar la base de datos
* Checklists de verificación de backups
* Cronograma de respaldo automatizado y manual