

Contenido

[🔹 RESUMEN EJECUTIVO - PROYECTO RIFAS SOLIDARIAS 3](#_Toc207975050)

[📊 ESTADO ACTUAL: COMPLETADO EXITOSAMENTE 3](#_Toc207975051)

[🎯 OBJETIVOS LOGRADOS 3](#_Toc207975052)

[✅ CAPÍTULO 1: INFRAESTRUCTURA Y BASE DE DATOS 3](#_Toc207975053)

[✅ CAPÍTULO 2: CRUDs BÁSICOS OPERATIVOS 3](#_Toc207975054)

[✅ DOCUMENTACIÓN Y TESTING 3](#_Toc207975055)

[🏗️ ARQUITECTURA IMPLEMENTADA 3](#_Toc207975056)

[🚀 FUNCIONALIDADES OPERATIVAS 4](#_Toc207975057)

[Instituciones 4](#_Toc207975058)

[Usuarios 4](#_Toc207975059)

[🔧 TECNOLOGÍAS IMPLEMENTADAS 4](#_Toc207975060)

[📈 MÉTRICAS DE ÉXITO 5](#_Toc207975061)

## 🔹 ****RESUMEN EJECUTIVO - PROYECTO RIFAS SOLIDARIAS****

### ****📊 ESTADO ACTUAL: COMPLETADO EXITOSAMENTE****

# ****🎯 OBJETIVOS LOGRADOS****

## ****✅ CAPÍTULO 1: INFRAESTRUCTURA Y BASE DE DATOS****

* **Base de datos MariaDB** configurada y funcionando
* **Schema completo** implementado (8 tablas principales)
* **Conexión Node.js + MySQL** estable
* **Variables de entorno** configuradas

## ****✅ CAPÍTULO 2: CRUDs BÁSICOS OPERATIVOS****

* **CRUD Instituciones** completo y funcional
* **CRUD Usuarios** completo y funcional
* **Validaciones robustas** implementadas
* **Manejo de errores** profesional

## ****✅ DOCUMENTACIÓN Y TESTING****

* **Swagger UI** funcionando correctamente
* **Endpoints documentados** con ejemplos
* **Testing manual** exitoso

# ****🏗️ ARQUITECTURA IMPLEMENTADA****

rifas-solidarias-backend/

├── src/

│ ├── controllers/

│ │ ├── institucionController.js ✅

│ │ └── usuarioController.js ✅

│ ├── routes/

│ │ ├── instituciones.js ✅

│ │ └── usuarios.js ✅

│ └── config/

│ ├── db.js ✅

│ └── swagger.js ✅

├── database/

│ └── schema/

│ └── rifas\_solidarias\_schema.sql ✅

├── .env ✅

├── index.js ✅

└── package.json ✅

# ****🚀 FUNCIONALIDADES OPERATIVAS****

## ****Instituciones****

* ✅ **Listar** todas las instituciones
* ✅ **Crear** nueva institución
* ✅ **Obtener** institución específica
* ✅ **Actualizar** institución
* ✅ **Eliminar** institución (con validaciones)

## ****Usuarios****

* ✅ **Listar** todos los usuarios
* ✅ **Crear** nuevo usuario (con encriptación)
* ✅ **Obtener** usuario específico
* ✅ **Actualizar** usuario
* ✅ **Eliminar** usuario
* ✅ **4 roles** disponibles: admin\_global, admin\_institucion, vendedor, comprador

# ****🔧 TECNOLOGÍAS IMPLEMENTADAS****

| **Tecnología** | **Estado** | **Uso** |
| --- | --- | --- |
| **Node.js + Express** | ✅ | Backend API |
| **MariaDB/MySQL** | ✅ | Base de datos |
| **bcrypt** | ✅ | Encriptación passwords |
| **CORS** | ✅ | Cross-origin requests |
| **Swagger** | ✅ | Documentación API |
| **dotenv** | ✅ | Variables entorno |

# ****📈 MÉTRICAS DE ÉXITO****

* **8 endpoints** funcionando correctamente
* **2 entidades principales** completamente operativas
* **100% uptime** en testing manual
* **Documentación completa** disponible
* **Validaciones robustas** implementadas
* **Arquitectura escalable** establecida