

# Proyecto Final CTT 2025

---

Título del Proyecto: Sistema de Rifas Online

Unidad Curricular: Registro Documental

Hito correspondiente: Diseño e Implementación de Base de Datos

Versión del Documento: 0.1

Fecha de emisión: 17/08/2025

Autor/es: Ruben Perurena, Federico Bisio, Claudia Martinez, Fabian Gonzalez

Revisor/Tutor: Juan Damian Pajares García

Estado: Entrega Hito 1

# Índice

Historial de cambios.....	3
1. Introducción.....	5
1.1 Propósito.....	5
1.2 Alcance.....	5
1.3 Referencias.....	5
1.4 Definiciones y acrónimos.....	5
2. Requerimientos.....	5
2.1 Requerimientos funcionales.....	5
2.2 Requerimientos no funcionales.....	5
2.3 Restricciones.....	6
2.4 Trazabilidad.....	6
3. Arquitectura y Diseño.....	6
3.1 Herramientas utilizadas.....	6
4. Plan de Implementación.....	6
4.1 Actividades a realizar.....	6
4.2 Cronograma (por hito).....	7
4.3 Responsables.....	7
5. Plan de Verificación y Validación.....	8
5.1 Casos de prueba asociados a cada requerimiento.....	8
5.2 Herramientas de testing utilizadas.....	8
5.3 Criterios de aceptación.....	8
6. Repositorio.....	9

## Historial de cambios

Versión	Fecha	Autor	Descripción del cambio	Revisor
0.1	17/08/2025	Ruben Perurena, Federico Bisio, Claudia Martinez, Fabian Gonzales	Creación inicial	Todo el grupo

## 1. Introducción

### 1.1 Propósito

El propósito de este documento es describir el diseño, requerimientos, implementación y validación del sistema de rifas online. Este sistema permitirá gestionar rifas digitales, incluyendo usuarios, reservas, pagos y asignación de números.

### 1.2 Alcance

El documento cubre el modelado de la base de datos, requerimientos funcionales y no funcionales, arquitectura y plan de pruebas. Aplica al desarrollo parcial del proyecto final en la unidad curricular de Bases de Datos y Programación.

### 1.3 Referencias

- ISO/IEC 9126
- ISO/IEC/IEEE 29148
- Manual de Gestión Documental CTT

### 1.4 Definiciones y acrónimos

Rifa: Sorteo en el que los usuarios compran números para optar a un premio.

Reserva: Asignación temporal de un número hasta que sea pagado.

MP: Mercado Pago, pasarela de pagos utilizada en el sistema.

## **2. Requerimientos**

### **2.1 Requerimientos funcionales**

RF-01: El sistema debe permitir el registro y autenticación de usuarios con Google.

RF-02: El sistema debe permitir la creación y gestión de rifas.

RF-03: El sistema debe permitir reservar números de una rifa.

RF-04: El sistema debe procesar pagos a través de una pasarela de pago (MP).

RF-05: El sistema debe asignar un número ganador al finalizar la rifa.

RF-06: El sistema debe enviar notificaciones por correo al usuario cuando compre un número de rifa.

RF-07: El sistema debe contar con un panel de administración para gestionar rifas, usuarios, reservas y pagos.

### **2.2 Requerimientos no funcionales**

RNF-01: El sistema debe ser accesible desde navegadores web modernos.

RNF-02: El sistema debe garantizar la consistencia de datos mediante integridad referencial.

RNF-03: El sistema debe permitir la trazabilidad de operaciones (reservas, pagos, rifas).

### **2.3 Restricciones**

- El sistema debe implementarse sobre MySQL o MariaDB.
- Los pagos estarán limitados a la pasarela Mercado Pago.
- Los tiempos de reserva estarán definidos en la configuración del sistema.

### **2.4 Trazabilidad**

Se elaborará una tabla que vincule cada requerimiento con los artefactos implementados, casos de prueba y evidencias (pendiente de completar).

## **3. Arquitectura y Diseño**

La arquitectura del sistema está centrada en una base de datos relacional. Se incluyen las siguientes entidades principales: usuarios, rifas, reservas, números de rifa y pagos. Cada entidad tiene relaciones bien definidas, garantizando la integridad referencial. El diseño se representará con un diagrama E-R (Modelo Entidad-Relación).

### **3.1 Herramientas utilizadas**

- Diagramas (UML, redes, bases de datos, etc.): Drawio. Realizaremos diagramas de estructura y comportamiento en UML.
- Descripción de componentes y relaciones: No definido

## 4. Plan de Implementación

### 4.1 Actividades a realizar

1. **Diseño de la base de datos:** creación de las tablas usuarios, rifas, numeros\_rifa, reservas, pagos.
2. **Desarrollo del backend:**
  - API para la gestión de rifas y compra de números.
  - Integración con servicio de correo electrónico (notificación automática al comprar número).
  - Autenticación de usuarios (correo y Google).
3. **Desarrollo del frontend:**
  - Interfaz de usuario para registro/login.
  - Panel de compra de números.
  - Panel de administración (gestión de rifas, ventas y reportes).
4. **Configuración del servidor y despliegue:**
  - Configuración en Docker (contenedores para backend, frontend, base de datos y servidor web).
  - Pruebas de despliegue en entorno de staging.
5. **Pruebas y corrección de errores.**
6. **Entrega final y documentación.**

### 4.2 Cronograma (por hito)

Hito	Actividad	Duración Estimada	Fecha
H1	Establecimiento del marco documental	1 semana	18/08/2025
H2	Gestión y actualización del documental	2 semanas	15/09/2025
H3	Documento final y entrega	2 semanas	3/11/2025

### 4.3 Responsables

- **Backend:** Rubén Perurena, Federico Bisio.
  - **Frontend:** Claudia Martínez.
  - **Base de datos y documentación:** Fabián González.
  - **Testing:** Federico Bisio.
  - **Validación:** Juan Daniel Pajarez.
  - **Supervisión y control de cambios:** Equipo completo en reuniones semanales.
- 

## 5. Plan de Verificación y Validación

### 5.1 Casos de prueba asociados a cada requerimiento

Requerimiento	Caso de prueba	Resultado esperado
Registro de usuario	Crear usuario con datos válidos	Usuario registrado en DB y sesión iniciada
Login	Iniciar sesión con Google o correo válido	Acceso concedido
Compra de número	Seleccionar número disponible y pagar	Número asignado al usuario y correo de confirmación enviado
Notificación por correo	Comprar ticket	Correo automático recibido en menos de 1 minuto
Panel de administración	Acceder como admin	Visualización de rifas, ventas y reportes

## **5.2 Herramientas de testing utilizadas**

- **PHPUnit: pruebas unitarias.**
- **Actualizar próximos.**

## **5.3 Criterios de aceptación**

- Todos los casos de prueba definidos deben ser ejecutados exitosamente.
- La aplicación debe funcionar en navegadores modernos (Chrome, Firefox, Edge).
- Las notificaciones por correo deben recibirse de forma confiable.
- El sistema debe soportar al menos 100 usuarios concurrentes en pruebas de carga.
- La interfaz debe ser intuitiva y sin errores críticos de usabilidad.

## **6. Repositorio**

Tanto la documentación como el proyecto lo subimos en un repositorio de GitHub. El enlace es el siguiente:

<https://github.com/Claudiamar1974/PROYECTO-FINAL-RIFAS-BLUE-CHERRY->