

**Nombre y Apellido:** Sebastian Celano

**Asignatura:** Proyecto de Desarrollo de Aplicaciones Web

**Carrera:** Ing. Informática

**Semestre:** 8vo

**Turno:** Mañana

**Periodo:**

2025.1

**Docente:** Silvia C. Vázquez N.

**Fecha:**

**18**/06/25

**Puntos Totales: 60**

**Puntos Correctos:**

1°P.: 2°P.:

3°P.: X E.F.:

E.F.:x

## Consigna:

Desarrolla e implementa un pequeño sistema web funcional que integre al menos las siguientes operaciones básicas:

1. Registrar información mediante formularios.
2. Visualizar los datos almacenados.
3. Editar y eliminar registros.

Ejemplos de temas permitidos (elige uno distinto a tu proyecto grupal):

* + Registro de mascotas para adopción.
  + Agenda de eventos personales.
  + Inventario doméstico.
  + Control de préstamos de libros entre amigos.
  + Lista de objetivos personales o metas académicas.

# Requisitos del examen:

## Parte 1 – Integración y Pruebas (50 puntos)

* + El sistema permite registrar, visualizar, editar y eliminar datos (20 pts)
  + Validaciones básicas en formularios (cliente y servidor) (10 pts)
  + Evita duplicados o errores básicos en la lógica (10 pts)
  + Capturas de pantalla que muestren el sistema en funcionamiento (10 pts)

## Parte 2 – Despliegue y Mantenimiento (10 puntos)

* + El sistema está publicado en una plataforma online (Netlify, Vercel, Render, etc.) Sólo se menciona cual sería la plataforma elegida (10 pts)

# Ejemplo de Desarrollo Individual – Registro de Mascotas para Adopción

## Descripción del Proyecto

Este proyecto consiste en un sistema web sencillo para registrar mascotas disponibles para adopción. Permite agregar mascotas con sus datos básicos, visualizarlas en una lista, editarlas o eliminarlas. Está desplegado online y cumple con validaciones básicas.

## Tecnologías Utilizadas

* + Frontend: HTML, CSS, JavaScript (Vanilla)
  + Backend: Node.js + Express
  + Base de Datos: MongoDB Atlas
  + Despliegue: Render.com **Parte 1 – Integración y Pruebas Funcionalidades:**

## Agregar Mascota

* + Formulario con campos: nombre, tipo (perro/gato), edad y contacto.
  + Validación de campos obligatorios en el cliente y servidor.

## Visualizar Mascotas

* + Lista dinámica que muestra las mascotas registradas en la base de datos.

## Editar Mascota

* + Botón "Editar" que carga los datos en el formulario y permite actualizar.

## Eliminar Mascota

* + Botón "Eliminar" con confirmación y borrado del registro.

## Pruebas Realizadas:

* Se probó el formulario dejando campos vacíos → muestra mensajes de error.
* Se probó enviar datos con edad negativa → rechaza con mensaje.
* Se probó conexión desconectando la base de datos → muestra error.
* Se corrigió un error donde los nombres con tilde causaban fallos.

## ● Parte 2 – Despliegue y Mantenimiento

**Plataforma Utilizada:** Se utilizó Render.com para el backend y Netlify para el frontend.

# **Sistema de Control de Préstamos de Libros.**

## Descripción del Proyecto:

Este proyecto consiste en un sistema web sencillo para registrar y gestionar préstamos de libros entre amigos. Permite agregar préstamos con información detallada, visualizarlos en una lista, editarlos o marcarlos como devueltos. El sistema incluye validaciones básicas para evitar errores.

## Tecnologías Utilizadas

* **Frontend**: HTML, CSS, JavaScript (Vanilla)
* **Backend**: Node.js + Express
* **Base de Datos**: MongoDB Atlas
* **Despliegue**: Render.com (backend) y Netlify (frontend)

## Parte 1 – Integración y Pruebas

### Funcionalidades implementadas:

1. **Agregar Préstamo de Libro**
   * Formulario con campos:
     + Título del libro (obligatorio)
     + Autor del libro (obligatorio)
     + Amigo que presta (obligatorio)
     + Fecha de préstamo (obligatorio, validación de formato)
     + Fecha límite de devolución
     + Estado (Prestado/Devuelto)
   * Validación de campos obligatorios en cliente y servidor
   * Validación de fechas (fecha de devolución no puede ser anterior al préstamo)
2. **Visualizar Préstamos**
   * Lista dinámica que muestra todos los préstamos registrados
   * Visualización diferenciada por estado (Prestado/Devuelto)
   * Opción de filtrar por estado
3. **Editar Préstamo**
   * Botón "Editar" que carga los datos en el formulario
   * Permite actualizar cualquier campo
   * Validaciones consistentes con el formulario de creación
4. **Marcar como Devuelto/Eliminar**
   * Botón "Marcar como Devuelto" que actualiza el estado
   * Botón "Eliminar" con confirmación para borrar el registro

### Pruebas Realizadas:

* Se probó el formulario dejando campos obligatorios vacíos → muestra mensajes de error
* Se probó enviar fechas inválidas (devolución anterior al préstamo) → rechaza con mensaje
* Se probó conexión con base de datos desconectada → muestra error adecuado
* Se verificó que no se puedan crear préstamos duplicados (mismo libro, mismo amigo, mismo día)
* Se comprobó que el cambio de estado a "Devuelto" registra la fecha actual automáticamente

## Parte 2 – Despliegue y Mantenimiento

**Plataforma elegida para despliegue**:

* Backend: Render.com
* Frontend: Netlify