

UNIVERSIDAD PRIVADA DOMINGO SAVIO

Ingeniería de Software II



UNIVERSIDAD PRIVADA
DOMINGO SAVIO

“PRÁCTICO Nro 2” *Reutilización*

ESTUDIANTE (S):

- Lisbeth Mileydi Barrientos Aparicio
- Daniel Ramiro Romero Romero
- Gabriel Elmer Quiroga Pereyra
- Nilton Mostacedo Fernandez

DOCENTE:

Ing Yanet Colque Alarcon

TARIJA, Mayo 2025

Sistema Centralizado para Gestión de Clínica Veterinaria (con enfoque en reutilización de software)

1. Introducción

En el desarrollo de sistemas de software modernos, la reutilización juega un rol fundamental en la mejora de la calidad, eficiencia y mantenimiento de los productos. Este caso de estudio plantea el diseño de una solución tecnológica para una clínica veterinaria que actualmente gestiona sus operaciones de manera manual. A través del uso de componentes reutilizables en React, se busca construir una base sólida para un sistema modular, flexible y escalable.

2. Descripción del problema

La clínica veterinaria presenta los siguientes desafíos:

- El historial médico de las mascotas se registra manualmente, con riesgo de pérdida o deterioro de la información.
- Las citas se agendan manualmente, provocando errores y conflictos de horario.
- El control de inventario se realiza de forma rudimentaria, causando desabastecimiento frecuente.

Estos problemas generan ineficiencia operativa y una experiencia negativa tanto para el personal como para los clientes. Es necesario un sistema digital centralizado y bien estructurado que permita automatizar y organizar estas tareas de forma eficiente.

3. Objetivo del caso de estudio

Diseñar una estrategia para el desarrollo de un sistema web utilizando React que aplique principios de reutilización de software en su arquitectura, con el fin de:

- Mejorar la calidad y consistencia de la interfaz.
- Reducir tiempos de desarrollo y costos de mantenimiento.
- Facilitar la extensión futura del sistema.

4. Estrategia de reutilización propuesta

El sistema se construirá en base a componentes reutilizables, hooks personalizados y patrones de diseño que promuevan la reutilización lógica y visual del código.

4.1 Componentes reutilizables (UI)

Se desarrollarán componentes genéricos reutilizables que se parametrizan mediante props:

- InputField: campo de texto reutilizable para nombre, edad, raza, stock, etc.
- Button: botón configurable con diferentes estilos y acciones.
- FormCard: contenedor estilizado para formularios.

- Table: tabla reutilizable para visualizar mascotas, productos, o citas.
- Modal: ventana emergente para registrar o editar elementos.

4.2 Hooks reutilizables (lógica)

- useForm: lógica para validación y manejo de formularios.
- useFetch: manejo estandarizado de peticiones al backend.
- useInventory: lógica común de inventario compartida por distintos módulos.

4.3 Vistas reutilizables

Las páginas de gestión de mascotas, citas e inventario estarán construidas sobre la misma estructura visual y funcional, basándose en los mismos componentes.

5. Aplicación concreta de la reutilización en el sistema

Módulo: Gestión de mascotas

- Se reutiliza FormCard con InputField para registrar mascotas.
- El componente Table muestra mascotas registradas con opciones para editar o eliminar.

Módulo: Agendamiento de citas

- Se emplean los mismos componentes de formulario y tabla, adaptando los campos requeridos.
- Modal se reutiliza para programar o modificar una cita.

Módulo: Inventario de productos

- Table se adapta para mostrar productos y stock.
- Modal se utiliza para ingresar nuevos productos, usando los mismos campos base.

6. Beneficios de la reutilización

6.1 Calidad del software

- La estandarización de componentes reduce errores y ambigüedades.
- Los componentes probados una vez pueden usarse en múltiples partes del sistema.

6.2 Eficiencia del desarrollo

- Se acelera el desarrollo al evitar la duplicación de código.
- Nuevos módulos pueden construirse rápidamente combinando componentes existentes.

6.3 Mantenimiento a largo plazo

- Si un componente necesita ser actualizado, el cambio se propaga automáticamente en todo el sistema.
- La modularidad permite localizar y corregir errores de manera más sencilla.

7. Herramientas utilizadas

- React: biblioteca principal para desarrollo del frontend.
- GitHub: para control de versiones, documentación del desarrollo y colaboración.
- Posiblemente otras herramientas complementarias como Vite o Create React App.

8. Plan de desarrollo

Etapas iniciales del proyecto:

- Diseño del esquema de la aplicación: módulos, rutas y estructura de carpetas.
- Identificación de componentes base reutilizables.
- Desarrollo de prototipos visuales sin funcionalidad (maquetado).
- Creación de repositorio Git y documentación del avance del diseño.
- Integración progresiva de lógica en componentes.
- Pruebas y validación del comportamiento reutilizable.

9. Conclusiones

Este caso de estudio propone una solución basada en el principio de reutilización para enfrentar los desafíos operativos de una clínica veterinaria. A través del uso de componentes reutilizables, se facilitará el desarrollo de un sistema más robusto, mantenible y escalable. La reutilización no solo permite ahorrar tiempo y esfuerzo, sino que también contribuye directamente a la calidad y sostenibilidad del software a largo plazo.