# **INFORME DEL PROYECTO: SERVICIOS TÉCNICOS (SPA REACT)**

NOMBRE: Bryan Belmar

CARRERA: ingeniería en informática

ASIGNATURA: Front End

FECHA: 03-07-2025

# Introducción

El presente informe describe el desarrollo de una aplicación web de servicios técnicos, construida como una SPA (Single Page Application) utilizando React. El objetivo principal es simular la presencia digital de una empresa dedicada a la reparación, mantenimiento y soporte técnico, ofreciendo una experiencia de usuario moderna, intuitiva y responsiva. El proyecto busca servir como ejemplo práctico de buenas prácticas en React, manejo de formularios, validación, persistencia local y diseño de interfaces.

# Desarrollo

## Diseño de la Solución

* **Framework:** React (funcional, hooks, componentes reutilizables)
* **Estilos:** CSS y Bootstrap 5
* **Persistencia:** localStorage para el CRUD de notas
* **Validación:** Validación de formularios en el cliente
* **Despliegue:** <https://perserver.netlify.app/>

# ESTRUCTURA DE COMPONENTES

* **Navbar (Navbar.jsx):**
  + Barra de navegación fija, permite cambiar de sección sin recargar la página.
  + Controla la sección activa con useState.
  + Usa botones
  + que actualizan el DOM dinámicamente.
* **Inicio (Inicio.jsx):**
  + Página principal con mensaje de bienvenida.
  + Tarjetas con imágenes y descripciones de servicios.
  + Solo presentación, sin lógica compleja.
* **Nosotros (Nosotros.jsx):**
  + Explica misión, visión y valores de la empresa.
  + Lista con íconos y textos explicativos.
* **Servicios (Servicios.jsx):**
  + Lista de servicios (reparaciones, mantenimiento, etc).
  + Presentación simple en tarjetas.
* **Contacto (Contacto.jsx):**
  + Formulario controlado con useState.
  + Validación de todos los campos requeridos.
  + Inputs: texto, email, teléfono, fecha, select, checkboxes.
  + Muestra errores personalizados.
  + Al enviar, descarga los datos como JSON.
  + Estados: form (datos), errores (validaciones).
  + Lógica: handleChange y handleSubmit.
* **Extra (Extra.jsx):**
  + CRUD de notas/tareas.
  + Persistencia en localStorage.
  + Permite editar y eliminar elementos.
  + Estados: items (lista), form (campos), modoEdicion y editIndex (edición).
* Si SÍ hay paquetes, continúa al siguiente paso.

1. **Buscar paquete más cercano**: El robot identifica el paquete más próximo a su ubicación actual.
2. **¿Camino al paquete?**: Verifica si existe una ruta para llegar al paquete.

* Si NO hay camino, termina con error.
* Si SÍ hay camino, continúa.

1. **Mover robot → Paquete**: El robot se desplaza hasta la ubicación del paquete.
2. **Recoger paquete**: El robot toma el paquete.
3. **¿Camino a entrega?**: Verifica si hay ruta al punto de entrega.

* Si NO hay camino, termina con error.
* Si SÍ hay camino, continúa.

1. **Mover robot → Entrega**: El robot lleva el paquete al punto de entrega.
2. **Dejar paquete**: El robot deposita el paquete.
3. **Volver a escanear**: El proceso se repite desde el paso 2.

# Conclusiones

El desarrollo de esta SPA con React permitió aplicar conceptos clave de desarrollo frontend moderno, como el uso de hooks, componentes reutilizables y validación de formularios. La estructura modular facilita la escalabilidad y el mantenimiento del código. Además, la integración de Bootstrap y CSS personalizado asegura una experiencia visual atractiva y adaptable a distintos dispositivos. Este proyecto es una excelente base para expandir funcionalidades, integrar servicios externos o servir como plantilla para otros emprendimientos digitales.

- Repositorio: https://github.com/bry4ns/spa-react

- Demo: https://perserver.netlify.app/