

Heurísticas de Nielsen

Análisis de Usabilidad: 10 Heurísticas de Nielsen

1. Visibilidad del Estado del Sistema

Evaluación: Bien cumplida. El sistema informa consistentemente a los usuarios sobre lo que está sucediendo.

- **Estados de Carga:** Componentes como `ReserveCalendar.jsx`, `RackSemanaL.jsx` y `ReportView.jsx` muestran indicadores de carga (spinners) y mensajes como "Cargando..." o "Calculando precio..." mientras se obtienen los datos.
- **Retroalimentación de Acciones:** En `ComprobanteList.jsx`, los botones muestran spinners durante operaciones como la descarga de PDF o el envío de correos. Además, utiliza notificaciones emergentes (toasts) para confirmar el éxito o fracaso de una acción.
- **Estado de Navegación:** El componente `Header.jsx` resalta la página activa, mostrando a los usuarios su ubicación actual dentro de la aplicación.

2. Coincidencia entre el Sistema y el Mundo Real

Evaluación: Bien cumplida. El sistema utiliza un lenguaje y conceptos familiares para el usuario.

- **Lenguaje y Terminología:** La interfaz utiliza español y términos específicos del dominio como "Rack Semanal", "Tarifas", "10 vueltas" y "Comprobantes", que son adecuados para la gestión de un negocio de karting.
- **Formato de Datos:** Los datos financieros en `ReportView.jsx` y `ComprobanteList.jsx` se formatean correctamente como Pesos Chilenos (CLP). Las fechas y horas también se presentan en un formato local (`es-ES`).
- **Convenciones del Mundo Real:** Los componentes `ReserveCalendar.jsx` y `RackSemanaL.jsx` imitan calendarios y planillas de reserva reales, lo que los hace intuitivos.

3. Control y Libertad del Usuario

Evaluación: Cumplida. Los usuarios pueden navegar y revertir acciones fácilmente.

- **Navegación Sencilla:** El encabezado persistente en `Header.jsx` permite a los usuarios salir de la vista actual y moverse a otra sección en cualquier momento.
- **Acciones Cancelables:** Los modales, como el de confirmación en `ComprobanteList.jsx`, incluyen botones de "Cancelar" o "Cerrar", proporcionando una

"salida de emergencia".

- **Acciones sin Compromiso:** El cálculo de precios en `ReserveForm.jsx` se realiza en tiempo real, permitiendo a los usuarios ver el costo antes de confirmar la creación de una reserva.

4. Consistencia y Estándares

Evaluación: Bien cumplida. La aplicación sigue las convenciones de la plataforma y las internas.

- **Framework de UI:** El proyecto utiliza Bootstrap de manera consistente para su diseño, componentes (botones, modales, alertas) y estilos en todos los archivos `.jsx`, asegurando una apariencia uniforme.
- **Consistencia Interna:** Las convenciones de nomenclatura para rutas, componentes y acciones son coherentes. Por ejemplo, las vistas de listado se nombran `...List.jsx` y los formularios `...Form.jsx`.
- **Diseño de Página:** La estructura general de la página es consistente, con un encabezado (`Header.jsx`) y un área de contenido principal definida en `App.jsx`.

5. Prevención de Errores

Evaluación: Bien cumplida. El diseño ayuda a prevenir errores comunes.

- **Validación de Entradas:** El formulario `ReserveForm.jsx` realiza validaciones del lado del cliente para campos obligatorios, formato de correo electrónico y RUT, previniendo el envío de datos inválidos.
- **Desactivación de Controles:** La hora de fin (`hora_fin`) en el formulario de reserva es un campo de solo lectura, calculado automáticamente por `reservationUtils.js`. Esto evita que los usuarios creen rangos de tiempo inconsistentes.
- **Selección Guiada:** En `RackSemana.jsx`, los usuarios no pueden seleccionar franjas horarias fuera del horario comercial, previniendo un error común de reserva.

6. Reconocimiento en lugar de Recuerdo

Evaluación: Cumplida. La interfaz minimiza la carga de memoria del usuario.

- **Navegación Visible:** Todas las secciones principales están siempre visibles en el `Header.jsx`, por lo que los usuarios no necesitan recordar las rutas.
- **Etiquetas Descriptivas:** En `ReserveCalendar.jsx`, los tipos de reserva están etiquetados de forma descriptiva (ej., "Normal (10 vueltas)"), no solo con identificadores.
- **Información Contextual:** Al ver los detalles de una reserva en `RackSemana.jsx`, toda la información relevante se muestra en un modal, por lo que el usuario no necesita recordarla de una pantalla anterior.

7. Flexibilidad y Eficiencia de Uso

Evaluación: Cumplida. El sistema se adapta tanto a usuarios novatos como a expertos.

- **Aceleradores:** Hacer clic en una franja horaria vacía en `RackSemana1.jsx` navega directamente al `ReserveForm.jsx` con la fecha y hora pre-rellenadas, agilizando el proceso.
- **Vistas Múltiples:** Los componentes de calendario (`ReserveCalendar.jsx` , `RackSemana1.jsx`) ofrecen múltiples vistas (Mes, Semana, Día, Agenda), permitiendo a los usuarios elegir la forma más eficiente de ver los datos.
- **Acciones Directas:** El componente `ReserveList.jsx` permite la eliminación directa desde la lista, una acción eficiente para usuarios expertos.

8. Diseño Estético y Minimalista

Evaluación: Cumplida. La interfaz es limpia y se centra en la información esencial.

- **UI Limpia:** La interfaz utiliza una paleta de colores simple y componentes estándar de Bootstrap, evitando el desorden visual. Archivos como `App.css` y `HomePage.css` muestran un enfoque en diseños limpios y espacios en blanco.
- **Sin Información Irrelevante:** Los diálogos y las vistas presentan solo la información necesaria para la tarea en cuestión. Por ejemplo, el componente `PriceBreakdown.jsx` solo muestra los componentes de precio relevantes.
- **Enfoque en el Contenido:** El diseño prioriza el contenido y la funcionalidad sobre la decoración excesiva.

9. Ayudar a los Usuarios a Reconocer, Diagnosticar y Recuperarse de Errores

Evaluación: Bien cumplida. Los mensajes de error son claros y constructivos.

- **Mensajes Claros:** Cuando ocurre un error, como en `ReserveCalendar.jsx` , se muestra un mensaje claro (ej., "No se pudo conectar al servidor backend. Usando datos de muestra.").
- **Lenguaje Sencillo:** Los mensajes de error en `ReserveForm.jsx` están en lenguaje sencillo (ej., "El nombre es obligatorio") y se muestran cerca del campo de entrada correspondiente.
- **Sin Códigos de Error:** La aplicación evita mostrar códigos de error crípticos al usuario.

10. Ayuda y Documentación

Evaluación: Parcialmente cumplida. Aunque la interfaz es intuitiva, no hay una sección de ayuda o documentación formal visible para el usuario final.

- **Ayuda Implícita:** La interfaz proporciona ayuda implícita a través de etiquetas claras, texto de ejemplo (placeholders como "RUT (Ej: 12345678-9)" en `ReserveForm.jsx`) y texto descriptivo en los botones.

- **Documentación:** El proyecto contiene un archivo `MEJORAS-DURACION.md` que documenta una característica específica. Esto es excelente para los desarrolladores, pero no es un sistema de ayuda en la aplicación para los usuarios finales.
- **Oportunidad de Mejora:** Una sección de ayuda simple orientada al usuario o información sobre herramientas (tooltips) en funciones complejas podría mejorar aún más