

Mini Guía Técnica

Rol: Frontend – UI / UX

Objetivo del rol

Diseñar y mejorar la **experiencia visual y de uso** del constructor de órdenes, **sin modificar la lógica de negocio ni el contrato de datos**.

Este rol se enfoca **exclusivamente en presentación, interacción visual y usabilidad**.

Contexto del proyecto

El sistema es un **constructor visual de órdenes** con drag & drop.

Actualmente existen:

- Paleta de campos
- Canvas de construcción
- Editor de propiedades
- Drag & drop funcional

La **lógica ya existe** y **no debe duplicarse ni reescribirse**.

Responsabilidades técnicas

Mejorar la experiencia visual del canvas

El canvas debe:

- verse claramente como un área de trabajo
- mostrar jerarquía visual entre campos
- indicar correctamente estados (hover, seleccionado, arrastrando)

Ejemplos:

- resaltar campo seleccionado
- mostrar sombra o borde al arrastrar
- separar visualmente campos

 **No cambiar cómo se crean o almacenan los campos.**

Diseñar estados visuales claros

Cada campo debe tener estados visibles:

Estado	Requisito
Normal	Campo visible, neutro
Hover	Indica que es interactivo
Seleccionado	Diferente al hover
Dragging	Retroalimentación clara
Deshabilitado	Si aplica en el futuro

3 Mejorar el FieldEditor (solo visual)

El editor de propiedades debe:

- ser claro y legible
- agrupar propiedades
- usar labels comprensibles
- evitar saturación visual

Ejemplo:


- secciones
- espaciados correctos
- títulos claros

 **No agregar lógica de actualización** (eso ya existe).

4 Usabilidad del drag & drop

Mejorar percepción visual:

- indicadores de zona válida de drop
- feedback cuando el usuario está sobre el canvas
- suavizar movimientos (transiciones)

 No tocar:

- handleDragEnd
- useDraggable
- useSortable

5 Consistencia visual

Debe existir coherencia entre:

- Paleta de campos
- Campos dentro del canvas
- Editor lateral

Ejemplos:

- tipografía consistente
- tamaños similares
- iconografía uniforme (si se usa)



Reglas de diseño

- No esconder funcionalidades
- No depender de animaciones para lógica
- La UI no debe afectar el JSON final
- El usuario debe entender qué pasa sin leer instrucciones



Archivos que puede modificar

- ✓ Estilos (CSS / inline styles / styled-components)
- ✓ Componentes visuales
- ✓ Layout
- ✓ Iconos
- ✓ Clases CSS



Archivos que NO debe modificar

- ✗ Estructura de fields
- ✗ props de los campos
- ✗ Drag & drop
- ✗ Contrato JSON
- ✗ Backend
- ✗ Lógica de actualización de campos

Checklist de validación (autoevaluación)

- El usuario identifica el campo seleccionado
- El canvas se siente como área de trabajo

- El editor es claro y ordenado
- No se rompió ninguna funcionalidad
- El JSON final no cambia
- No se agregó lógica innecesaria

🏠 5ª Frase guía para IA (copiar y pegar)

Estoy trabajando en el UI/UX de un constructor de formularios en React. No debo tocar lógica, drag & drop ni estructura de datos. Mi objetivo es mejorar estados visuales, claridad, jerarquía y experiencia de usuario sin afectar el contrato JSON.

🏠 Regla de oro

Si la UI se puede cambiar sin tocar el JSON, el trabajo está bien hecho.