

## Introducción

Este documento describe un ejercicio práctico para el desarrollo de componentes en React. Se trabajará con un conjunto de datos simulados que representan la proyección diaria de productos. El objetivo es construir dos componentes principales: un Grid de Proyección y un Resumen de datos.

## Datos Proporcionados

Se proporcionará un listado de datos con la siguiente estructura JSON:

```
Unset
{
  "CenterCode": "BCV",
  "Reference": "210001000004R20",
  "VisibleForecastedDate": "2025-03-21T00:00:00",
  "NetFlow": 1513.0,
  "GreenZone": 121.768285,
  "YellowZone": 121.768285,
  "RedZone": 194.829256,
  "MakeToOrder": 0
}
```

Cada objeto representa la información simulada de un producto en un día específico.

## Campos de Datos

- **CenterCode:** Código del centro donde se encuentra el producto.
- **Reference:** Código identificador del producto (Referencia).
- **VisibleForecastedDate:** Fecha simulada.
- **NetFlow:** Valor de inventario neto por día.
- **GreenZone:** Rango para pintar de color verde.
- **YellowZone:** Rango para pintar de color amarillo.
- **RedZone:** Rango para pintar de color rojo.
- **MakeToOrder:** Pedido de Abastecimiento por día.

## Ejercicio

### Componentes a Desarrollar

#### 1. Grid de Proyección

Construir un componente tipo tabla que:

- Muestre los datos organizados por producto y fecha.
- Cada celda muestra el valor de `MakeToOrder``.
- El color de fondo de cada celda se determina según el cálculo `NetFlow + MakeToOrder``, comparado con los rangos `RedZone``, `YellowZone`` y `GreenZone``.
  - **Rojo:** Si `1 <= (NetFlow + MakeToOrder) <= RedZone``.
  - **Amarillo:** Si `RedZone < (NetFlow + MakeToOrder) <= RedZone + YellowZone``.
  - **Verde:** Si `RedZone + YellowZone < (NetFlow + MakeToOrder) <= RedZone + YellowZone + GreenZone``.
  - **Negro:** Si `(NetFlow + MakeToOrder) == 0``.
  - **Azul:** Si `(NetFlow + MakeToOrder) > RedZone + YellowZone + GreenZone``.
- Al editar el valor de `MakeToOrder`` de una celda:
  - Se recalcula únicamente el color de esa celda.
  - No se modifiquen ni recalculen las celdas siguientes en la misma fila.
  - El nuevo color se determina en tiempo real basado en el nuevo valor ingresado y el `NetFlow`` correspondiente.
  - El Componente 2 (Resumen) se recalcula tomando en cuenta el nuevo valor de la celda.

#### Ejemplo:

Si `NetFlow = 100``, `MakeToOrder = 23``, `RedZone = 50``, `YellowZone = 55``, `GreenZone = 55``, el color de la celda es **verde**.

**Importante:** Cada celda del grid debe tener un `backgroundColor`` según la regla anterior. El dato que se muestra es `MakeToOrder``, pero el que define el color es `NetFlow + MakeToOrder``.

CenterCode	Reference	3/21/2025	3/22/2025	3/23/2025	3/24/2025	3/25/2025	3/26/2025	3/27/2025	3/28/2025	3/29/2025	3/30/2025	3/31/2025
BCV	210001000004R20	30	0	0	0	0	0	52	0	0	0	0
BCV	210001000006R20	0	0	0	0	0	28	0	0	0	0	36
BCV	210002000007R1	0	16	0	0	43	0	0	0	0	6	15
BCV	210002000008R5	10	0	0	0	0	0	0	0	2	0	0
BCV	210002000017R11	37	0	0	0	8	20	0	0	0	0	44
BCV	210001000001R30	39	0	0	0	0	0	0	6	0	0	0
BCV	210001000001R5	47	0	0	0	0	0	41	0	0	0	13
BCV	210001000002R1	9	0	29	0	0	28	34	0	0	0	0
BCV	210001000003R25	0	0	34	16	0	0	0	0	0	0	28
BCV	210001000007R2	16	0	0	0	0	43	0	0	0	0	22
BCV	210001000014R40	0	0	20	0	0	0	46	0	0	49	0
BCV	210001000019R5	13	0	15	0	0	0	15	13	0	35	0
BCV	210002000001R1	16	0	0	0	0	0	0	0	0	0	0
BCV	210002000003R1	0	0	0	0	54	19	15	0	0	0	0
BCV	210002000014R30	0	0	0	3	0	0	0	0	0	0	0
BCV	210002000015R5	0	10	0	0	0	0	41	0	0	0	0
BCV	210003000001R30	0	0	0	12	0	0	55	0	0	0	0
BCV	210003000001R5	32	9	26	55	0	0	0	0	0	0	41
BCV	210003000002R1	2	2	0	0	0	0	0	0	0	0	17
BCV	210004000001R1	0	0	0	40	0	0	0	35	53	0	0

## 2. Resumen

Crear un componente de resumen que muestre:

- Cantidad de filas por cada color en la columna seleccionada.
- Presentación en porcentaje de cada color.


### Funcionalidades:

- Calcular la cantidad de filas por cada color y su porcentaje perteneciente a la columna de la celda seleccionada.
- Actualizarse la información al seleccionar una celda en una columna diferente de fecha en el grid de proyección.
- Solo cargar información si la columna seleccionada es una de las columnas de proyección (con fechas).

Azul	0	0%
Verde	4	20%
Amarillo	4	20%
Rojo	7	35%
Negro	5	25%
Total	20	

## Resultado Final

Ambos componentes deben estar montados en una página y contar con las funcionalidades descritas para considerar el ejercicio exitoso.



CenterCode	Reference	3/21/2025	3/22/2025	3/23/2025	3/24/2025	3/25/2025	3/26/2025	3/27/2025	3/28/2025	3/29/2025	3/30/2025	3/31/2025
BCV	210001000004R20	30	0	0	0	0	0	52	0	0	0	0
BCV	210001000006R20	0	0	0	0	0	26	0	0	0	0	36
BCV	210002000007R1	0	15	0	0	43	0	0	0	0	0	15
BCV	210002000008R5	19	0	0	0	0	0	0	0	2	0	0
BCV	210002000017R11	37	0	0	0	8	20	0	0	0	0	44
BCV	210001000001R30	39	0	0	0	0	0	0	6	0	0	0
BCV	210001000001R5	47	0	0	0	0	41	0	0	0	0	13
BCV	210001000002R1	9	0	29	0	0	20	34	0	0	0	0
BCV	210001000003R25	0	0	34	16	0	0	0	0	0	0	20
BCV	210001000007R2	16	0	0	0	0	43	0	0	0	22	0
BCV	210001000014R40	0	0	20	0	0	0	46	0	0	49	0
BCV	210001000015R5	13	0	15	0	0	15	13	0	0	35	0
BCV	210002000001R1	15	0	0	0	0	0	0	0	0	0	0
BCV	210002000003R1	0	0	0	0	54	19	15	0	0	0	0
BCV	210002000014R30	0	0	0	0	3	0	0	0	0	0	0
BCV	210002000015R5	0	0	0	0	0	0	41	0	0	0	0
BCV	210003000001R30	0	0	0	12	0	0	55	0	0	0	0
BCV	210003000001R5	32	9	26	55	0	0	0	0	0	41	0
BCV	210003000002R1	2	2	0	0	0	0	0	0	0	17	0
BCV	210004000001R1	0	0	0	40	0	0	0	35	53	0	0

Azul	0	0%
Verde	4	20%
Amarillo	4	20%
Rojo	7	35%
Negro	5	25%
Total	20	

## Entrega

Al finalizar, enviar un archivo comprimido (`.zip`) al correo electrónico desde el cual se recibió este documento con el asunto "Prueba Desarrollo WA 2025". El archivo debe contener:

- El proyecto React completo.
- `package.json`.
- Código fuente.
- Un `README` básico con instrucciones para ejecutar el proyecto (`npm install + npm start`).

## Criterios de Aceptación y Evaluación

### 1. Diseño y Creatividad de la Interfaz Gráfica (35%)

- Diseño limpio y profesional.
- Buena UX y disposición.
- Uso adecuado de colores, espaciado y componentes visuales.
- Se permite el uso de librerías de terceros (ej., Material UI, Ant Design).

### 2. Funcionalidad y Exactitud (25%)

- Lógica de color correcta y conforme a las reglas.
- Edición funcional sin errores.
- Resumen reflejando correctamente los datos.

### 3. Rendimiento y Optimización (25%)

- Fluidez en la navegación.
- Uso eficiente del renderizado.
- Buen manejo del estado

### 4. Código y Buenas Prácticas (15%)

- Código modular y limpio.
- Buena organización del proyecto.
- Comentarios útiles y estructura clara.

Para ser considerado exitoso, el proyecto debe cumplir al menos el 80% de los requisitos funcionales.

**¡Muchos éxitos!**