

CRUD de Productos y Promociones

Este documento explica cómo funciona el sistema de gestión (crear, leer, actualizar, eliminar) de productos y promociones en la aplicación.

Índice

- [Productos](#)
 - [Endpoints de API](#)
 - [Componentes](#)
 - [Validaciones](#)
 - [Promociones](#)
 - [Endpoints de API](#)
 - [Componentes](#)
 - [Validaciones](#)
 - [Componentes compartidos](#)
-

Productos

Endpoints de Productos

Listar productos

```
GET /api/productos?bodegaId=BOD_001
```

Retorna todos los productos de una bodega.

Respuesta exitosa:

```
{
  "success": true,
  "ok": true,
  "data": [...],
  "total": 10
}
```

Crear producto

```
POST /api/productos
```

Cuerpo de la petición:

```
{
  "bodegaId": "BOD_001",
  "nombre": "Arroz Premium 5kg",
  "sku": "ARR-001",
  "categoria": "Granos",
  "precio": 15000,
  "stock": 100,
  "descripcion": "Arroz de alta calidad"
}
```

Obtener un producto

```
GET /api/productos/[id]
```

Retorna un producto específico por su ID.

Actualizar producto

```
PUT /api/productos/[id]
```

Cuerpo de la petición:

```
{
  "nombre": "Nuevo nombre",
  "precio": 18000
}
```

Solo se envían los campos que se quieren actualizar.

Eliminar producto

```
DELETE /api/productos/[id]
```

Elimina permanentemente un producto.

Componentes de Productos

- **ProductosClient.tsx** : Componente principal que muestra la lista de productos con opciones para crear, editar, duplicar y eliminar.
- **ProductForm.tsx** : Formulario reutilizable para crear y editar productos.

Validaciones de Productos

Campo	Regla
nombre	Obligatorio, 2-200 caracteres
sku	Obligatorio, máximo 50 caracteres
precio	Obligatorio, debe ser ≥ 0
stock	Debe ser ≥ 0
descripcion	Opcional, máximo 1000 caracteres

Promociones

Endpoints de Promociones

Listar promociones

```
GET /api/promociones?bodegaId=BOD_001
```

Retorna todas las promociones de una bodega con estado calculado dinámicamente.

Respuesta exitosa:

```
{
  "success": true,
  "ok": true,
  "data": [...],
  "total": 5,
  "activas": 2
}
```

Crear promoción

```
POST /api/promociones
```

Cuerpo de la petición:

```
{
  "bodegaId": "BOD_001",
  "nombre": "Descuento de Verano",
  "tipo": "porcentaje",
  "valor": 15,
  "fechaInicio": "2026-03-01T00:00:00Z",
  "fechaFin": "2026-03-31T23:59:59Z",
  "aplicaA": "categoria",
  "categoriaProductos": ["Granos", "Lácteos"]
}
```

Obtener una promoción

```
GET /api/promociones/[id]
```

Retorna una promoción específica con estado calculado.

Actualizar promoción

```
PUT /api/promociones/[id]
```

Cuerpo de la petición:

```
{
  "nombre": "Descuento Extendido",
  "valor": 20
}
```

Eliminar promoción

```
DELETE /api/promociones/[id]
```

Elimina permanentemente una promoción.

Componentes de Promociones

- **PromocionesClient.tsx** : Componente principal con pestañas para activas, programadas y finalizadas.
- **PromocionForm.tsx** : Formulario reutilizable para crear y editar promociones.

Validaciones de Promociones

Campo	Regla
nombre	Obligatorio, 3-100 caracteres
tipo	"porcentaje" o "monto_fijo"
valor	Obligatorio, > 0, si es porcentaje <= 100
fechaInicio	Obligatoria, fecha válida
fechaFin	Obligatoria, posterior a fechaInicio
aplicaA	"categoria", "producto" o "todos"

Estados de Promociones

El estado se calcula automáticamente basado en las fechas:

- **Activa**: La fecha actual está entre fechaInicio y fechaFin
- **Programada**: La fecha actual es anterior a fechaInicio
- **Finalizada**: La fecha actual es posterior a fechaFin

Componentes Compartidos

ConfirmDialog

Modal de confirmación para acciones destructivas (eliminar).

```
import ConfirmDialog from "@components/ConfirmDialog"

<ConfirmDialog
  isOpen={isOpen}
  onClose={() => setIsOpen(false)}
  onConfirm={handleDelete}
  title="Eliminar producto"
  message="¿Estás seguro? Esta acción no se puede deshacer."
  confirmText="Eliminar"
  cancelText="Cancelar"
  variant="danger"
  isLoading={isDeleting}
/>
```

Props:

Prop	Tipo	Descripción
isOpen	boolean	Controla visibilidad
onClose	() => void	Función para cerrar
onConfirm	() => void	Función al confirmar
title	string	Título del diálogo
message	string	Mensaje de confirmación
confirmText	string	Texto del botón confirmar
cancelText	string	Texto del botón cancelar
variant	"danger" "warning"	Estilo visual
isLoading	boolean	Estado de carga

Funciones de Validación

```
import { validarProducto, validarPromocion } from "@lib/validations"

// Validar producto
const resultado = validarProducto({
  nombre: "Mi producto",
  sku: "SKU-001",
  precio: 1000
}, true) // true = es creación

if (!resultado.esValido) {
  console.log(resultado.errores)
}
```

Permisos

- Para crear, editar o eliminar productos/promociones, el usuario debe tener uno de estos roles:
- **ADMIN:** Acceso completo
 - **BODEGUERO:** Solo para su bodega asignada

Ejemplos de Uso

Crear un producto

```
const response = await fetch('/api/productos', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    bodegaId: 'BOD_001',
    nombre: 'Nuevo Producto',
    sku: 'NP-001',
    categoria: 'General',
    precio: 5000,
    stock: 10,
  }),
})

const result = await response.json()
if (result.ok || result.success) {
  console.log('Producto creado:', result.data)
}
```

Actualizar una promoción

```
const response = await fetch(`/api/promociones/${promoId}`, {
  method: 'PUT',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    valor: 25, // Nuevo valor
  }),
})

const result = await response.json()
if (result.ok || result.success) {
  console.log('Promoción actualizada:', result.data)
}
```

Historial de Cambios

Fecha	Versión	Descripción
2026-02-23	1.0	Implementación inicial del CRUD completo

Archivos Relacionados

```
app/
├── api/
│   ├── productos/
│   │   ├── route.ts           # GET, POST
│   │   └── [id]/
│   │       └── route.ts       # GET, PUT, DELETE
│   └── promociones/
│       ├── route.ts          # GET, POST
│       └── [id]/
│           └── route.ts       # GET, PUT, DELETE
├── components/
│   ├── ProductosClient.tsx
│   ├── ProductForm.tsx
│   ├── PromocionesClient.tsx
│   ├── PromocionForm.tsx
│   └── ConfirmDialog.tsx
├── lib/
│   └── validations.ts
```