# 📘 Requerimientos Funcionales — Sistema de Budget y Forecast

🗓️ Fase: Relevamiento

Proyecto: Sistema de Presupuesto y Forecast para Empresa Agroindustrial

Semana actual: Revisión de reglas, modelado preliminar y definición de funcionalidades

## 1. Funcionalidades Principales por Perfil

### 👤 Comerciales

- Carga de datos presupuestarios por periodo y unidad (producto, zona, etc.).

- Edición directa en grilla estilo Excel (ag-grid, handsontable).

- Validación básica de campos requeridos.

- Visualización de forecast histórico.

- Envío de versión para revisión/validación.

### 🧮 Analistas Financieros

- Revisión y consolidación de datos ingresados por comerciales.

- Edición y ajustes con fórmulas (márgenes, ponderadores, etc.).

- Validaciones de consistencia (sumatorias, formatos).

- Comparativos entre versiones (actual, anteriores).

- Envío de versión a aprobación gerencial.

### 🧑‍💼 Gerentes

- Aprobación o rechazo de versiones.

- Visualización consolidada (por zona, unidad, etc.).

- Reportes de KPIs: ejecución vs forecast vs presupuesto.

- Comentarios y observaciones por versión.

## 2. Modelo Conceptual Preliminar

### Entidades Tentativas

* 🔹 Usuario
* - id
* - nombre
* - email
* - rol (comercial, analista, gerente)
* 🔹 ObjetoPresupuestario
* - id
* - zona
* - producto
* - unidad
* - tipo (venta, costo, etc.)
* 🔹 Periodo
* - id
* - año
* - mes
* 🔹 Valor
* - id
* - valor\_numérico
* - tipo\_valor (real, estimado)
* - comentarios
* - FK a: ObjetoPresupuestario, Periodo, Version
* 🔹 Version
* - id
* - nombre
* - estado (borrador, enviado, aprobado)
* - fecha\_creación
* - FK a: Usuario
* 🔹 Aprobacion
* - id
* - version\_id
* - usuario\_id
* - estado (pendiente, aprobado, rechazado)
* - fecha
* - comentarios

### Relaciones

- Un Usuario puede tener muchas Versiones.

- Una Version contiene múltiples Valores.

- Los Valores están vinculados a ObjetoPresupuestario y Periodo.

- Una Version puede tener múltiples Aprobaciones.

## 3. Consideraciones Técnicas

### Frontend (React 17 + Redux)

- Uso de ag-grid o handsontable para edición tipo Excel.

- Manejo de estados con Redux clásico por módulo (usuarios, versiones, etc.).

- Librerías complementarias: react-datepicker, react-select, highcharts, chart.js.

- Carga condicional de componentes por perfil.

- Validaciones dinámicas según reglas del backend.

### Backend (Node.js + PostgreSQL)

- Arquitectura modular: dominios por entidad clave.

- Middleware de autenticación/autorización por rol.

- Auditoría de cambios por versión.

- Consultas agregadas para dashboards de gerente.

- Reglas de negocio:

- - Validar que una versión tenga todos los valores cargados antes de enviarse.

- - Control de estados por perfil.

### General

- Carpeta /docs para documentación técnica.

- Documento actual: /docs/01\_requerimientos.md

- Preparar migración inicial desde Excel a base de datos con script.

- Posibilidad de control de versiones presupuestarias (v1, v2...).