

A continuación, presento la **Documentación Actualizada** lista para copiar y pegar en Google Docs. He marcado los cambios críticos con la etiqueta **[ACTUALIZADO]** o **[NUEVO]** en negrita para facilitar la trazabilidad.

# 1. DOCUMENTO DE DISEÑO DE JUEGO (GDD) - v1.1

**Nombre del Proyecto:** ElectroNova Inc.

**Versión:** 1.1 (Post-Implementación de Cadena de Suministro)

## Mecánica Central (Core Loop) **[ACTUALIZADO]**

El estudiante gestiona el flujo completo de valor. Ya no es producción "mágica", ahora requiere gestión de insumos:

1. **Compras (Procurement):** Adquisición de Materia Prima (MP) a costo de mercado.
2. **Producción (Manufacturing):** Transformación de MP en Producto Terminado (PT).  
**[REGLA:]** No se puede producir si no hay stock de MP.
3. **Venta (Market):** El algoritmo decide las ventas basándose en precio y marketing.
4. **Finanzas:** Recaudación y pago de costos operativos.

## Sistema de Mercado y Precios **[ACTUALIZADO]**

El mercado ya no acepta cualquier precio. Se ha implementado una "Barrera de Resistencia":

- **Precio Máximo Aceptable:** \$300.00.
- **Mecánica de Rechazo:** Si el precio supera los \$300, la demanda se contrae exponencialmente (Elasticidad Precio Demanda agresiva). Un precio de \$1,000 genera ventas cercanas a cero, incluso siendo monopolio.
- **Sensibilidad:** El algoritmo compara precios relativos con una sensibilidad cuadrática (
- Sensitivity=3
- *Sensitivity=3*
- ).

## Estructura de Costos (Hard Data) **[NUEVO]**

Para garantizar el balance financiero, se establecen los siguientes costos base:

- **Costo Materia Prima (MP):** \$15.00 / unidad.
- **Costo Manufactura:** \$35.00 / unidad.
- **Costo Total Unitario (COGS):** \$50.00.
- **Costo Almacenamiento:** \$0.20 / unidad / ronda.

## 2. TECHNICAL DESIGN DOCUMENT (TDD)

### - v1.1

#### Stack Tecnológico

- **Backend:** Node.js + Express + Mongoose.
- **Frontend:** React + Vite + Tailwind CSS + Recharts.
- **Real-time:** Socket.io.
- **Deployment:** Render (API) + Vercel (Client).

#### Esquemas de Datos (Data Modeling) [ACTUALIZADO]

##### Company Schema (Empresa):

codeJavaScript

```
{  
  financials: { cash: Decimal128, assets: Decimal128 },  
  // [NUEVO] Gestión de Materia Prima  
  rawMaterials: {  
    units: Number, // Stock físico de MP  
    averageCost: Decimal128  
  },  
  inventory: [ /* Array de Lotes FIFO */ ],  
  history: [ /* Array de snapshots por ronda */ ]  
}
```

##### Decision Schema (Decisión del Estudiante):

codeJavaScript

```
{  
  price: Decimal128,
```

```
marketing: Decimal128,  
// [NUEVO] Compras  
procurement: { units: Number },  
// [ACTUALIZADO] Producción vinculada a MP  
production: { units: Number }  
}
```

## Algoritmo de Procesamiento de Ronda (Logic Flow) [ACTUALIZADO]

El servidor ejecuta la ronda en 3 Fases Estrictas para garantizar coherencia contable:

### 1. Fase Operativa (Pre-Mercado):

- Se procesa la compra de MP (Caja
  - ↓
  - ↓
  - , MP
  - ↑
  - ↑
  - ).
  - Se valida la Producción: `ActualProduction = Min(DesiredProduction, AvailableMP)`
  - Se consume MP y se paga Manufactura (Caja
    - ↓
    - ↓
    - , MP
    - ↓
    - ↓
    - , Inventario PT
    - ↑
    - ↑
    - ).

2.

### 3. Fase de Mercado (Motor ECPCIM v1.1):

- Se calcula el Score Competitivo.
- [NUEVO] Se aplica penalización por "Barrera de Precio" (
- $P > 300$
- $P > 300$
- ).
- Se determinan las unidades vendidas  $\min(\text{Demanda}, \text{Stock})$ .

4.

#### 5. Fase Financiera (Cierre):

- Se ejecuta la venta (Caja
- ↑
- ↑
- , Inventario
- ↓
- ↓
- usando FIFO).
- Se cobran costos de almacenamiento y marketing.
- Se actualiza el Historial y se avanza la ronda global.

6.

---

## 3. BIBLIA DE ESTRATEGIA EDTECH (Guía del Estudiante)

### Guía de Operaciones [ACTUALIZADO]

- **La Regla de Oro de la Producción:** "Sin materiales, no hay fábrica". Antes de producir, debes asegurarte de comprar suficiente Materia Prima (MP). Si pides producir 1,000 unidades pero tienes 0 MP, tu fábrica se detendrá y perderás ventas.
- **Estructura de Costos:**
  - Comprar MP: \$15.00/u.
  - Producir: \$35.00/u.
  - **Costo Base Total:** \$50.00/u.

- *Tip:* Tu precio de venta debe cubrir estos \$50 + Marketing + Almacenaje para ser rentable.

•

## Alerta de Mercado: La Barrera de Precio [NUEVO]

El mercado de ElectroNova tiene un límite de tolerancia.

- **Zona Segura:** Precios entre \$50 y \$300.
- **Zona de Rechazo:** Precios superiores a **\$300**.
- **Consecuencia:** Si intentas especular con precios abusivos (ej. \$1,000), los clientes rechazarán tu producto masivamente, dejándote con el inventario lleno ("Hueso") y sin liquidez. ¡Cuidado con la avaricia!

## 4. PLAN DE DESARROLLO ACTUALIZADO (ROADMAP v1.2)

**Estado Actual:** Versión 1.1 (Estable en Producción).

**Meta:** Cerrar la brecha funcional completa (Logística y Multi-Plaza).

### FASE 1: Cadena de Suministro Básica (COMPLETADA ✓)

- Implementación de Materia Prima en DB.
- Lógica de restricción de producción (Sin MP no hay PT).
- Ajuste del algoritmo de mercado (Barrera de precio \$300).
- Dashboard actualizado con inputs de Compras.

### FASE 2: Logística y Tiempos (PRÓXIMO OBJETIVO)

*Objetivo:* Introducir el concepto de "Lead Time" (Tiempo de espera).

- **Tarea 2.1 (Backend):** Modificar el modelo `Company` para soportar `inTransit` (mercancía viajando).
- **Tarea 2.2 (Lógica):** Implementar tipos de envío en `DecisionModal`:
  - *Aéreo:* Llega en la misma ronda (Costo alto).
  - *Terrestre:* Llega en la siguiente ronda (Costo bajo).

•

- **Tarea 2.3 (Processor):** La MP comprada no debe sumarse inmediatamente al stock, sino entrar en una cola de espera según el transporte elegido.

## FASE 3: Segmentación de Mercado (Multi-Plaza)

*Objetivo: Romper el mercado único en 3 zonas con comportamientos distintos.*

- **Tarea 3.1 (Data):** Crear configuración de Plazas (Norte, Sur, Centro) en `GlobalConfig`.
  - *Norte:* Alta calidad, paga bien.
  - *Sur:* Sensible al precio.
- **Tarea 3.2 (Logística):** El estudiante debe decidir a qué plaza enviar su producto terminado.
- **Tarea 3.3 (Motor):** Ejecutar el algoritmo `marketEngine` tres veces por ronda (una por cada plaza) y sumar los resultados.

## FASE 4: Pulido y Eventos (Final)

- **Tarea 4.1:** Implementar eventos aleatorios (Huelgas que bloquean envíos terrestres).
- **Tarea 4.2:** Reportes financieros detallados (Estado de Resultados desglosado).