

# TECHNICAL DESIGN DOCUMENT (TDD)

## Stack Tecnológico:

- **Base de Datos:** MongoDB (Transaccional para inventarios).
- **Backend:** Node.js + Express.
- **Comunicación:** Socket.io para actualizaciones del cronómetro y ranking en tiempo real.
- **Frontend:** React (Hooks: `useContext` para el estado global del juego).

## Esquemas Clave (Data Modeling):

```
const CompanySchema = new Schema({
  name: String,
  cash: { type: Schema.Types.Decimal128, default: 500000.00 }, // Capital inicial propuesto
  loanInterest: { type: Schema.Types.Decimal128, default: 0.05 }, // Tasa admin
  inventoryMP: { alfa: Number, beta: Number, omega: Number },
  inventoryPT: [{ plaza: String, gama: String, units: Number, age: Number }],
  ethicsScore: { type: Number, min: 0, max: 100, default: 100 }
});
```

## Algoritmo de Procesamiento de Ronda:

1. **Validación de Saldo:** Si  $E_{egresos} > S_{saldo}$  \implies  $S_{saldo} = S_{saldo} - (E_{egresos} \cdot 1.05)$ .
2. Cálculo de Score Competitivo (CS):  
$$CS_i = w_1 \left( \frac{P_{min}}{P_i} \right) + w_2 (Q_i) + w_3 (D_i) + w_4 (M_i)$$
3. **Distribución de Mercado:** La demanda total de la plaza se reparte proporcionalmente al  $CS$  de cada empresa activa.

**Seguridad:** Implementación de CORS, Helmet.js y sanitización de inputs para prevenir Inyección NoSQL.