**🚀 COMANDOS INICIALES**

**Iniciar el proyecto completo:**

# Iniciar todos los servicios (frontend + backend + MongoDB)

sudo supervisorctl start all

# O individualmente:

sudo supervisorctl start frontend

sudo supervisorctl start backend

**Ver estado de los servicios:**

# Ver estado de todos los servicios

sudo supervisorctl status

# Reiniciar servicios si es necesario

sudo supervisorctl restart all

sudo supervisorctl restart frontend

sudo supervisorctl restart backend

**Logs para debugging:**

# Ver logs del backend

tail -f /var/log/supervisor/backend.\*.log

# Ver logs del frontend

tail -f /var/log/supervisor/frontend.\*.log

# Ver logs en tiempo real

sudo supervisorctl tail -f backend

sudo supervisorctl tail -f frontend

**📁 ESTRUCTURA DE CARPETAS IMPORTANTES**

**🎮 Frontend (React)**

/app/frontend/

├── src/

│ ├── components/

│ │ ├── InventoryPanel.jsx # 🎯 Componente principal del juego

│ │ ├── InventoryItem.jsx # Items arrastrables

│ │ ├── GridSlot.jsx # Celdas del grid

│ │ └── ui/ # Componentes shadcn

│ ├── data/

│ │ └── mock.js # 🎮 NIVELES Y CONFIGURACIÓN DEL JUEGO

│ ├── services/

│ │ └── gameAPI.js # 🔗 Conexión con backend

│ ├── utils/

│ │ └── itemUtils.js # Lógica de formas y rotaciones

│ └── App.js # App principal

├── package.json

└── .env # Variables de entorno frontend

**⚙️ Backend (FastAPI)**

/app/backend/

├── models/

│ └── player.py # 🏆 Modelos de datos (jugador, progreso, sesiones)

├── routes/

│ └── game.py # 🎯 API ENDPOINTS PRINCIPALES

├── database.py # 🔗 Conexión MongoDB

├── server.py # 🚀 Servidor principal

├── requirements.txt # Dependencias Python

└── .env # Variables de entorno backend

**🔧 ARCHIVOS CLAVE PARA MODIFICAR**

**🎮 Para ajustar niveles:**

# Editar niveles, obstáculos, items

nano /app/frontend/src/data/mock.js

**🎯 Para modificar lógica del juego:**

# Componente principal del juego

nano /app/frontend/src/components/InventoryPanel.jsx

**⚙️ Para ajustar API del backend:**

# Endpoints principales del juego

nano /app/backend/routes/game.py

**🏆 Para modificar sistema de puntuación:**

# Buscar función calculate\_stars() en:

nano /app/backend/routes/game.py

**🌐 ACCESO A LA APLICACIÓN**

**URLs importantes:**

# Juego principal

http://localhost:3000

# API del backend

http://localhost:8001/api

# Health check del backend

http://localhost:8001/api/health

# Docs interactivas de la API

http://localhost:8001/docs

**🛠️ COMANDOS DE DESARROLLO**

**Instalar nuevas dependencias:**

# Frontend (usar yarn, NO npm)

cd /app/frontend

yarn add nombre-paquete

# Backend

cd /app/backend

pip install nuevo-paquete

# Actualizar requirements.txt

pip freeze > requirements.txt

**Reiniciar después de cambios:**

# Después de cambios en .env o nuevas dependencias

sudo supervisorctl restart all

# Solo para cambios de código (hot reload automático)

# No necesitas reiniciar, se recarga solo

**📊 BASE DE DATOS**

**MongoDB:**

# Conectar a MongoDB (opcional, para debug)

mongo

# Ver bases de datos

show dbs

# Usar la base del juego

use puzzle\_inventory

# Ver colecciones

show collections

# Ver jugadores

db.players.find()

# Ver progreso

db.player\_progress.find()

**🚨 TROUBLESHOOTING COMÚN**

# Si el frontend no carga

sudo supervisorctl restart frontend

tail -f /var/log/supervisor/frontend.\*.log

# Si el backend no responde

sudo supervisorctl restart backend

tail -f /var/log/supervisor/backend.\*.log

# Si MongoDB no conecta

sudo systemctl status mongod

sudo systemctl restart mongod

**🎯 RESUMEN RÁPIDO**

**Para iniciar todo:**

sudo supervisorctl start all

**Para ver el juego:**

http://localhost:3000

**Archivos más importantes:**

* /app/frontend/src/data/mock.js - Configuración de niveles
* /app/frontend/src/components/InventoryPanel.jsx - Lógica principal
* /app/backend/routes/game.py - API del juego

¡Con esto tienes todo lo necesario para manejar **RESIDENT INVENTORY**!