NexOS — Gameplan de Agentes y Subagentes

Escenario: operar TODO NexOS con **1 solo número de WhatsApp hoy**, dejando la **arquitectura preparada** para añadir **múltiples números mañana** (por operatividad) **sin rediseñar**.

---

0. Principios NexOS (canónicos)

- **Zero-UI**: WhatsApp es canal, no producto.
- **Event-driven**: mensajes, audios, imágenes, fechas.
- **SSOT**: memoria centralizada (Airtable/DB).
- **Idempotencia**: eventos y acciones no se duplican.
- **Flags/Estados**: el sistema sabe siempre "qué ya se hizo".
- **Fix mínimo**: escalar = añadir canales, no lógica.

---

1. Arquitectura lógica

1.1 Orquestador NexOS (Cerebro)

Responsabilidades: - Identificar contacto y contexto. - Clasificar **intención**, **planeta** (producto) y **fase**. - Enrutar al **subagente** correcto. - Aplicar políticas globales (tono, permisos, horarios). - Ejecutar acciones con idempotencia. - Persistir estados y flags en SSOT.

1.2 Subagentes (Planetas)

- Uno por producto (RentOS, IngresOS, etc.).
- KB y reglas **solo** de su dominio.
- Proponen acciones; **no** deciden el canal.

  Regla: los subagentes **no hablan directo**; el Orquestador gobierna.

---

2. Gameplan por fases

Fase A — Base (1 número)

- Un único canal WhatsApp.
- Router activo.
- SSOT operativo.
- Idempotencia completa.

Fase B — Primer planeta (RentOS)

- Subagente RentOS v1.
- Estados y flags operativos.

Fase C — Nuevos planetas (IngresOS, etc.)

- Añadir subagente.
- Sin tocar router ni flujos base.

---

3. SSOT (esquema mínimo)

Contacts

- contact_id (PK)
- phone_e164 (unique)
- name
- org_id

Conversations

- conversation_id (PK)
- contact_id (FK)
- channel_id (FK)
- active_planet
- phase (lead | onboarding | ops | support)
- status
- locks (json)

Events (idempotencia)

- event_id (PK)
- conversation_id
- provider
- message_id
- type
- payload_hash
- processing_status

Actions (trazabilidad)

- action_id (PK)
- event_id (FK)
- planet
- action_type
- idempotency_key (unique)
- status

Channels (clave para escalar)

- channel_id (PK)
- provider
- phone_number_id
- display_name
- mode (brain | ops)
- default_planet

- active

---

4. Router (decisión)

Inputs: mensaje, historial, canal, señales lingüísticas.

Outputs: - planet - phase - intent - confidence - needs_handoff

Persistencia: si la confianza es alta → fijar `active_planet`.

---

5. Contrato Orquestador ↔ Subagente

Entrada: contexto + intent + mensaje + restricciones.

Salida: - assistant_message - actions[] - state_updates - flags

El Orquestador valida y ejecuta.

---

6. Idempotencia

- Evento repetido → se ignora.
- Acción con misma `idempotency_key` → no se repite.

Ejemplo conceptual: `send_message:conversation_id:template:yyyy-mm-dd`

---

7. Operar con 1 número (HOY)

- `Channels` tiene 1 registro (`mode=brain`).
- Separación lógica por `phase`, `locks` y `active_planet`.
- No mezclar comercial con operativa aunque sea el mismo número.

---

8. Escalar a multi-número (MAÑANA)

Qué cambia

- Añadir nuevos registros en `Channels` (p. ej. `mode=ops`).
- Mapear `phone_number_id` entrante → `channel_id`.

Qué NO cambia

- Router
- Subagentes
- SSOT

- Idempotencia

Regla de envío

- Si `phase=ops` y existe canal `ops` del planeta → enviar por ahí.
- Si no → canal `brain`.

---

9. Workflows n8n (lógico)

WF-01 Inbound WhatsApp 1. Webhook 2. Normalizar 3. Upsert Contact 4. Upsert Conversation 5. Insert Event 6. Router 7. Subagente 8. Registrar Actions 9. Ejecutar con idempotencia 10. Marcar Event processed

WF-02 Scheduler - Fechas → Event interno → Orquestador → Actions

---

10. Checklist

- [ ] SSOT creado
- [ ] Router estable
- [ ] Subagente RentOS
- [ ] Tabla Channels preparada
- [ ] Idempotencia verificada

---

11. Decisión final

- Empezar con **1 número** es correcto.
- El **cerebro es único**.
- Escalar = **añadir canales**, no productos.

---

12. Próximo paso

Definir planetas activos y fases exactas del router.