```
# Ninjō — Product Blueprint v1

## 1. Vision de producto

Ninjō es un AI-native operating system para revenue generation through conversational
channels.

No es:
- un chatbot
- un SaaS tradicional
- una agencia

Es:
> un sistema que opera, optimiza y escala funnels de revenue usando AI workers

### Core promise
Connect your channels → deploy an AI operator → generate and optimize revenue.

### Core loop
Deploy → Generate conversations → Convert → Learn → Improve → Repeat.

### Diferencial clave
- No es solo configuración, es ejecución continua
- No depende del usuario, propone, actúa y mejora
- Aprende cross-client, creando network effects en prompts y workflows

---

## 2. Principios de producto

1. Opinionated over flexible
   - Guiar al usuario > dejarlo hacer cualquier cosa

2. Progressive complexity
   - No mostrar todo upfront
   - Unlock por etapas

3. Runtime over setup
   - El valor está en operar, no en configurar

4. One interface
   - Un solo operador visible

5. Default to action
   - Siempre empujar al siguiente paso

---

## 3. Customer journey

### Resumen
```

```
Entry → Onboarding → Setup → Creation loop → Launch → First result → Ongoing loop →
Expansion

### Dos funnels de entrada
#### Self-serve funnel
- Ads / Landing / Referral
- Web mobile signup
- Crea credenciales
- Crea agente
- Activa
- Paga cuando corresponde
- Recibe notificaciones / handoffs
- Expande

#### High-ticket closing funnel
- Slack assisted sale
- Guided discovery
- Configuración asistida
- Activación con acompañamiento
- Luego puede migrar a self-serve si hace sentido

### Core journey unificado
Ambos funnels convergen en:
- onboarding
- activation
- expansion

---

## 4. Arquitectura

### High-level
Operator UX layer
↓
Journey Engine (state + rules)
↓
Cortex (skills) + subagents
↓
Integrations / Channels
↓
Memory & Data layer

### Componentes

#### 4.1 Operator
- Interfaz conversacional
- Mantiene contexto
- Guía al usuario
- Decide next step

#### 4.2 Journey Engine
- Estado actual del usuario
- Qué falta
- Qué puede hacer
- Qué no puede hacer
- Cuándo avanzar

Este es el cerebro de producto.

#### 4.3 Cortex
- Ejecución determinística
- Tasks repetibles
- Outputs estructurados

#### 4.4 Subagents
- Tareas abiertas
- Exploración
- Edge cases

#### 4.5 Memory / State
- User profile
- Business context
- Agent config
- Performance data
- Improvements

---

## 5. Journey state machine

### Estados
1. Entry
   - Objetivo: llevar a onboarding
   - Acción: explicar valor

2. Onboarding
   - Objetivo: entender goal
   - Inputs: use case, objetivo

3. Setup
   - Objetivo: crear draft del agente
   - Inputs: contexto mínimo

4. Creation loop
   - Objetivo: confianza + calidad
   - Acciones: simulation, role play, edits, suggestions

5. Launch
   - Objetivo: activar agente
   - Chequeos: canales, contexto, readiness

6. Live pre wow
   - Objetivo: llegar a primer resultado
   - Acciones: optimizaciones, sugerencias

7. Wow moment
   - Objetivo: mostrar valor claro
   - Outputs: booking, sale, qualified lead, correct handoff, engagement win
```

```
8. Ongoing
   - Objetivo: mantener engagement
   - Acciones: alerts, reports, improvements

9. Expansion
   - Objetivo: aumentar revenue
   - Acciones: más canales, más agentes, más autonomía

---

## 6. Tipos de agente

### 3 arquitecturas de autonomía
1. Hybrid AI / Human
   - Parte AI, parte humano
   - Contact limits
   - Handoff alerts

2. Progressive Human / AI
   - Arranca más humano
   - AI aprende de respuestas humanas
   - Reduce intervención progresivamente

3. Full AI end to end
   - AI hace todo
   - Solo excepciones humanas

### Casos de uso
Estos tipos pueden correr:
- High-ticket appointment setting
- Low-ticket closing
- Onboarding
- Community building / management

---

## 7. Wow moments

El wow no es “el agente está creado”.

El wow real es algo visible y atribuible:
- First booking
- First sale
- First qualified lead
- First correct handoff
- First engagement win

### Regla
El producto debe empujar al usuario al primer wow lo antes posible.

---

## 8. Setup y recomendaciones contextuales
```

```
### No forzar integraciones
CRM, Calendly, GHL, Fathom y Meta attribution no son obligatorios para onboarding.

### Recomendación contextual
Si el cliente ya tiene:
- link de agenda
- booking flow
- CRM
- GHL
- Fathom
- Meta ads

el sistema recomienda integrarlo para mejorar atribución, seguimiento y visibilidad del
funnel.

### Dónde aparece
- Durante setup, como recomendación
- Después del primer valor, en alerts y follow-ups

### Follow-up lógico
- por lead individual
- o agregado con varios leads
- preguntando si logró agendar
- ofreciendo integrar calendario / CRM si ya hay tracción

---

## 9. Ongoing value

### Reportes proactivos
- Daily summary
- Hot leads alerts
- Objection insights
- Attribution reports
- Funnel health
- Auto-improvements
- Context recommendations

### Alertas
- Lead caliente sin respuesta
- Link enviado, sin booking
- Handoff necesario
- Conversación estancada
- Caída de conversión
- Nuevo patrón de objeción
- Señal de satisfacción

### Loop de mejora
1. Detectar señal
2. Recomendar acción
3. Aplicar ajuste
4. Medir impacto
5. Repetir
```

~~~
---

## 10. Flowchart del customer journey

```mermaid
flowchart TD
  A1[Self-serve funnel: Ads / Landing / Referral] --> B1[Web mobile signup]
  A2[High-ticket closing funnel: Slack assisted sale] --> B2[Guided discovery in Slack]

  B1 --> C[Core journey]
  B2 --> C

  C --> D[Create credentials / account]
  D --> E[Choose use case]
  E --> E1[High-ticket appointment setting]
  E --> E2[Low-ticket closing]
  E --> E3[Onboarding]
  E --> E4[Community building / management]

  E1 --> F[Choose agent type]
  E2 --> F
  E3 --> F
  E4 --> F

  F --> F1[Hybrid AI / Human]
  F --> F2[Progressive Human / AI]
  F --> F3[Full AI end to end]

  F1 --> G[Import context]
  F2 --> G
  F3 --> G

  G --> H[Connect channels]
  H --> H1[IG]
  H --> H2[WhatsApp]
  H --> H3[Slack]
  H --> H4[Telegram]
  H --> H5[Ninjo mobile app]

  G --> I[Optional recommendations]
  I --> I1[CRM]
  I --> I2[Calendly]
  I --> I3[GHL]
  I --> I4[Fathom]
  I --> I5[Meta ads / attribution]

  H --> J[Define rules]
  I --> J

  J --> K[Simulation / test]
  K --> L{Simulation OK?}
  L -->|No| J
  L -->|Yes| M[Activation gate]
~~~

~~~~
  M --> N{Self-serve?}
  N -->|Yes| N1[Paywall / activation payment]
  N -->|No| N2[Assisted activation in Slack]

  N1 --> O[Launch live agent]
  N2 --> O

  O --> P[First wow moment]
  P --> P1[First booking]
  P --> P2[First sale]
  P --> P3[First qualified lead]
  P --> P4[First successful handoff]
  P --> P5[First engagement win]

  P --> Q[Ongoing value]

  Q --> Q1[Daily summary]
  Q --> Q2[Hot leads alerts]
  Q --> Q3[Objection insights]
  Q --> Q4[Attribution reports]
  Q --> Q5[Funnel health]
  Q --> Q6[Auto-improvements]

  Q2 --> R{Lead has link sent?}
  R -->|Yes| R1[Follow-up: did they book?]
  R -->|No| R2[No booking follow-up]

  R1 --> R3[Individual follow-up]
  R1 --> R4[Aggregate follow-up]
  R1 --> R5[Suggest CRM / Calendly / GHL integration]

  Q --> S[Expansion]
  S --> S1[Add more context]
  S --> S2[Add more channels]
  S --> S3[Add more agents]
  S --> S4[Increase autonomy]
  S --> S5[Add new use case]
```

---

## 11. Operator prompt for end to end testing

### Operator role
The Operator is not a chatbot. It is the product layer that guides the user end to end
through creating, launching, and operating an AI agent that generates revenue through
conversations.

### Core behavior
- Always know the current stage
- Move the user to the next stage
- Ask 1 to 2 questions at a time
- Suggest defaults when possible
~~~~

```
- Prefer action over explanation
- Use Cortex when a task can be executed deterministically
- Spawn subagents for edge cases or open exploration

### Stage rules

#### Onboarding
- Understand the user's goal
- Recommend a use case
- Recommend an agent type

#### Setup
- Collect minimal context
- Offer, audience, goal
- Do not ask unnecessary questions

#### Agent creation
- Allow simulation, role play, direct edits
- If simulation fails, suggest improvements automatically

#### Launch
- Ensure readiness: channels, context, behavior rules
- Guide activation

#### Live pre wow
- Monitor early activity
- Suggest improvements if no results

#### Wow moment
- Highlight the result
- Explain why it happened
- Reinforce value

#### Ongoing
- Provide daily summaries, hot leads alerts, insights
- Suggest actions continuously

#### Expansion
- Recommend more channels, more agents, higher autonomy, new use cases

### Execution policy
- If a task can be executed via Cortex, use it
- If not, spawn a specialized agent
- Always summarize results clearly

### Output style
- Concise
- Structured
- Proactive
- Always end with a next step or question

### Goal
The Operator does not answer questions as an endpoint
The Operator guides deployment, generates results, and improves them continuously
```

```
---

## 12. What is still missing

1. North Star Metric
   - Suggested: revenue influenced
   - Secondary: booked calls, qualified leads, conversion rate

2. Activation definition
   - Example: agent live plus first conversations or first result

3. Guardrails
   - What the operator can do alone
   - When it must ask for confirmation
   - When it must escalate to a human

4. Improvement policy
   - What it can auto-improve
   - What it can suggest
   - What it must block

5. UX contract
   - Response format
   - Tone
   - Output standards

6. Data model
   - Agents, conversations, results, improvements

7. Feedback loops
   - Prompts, workflows, recommendations, attribution

---

## 13. Summary

Ninjō is being designed as an operating system for revenue on social channels.

The product is not just setup.
It is:
- guided creation
- controlled activation
- measurable wow moments
- ongoing optimization
- expansion with more context

The core design rule is simple:

> connect → create → simulate → launch → show value → learn → optimize → scale
```