

# MVP Definition: Project Showcase App

Autor: Esteban de Santiago García

Versión: 1.1

**Última actualización: 2025-10-20**

---

## 1. Objetivo del Producto

---

Crear una aplicación pública para mostrar, de forma clara y atractiva, los agentes de IA desarrollados y los equipos que los construyeron. La app permitirá explorar agentes, conocer al equipo y su desarrollador principal, y entender el propósito y capacidades de cada agente.

## 2. Visión de Alto Nivel

---

Una experiencia intuitiva (Flutter) que consume datos desde una API REST (Spring Boot) y una BaaS (Supabase), centrada en la visualización de agentes y equipos, con navegación fluida y contenido estructurado proveniente de especificaciones de agente.

---

## 3. Alcance del MVP (Core Features)

---

Prioriza el valor y la transparencia, guiado por el Agente de Product Owner (principios: value\_maximization, clarity\_and\_transparency).

### Épica 1: Visualización de Agentes

- Feature: Pantalla principal con lista/galería de agentes (nombre, breve descripción, equipo, chip de capacidades clave).
- Feature: Detalle de agente con información completa (nombre, descripción, rol, capacidades, equipo, desarrollador principal, enlaces a especificación y repositorio si existen).

### Criterios de Aceptación (CA):

- CA1: Al abrir la app se muestra la lista de agentes con al menos nombre y equipo.
- CA2: Al pulsar un agente se navega a su detalle y se visualizan descripción, rol y capacidades.
- CA3: Si hay un enlace a especificación (archivo XML/MD), se muestra un botón para abrirlo (en navegador/app WebView).

## Épica 2: Visualización de Equipos y Desarrolladores

- Feature: Pantalla con lista de equipos participantes.
- Feature: Detalle de equipo con miembros.
- Feature: Desde detalle de agente, se puede abrir el perfil básico del desarrollador principal (nombre y rol).

### Criterios de Aceptación (CA):

- CA4: La lista de equipos muestra nombre y cantidad de agentes relacionados.
- CA5: El detalle de equipo lista miembros con nombre y rol.
- CA6: Desde el detalle de agente existe un enlace al desarrollador principal que abre su ficha básica.

## Fuera de Alcance (MVP)

- Métricas de desempeño/asistencia/progreso por desarrollador.
- Visualización de grabaciones de sesiones.
- Explorador de carpetas de integrantes.
- Autenticación.

Estas capacidades pueden considerarse Post-MVP y se modelan como adjuntos/enlaces no visibles en MVP.

---

## 4. Stack Tecnológico (Confirmado)

- Frontend: Flutter
- Backend: Java con Spring Boot (API REST, read-only en MVP)
- BaaS/DB: Supabase

- Comunicación: API REST (JSON)
- 

## 5. Modelo de Datos (Supabase) — Propuesta MVP

---

Tablas base:

- agents
  - id (uuid)
  - name (text)
  - description (text)
  - role (text)
  - capabilities (text[])
  - team\_id (uuid, fk [teams.id](#))
  - owner\_id (uuid, fk [members.id](#)) — desarrollador principal
  - repo\_url (text, nullable)
  - spec\_url (text, nullable) — enlace a XML/MD de especificación del agente
  - created\_at (timestampz, default now())
- teams
  - id (uuid)
  - name (text)
  - description (text, nullable)
- members
  - id (uuid)
  - name (text)
  - role (text)
  - avatar\_url (text, nullable)
- team\_members (para equipos > 1 miembro)
  - team\_id (uuid, fk [teams.id](#))
  - member\_id (uuid, fk [members.id](#))

- attachments (post-MVP utilizable más tarde)
  - id (uuid)
  - agent\_id (uuid, fk [agents.id](#))
  - type (enum: 'spec' | 'video' | 'doc' | 'url')
  - title (text)
  - url (text)
  - visible\_in\_mvp (boolean, default false)

Consultas frecuentes:

- Listar agentes con join a team y owner (para mostrar en la lista y detalle).
  - Listar equipos con conteo de agentes.
- 

## 6. API REST (Spring Boot) — Contrato MVP (Read-Only)

---

- GET /api/agents
  - 200 OK: [{ id, name, description (short), team: { id, name }, owner: { id, name, role } }]
- GET /api/agents/{id}
  - 200 OK: { id, name, description, role, capabilities, team, owner, repoUrl, specUrl }
- GET /api/teams
  - 200 OK: [{ id, name, agentsCount }]
- GET /api/teams/{id}
  - 200 OK: { id, name, description, members: [{ id, name, role }], agents: [{ id, name }] }

Notas:

- Filtrado/búsqueda simple (query params) opcional si entra en capacidad del Sprint.
  - Administración (POST/PUT/DELETE) queda fuera del MVP.
-

## 7. Flutter — Navegación y Pantallas

---

Pantallas MVP:

- HomeAgentesPage (lista/galería) — consumo de GET /api/agents
- AgenteDetallePage — consumo de GET /api/agents/{id}
- EquiposPage — consumo de GET /api/teams
- EquipoDetallePage — consumo de GET /api/teams/{id}
- DesarrolladorDetalleSheet/Dialog — datos desde agent.owner o desde members por id

UX Consideraciones:

- Búsqueda simple por nombre de agente (opcional).
- Chips de capacidades en detalle de agente.
- Botón “Ver especificación” si specUrl existe.

---

## 8. Principios y Gobernanza (Agente Product Owner)

---

Basados en el agente product-owner-agent:

- Value Maximization (crítico): priorizar épicas/features que impacten la visualización clara de agentes y equipos.
- Clarity & Transparency (alta): backlog visible, historias claras y testables, contrato de API documentado.
- DoR (Definition of Ready): historia con propósito, datos de ejemplo, criterios de aceptación, dependencias identificadas.
- DoD (Definition of Done): UI implementada, integración con API, estados vacíos/errores, pruebas básicas pasadas, documentación de endpoint y pantallas actualizada.
- Cadencia: Refinamiento semanal; revisión de incremento al final del Sprint con stakeholders.

---

## 9. Métricas de Éxito (MVP)

---

- Adopción: visitantes únicos (si Web) o instalaciones/usuarios activos (si móvil) primer mes.
  - Engagement: vistas de detalle de agente, tiempo medio de sesión.
  - Feedback cualitativo de stakeholders sobre claridad y utilidad.
- 

## 10. Roadmap y Hitos (tentativo)

---

- Sprint 1: Arquitectura, esquema Supabase, contrato de API, esqueleto Flutter (Home + navegación).
- Sprint 2: Listado y detalle de agentes; estilos base.
- Sprint 3: Listado y detalle de equipos; ficha básica de desarrollador; mejoras UX.
- Sprint 4: Pruebas, hardening, documentación y despliegue.

Post-MVP:

- Adjuntos visibles (grabaciones, docs), explorador de materiales, métricas por desarrollador (con revisión de privacidad), autenticación básica si se requiere.