**Documento Maestro de Especificación Funcional y Arquitectura - Antigravity**

**1. Resumen Ejecutivo del Sistema**

**Nombre del Proyecto:** Antigravity (Gofo Knowledge Center)

**Propósito:** Antigravity es un **Centro de Conocimiento Centralizado** diseñado para optimizar el flujo de trabajo de los agentes de soporte (MQA & WAP System). Su objetivo principal es proporcionar acceso rápido y estructurado a:

1. **Casos de Soporte (MQA):** Guiones (scripts) oficiales, condiciones y procedimientos para manejar incidencias de clientes.
2. **Prompts de WhatsApp (WAP):** Plantillas de mensajes predefinidos (bilingües) para una comunicación rápida y estandarizada.

El sistema incluye un **Panel de Administración** para gestionar usuarios, contenidos y visualizar métricas de uso, asegurando que los agentes siempre tengan acceso a la información más actualizada.

**Stack Tecnológico:**

- **Framework Principal:** Next.js 16.0.7 (App Router).
- **Lenguaje:** TypeScript.
- **Base de Datos:** PostgreSQL (gestionada vía Prisma ORM 5.22.0).
- **Estilos:** Tailwind CSS 3.4.18 + Lucide React (Iconos).
- **Autenticación:** Sistema personalizado basado en Cookies y JWT (simulado/simplificado para este MVP).

---

**2. Arquitectura y Estructura del Proyecto**

El proyecto sigue una arquitectura moderna basada en **Next.js App Router**, separando claramente las responsabilidades entre el servidor y el cliente.

Estructura de Directorios Clave

- **src/app**: Contiene las rutas de la aplicación (Pages). Sigue el patrón de enrutamiento basado en carpetas de Next.js.
    - src/app/api: Endpoints de backend (API Routes) para operaciones CRUD y lógica de negocio.
    - src/app/admin: Rutas protegidas para administradores.
    - src/app/mqa y src/app/prompts: Rutas principales para los agentes.
- **src/components**: Componentes de UI reutilizables (Modales, Tablas, Navegación).
- **src/services**: Capa de abstracción para llamadas a la API (api.ts). Centraliza la lógica de "fetch" para mantener los componentes limpios.
- **src/lib**: Utilidades del sistema (Cliente de Prisma, Manejador de Errores de API).

- **prisma**: Definición del esquema de base de datos (schema.prisma) y migraciones.

Patrones de Diseño

1. **Server Components vs Client Components:**
    - Las **Páginas (page.tsx)** son mayoritariamente *Server Components*. Se encargan de obtener datos directamente de la base de datos (vía Prisma) para un renderizado inicial rápido y SEO-friendly.
    - Los **Componentes Interactivos** (Tablas, Buscadores, Modales) son *Client Components* ('use client') que manejan el estado y la interacción del usuario.
2. **Service Layer:** Se utiliza un patrón de servicio (src/services/api.ts) para desacoplar la lógica de red de los componentes de UI, facilitando el mantenimiento y las pruebas.
3. **Centralized Error Handling:** Un wrapper (apiHandler) envuelve todas las rutas de API para estandarizar las respuestas de error y el manejo de excepciones.

---

**3. Modelo de Datos (Entidades de Negocio)**

El modelo de datos está definido en schema.prisma y representa el núcleo del negocio.

Entidades Principales

1. **User (Usuario)**
    - **Descripción:** Representa a las personas que acceden al sistema.
    - **Atributos Clave:**
        - role: Define si es 'admin' (gestor) o 'agent' (usuario final).
        - email y password: Credenciales de acceso.
        - mustChangePassword: Bandera de seguridad para forzar cambio de clave en el primer inicio.
        - preferences: Configuración personal (tema, idioma, vista predeterminada).
    - **Relaciones:** Tiene muchas Interactions (historial de uso).
2. **SupportCase (Caso de Soporte)**
    - **Descripción:** Un escenario de soporte específico (ej: "Cliente reporta tarjeta bloqueada").
    - **Atributos Clave:**
        - title_es / title_en: Título en español e inglés.
        - script_official: El guion exacto que el agente debe leer.
        - condition: Criterios que deben cumplirse para aplicar este caso.
        - crm_codes: Códigos internos para registrar en el CRM.

- **Relaciones:** Puede recomendar múltiples WhatsappPrompts.

3. **WhatsappPrompt (Prompt de WhatsApp)**
    - **Descripción:** Una plantilla de mensaje de texto para enviar por chat.
    - **Atributos Clave:**
        - content_es / content_en: El texto del mensaje.
        - category: Clasificación (ej: "Saludo", "Cierre", "Error").
    - **Relaciones:** Puede estar asociado a varios SupportCases.
4. **Interaction (Interacción)**
    - **Descripción:** Registro de auditoría que rastrea cuándo un usuario utiliza un recurso.
    - **Atributos Clave:**
        - type: Si se usó un 'case' o un 'prompt'.
        - userId: Quién lo usó.
        - timestamp: Cuándo ocurrió.

---

**4. Desglose Funcional por Módulos**

A. Módulo de Espacio de Trabajo (Agent Workspace)

- **Ruta:** /mqa
- **Funcionalidad:**
    - **Visualización de Casos:** Lista de tarjetas con casos de soporte.
    - **Búsqueda Inteligente:** Filtrado por título, categoría o palabras clave.
    - **Vista Detallada:** Al hacer clic, despliega scripts, condiciones y prompts recomendados.
    - **Copiado Rápido:** Botones para copiar scripts al portapapeles.
- **Lógica de Fondo:** Al copiar o expandir un caso, se dispara una llamada a /api/usage para registrar una Interaction, alimentando las analíticas.

B. Módulo de Prompts (WAP System)

- **Ruta:** /prompts
- **Funcionalidad:**
    - **Galería de Plantillas:** Visualización de mensajes de WhatsApp en formato grilla o lista.
    - **Filtrado:** Por categoría (Saludo, Despedida, etc.).
    - **Copiado One-Click:** Copia el texto en inglés o español.

- **Lógica de Fondo:** Similar al Workspace, cada copiado registra una interacción para medir qué templates son los más usados.

C. Módulo de Administración de Usuarios

- **Ruta:** /admin/users
- **Funcionalidad:**
    - **Gestión CRUD:** Crear, Editar y Eliminar usuarios.
    - **Control de Acceso:** Asignar roles (Admin/Agent).
    - **Reset de Contraseña:** El admin asigna contraseñas temporales.
- **Lógica de Fondo:**
    - Al crear un usuario, el sistema asigna una contraseña temporal basada en el rol ('admin' o 'agent') y activa la bandera mustChangePassword.

D. Módulo de Gestión de Contenidos (CMS)

- **Rutas:** /admin/cases, /admin/prompts
- **Funcionalidad:**
    - Permite a los administradores mantener actualizada la base de conocimiento.
    - Formularios para redactar scripts y templates en dos idiomas.

E. Módulo de Analíticas

- **Ruta:** /admin/analytics
- **Funcionalidad:**
    - Dashboard con métricas de desempeño.
    - Ranking de usuarios más activos.
    - Recursos (Casos/Prompts) más utilizados.

---

**5. Flujos de Usuario Críticos (User Journeys)**

1. Proceso de Onboarding de Nuevo Agente

1. **Admin** accede a /admin/users y hace clic en "Nuevo Usuario".
2. **Admin** llena los datos básicos (Nombre, Email, Rol: Agent).
3. **Sistema** crea el usuario con contraseña temporal "agent" y marca mustChangePassword = true.
4. **Agente** intenta iniciar sesión por primera vez con la clave temporal.
5. **Sistema** detecta la bandera y **bloquea el acceso**, mostrando el modal "Cambio de Contraseña Requerido".
6. **Agente** ingresa una nueva contraseña segura (validada por medidor de fuerza).

7. **Sistema** actualiza la credencial, limpia la bandera y permite el acceso al Workspace.

2. Flujo de Atención de Soporte (MQA)

1. **Agente** recibe una llamada/chat de un cliente con un problema.
2. **Agente** navega a /mqa y usa la barra de búsqueda (ej: escribe "tarjeta").
3. **Sistema** filtra los casos relevantes en tiempo real.
4. **Agente** selecciona el caso correcto. Se despliega el "Script Oficial".
5. **Agente** lee el script al cliente. Si necesita enviar un WhatsApp, ve la sección "Prompts Recomendados" en la misma tarjeta.
6. **Agente** hace clic en copiar un prompt.
7. **Sistema** copia el texto al portapapeles y registra silenciosamente una Interaction en la base de datos.

---

**6. Glosario de Términos**

- **MQA (Monitoring Quality Assurance):** En este contexto, se refiere al sistema de gestión de calidad y guiones de soporte.
- **WAP (WhatsApp Prompts):** Sistema de plantillas predefinidas para mensajería instantánea.
- **Script:** Texto literal que el agente debe verbalizar o escribir al cliente.
- **Prompt:** Plantilla de mensaje corta y reutilizable.
- **Interaction (Interacción):** La unidad de medida de uso del sistema. Cada vez que un agente "usa" un recurso (lo copia o lo abre), cuenta como una interacción.
- **Server Action / API Handler:** Funciones que se ejecutan en el servidor para procesar datos de manera segura (ej: guardar en base de datos) antes de devolver una respuesta al navegador.