

ClueQuest Mind Map – Blueprint Profundo

Este documento expande el mapa mental original de ClueQuest con una numeración jerárquica (1.1, 1.1.1, 1.1.1.1...) para detallar cada rama hasta llegar a sugerencias técnicas, repositorios de inspiración y oportunidades de mejora. Se incluyen opiniones de expertos de diferentes áreas para enriquecer cada apartado.

1. Concepto & Estrategia

1.1 Brief / One-Pager

1.1.1 Elevator pitch

1.1.1.1 Variación para eventos

ClueQuest convierte cualquier fiesta o festival en una aventura interactiva con narrativa, roles y avatares generados por IA. Los asistentes participan en una búsqueda del tesoro con QR/AR que mezcla gamificación y storytelling.

1.1.1.2 Variación para educación

Se trata de una plataforma que gamifica el aprendizaje: los estudiantes resuelven retos geolocalizados y QR mientras siguen una historia ramificada. El profesorado controla el ritmo y monitoriza el progreso de la clase.

1.1.1.3 Variación para empresas

Un SaaS para team-building que combina rol-play, retos colaborativos y métricas en tiempo real. El administrador diseña la experiencia sin código, integra la dinámica con Slack/Teams y analiza el rendimiento de los equipos.

1.1.2 Unique Selling Proposition (USP)

1.1.2.1 Narrativa y rol-play no-code

El motor de aventuras permite crear historias ramificadas con personajes y roles personalizados sin conocimientos de programación. Inspirado en herramientas como Twine e Ink, se puede adaptar fácilmente a cada evento.

1.1.2.2 Contenido generado por IA

La plataforma integra avatares generados por IA (p. ej. DALL·E【718000835024873†screenshot】), generación de diálogos y preguntas dinámicas. Estas capacidades incrementan la personalización y mantienen la frescura de cada aventura.

1.1.2.3 Híbrido físico-digital

Combina experiencias físicas (códigos QR, geolocalización, beacons) con contenido digital (AR, mapas, ranking en tiempo real). Repos como *moodle-mod_treasurehunt* permiten ver ejemplos de rutas y pistas en GIS【735571247025979†screenshot】.

1.1.3 Casos de uso

1.1.3.1 Eventos sociales

En bodas, cumpleaños o festivales, el anfitrión crea una narrativa temática (p. ej. hadas y duendes). Los asistentes se registran, eligen un rol y recorren el venue escaneando QR. Repositorios como *ScavengerHunt* de Simon Carpentier implementan roles de administrador y juez y un flujo completo de cazas【719364188597829†screenshot】.

1.1.3.2 Educación

Profesores pueden transformar una lección en un juego de pistas. El módulo open-source *moodle-mod_treasurehunt* guía a los usuarios para crear “camino” con etapas y pistas que apuntan a la siguiente localización【735571247025979†screenshot】. La narrativa integra preguntas de opción múltiple para reforzar el aprendizaje.

1.1.3.3 Corporativo

Equipos de trabajo compiten o colaboran resolviendo retos. El proyecto *Scavenger Hunt for Jira/Confluence* publicado por Triple E IT demuestra cómo integrar un caza-tesoros en herramientas empresariales y gestionar múltiples entornos y autenticación【481949067328864†screenshot】.

1.1.4 Ruta de implantación

1.1.4.1 MVP inicial

Implementar registro básico, selección de roles, generación y lectura de QR firmados y un ranking simple. Reutilizar conceptos del proyecto *Serious Scavenger Hunt* para la gestión de stunts y equipos【719364188597829†screenshot】.

1.1.4.2 Crecimiento incremental

Añadir avatares IA y narrativa ramificada utilizando motores como Ink. Incluir minijuegos (trivia, foto-prueba) y un dashboard de organizador en Supabase.

1.1.4.3 Integración completa

Escalar la arquitectura con microservicios (p. ej. Node/Express + Supabase Functions) y habilitar plugins para AR avanzada (Mind AR) y generadores de aventuras procedurales.

1.2 Propuesta de Valor por Segmento

1.2.1 Organizadores

1.2.1.1 Herramientas de configuración

Panel no-code para elegir tema, historia, roles, tipos de retos, duración y premios. Permite personalizar cada elemento y aplicar plantillas.

1.2.1.2 Dashboard en tiempo real

Visualiza progresos, puntuaciones y rutas de los participantes. Los repositorios *ScavengerHunt* y *moodle-mod_treasurehunt* sirven de referencia para gestionar etapas y ver tablas de progreso【719364188597829†screenshot】【735571247025979†screenshot】.

1.2.1.3 Onboarding simplificado

Asistente paso a paso (“wizard”) que guía al anfitrión por los 5 pasos de configuración (1: tema; 2: narrativa; 3: roles; 4: retos; 5: premios y seguridad).

1.2.2 Empresas

1.2.2.1 Integración corporativa

Conexión con Slack, Microsoft Teams o Atlassian Forge para enviar notificaciones y registrar resultados. El código de *Scavenger Hunt for Jira/Confluence* muestra cómo empaquetar funciones para distintos entornos【481949067328864†screenshot】.

1.2.2.2 Analítica de rendimiento

Métricas de participación, colaboración y liderazgo para equipos. Utiliza tablas de clasificación en tiempo real y heatmaps de actividad.

1.2.3 Educadores

1.2.3.1 Módulos gamificados

Creamos aventuras educativas basadas en materias (historia, ciencias, idiomas). *moodle-mod_treasurehunt* sirve de inspiración: cada “camino” incluye pistas, preguntas y validaciones【735571247025979†screenshot】.

1.2.3.2 Herramientas de seguimiento

Panel para ver progreso de cada estudiante, calificaciones y participación. Integración con LMS vía LTI o API.

1.2.4 Participantes

1.2.4.1 Personalización del jugador

Cada usuario elige su rol y avatar IA; puede compartir logros y coleccionar criaturas en un inventario digital.

1.2.4.2 Recompensas sociales

Badges, certificados y ranking público para fomentar la competición sana y la colaboración. Se pueden extender con NFTs en fases avanzadas.

1.3 Principios de Diseño

1.3.1 Modularidad

1.3.1.1 Arquitectura de microservicios

Separar el motor de aventuras, el servicio de IA, el módulo de AR y el backend de analíticas. Cada servicio cuenta con APIs bien definidas y escalables. **Ejemplo:** replicar la estructura de repos *ScavengerHunt* donde el módulo *ScavengerHunt.Web* se separa de la lógica de negocio【719364188597829†screenshot】.

1.3.1.1.1 Implementación Node/Supabase

Utilizar Next.js para el frontend y Node.js/Express para los microservicios; emplear Supabase Functions para lógica serverless. Esto permite desplegar funciones cercanas al usuario con baja latencia.

1.3.1.1.2 GraphQL

Permitir consultas flexibles y reducir las peticiones. Se puede integrar [Apollo Server](#) en los microservicios.

1.3.1.2 Arquitectura de plugins

Diseñar la aplicación para que nuevos tipos de retos (p. ej. AR sin marcadores, juegos con IoT, puzles matemáticos) se añadan como plugins. Un ejemplo es la estructura del proyecto *moodle-mod_treasurehunt*, donde los tipos de pistas son módulos separados.

1.3.1.3 Feature flags

Permitir habilitar o deshabilitar funciones para grupos de usuarios y hacer pruebas A/B de nuevas mecánicas.

1.3.2 Escalabilidad

1.3.2.1 Despliegue serverless

Utilizar Vercel o Supabase Edge Functions para escalar automáticamente según el número de participantes. Los servidores sólo se activan cuando se invocan endpoints (pay-as-you-go).

1.3.2.2 Caché y tiempo real

Cachear assets (modelos 3D, avatares) en CDN. Emplear Supabase Realtime (basado en Postgres logical replication) para publicar actualizaciones de rankings y premios.

1.3.2.3 Optimización de AR

Seleccionar modelos 3D optimizados (bajo poligonado y texturas comprimidas). Considerar Mind AR【718000835024873†screenshot】 para AR web ligera y *ar-savenger-hunt* como inspiración para tours【718000835024873†screenshot】.

1.3.3 Personalización

1.3.3.1 Temas

Definir tokens de diseño (colores, tipografías, iconografía) que se puedan sobrescribir. El anfitrión puede escoger entre plantillas o subir sus propios assets.

1.3.3.2 Narrativa

Admitir importación de ficheros Twine/Ink para historias ramificadas y permitir su edición visual. Un plugin interpreta la narrativa y la conecta con los retos y QRs.

1.3.3.3 Roles

Permitir crear roles customizados con perks (pistas adicionales, multiplicadores de puntos). Las definiciones se guardan en la tabla `roles` y se aplican al calcular la puntuación.

1.3.4 Accesibilidad

1.3.4.1 PWA offline

El servicio debe funcionar sin conexión; las capturas y acciones se sincronizan cuando hay internet. Esta práctica se inspira en la compatibilidad de *moodle-mod_treasurehunt* con mobile【735571247025979†screenshot】.

1.3.4.2 Multilenguaje

Internacionalización (i18n) completa y soporte de alfabetos no latinos. Generar traducciones con IA y permitir que los organizadores ajusten las cadenas.

1.3.4.3 Compatibilidad multi-dispositivo

Garantizar UI responsive y controles adaptados a teclado, ratón y pantalla táctil. Incluir soporte para AR en navegadores modernos y en apps nativas (ARKit/ARCore).

1.3.5 Seguridad

1.3.5.1 Firmas en QR

Códigos HMAC para evitar falsificación. Cada QR incluye un timestamp y un hash, verificado por el backend antes de conceder puntos.

1.3.5.2 Anti-fraude geolocalización

Comprobar la localización mediante GPS, Wi-Fi y Bluetooth; usar “bounding boxes” como en *moodle-mod_treasurehunt* [735571247025979†screenshot]. Detectar patrones de escaneo anómalos con modelos ML.

1.3.5.3 Privacidad de datos

Encriptar datos en tránsito y en reposo (TLS y AES-256). Cumplir con GDPR y ofrecer borrado bajo solicitud. Implementar autenticación multifactor para anfitriones y roles críticos.

1.4 Roadmap de Fases

1.4.1 Fase 1 – MVP

1.4.1.1 Registro y roles

Crear tablas users y roles, implementar login social (Google/Apple) y selección de rol. Basarse en la estructura de *ScavengerHunt* para asignar roles y permisos [719364188597829†screenshot].

1.4.1.2 Hunt básico

Generar QR firmados y asociarlos con retos simples (p. ej. preguntas de múltiple opción). Al escanear, se actualiza la tabla captures y el ranking.

1.4.1.3 Dashboard inicial

Panel para anfitrión con listado de QRs, participantes y puntuaciones. Inspirado en la vista de resumen de *moodle-mod_treasurehunt* y su editor GIS [735571247025979†screenshot].

1.4.2 Fase 2 – Crecimiento

1.4.2.1 Motor de narrativa

Incorporar un editor visual para historias ramificadas y su parser (Twine/Ink). Conectar cada nodo con retos y recompensas.

1.4.2.2 AR y VR básicos

Incluir overlays con Mind AR o A-Frame para mostrar criaturas flotantes. El repo *ar-scavenger-hunt* enseña cómo posicionar objetos en puntos de interés dentro de un edificio【718000835024873†screenshot】.

1.4.2.3 Minijuegos IA

Generar preguntas temáticas con LLMs y validar respuestas. Implementar reconocimiento de objetos para retos fotográficos y análisis de audio para pistas sonoras.

1.4.3 Fase 3 – Expansión

1.4.3.1 Integraciones empresariales

Implementar conectores con Slack/Teams y Atlassian; usar ejemplos de *Scavenger Hunt for Jira/Confluence* para paquetes específicos【481949067328864†screenshot】.

1.4.3.2 Marketplace de contenido

Permitir a diseñadores y desarrolladores vender temas, narrativas, minijuegos y modelos 3D. Gestionar comisiones y licencias.

1.4.3.3 AR avanzada

Explorar AR sin marcadores (ARCore/ARKit) y filtrado facial. Añadir máscaras y efectos en tiempo real.

1.4.4 Fase 4 – Innovación

1.4.4.1 Integración IoT

Conectar luces, sonido y vibración del venue (p. ej. Philips Hue, IoT devices) para crear eventos multisensoriales. Activar dispositivos según el progreso de los jugadores.

1.4.4.2 Blockchain y NFTs

Emitir certificados coleccionables como NFT para logros raros. Considerar usar cadenas con baja huella de carbono (p. ej. Polygon). Dejar la integración como módulo opcional.

1.4.4.3 Generación procedural

Utilizar IA generativa para crear aventuras completas (narrativa, retos, assets) a partir de parámetros (tema, duración, número de jugadores).

1.5 Inspiración en Repositorios

1.5.1 Hunts & QR

1.5.1.1 Serious Scavenger Hunt – [scarpentier/ScavengerHunt](#)

Proyecto open-source en C# y JavaScript que permite crear y gestionar scavenger hunts con roles de administrador y juez. Ofrece un ejemplo de arquitectura basada en MVC y plantillas Razor【719364188597829†screenshot】.

1.5.1.1.1 Características relevantes

- Gestión de equipos y “stunts” (pruebas) con roles diferenciados (admin, judge).
- Vistas de progreso y puntuación para cada equipo.
- Flujo de autenticación y autorización en .NET.

1.5.1.2 moodle-mod_treasurehunt – [juacas/moodle-mod_treasurehunt](#)

Módulo para Moodle que permite tesoros geolocalizados en exteriores, interiores (QR) y virtuales. Incluye editor de mapas GIS, creación de caminos y pistas con preguntas asociadas【735571247025979†screenshot】.

1.5.1.2.1 Características relevantes

- Permite crear varios “caminos” y etapas por camino.
- Las pistas pueden incluir preguntas para doble-chequeo.
- Editor GIS para trazar localizaciones exactas.
- Soporta modalidades individual y grupal, con ajustes de penalización y puntuación.

1.5.1.3 Scavenger Hunt para Jira/Confluence – [TripleEIT](#)

Repositorio de Triple E IT que implementa un juego de caza para las plataformas de Atlassian. Muestra cómo dividir el proyecto en tres repos (core, Jira y Confluence) y aprovechar la plataforma Forge【481949067328864†screenshot】.

1.5.1.3.1 Características relevantes

- Utiliza la plataforma Forge para alojar el código en la nube de Atlassian.
- Separa el núcleo funcional de las dependencias de Jira/Confluence.
- Incluye autenticación, despliegues automáticos y permisos integrados.

1.5.2 Narrativa & Juegos

1.5.2.1 Twine

Herramienta open-source para crear historias interactivas en formato HTML. Permite exportar historias que pueden ser interpretadas por el motor de la plataforma.

1.5.2.2 Ink by Inkle

Librería para diálogos interactivos y narrativa no lineal, con sintaxis sencilla. Se puede compilar a JSON para integrarse con motores web.

1.5.2.3 Godot Dialogue System

Complemento para Godot Engine que soporta diálogos tipo árbol; útil como referencia para crear plugins de narrativa.

1.5.3 Avatares & AI

1.5.3.1 Ready Player Me

SDK que genera avatares 3D multiplataforma. Puede integrarse para generar avatares en juegos VR/AR y exportarlos a GLB.

1.5.3.2 OpenAI Images (DALL·E)

API para generación y edición de imágenes. Se puede emplear para crear retratos estilizados de los usuarios o transformar selfies en personajes de fantasía.

1.5.3.3 Leonardo.ai

Plataforma que permite elegir estilos artísticos diversos y generar imágenes de alta calidad. Útil para ofrecer variedad de estilos de avatar.

1.5.4 Herramientas AR

1.5.4.1 Mind AR / AR.js

Frameworks de realidad aumentada para web que permiten reconocimiento de imágenes y marcadores. El repositorio *ar-scavenger-hunt* muestra un caso de uso en bibliotecas【718000835024873†screenshot】.

1.5.4.2 H5P AR Scavenger

Librería de código abierto para crear cazas con AR y QR sin conocimientos de programación. Incluye soporte para marcadores personalizados y es utilizada por educadores.

1.6 Implementaciones de IA

1.6.1 Avatares

1.6.1.1 Generación de avatares con DALL·E

Integra la API de OpenAI para transformar selfies en estilos artísticos (realista, cartoon, 8-bit). Debe respetar la privacidad del usuario y eliminar fotos originales después de la transformación.

1.6.1.2 Leonardo.ai para estilos creativos

Permite crear avatares con diferentes paletas y estilos. Puede ser ofrecido como complemento premium.

1.6.1.3 Ready Player Me 3D

Genera avatares 3D listos para AR/VR. Estos modelos se pueden cargar en escenas de A-Frame o Unity.

1.6.2 Narrativa dinámica

1.6.2.1 IA generativa de diálogos

Utilizar LLMs (ChatGPT o modelos open-source como Llama) para generar respuestas y ramificar historias basándose en las decisiones del jugador. Añadir filtros éticos para evitar contenido inapropiado.

1.6.2.2 Ajuste de dificultad

Aplicar aprendizaje automático para ajustar la dificultad de retos basándose en el rendimiento individual. La IA decide cuándo proporcionar pistas adicionales.

1.6.2.3 Auto-recapitulación

Después de un evento, generar automáticamente un resumen narrativo y visual con los puntos clave y logros de los participantes.

1.6.3 Retos inteligentes

1.6.3.1 Generación de trivia

Usar un modelo generativo para crear preguntas relacionadas con el tema. El sistema valida respuestas, explica la correcta y ajusta la puntuación.

1.6.3.2 Validación de fotos

Algoritmos de visión artificial detectan objetos o situaciones (p. ej. “selfie con un cartel específico”). La puntuación se concede si los criterios se cumplen.

1.6.3.3 Análisis de audio

Emplear modelos de reconocimiento de voz para transcribir mensajes secretos y validar adivinanzas pronunciadas por los usuarios.

1.6.4 *Anti-fraude y analítica*

1.6.4.1 Detección de patrones irregulares

Modelos de clasificación detectan secuencias sospechosas (escaneos repetidos, cambios rápidos de ubicación). Se actúa bloqueando o revisando manualmente.

1.6.4.2 Sensor fusion

Combinar información de GPS, Wi-Fi y Bluetooth para verificar la presencia física del usuario. Usar “testigos digitales” (otros jugadores) como verificación cruzada.

1.6.4.3 Heurísticas de abuso

Limitar el número de escaneos por minuto y usar tokens únicos por código para evitar reusos.

1.7 **Settings del Anfitrión**

1.7.1 *Tema visual*

1.7.1.1 Selección de plantillas

Ofrecer temas preconfigurados (fantasía, misterio, corporativo, sci-fi, educativo). Cada tema define paletas y tipografías.

1.7.1.2 Constructor de temas

Permitir al anfitrión personalizar colores, fuentes y fondos vía un panel. Guardar presets para futuros eventos.

1.7.2 *Narrativa*

1.7.2.1 Historias prehechas

Biblioteca de historias para arrancar rápidamente (ej. “Bosque encantado”, “Operación hacker”, “Juego de detectives”).

1.7.2.2 IA generativa de historias

El anfitrión define parámetros (tema, tono, duración) y la IA sugiere una narrativa completa con nodos y diálogos. El usuario puede editarla antes de publicarla.

1.7.3 Roles y avatares

1.7.3.1 Definición de roles

Crear roles únicos asignando nombre, descripción y perks. Se pueden crear roles colaborativos o competitivos (ej. “Mago” con pistas extra; “Guerrero” con más puntos por respuestas rápidas).

1.7.3.2 Gestión de avatares

Activar o desactivar la generación de avatares IA según el plan contratado. Elegir estilos disponibles y establecer límites de generación.

1.7.4 Retos y dinámica

1.7.4.1 Número y distribución de QR

Definir cuántos códigos se usarán y dónde se colocarán. Ajustar el nivel de dificultad y la longitud del recorrido.

1.7.4.2 Tipos de retos

Seleccionar qué retos estarán habilitados: trivia, foto, audio, AR, puzzles colaborativos. Cada tipo tiene sus propias opciones (tiempo límite, número de intentos, puntuación).

1.7.4.3 Duración y ritmo

Establecer la duración total del evento y la ventana de tiempo para cada reto. Incluir descansos o eventos sorpresa.

1.7.5 Premios y cierre

1.7.5.1 Visibilidad de ranking

Permitir ocultar o mostrar el ranking durante la partida; mostrarlo al final o en tiempo real.

1.7.5.2 Recompensas digitales

Configurar badges, certificados y NFTs. Establecer requisitos para cada recompensa (número de criaturas atrapadas, minijuegos completados).

1.7.5.3 Recompensas físicas

Registrar premios físicos y asignarlos al ganador o por sorteo. Gestionar la logística de entrega.

1.7.6 Seguridad y acceso

1.7.6.1 Modo de acceso

Elegir entre modo privado (solo por invitación) o público (cualquier persona con el código). Se pueden generar códigos únicos para invitados.

1.7.6.2 Límite de dispositivos

Establecer cuántos dispositivos pueden usar la misma cuenta para evitar trampas. Asociar cada usuario a un dispositivo principal.

1.7.6.3 Autenticación multifactor

Ofrecer MFA para anfitriones y usuarios con permisos avanzados. Aumenta la seguridad en la administración del evento.

2. Experiencia & Producto

2.1 Wireframes / Flujos

2.1.0 Configuración del anfitrión

Antes de que empiece la fiesta, el anfitrión pasa por un “wizard” con cinco pasos:

1. **Tema** – Seleccionar plantilla o crear un tema personalizado.
2. **Narrativa** – Elegir historia prehecha o generar una nueva con IA y editarla.
3. **Roles** – Definir roles disponibles y perks.
4. **Retos** – Seleccionar tipos de retos y generar códigos QR/NFC.
5. **Premios y Seguridad** – Establecer ranking, recompensas, accesos y límites de dispositivos.

2.1.1 P1 – Pantalla de bienvenida

- **Objetivo:** captar la atención y presentar el tema.
- **Acciones:** “Entrar a la aventura”, “Cambiar idioma”.
- **Consideraciones de UX:** CTA única, copy adaptado al tema.
- **IA:** adaptador de texto que ajusta el tono según el público objetivo.

2.1.2 P2 – Login express

- **Objetivo:** permitir acceso rápido con SSO o modo invitado.
- **Estados:** logueado, en espera, offline.
- **Críticas:** evitar formularios innecesarios, mostrar fallos de forma clara.

2.1.3 P3 – Historia introductoria

- **Objetivo:** crear curiosidad y establecer la misión.
- **Acciones:** avanzar, saltar historia.

- **IA:** ajustar longitud del texto según la interacción previa (skip rápido → narrativa concisa).

2.1.4 P4 – Selección de rol

- **Objetivo:** permitir al jugador identificarse con un rol y personalizar su experiencia.
- **Contenido:** tarjetas animadas con perks.
- **IA:** sugerir roles equilibrados si un equipo queda desequilibrado.

2.1.5 P5 – Avatar IA

- **Objetivo:** generar un avatar que aumente el engagement.
- **Flujo:** subir selfie → seleccionar estilo → preview → guardar.
- **IA:** filtros de moderación para contenido inapropiado; estilos sugeridos.

2.1.6 P6 – Mapa / Lista de QR

- **Objetivo:** guiar la exploración.
- **Funciones:** mostrar ubicación de QR, progreso, rutas sugeridas.
- **Críticas:** ofrecer alternativa de lista para personas con discapacidades visuales; no saturar con demasiada información en pantalla.

2.1.7 P7 – Retos / Minijuegos

- **Objetivo:** proporcionar desafíos breves (≤ 90 s) que mantengan el ritmo.
- **Tipos:** trivia generada, foto-prueba con IA, audio secreto, puzzles colaborativos.
- **Críticas:** asegurar la accesibilidad (uso sin audio, imágenes alt) y proporcionar opciones de ayuda.

2.1.8 P8 – Inventario y progreso

- **Objetivo:** mostrar la colección y fomentar la inversión del jugador.
- **Funcionalidades:** lista de criaturas capturadas, sets, nivel del jugador, logros.
- **IA:** sugerir próximos QRs para completar sets, ajustar dificultad.

2.1.9 P9 – Ranking y cierre

- **Objetivo:** celebrar a los ganadores y concluir la aventura.

- **Elementos:** tabla de clasificación, premios digitales y físicos, compartir resultados en redes sociales.
- **Críticas:** evitar humillar a los últimos puestos; reconocer logros adicionales como “mejor equipo colaborativo”.

2.2 Storyboard

El flujo narrativo estándar sigue estos pasos:

QR de inicio → **Login** → **Narrativa** → **Elección de rol** → **Avatar** → **Exploración** → **Retos** → **Recompensa** → **Ceremonia final**.

Cada flecha puede ramificarse según decisiones y nivel del jugador. Se plantean variantes de historia para distintos temas (fantasía, educación, corporativo) y se sugiere un modo “salto” que acorta la historia para usuarios impacientes.

2.3 UX Heurísticas

- **Hooked:** cada pantalla debería corresponder a un paso del ciclo (disparador, acción, recompensa, inversión). Se definen micro-recompensas (confeti, sonidos) tras cada reto.
- **Indistractable:** limitar la duración de cada interacción y eliminar distracciones (no más de dos CTAs por pantalla). Incluir pausa para permitir a los jugadores socializar sin penalización.
- **Inclusividad:** ofrecer modo oscuro, tamaños de letra ajustables y soporte para lectores de pantalla.

2.4 Assets y branding

- **Estilos:** proveer tres líneas de arte (cartoon vibrante, pixel retro, minimal corporativo) con posibilidad de crear combinaciones.
- **Paquete audiovisual:** loops de música ambiental, efectos de sonido de escaneo y victoria.
- **Modelos 3D:** criaturas y objetos optimizados (< 1.2 MB) compatibles con Mind AR/A-Frame.
- **Tipografías:** fuentes accesibles (WCAG AA) con variantes serif y sans.

2.5 Pricing y monetización

- **Evento único:** pago por participante (ej. 0,99 €) que incluye QR, ranking y roles básicos.
- **Pro mensual:** suscripción que desbloquea IA generativa, AR básica y análisis avanzados.

- **Enterprise:** paquete marca blanca, con soporte 24/7, SSO, SLA y acceso a API/LLM privado.
- **Complementos:** venta de temas premium, narrativas exclusivas y modelos 3D.

3. Arquitectura & Tecnología

3.1 Frontend

Construido con Next.js y Tailwind, generando una PWA que funciona offline. Integra WebXR para AR y soporta exportación a apps nativas mediante React Native si se necesita ARKit/ARCore.

3.2 Backend y microservicios

El backend se compone de microservicios serverless desplegados en Supabase Functions o Vercel. Incluye servicios de autenticación, aventuras, IA, AR, pagos y analíticas.

3.3 Datos & Esquema

Tablas principales: users, events, scenes, roles, qr_codes, challenges, captures, leaderboard, media_assets, billing. Cada tabla dispone de políticas RLS para restringir acceso según rol y organización.

3.4 APIs

Rutas REST y GraphQL para: autenticación (/auth/login), gestión de eventos (/events), edición de escenas (/events/:id/scenes), ejecución de juego (/hunt/scan, /challenge/submit), ranking (/events/:id/leaderboard) y generación de avatares (/avatar/generate). Los endpoints disponen de límites de uso y verificación HMAC.

3.5 Seguridad

Implementa controles de acceso basados en roles (RBAC), encriptación de datos y anti-fraude. Incluye auditoría de acciones, detección de bots y pruebas de penetración periódicas.

3.6 Variables de entorno

Ejemplo .env:

```
NEXT_PUBLIC_APP_NAME=ClueQuest
NEXT_PUBLIC_BASE_URL=https://app.mi-cluequest.com
SUPABASE_URL=https://xxxxxxxxx.supabase.co
SUPABASE_SERVICE_ROLE_KEY=....
NEXT_PUBLIC_SUPABASE_ANON_KEY=....
STRIPE_PUBLIC_KEY=pk_live_....
STRIPE_SECRET_KEY=sk_live_....
OPENAI_API_KEY=....
```


JWT_SECRET=.....

QR_HMAC_SECRET=.....

3.7 Repositorios de referencia

- **Serious Scavenger Hunt**[719364188597829†screenshot]: arquitectura MVC y gestión de equipos.
- **moodle-mod_treasurehunt**[735571247025979†screenshot]: tesoros geolocalizados, editor de mapas y preguntas.
- **TripleEIT Scavenger Hunt for Jira/Confluence**[481949067328864†screenshot]: integración en plataformas empresariales.
- **ar-scamenger-hunt**[718000835024873†screenshot]: uso de Mind AR y A-Frame para tours AR.

4. Operación & Delivery

4.1 Métricas

Métricas clave: total de escaneos, escaneos por usuario, tiempo medio por reto, tasa de conversión (inicio → primer QR), porcentaje de finalización, NPS post-evento y engagement en redes sociales.

4.2 Riesgos y mitigaciones

1. **Cobertura GPS indoor**: usar beacons y códigos alternativos para interiores; permitir validación manual mediante foto o testigo digital.
2. **Carga IA en vivo**: procesar la generación de avatares en cola y ofrecer un avatar genérico de reserva si se supera el tiempo umbral.
3. **Conectividad**: la PWA almacena las capturas offline y sincroniza cuando vuelve la conexión.
4. **Abuso / trampas**: implementar modelos de detección de anomalías y límites de uso por minuto.

4.3 QA / Test Plan

Happy path: registro → rol → avatar → 3 escaneos → reto → ranking → final.

Edge cases: sin señal, QR expirado, GPS fuera de radio, selfie inválida, fraude.

Performance: pruebas de carga con 200 escaneos/min y 1 000 usuarios simultáneos.

4.4 Checklist de lanzamiento

1. Dominio configurado con SSL.

2. Stripe en modo live y webhook configurado.
3. Políticas legales de Términos de uso y Privacidad actualizadas.
4. Manual de instrucciones para imprimir y distribuir QRs.
5. Evento de demostración pública (link compartido) para feedback inicial.

5. Mesa redonda – Opiniones de expertos (actualización)

Esta ronda incorpora la primera crítica constructiva de 15 profesionales y profundiza con nuevas sugerencias adaptadas a cada rama detallada.

5.1 Desarrollador

1. Aconseja definir una arquitectura hexagonal, desacoplando dominio, infraestructura y presentación.
2. Sugiere implementar pruebas unitarias en cada microservicio y pruebas end-to-end para el flujo completo.
3. Recomienda utilizar herramientas de control de versiones monorepo (p. ej. Nx) para gestionar módulos de frontend y backend.

5.2 Publicista

1. Destaca la importancia de comunicar la USP claramente en todas las piezas promocionales: “Crea aventuras únicas sin escribir código”.
2. Propone campañas de marketing estacionales (Halloween, Verano) con historias temáticas y descuentos.
3. Sugiere alianzas con recintos culturales y festivales para lanzar experiencias patrocinadas.

5.3 Director de Arte

1. Recomienda crear un manual de identidad que defina el uso de logotipo, colores, tipografías y animaciones comunes.
2. Aconseja desarrollar un sistema de componentes UI escalable que se adapte a diferentes temas sin perder coherencia.
3. Sugiere usar animaciones sutiles (parallax, fades) para mejorar la inmersión sin afectar el rendimiento.

5.4 Experto en Juegos Inmersivos

1. Propone implementar una curva de dificultad adaptativa que mantenga el interés tanto de principiantes como de expertos.
2. Sugiere diseñar retos cooperativos y competitivos para fomentar distintas dinámicas de grupo.
3. Recomienda integrar puzzles narrativos alineados con el tema (p. ej. descifrar runas en eventos de fantasía).

5.5 Especialista en Vibecoding

1. Insiste en la importancia de micro-recompensas constantes (sonidos, colores, vibraciones) para mantener el “vibe” del usuario.
2. Sugiere ajustar la duración de cada ciclo Hooked (2-3 minutos) para evitar la procrastinación y mantener la atención.
3. Recomienda coordinar la música de fondo con los momentos de tensión y recompensa.

5.6 Diseñador UX

1. Aconseja simplificar la navegación y minimizar el número de pasos hasta empezar el juego.
2. Propone incluir tutoriales interactivos opcionales para usuarios nuevos.
3. Recomienda implementar un modo de alto contraste para eventos nocturnos o usuarios con visión reducida.

5.7 Arquitecto de Software

1. Recomienda utilizar infraestructura serverless y bases de datos distribuidas para lograr alta disponibilidad.
2. Sugiere emplear GraphQL con persistencia de consultas para optimizar el rendimiento.
3. Aconseja implementar feature flags y despliegues canary para lanzar funciones a un porcentaje de usuarios antes del release completo.

5.8 Especialista en AR/VR

1. Propone investigar AR sin marcadores (plane detection) para eventos en espacios abiertos.

2. Recomienda priorizar modelos 3D de baja carga poligonal para garantizar compatibilidad en móviles.
3. Sugiere añadir un modo de recorrido virtual (VR) para quienes no puedan asistir presencialmente.

5.9 Experto en Gamificación

1. Sugiere tablas de clasificación en tiempo real con categorías variadas (mejor colaborador, más creativo).
2. Aconseja desarrollar logros con niveles y coleccionables para incentivar la rejugabilidad.
3. Propone retos diarios o semanales para seguir usando la plataforma fuera de los eventos principales.

5.10 Product Manager

1. Insiste en validar el MVP con un grupo piloto antes de escalar.
2. Sugiere priorizar características basándose en datos de uso y feedback.
3. Recomienda medir el coste por adquisición y el retorno por evento para ajustar la estrategia de precios.

5.11 Estratega de Branding

1. Aconseja crear un kit de marca editable para que los clientes mantengan coherencia al personalizar su evento.
2. Propone definir una voz de marca aventurera y cercana, con toque de humor generacional.
3. Sugiere aprovechar el contenido generado por usuarios en redes como parte de la narrativa de marca.

5.12 Especialista en Seguridad

1. Propone realizar auditorías de seguridad periódicas e incluir un programa de bug bounty.
2. Recomienda cifrar toda la información personal y anonimizar los datos analíticos.
3. Sugiere desplegar autenticación multifactor y alertas en tiempo real ante actividades sospechosas.

5.13 Especialista en Monetización

1. Sugiere explorar modelos de suscripción flexibles con complementos premium.
2. Recomienda alianzas con recintos y marcas para patrocinios y reparto de ingresos.
3. Aconseja ofrecer “packs” de personalización (temas, avatares, narrativa) como microtransacciones.

5.14 Experto en IA Generativa

1. Propone entrenar modelos con datos anónimos de eventos para personalizar mejor las recomendaciones.
2. Recomienda desarrollar NPCs conversacionales que adapten sus respuestas según el progreso del jugador.
3. Sugiere crear un sistema de seguridad para revisar automáticamente los outputs de IA y evitar sesgos o contenidos impropios.

5.15 Tester QA

1. Aconseja automatizar pruebas de regresión en cada despliegue usando frameworks como Playwright.
2. Propone ejecutar pruebas en una amplia variedad de dispositivos y condiciones de red para replicar escenarios reales.
3. Recomienda recopilar feedback de eventos reales para descubrir bugs y fricciones no identificadas en laboratorio.