Tabla de contenidos

a. Tabla de contenidos

b. Introducción del proyecto

c. Resumen ejecutivo.

d. Antecedentes.

e. Visión

f. Misión

g. Descripción de caso de estudio.

h. Objetivos.

i. Público beneficiado.

j. Resultados esperados.

k. Desarrollo.

1. REQUERIMIENTOS

1.1. Propósito

1.2. Alcance del producto/Software

1.3. Reglas de negocio

1.5. Clase y características de usuarios

1.6 Funcionalidades del producto

1.6.1. Requerimientos funcionales

1.6.2. Requerimientos no funcionales

1.6.2. Requerimiento de interfaces externas

1.7. Arquitectura del Software

1.8. Modelo de casos de uso del sistema

1.8.1. Diagramas de caso de uso del sistema

1.8.2. Documentación de elementos del Modelo de Casos de uso

1.9 Especificaciones de Casos de Uso

1.9.1. Especificación de Alto Nivel

1.9.2. Especificación Expandida

1.9.3. Diseño de prototipos

1.10 Priorización de Casos de Uso.

1.10.1 Clasificación de los Casos de Uso

1.11 Flujo General de Navegación

1.12 Esquema de Seguridad

1.13 Glosario

2. ANÁLISIS Y DISEÑO

2.1. Modelo de Análisis

2.1.1 Arquitectura del Sistema

2.1.2 Realización de Casos de Uso – Análisis

2.1.3 Modelo Conceptual

2.1.4 Modelo Lógico

2.1.5 Modelo Físico

2.1.6 Metas y Restricciones de la Arquitectura

2.2. Modelo de Diseño

2.2.1 Arquitectura del Software

2.2.1. Vista de Capas y Subsistemas

2.2.2 Realización de Casos de Uso – Modelo de Diseño

2.2.2.1 Diagrama de Clases

2.2.2.2 Diagrama de secuencia

2.2.3. Programación diagramas de clases

2.2.3.1. Código del DC – Nombre del DC 01

2.2.3.2. Código del DC – Nombre del DC 01

2.2.3.3. Código del DC – Nombre del DC 01

2.2.3.4. Código del DC – Nombre del DC 01

2.3. Diagrama de Despliegue

2.4. Diagrama de Componentes

2.5. Vista de Integración del Software

m) Conclusiones.

n) Recomendaciones.

o) Referencias.

a. Introducción del proyecto

El dominio del inglés es hoy un requisito clave en la educación, el trabajo y la cultura digital. Sin embargo, para muchos hispanohablantes la adquisición de vocabulario es la base de la comprensión lectora y la comunicación efectiva y sigue siendo una barrera. Por eso sepropone una solución web centrada en el usuario para acelerar ese proceso: una plataforma interactiva que facilita el aprendizaje y la retención de vocabulario a través de micro-lecciones, práctica espaciada, ejemplos contextualizados y retroalimentación inmediata.

El propósito de este proyecto es **diseñar y desarrollar una aplicación web responsiva** que permita a estudiantes de distintos niveles (secundaria, universitarios y autodidactas adultos) ampliar su léxico de forma progresiva y medible. La plataforma prioriza buenas prácticas de usabilidad y accesibilidad, integra recursos auditivos para la pronunciación, y emplea dinámicas de gamificación (puntos, rachas, logros) para mantener la motivación.

b. Resumen Ejecutivo

c. Antecedentes

1) Contexto pedagógico: por qué centrarse en vocabulario

El vocabulario es uno de los principales predictores de la comprensión lectora y de la fluidez comunicativa en una segunda lengua. Para desenvolverse con comodidad en textos y conversaciones cotidianas se requiere una amplia cobertura léxica de alta frecuencia. En hispanohablantes, las dificultades más comunes no provienen de la gramática básica sino de la escasez de léxico útil y la baja retención en el tiempo, especialmente sin apoyo sistemático.

2) Fundamentos cognitivos del diseño

El aprendizaje de vocabulario está condicionado por la “curva del olvido”: sin repaso, la memoria decae rápidamente. Dos principios sólidos la mitigan y guían el diseño del proyecto:

Práctica espaciada (spaced repetition): distribuir los repasos en intervalos crecientes favorece la memoria de largo plazo.

Recuperación activa (retrieval practice): evocación mediante pruebas breves produce un aprendizaje más durable que la relectura pasiva.

De estos principios se desprende que una plataforma efectiva debe combinar repetición espaciada + evaluación activa, con retroalimentación inmediata y seguimiento del desempeño.

3) Estado del arte: soluciones digitales relevantes

Anki: popularizó el uso de tarjetas con algoritmos de repetición espaciada (SRS) y configuraciones de dificultad basadas en el rendimiento del usuario.

Duolingo: integra repasos adaptativos y elementos de gamificación (puntos, rachas) para sostener la motivación.

Quizlet y Memrise: ofrecen estudio de términos con modos de práctica variados, audio y ejercicios de reconocimiento/producción.

Estas soluciones validan el valor del SRS y la gamificación; sin embargo, suelen abordar múltiples destrezas a la vez y no siempre proporcionan métricas detalladas de retención léxica ni currículos transparentes por frecuencia o nivel.

e. Vision

Ser la plataforma web de referencia para hispanohablantes que dominan el vocabulario del inglés con sesiones breves, continuas y medibles, elevando su comprensión y comunicación real en estudios, trabajo y vida diaria.

f. Mision

Diseñar y operar una aplicación web accesible que:

* centre el aprendizaje en vocabulario útil organizado por temas y frecuencia, alineado a CEFR;
* combine repetición espaciada y recuperación activa con audio y ejemplos contextualizados;
* ofrezca analítica clara (palabras dominadas, retención, racha, metas semanales) para guiar el progreso;
* utilice gamificación responsable para sostener el hábito sin distraer del objetivo;
* garantice accesibilidad (WCAG 2.1 AA) y una experiencia simple para estudiantes A1–B2 en cualquier dispositivo.

g. Descripcion del caso de estudio

El caso describe el desarrollo e implementación de una aplicación web *mobile-first* enfocada exclusivamente en el aprendizaje y la retención de vocabulario en inglés para hispanohablantes de nivel A1–B2. La plataforma organiza palabras de alta frecuencia por temas y niveles CEFR, y ofrece micro-sesiones de 5–10 minutos con tarjetas de estudio, repetición espaciada, ejercicios de escucha y dictado, y ejemplos en contexto. El usuario se registra, elige su nivel y temas de interés, y desde el teléfono realiza sesiones breves en cualquier momento; el sistema programa automáticamente los repasos y muestra un panel de progreso con palabras dominadas, racha y metas semanales.

h. Objetivos.

**Objetivo general**

Diseñar, desarrollar y validar un MVP web que permita a hispanohablantes de nivel A1–B2 aprender y retener vocabulario de alta frecuencia de forma medible, alcanzando ≥ 70% de retención a 4 semanas y ≥ 3 sesiones por semana por usuario durante el piloto.

**Objetivos específicos**

1. Aprendizaje. Lograr que los usuarios alcancen ≥ 70% de retención del vocabulario practicado en un postest (o estimación del SRS) al finalizar el piloto.
2. Uso y hábito. Conseguir que ≥ 60% de los participantes realicen 3 o más sesiones semanales de 5–10 minutos.
3. Contenido. Publicar un banco inicial de ~800 palabras (rango 600–1,000) organizadas por temas y niveles CEFR, con audio y ejemplos en contexto.
4. Funcionalidad clave. Implementar repetición espaciada (SRS), recuperación activa, ejercicios de escucha/dictado y un programador de repasos adaptativo.
5. Analítica y progreso. Ofrecer un panel con métricas en tiempo real (palabras dominadas, racha, tasa de aciertos, tiempo de estudio) y definir la regla de “palabra dominada”.
6. Experiencia y accesibilidad. Alcanzar SUS ≥ 75 y cumplir WCAG 2.1 AA en navegación, contraste, uso por teclado y descripciones.
7. Rendimiento y fiabilidad. Lograr tiempo de carga < 2.5 s en 4G y disponibilidad ≥ 99% durante el piloto.
8. Operación y calidad. Disponer de un panel de administración para gestionar vocabulario y módulos (CRUD), manteniendo incidencias críticas < 1/semana y tiempos de resolución ≤ 7 días.

i. Público beneficiado.

Beneficiarios directos

Estudiantes de secundaria (A1–A2): refuerzo de vocabulario básico y vocabulario académico inicial.

Estudiantes de educación superior y técnica (A2–B1): léxico para lecturas, presentaciones y trabajos.

Jóvenes profesionales y trabajadores (A2–B2): vocabulario funcional para correo, reuniones y atención al cliente.

Autodidactas adultos (A1–B2): aprendizaje flexible en micro-sesiones de 5–10 minutos desde el teléfono.

Beneficiarios indirectos

Docentes de inglés: apoyo para tareas de práctica, seguimiento de progreso y selección de temarios por nivel/tema.

Instituciones educativas y centros de idiomas: herramienta ligera para complementar cursos y medir avance léxico.

Familias y tutores: visibilidad simple del progreso (palabras dominadas, racha y metas semanales).

Equipos de RR. HH./formación interna (opcional): módulos temáticos para roles con contacto internacional.

j. Resultados esperados.

Al finalizar el MVP de JIM: Proyecto Inglés se espera contar con una aplicación web priorizada para navegador en PC (desktop-first), estable y lista para uso real. El producto incluirá la experiencia de estudio con tarjetas, audio y dictado, organización del vocabulario por temas y niveles, y un panel de progreso visible para cada usuario, junto con un panel de administración que permita gestionar palabras, categorías y contenidos de forma ágil. Estará desplegado en un entorno de producción con CI/CD, monitoreo y registro de errores, cumpliendo buenas prácticas de seguridad, accesibilidad y compatibilidad con las versiones actuales de los principales navegadores de escritorio. Además, el sistema incorporará analítica esencial para entender el uso y orientar mejoras, documentación clara para operación y mantenimiento, y un proceso de despliegue reproducible con opciones de respaldo y recuperación. En conjunto, el resultado esperado es un producto funcional, mantenible y escalable, apto para iterar rápidamente en siguientes versiones.

k. Desarrollo.

1. REQUERIMIENTOS

1.1. Propósito

El propósito es alinear a los interesados en torno al alcance del MVP, delimitar lo que entra y lo que queda fuera de esta versión, y anticipar restricciones técnicas y de negocio (rendimiento, accesibilidad, seguridad y compatibilidad en navegadores de escritorio). Asimismo, establece criterios de aceptación medibles y la trazabilidad entre requisitos y casos de prueba, con el fin de reducir ambigüedades y evitar el incremento no controlado del alcance.

1.2. Alcance del producto/Software

Sistema web priorizado para navegador en PC (desktop-first) que permite estudiar y gestionar vocabulario de inglés. Incluye:

* Aplicación de estudio con navegación por módulos temáticos y niveles CEFR (A1–B2); sesiones con tarjetas, audio de pronunciación y dictado; programador de repaso (SRS) activo durante el MVP.
* Panel de progreso por usuario (palabras estudiadas/dominadas, racha, tasa de aciertos básicas).
* Panel de administración con CRUD de palabras, categorías, niveles y archivos de audio; importación de lotes (CSV/JSON) con validaciones.
* Compatibilidad de escritorio: últimas dos versiones de Chrome, Edge, Firefox y Safari; accesibilidad esencial (WCAG 2.1 AA en flujos críticos).
* Analítica básica de uso (sesiones, errores, eventos clave) y registro de actividad para auditoría mínima.

1.3. Reglas de negocio

* Estructura del contenido: cada palabra pertenece a un tema y a un nivel CEFR; contiene lema, categoría gramatical, definición simple, traducción/es, ejemplo en contexto y audio.
* Unicidad: no se permiten duplicados por combinación *(lema + nivel + tema + variante)*.
* Estado de ítems: *borrador*, *listo*, *publicado*; solo *publicado* aparece a los usuarios.
* Sesiones de estudio: duración objetivo 5–10 min; máximo 20–30 tarjetas por sesión.
* Nuevas palabras por día: límite por usuario configurable (por defecto 20); el resto se agenda como repaso.
* Dominio de palabra: se marca “dominada” con ≥ 3 aciertos consecutivos o probabilidad estimada ≥ 0.8 (según el modelo de repaso).
* Programación de repasos: los intervalos se amplían con aciertos y se reinician/acortan con errores; los fallos tienen prioridad en la siguiente sesión.
* Racha: aumenta con actividad válida en el día (≥ 1 sesión finalizada); se conserva con “día de gracia” semanal configurable.
* Audio: formato recomendado MP3/OGG, duración ≤ 5 s, 1–2 variantes por palabra (UK/US) cuando estén disponibles.
* Permisos:

Usuario: estudiar, ver progreso propio.

Editor/Administrador: gestionar contenido (CRUD, importación), publicar, revisar audio; acceso a métricas agregadas.

* Importación de contenido: archivos deben incluir columnas mínimas (lema, tema, nivel, definición, ejemplo, ruta de audio); los registros inválidos se rechazan con reporte.
* Idiomas de interfaz: español en el MVP; textos e indicaciones se almacenan como recursos para futura localización.
* Privacidad mínima: se muestran métricas individuales solo al propio usuario; en administración se ven métricas agregadas (sin datos personales).

1.5. Clase y características de usuarios

**Usuario final (estudiante/autodidacta A1–B2)**

* Usa **PC** con teclado y ratón; prefiere sesiones cortas y repetibles.
* Necesita interfaz **clara**, feedback inmediato y audio accesible.
* Valora un **panel simple** con progreso y próximas tareas.

**Administrador / Editor de contenido**

* Carga y mantiene vocabulario, categorías y audio desde **escritorio**.
* Requiere vistas **tabulares**, filtros/búsquedas, **importación** de lotes y control de estados (borrador/publicado).
* Necesita **validaciones** y reportes de errores de carga, además de métricas agregadas para revisar uso y calidad del contenido.

1.6 Funcionalidades del producto

Aplicación web **priorizada para navegador en PC (desktop-first)** que permite estudiar vocabulario de inglés por **categorías** y **módulos**. El usuario selecciona una categoría o módulo y realiza sesiones con **tarjetas**, **audio de pronunciación** y **dictado**; el sistema programa **repasos** y muestra un **panel de progreso**. Incluye un **panel de administración** para gestionar palabras, categorías y módulos, además de **analítica básica** y registro de actividad.

1.6.1. Requerimientos funcionales

* **RF-01 Autenticación básica: registro, inicio/cierre de sesión y recuperación de contraseña.**
* **RF-02 Perfil de usuario: ver/editar datos mínimos (nombre, correo) y preferencias de categorías/módulos.**
* **RF-03 Exploración de contenidos: listado y filtro de categorías y módulos; detalle de cada módulo (descripción y tamaño).**
* **RF-04 Sesión de estudio: presentación de tarjetas con definición, ejemplo y audio; controles por teclado.**
* **RF-05 Dictado/escucha: ejercicio breve por módulo (entrada de texto, verificación y retroalimentación).**
* **RF-06 Programación de repasos (SRS): reaparición de ítems en función del desempeño del usuario.**
* **RF-07 Progreso del usuario: vista con palabras estudiadas/dominadas, racha y tasa de aciertos básicas.**
* **RF-08 Historial de actividad: sesiones realizadas (fecha, duración aproximada, categorías/módulos practicados).**
* **RF-09 Búsqueda: local por palabra/lema dentro del contenido disponible.**
* **RF-10 Administración—contenido: CRUD de palabras, categorías y módulos; asociación palabra↔módulo/categoría.**
* **RF-11 Administración—medios: carga/validación de audio (formato, tamaño) y asociación a palabras.**
* **RF-12 Importación masiva: carga de lotes (CSV/JSON) con validación de campos obligatorios y reporte de errores.**
* **Campos mínimos: *lema, categoría, módulo, definición, ejemplo, ruta de audio*.**
* **RF-13 Estados editoriales: *borrador*, *listo*, *publicado* (solo *publicado* es visible al usuario final).**
* **RF-14 Analítica básica: métricas agregadas (usuarios activos, sesiones, errores) visibles en un panel interno.**
* **RF-15 Registro y auditoría: log de acciones administrativas (alta/edición/baja de contenido).**
* **RF-16 Accesibilidad esencial: navegación por teclado, foco visible y etiquetas para lectores de pantalla en flujos críticos.**

1.6.2. Requerimientos no funcionales

* **RNF-01 Rendimiento (desktop):** tiempo de carga inicial p75 **< 2.5 s** en conexión equivalente a 4G; interacción fluida en listas/tablas.
* **RNF-02 Estabilidad:** disponibilidad objetivo **≥ 99%** en el entorno de producción del MVP.
* **RNF-03 Seguridad:** contraseñas con **hash+salt**, protección ante XSS/CSRF, validación de entrada en servidor.
* **RNF-04 Compatibilidad:** soporte para las **2 últimas versiones** de Chrome, Edge, Firefox y Safari en escritorio; resolución mínima objetivo **1366×768**.
* **RNF-05 Accesibilidad:** cumplimiento **WCAG 2.1 AA** en flujos de estudio, autenticación y administración.
* **RNF-06 Escalabilidad inicial:** capacidad para **≥ 1,000** usuarios registrados y **≥ 50,000** ítems de vocabulario en BD sin degradación perceptible.
* **RNF-07 Mantenibilidad:** código con lint/format, **tests** básicos y **CI/CD** para build y despliegue reproducible.
* **RNF-08 Observabilidad:** **logging** estructurado, captura de errores (5xx) y métricas de uso con alertas mínimas.
* **RNF-09 Privacidad:** almacenamiento mínimo de datos personales y métricas **agregadas** en vistas administrativas.
* **RNF-10 Internacionalización preparada:** textos de interfaz externalizados (al menos español en MVP).

1.6.2. Requerimiento de interfaces externas

* **RIE-01 Almacenamiento/serving de audio:** repositorio o **CDN** accesible por **HTTPS** (GET), con control de tipo MIME (MP3/OGG) y tamaño máximo por archivo.
* **RIE-02 Analítica/monitorización:** integración con un servicio de métricas/errores (self-hosted o tercero) vía **SDK JavaScript** y/o **webhooks**; eventos mínimos: inicio de sesión, inicio/fin de sesión de estudio, error de aplicación.
* **RIE-03 Correo (recuperación de contraseña):** servidor **SMTP** configurable (host, puerto, TLS, credenciales) para envío de correos transaccionales; plantillas editables.

1.7. Arquitectura del Software

**Estilo.** Cliente–servidor con **SPA en navegador (desktop-first)** que consume una **API REST**.  
**Despliegue lógico (C4 – Contenedores):**

* **Navegador (SPA):** interfaz de estudio por **categorías/módulos**, tarjetas, audio/dictado y panel de progreso.
* **API de aplicación (monolito modular):** expone endpoints para autenticación, estudio, programación de repasos y administración de contenido.
* **Base de datos relacional:** persistencia de palabras, categorías, módulos, sesiones/eventos de estudio y progreso.
* **Almacenamiento de audio:** repositorio/CDN para servir archivos de pronunciación.

**Patrón interno del backend.** **Arquitectura Hexagonal (Puertos y Adaptadores) / Clean**:

* **Dominio:** reglas y modelos (Palabra, Categoría, Módulo, SesiónEstudio, EventoEstudio, ProgramaciónRepaso **SRS**).
* **Aplicación (servicios):** casos de uso (iniciar sesión de estudio, siguiente tarjeta, registrar respuesta, calcular próximo repaso, CRUD de contenido).
* **Infraestructura (adaptadores):** REST controllers, repositorios a BD, acceso a almacenamiento de audio y correo (recuperación).

**Módulos principales de dominio:**

* **Contenido:** gestión de palabras, categorías, módulos y asociación palabra↔módulo.
* **Estudio:** flujo de tarjetas, audio/dictado y validación de respuestas.
* **Repaso (SRS):** cálculo de intervalos y actualización de **ProgramaciónRepaso** según desempeño.
* **Progreso:** agregación de métricas por usuario (palabras estudiadas/dominadas, racha básica).
* **Administración:** altas/ediciones/bajas e importación de lotes (CSV/JSON).

**Flujo típico (texto):**  
SPA (PC) → API REST → Servicio de Aplicación → Dominio → Repositorio → BD  
← respuesta con ítem de estudio / estado de progreso

1.8. Modelo de casos de uso del sistema

**Actores**

* **Usuario** (estudiante/autodidacta): practica por **categorías** y **módulos**, consulta su progreso.
* **Administrador/Editor**: gestiona palabras, categorías, módulos y audio; publica contenido; revisa métricas.
* **Servicios externos** (actores secundarios):
  + **SMTP** (recuperación de contraseña).
  + **CDN/Repositorio de audio** (servir pronunciaciones).
  + **Analítica/errores** (registro de eventos y fallos).

**Casos de uso por actor**

**Usuario**

* **CU-01** Registrarse
* **CU-02** Iniciar sesión / Cerrar sesión
* **CU-03** Recuperar contraseña (vía correo)
* **CU-04** Explorar **categorías** y **módulos**
* **CU-05** Iniciar **sesión de estudio** (selección de módulo)
* **CU-06** Resolver **tarjeta** con definición, ejemplo y **audio** / **dictado** (registrar acierto/error)
* **CU-07** Finalizar sesión de estudio (guardar resumen)
* **CU-08** Ver **progreso** personal (palabras estudiadas/dominadas, racha)
* **CU-09** **Buscar** palabra

**Administrador/Editor**

* **CU-10** Gestionar **palabras** (CRUD)
* **CU-11** Gestionar **categorías** y **módulos** (CRUD)
* **CU-12** **Cargar/validar audio** y asociarlo a palabras
* **CU-13** **Importar** lote (CSV/JSON) con reporte de errores
* **CU-14** **Publicar** contenido (cambiar estado: borrador/listo/publicado)
* **CU-15** Ver **métricas agregadas** (usuarios, sesiones, errores)
* **CU-16** Ver **registro de acciones** administrativas

1.8.1. Diagramas de caso de uso del sistema

Diagrama general — casos de uso

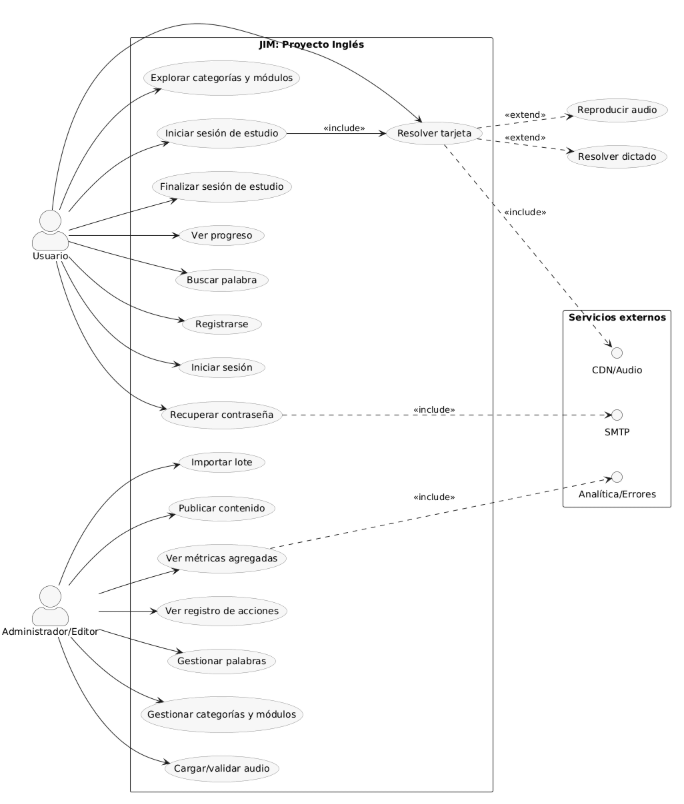


Diagrama — actor “Usuario”

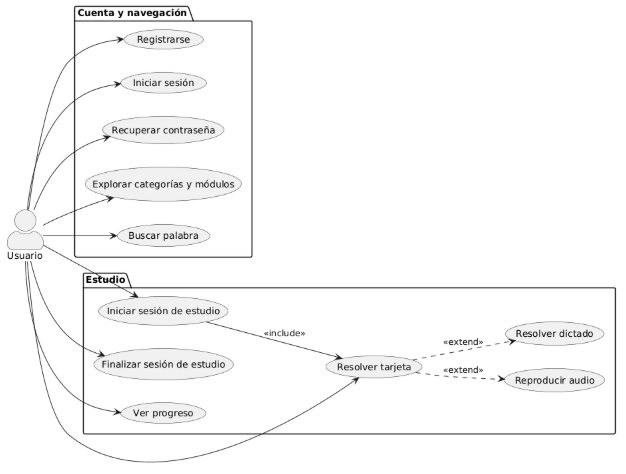
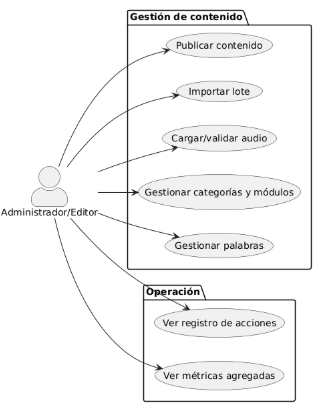


Diagrama — actor “Administrador/Editor”



1.8.2. Documentación de elementos del Modelo de Casos de uso

**1. Actores (diccionario)**

**A1. Usuario (estudiante/autodidacta)**

* **Rol:** Practica vocabulario, consulta su progreso.
* **Interacciones clave:** CU-04, CU-05, CU-06, CU-07, CU-08, CU-09, CU-01, CU-02, CU-03.
* **Características:** Usa PC (teclado/ratón); sesiones breves.

**A2. Administrador/Editor**

* **Rol:** Gestiona palabras, categorías, módulos y audio; publica contenido; revisa métricas.
* **Interacciones clave:** CU-10…CU-16.
* **Características:** Prefiere vistas tabulares, filtros y carga por lotes.

**A3. Servicios externos (secundarios)**

* **SMTP:** envío de correo para recuperación (CU-03).
* **CDN/Repositorio de audio:** servir pronunciaciones (CU-06).
* **Analítica/Errores:** recolección de eventos y fallos (CU-15).

**2. Especificación de Casos de Uso**

Formato por caso: **[ID] Nombre — Actor primario | Descripción | Precondiciones | Postcondiciones | Flujo básico | Alternos/Excepciones | Reglas/Datos | Prioridad**

**[CU-01] Registrarse — Usuario**

* **Descripción:** Crear cuenta con correo y contraseña.
* **Pre:** No autenticado.
* **Post:** Cuenta creada; sesión iniciada u opción de iniciar.
* **Flujo:** (1) Abrir “Crear cuenta” → (2) Ingresar datos → (3) Validar → (4) Guardar usuario → (5) Confirmación.
* **Alternos:** A1 correo ya registrado → mensaje; A2 validación falla → mostrar errores.
* **Reglas/Datos:** Usuario, validaciones básicas; RNF de seguridad.
* **Prioridad:** Media.

**[CU-02] Iniciar sesión — Usuario**

* **Descripción:** Autenticarse con credenciales.
* **Pre:** Usuario existente.
* **Post:** Sesión activa.
* **Flujo:** (1) Ingresar correo/contraseña → (2) Validar → (3) Crear sesión → (4) Redirigir a inicio.
* **Alternos:** Credenciales inválidas → error controlado; bloqueo por intentos.
* **Reglas/Datos:** Auth; rate-limit.
* **Prioridad:** Alta.

**[CU-03] Recuperar contraseña — Usuario**

* **Descripción:** Solicitar enlace de restablecimiento por correo.
* **Pre:** Cuenta registrada.
* **Post:** Token enviado por **SMTP**; contraseña actualizable.
* **Flujo:** (1) Ingresar correo → (2) Generar token → (3) Enviar email → (4) Confirmar envío.
* **Alternos:** Correo no registrado → respuesta genérica (no revelar existencia).
* **Reglas/Datos:** Token con expiración; SMTP.
* **Prioridad:** Media.

**[CU-04] Explorar categorías y módulos — Usuario**

* **Descripción:** Navegar categorías y abrir detalle de un módulo.
* **Pre:** Autenticado; módulos **publicados**.
* **Post:** Módulo seleccionado para estudiar; evento de vista registrado.
* **Flujo:** (1) Abrir “Explorar” → (2) Ver categorías → (3) Filtrar/buscar → (4) Abrir módulo.
* **Alternos:** Sin resultados → mostrar vacíos y sugerencias.
* **Reglas/Datos:** Categoría, Módulo (estado).
* **Prioridad:** Alta.

**[CU-05] Iniciar sesión de estudio — Usuario**

* **Descripción:** Comenzar una sesión en un módulo.
* **Pre:** Autenticado; módulo **publicado**.
* **Post:** SesiónEstudio creada; primer ítem listo.
* **Flujo:** (1) “Empezar” → (2) Crear SesiónEstudio → (3) Obtener primer ítem (SRS) → (4) Mostrar tarjeta.
* **Alternos:** Sin ítems pendientes → sugerir repaso u otro módulo.
* **Reglas/Datos:** SesiónEstudio, ProgramaciónRepaso.
* **Prioridad:** Alta.

**[CU-06] Resolver tarjeta (audio/dictado) — Usuario**

* **Descripción:** Responder tarjeta, escuchar audio y/o dictado; registrar acierto/error.
* **Pre:** Sesión de estudio activa; palabra **publicada**.
* **Post:** EventoEstudio guardado; SRS actualizado; siguiente tarjeta o fin.
* **Flujo:** (1) Ver tarjeta → (2) (Opcional) reproducir audio → (3) Responder → (4) Validar → (5) Feedback → (6) Registrar evento → (7) Recalcular repaso → (8) Siguiente/fin.
* **Alternos:** Falla de audio → continuar sin audio; respuesta vacía → pedir confirmación de salto.
* **Reglas/Datos:** Palabra, Audio, EventoEstudio, ProgramaciónRepaso; **RF-04, RF-05, RF-06**.
* **Prioridad:** Crítica.

**[CU-07] Finalizar sesión de estudio — Usuario**

* **Descripción:** Cerrar sesión y ver resumen.
* **Pre:** ≥ 1 tarjeta respondida.
* **Post:** Sesión cerrada; resumen visible; racha actualizada.
* **Flujo:** (1) Pulsar “Finalizar” → (2) Cerrar sesión → (3) Mostrar resumen (aciertos/tiempo).
* **Alternos:** Cierre inesperado → reanudar desde último estado si es posible.
* **Reglas/Datos:** SesiónEstudio, racha.
* **Prioridad:** Alta.

**[CU-08] Ver progreso — Usuario**

* **Descripción:** Ver palabras estudiadas/dominadas, racha y aciertos.
* **Pre:** Autenticado.
* **Post:** Sin cambios de estado.
* **Flujo:** (1) Abrir “Progreso” → (2) Consultar métricas.
* **Alternos:** Sin datos → mensajes de arranque.
* **Reglas/Datos:** Agregados de EventoEstudio; regla de “dominada”.
* **Prioridad:** Media.

**[CU-09] Buscar palabra — Usuario**

* **Descripción:** Buscar por lema dentro del contenido disponible.
* **Pre:** Autenticado; índice disponible.
* **Post:** Lista de coincidencias.
* **Flujo:** (1) Ingresar término → (2) Mostrar resultados → (3) Ir a detalle/módulo.
* **Alternos:** Sin resultados → alternativas/sugerencias.
* **Reglas/Datos:** Índice de palabras; normalización.
* **Prioridad:** Media.

**Administración**

**[CU-10] Gestionar palabras (CRUD) — Administrador/Editor**

* **Descripción:** Crear/editar/eliminar palabras (lema, definición, ejemplo, etiqueta gramatical opcional) y asociaciones a módulos.
* **Pre:** Sesión con rol **Editor/Admin**.
* **Post:** Palabra persistida; acción registrada.
* **Flujo:** (1) Abrir tabla → (2) Crear/editar → (3) Validar → (4) Guardar → (5) Asociar a módulos.
* **Alternos:** Campos inválidos → reporte de errores.
* **Reglas/Datos:** Unicidad; estados editoriales; **RF-10, RF-15**.
* **Prioridad:** Alta.

**[CU-11] Gestionar categorías y módulos (CRUD) — Administrador/Editor**

* **Descripción:** Crear/editar/eliminar categorías y módulos; asignar descripciones y tamaño estimado.
* **Pre:** Rol Editor/Admin.
* **Post:** Estructura actualizada.
* **Flujo:** (1) Abrir administración → (2) CRUD categoría/módulo → (3) Guardar.
* **Alternos:** Eliminación con dependencias → bloqueo/confirmación.
* **Reglas/Datos:** Integridad referencial; **RF-10**.
* **Prioridad:** Alta.

**[CU-12] Cargar/validar audio — Administrador/Editor**

* **Descripción:** Subir archivo MP3/OGG y asociarlo a una palabra.
* **Pre:** Palabra existente; archivo válido.
* **Post:** Audio cargado en repositorio/CDN; URL guardada.
* **Flujo:** (1) Seleccionar archivo → (2) Validar formato/tamaño → (3) Subir → (4) Asociar → (5) Confirmar.
* **Alternos:** Fallo de subida → reintento; archivo inválido → rechazo con causa.
* **Reglas/Datos:** RIE-01; **RF-11**.
* **Prioridad:** Media.

**[CU-13] Importar lote (CSV/JSON) — Administrador/Editor**

* **Descripción:** Cargar palabras en bloque con validación y reporte de errores.
* **Pre:** Plantilla con columnas mínimas: *lema, categoría, módulo, definición, ejemplo, ruta\_audio*.
* **Post:** Registros válidos creados en **borrador**; reporte de rechazados.
* **Flujo:** (1) Subir archivo → (2) Previsualizar → (3) Validar → (4) Importar → (5) Descargar reporte.
* **Alternos:** Archivo mal formado → abortar con detalles.
* **Reglas/Datos:** **RF-12**, unicidad, estados.
* **Prioridad:** Media.

**[CU-14] Publicar contenido — Administrador/Editor**

* **Descripción:** Cambiar estado a **publicado** para que sea visible a usuarios.
* **Pre:** Ítems en estado **listo**.
* **Post:** Contenido disponible en exploración/estudio.
* **Flujo:** (1) Seleccionar ítems → (2) Publicar → (3) Confirmar.
* **Alternos:** Ítems incompletos → bloqueo con motivo.
* **Reglas/Datos:** Estados editoriales; **RF-13**.
* **Prioridad:** Alta.

**[CU-15] Ver métricas agregadas — Administrador/Editor**

* **Descripción:** Consultar panel con usuarios activos, sesiones, errores.
* **Pre:** Rol autorizado; integraciones de analítica activas.
* **Post:** Sin cambios; solo lectura.
* **Flujo:** (1) Abrir panel → (2) Filtros → (3) Visualizar.
* **Alternos:** Falta de datos → placeholders.
* **Reglas/Datos:** RNF-08; privacidad agregada.
* **Prioridad:** Media.

**[CU-16] Ver registro de acciones — Administrador/Editor**

* **Descripción:** Auditar acciones administrativas (altas/ediciones/bajas).
* **Pre:** Rol autorizado.
* **Post:** Sin cambios; solo lectura.
* **Flujo:** (1) Abrir “Registro” → (2) Filtrar por actor/fecha → (3) Revisar.
* **Alternos:** Sin registros → vacío informativo.
* **Reglas/Datos:** **RF-15**; retención acorde a políticas.
* **Prioridad:** Media.

**3. Relaciones entre Casos de Uso**

* **<<include>>**:
  + **CU-05 Iniciar sesión de estudio** incluye **CU-06 Resolver tarjeta**.
  + **CU-03 Recuperar contraseña** incluye uso de **SMTP**.
* **<<extend>>**:
  + **CU-06 Resolver tarjeta** puede **extender**:
    - *Reproducir audio* (cuando hay audio disponible).
    - *Resolver dictado* (cuando el ítem lo requiere).
* **Dependencias externas:**
  + **CDN/Audio** en CU-06 para servir pronunciaciones.
  + **Analítica/Errores** en CU-15 para panel de métricas.

1.9 Especificaciones de Casos de Uso

**CU-01 — Registrarse**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Usuario |
| **Propósito** | Crear una cuenta con correo y contraseña. |
| **Precondiciones** | No autenticado. |
| **Postcondiciones** | Usuario creado; sesión iniciada o listo para iniciar sesión. |
| **Flujo principal** | 1) Abrir “Crear cuenta” → 2) Ingresar correo/contraseña → 3) Validar → 4) Guardar → 5) Confirmación. |
| **Alternos/Excepciones** | Correo ya registrado → mensaje; validación falla → mostrar errores. |
| **Reglas/Refs.** | RF-01, RNF-03 |
| **Datos** | Usuario{correo, hash\_password, nombre?} |
| **Criterios de aceptación** | Registro único y redirección correcta. |
| **Prioridad** | Media |

**CU-02 — Iniciar sesión**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Usuario |
| **Propósito** | Autenticarse en el sistema. |
| **Precondiciones** | Usuario existente. |
| **Postcondiciones** | Sesión activa; acceso al inicio. |
| **Flujo principal** | 1) Ingresar credenciales → 2) Validar → 3) Crear sesión (cookie/token) → 4) Redirigir. |
| **Alternos/Excepciones** | Credenciales inválidas; bloqueo temporal por intentos fallidos. |
| **Reglas/Refs.** | RF-01, RNF-03 |
| **Datos** | Usuario, Sesión |
| **Criterios de aceptación** | Login exitoso bajo HTTPS en navegadores soportados. |
| **Prioridad** | Alta |

**CU-03 — Recuperar contraseña**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Usuario |
| **Propósito** | Restablecer contraseña vía enlace enviado por correo. |
| **Precondiciones** | Cuenta registrada. |
| **Postcondiciones** | Token emitido; contraseña actualizada. |
| **Flujo principal** | 1) Ingresar correo → 2) Generar token con expiración → 3) Enviar email (SMTP) → 4) Abrir enlace → 5) Definir nueva contraseña. |
| **Alternos/Excepciones** | Correo no registrado → respuesta neutra. |
| **Reglas/Refs.** | RF-01, RIE-03, RNF-03 |
| **Datos** | TokenRecuperación |
| **Criterios de aceptación** | Enlace con expiración; cambio registrado. |
| **Prioridad** | Media |

**CU-04 — Explorar categorías y módulos**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Usuario |
| **Propósito** | Navegar categorías y abrir detalle de un módulo. |
| **Precondiciones** | Autenticado; módulos **publicados**. |
| **Postcondiciones** | Módulo seleccionado. |
| **Flujo principal** | 1) Abrir “Explorar” → 2) Ver categorías → 3) Filtrar/buscar → 4) Abrir módulo. |
| **Alternos/Excepciones** | Sin resultados → vacíos y sugerencias. |
| **Reglas/Refs.** | RF-03, RF-13 |
| **Datos** | Categoría, Módulo(estado) |
| **Criterios de aceptación** | Listas paginadas y rápidas en desktop. |
| **Prioridad** | Alta |

**CU-05 — Iniciar sesión de estudio**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Usuario |
| **Propósito** | Comenzar estudio en un módulo. |
| **Precondiciones** | Autenticado; módulo **publicado**. |
| **Postcondiciones** | SesiónEstudio creada; primer ítem listo. |
| **Flujo principal** | 1) “Empezar” → 2) Crear SesiónEstudio → 3) Obtener primer ítem (SRS) → 4) Mostrar tarjeta. |
| **Alternos/Excepciones** | Sin ítems pendientes → sugerir repaso u otro módulo. |
| **Reglas/Refs.** | RF-04, RF-06 |
| **Datos** | SesiónEstudio |
| **Criterios de aceptación** | Inicio en <1 s con primer ítem cargado. |
| **Prioridad** | Alta |

**CU-06 — Resolver tarjeta (audio/dictado)**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Usuario |
| **Propósito** | Responder tarjeta, registrar acierto/error y actualizar repaso. |
| **Precondiciones** | SesiónEstudio activa; palabra **publicada**. |
| **Postcondiciones** | EventoEstudio guardado; SRS actualizado; siguiente tarjeta o fin. |
| **Flujo principal** | 1) Ver tarjeta → 2) (Opc.) reproducir audio → 3) Responder/Dictado → 4) Validar → 5) Feedback → 6) Registrar evento → 7) Recalcular repaso → 8) Siguiente/fin. |
| **Alternos/Excepciones** | Falla de audio → continuar sin audio; respuesta vacía → confirmar salto. |
| **Reglas/Refs.** | RF-04, RF-05, RF-06 |
| **Datos** | Palabra, Audio, EventoEstudio, ProgramaciónRepaso |
| **Criterios de aceptación** | Atajos de teclado operan; validación <200 ms. |
| **Prioridad** | Crítica |

**CU-07 — Finalizar sesión de estudio**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Usuario |
| **Propósito** | Cerrar sesión y ver resumen. |
| **Precondiciones** | ≥ 1 tarjeta respondida. |
| **Postcondiciones** | Sesión cerrada; resumen visible; racha actualizada. |
| **Flujo principal** | 1) Pulsar “Finalizar” → 2) Cerrar SesiónEstudio → 3) Mostrar resumen (aciertos/tiempo). |
| **Alternos/Excepciones** | Cierre inesperado → reanudar o marcar parcial. |
| **Reglas/Refs.** | RF-07, RF-08 |
| **Datos** | SesiónEstudio(resumen) |
| **Criterios de aceptación** | Resumen persistido y visible en Historial. |
| **Prioridad** | Alta |

**CU-08 — Ver progreso**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Usuario |
| **Propósito** | Consultar palabras estudiadas/dominadas, racha y aciertos. |
| **Precondiciones** | Autenticado. |
| **Postcondiciones** | Solo lectura. |
| **Flujo principal** | 1) Abrir “Progreso” → 2) Ver métricas. |
| **Alternos/Excepciones** | Sin datos → mensajes de arranque. |
| **Reglas/Refs.** | RF-07 |
| **Datos** | Agregados de EventoEstudio |
| **Criterios de aceptación** | Carga <1.5 s; coherencia con sesiones. |
| **Prioridad** | Media |

**CU-09 — Buscar palabra**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Usuario |
| **Propósito** | Encontrar palabra por lema y acceder a su módulo. |
| **Precondiciones** | Autenticado; índice disponible. |
| **Postcondiciones** | Resultados listados; navegación al módulo. |
| **Flujo principal** | 1) Ingresar término → 2) Ver coincidencias → 3) Ir a detalle/módulo. |
| **Alternos/Excepciones** | Sin resultados → sugerencias. |
| **Reglas/Refs.** | RF-09 |
| **Datos** | ÍndicePalabras |
| **Criterios de aceptación** | Búsqueda tolerante a acentos/casos. |
| **Prioridad** | Media |

**CU-10 — Gestionar palabras (CRUD)**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Administrador/Editor |
| **Propósito** | Alta/edición/baja de palabras y asociaciones a módulos. |
| **Precondiciones** | Rol Editor/Admin. |
| **Postcondiciones** | Cambios persistidos; acción auditada. |
| **Flujo principal** | 1) Abrir tabla → 2) Crear/editar (lema, definición, ejemplo, etiqueta opc.) → 3) Validar → 4) Guardar → 5) Asociar a módulos. |
| **Alternos/Excepciones** | Campos inválidos/duplicados → rechazo con detalle. |
| **Reglas/Refs.** | RF-10, RF-15; unicidad por lema+módulo/categoría. |
| **Datos** | Palabra, WordModule, AcciónAdmin |
| **Criterios de aceptación** | Operaciones <800 ms; validaciones visibles en UI. |
| **Prioridad** | Alta |

**CU-11 — Gestionar categorías y módulos (CRUD)**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Administrador/Editor |
| **Propósito** | Crear/editar/eliminar categorías y módulos. |
| **Precondiciones** | Rol Editor/Admin. |
| **Postcondiciones** | Estructura actualizada. |
| **Flujo principal** | 1) Abrir administración → 2) CRUD categoría/módulo (nombre, descripción, tamaño, estado) → 3) Guardar. |
| **Alternos/Excepciones** | Eliminación con dependencias → confirmar o bloquear. |
| **Reglas/Refs.** | RF-10, RF-13 |
| **Datos** | Categoría, Módulo |
| **Criterios de aceptación** | Integridad referencial mantenida. |
| **Prioridad** | Alta |

**CU-12 — Cargar/validar audio**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Administrador/Editor |
| **Propósito** | Subir audio y asociarlo a una palabra. |
| **Precondiciones** | Palabra existente; archivo válido. |
| **Postcondiciones** | Audio en repositorio/CDN; URL guardada. |
| **Flujo principal** | 1) Seleccionar archivo → 2) Validar formato/tamaño → 3) Subir → 4) Guardar URL → 5) Confirmar. |
| **Alternos/Excepciones** | Fallo de subida → reintento; formato inválido → rechazo. |
| **Reglas/Refs.** | RF-11, RIE-01 |
| **Datos** | Audio{url, duracion}, Palabra |
| **Criterios de aceptación** | Validación previa al upload; vista previa reproducible. |
| **Prioridad** | Media |

**CU-13 — Importar lote (CSV/JSON)**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Administrador/Editor |
| **Propósito** | Alta masiva con validación y reporte. |
| **Precondiciones** | Plantilla con columnas mínimas (*lema, categoría, módulo, definición, ejemplo, ruta\_audio*). |
| **Postcondiciones** | Registros válidos en **borrador**; reporte de errores. |
| **Flujo principal** | 1) Subir archivo → 2) Previsualizar → 3) Validar → 4) Importar → 5) Descargar reporte. |
| **Alternos/Excepciones** | Archivo mal formado → abortar con detalles. |
| **Reglas/Refs.** | RF-12 |
| **Datos** | LoteImportación, AcciónAdmin |
| **Criterios de aceptación** | Job seguro; no duplicar registros. |
| **Prioridad** | Media |

**CU-14 — Publicar contenido**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Administrador/Editor |
| **Propósito** | Hacer visible contenido a usuarios. |
| **Precondiciones** | Ítems en estado **listo**. |
| **Postcondiciones** | Estado **publicado**; visible en exploración/estudio. |
| **Flujo principal** | 1) Seleccionar ítems → 2) Publicar → 3) Confirmar. |
| **Alternos/Excepciones** | Ítems incompletos → bloqueo con motivo. |
| **Reglas/Refs.** | RF-13 |
| **Datos** | EstadoEditorial |
| **Criterios de aceptación** | Efecto inmediato en listas del usuario. |
| **Prioridad** | Alta |

**CU-15 — Ver métricas agregadas**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Administrador/Editor |
| **Propósito** | Consultar panel de uso y errores. |
| **Precondiciones** | Rol autorizado; analítica activa. |
| **Postcondiciones** | Solo lectura. |
| **Flujo principal** | 1) Abrir panel → 2) Aplicar filtros → 3) Visualizar KPIs. |
| **Alternos/Excepciones** | Falta de datos → placeholders. |
| **Reglas/Refs.** | RF-14, RNF-08 |
| **Datos** | Métricas agregadas |
| **Criterios de aceptación** | No exponer PII; carga <2 s. |
| **Prioridad** | Media |

**CU-16 — Ver registro de acciones**

| **Campo** | **Detalle** |
| --- | --- |
| **Actor** | Administrador/Editor |
| **Propósito** | Auditar operaciones administrativas. |
| **Precondiciones** | Rol autorizado. |
| **Postcondiciones** | Solo lectura. |
| **Flujo principal** | 1) Abrir “Registro” → 2) Filtrar por actor/fecha/entidad → 3) Revisar entradas. |
| **Alternos/Excepciones** | Sin registros → vacío informativo. |
| **Reglas/Refs.** | RF-15 |
| **Datos** | AcciónAdmin{usuario, entidad, tipo, timestamp} |
| **Criterios de aceptación** | Orden cronológico; exportación (opcional) si se define. |
| **Prioridad** | Media |

1.9.1. Especificación de Alto Nivel

**Alcance funcional por módulos**

| **Módulo** | **Descripción** | **Principales entradas** | **Principales salidas** |
| --- | --- | --- | --- |
| Autenticación & Perfil | Registro, inicio de sesión, perfil básico. | email/clave o SSO, datos perfil | token sesión, perfil |
| Catálogo de Módulos | Listar/buscar categorías y lecciones. | filtros, búsqueda | lista de módulos/lecciones |
| Práctica/Ejercicios | Ejercicios de vocabulario, gramática, listening, speaking. | respuestas, audio del usuario (opcional) | corrección, puntaje, sugerencias |
| Retroalimentación | Feedback inmediato + explicación. | intento del usuario | resultado, explicación, hint |
| Progreso & Analítica | Seguimiento por módulo/competencia. | intentos, tiempo | tablero de progreso, recomendaciones |
| Gamificación (opcional MVP+) | Puntos, rachas, badges. | acciones del usuario | métricas de juego |
| Notificaciones | Recordatorios (web push/email). | preferencia de usuario | notificaciones |
| Administración de Contenido (CMS) | Crea/edita módulos, lecciones, bancos de ítems. | contenido, metadatos | contenido publicado/estado |
| Integraciones | TTS/STT, correo/push, (opcional pagos). | textos/audio, plantillas | audio sintético, envíos |

**Roles de usuario**

| **Rol** | **Permisos clave** |
| --- | --- |
| Invitado | Navega catálogo limitado, prueba 1–2 ejercicios. |
| Estudiante | Acceso completo a práctica, progreso, notificaciones. |
| Admin/Editor | CRUD de contenido, gestión de usuarios, métricas. |

**Datos clave (entidades)**

Usuario, Módulo, Lección, Actividad (ítem), Intento, Resultado, Progreso, Recurso (audio/imagen), Notificación, Parámetro del sistema, Rol/Permiso.

**Reglas de negocio esenciales (resumen)**

1. El usuario **elige módulo/categoría** libremente (no hay prerequisitos de nivel).
2. Cada **Actividad** define tipo (MCQ, completar, dictado, speaking), puntaje y criterios de corrección.
3. **Feedback inmediato** obligatorio tras enviar la respuesta.
4. **Intento** queda registrado con sello de tiempo, fuente (web/app), duración.
5. **Progreso** se calcula por módulo (≥70% = “competencia alcanzada”, configurable).
6. **Contenido publicado** no editable sin nueva versión.
7. Invitado no acumula progreso persistente.
8. Datos personales sujetos a consentimiento y eliminación bajo solicitud.

**Interfaces (vista 10,000 pies)**

* **Cliente web responsivo** (SPA).
* **API** REST/JSON (o GraphQL) para catálogo, práctica, progreso y CMS.
* **BD relacional** para datos transaccionales; **almacenamiento de medios** para audio/imagen.
* **Servicios externos**: TTS/STT, correo/push (y pagos si aplica).

**Atributos de calidad (metas)**

| **Atributo** | **Meta inicial** |
| --- | --- |
| Disponibilidad | 99.5% mensual |
| Rendimiento | P95 < 300 ms para GET catálogo; P95 < 800 ms envío de respuesta |
| Seguridad | OWASP ASVS nivel básico; MFA opcional admin |
| Privacidad | Minimización de datos; retención de intentos 18 meses (configurable) |
| Accesibilidad | WCAG 2.1 AA en UI |
| Portabilidad | Infra agnóstica; contenedores |

**Suposiciones y restricciones**

* MVP web (mobile-first). App nativa: futuro.
* Conectividad requerida para práctica con audio.
* Idioma UI: ES; contenido didáctico: EN (con indicaciones en ES).

1.9.2. Especificación Expandida

**1) Requerimientos funcionales detallados (con criterios de aceptación)**

| **ID** | **Descripción** | **Prioridad** | **Criterios de aceptación (resumen)** |
| --- | --- | --- | --- |
| FR-01 | Registro e inicio de sesión | Alta | Dado email/clave válidos → crea sesión y devuelve token; errores claros en 400/401. |
| FR-02 | Ver catálogo de módulos | Alta | GET /modules devuelve lista paginada con id,title,tags,stats; P95 <300 ms con 10k módulos. |
| FR-03 | Buscar/filtrar módulos | Media | q, tag, skill filtran resultados; sin resultados → mensaje “no se encontraron…”. |
| FR-04 | Iniciar práctica | Alta | POST /sessions crea sesión de práctica con moduleId y la primera actividad. |
| FR-05 | Enviar respuesta | Alta | POST /attempts guarda intento; devuelve isCorrect, explicación, y siguiente actividad (si existe). |
| FR-06 | Feedback inmediato | Alta | Para MCQ/Texto: explicación + corrección; para audio: score y transcripción (si STT activo). |
| FR-07 | Ver progreso | Alta | GET /progress/me muestra % por módulo y racha (si aplica); exactitud ±1%. |
| FR-08 | Recomendaciones | Media | Tras 5+ intentos en un módulo, GET /recommendations sugiere próxima lección según errores frecuentes. |
| FR-09 | CMS: CRUD contenido | Alta (Admin) | Editor puede crear/editar borradores; publicar crea nueva versión inmutable. |
| FR-10 | Notificaciones | Media | Usuario opt-in/out; se registran y envían recordatorios diarios/semanales configurables. |

Nota: Ajusta IDs a tu lista oficial si ya la tienes.

**2) Flujos clave (resumen paso a paso)**

**UC-01 Practicar por módulo**

1. Usuario elige módulo → 2) Sistema crea sesión → 3) Muestra Actividad N → 4) Usuario responde → 5) Sistema corrige y da feedback → 6) Avanza hasta fin → 7) Registra progreso.

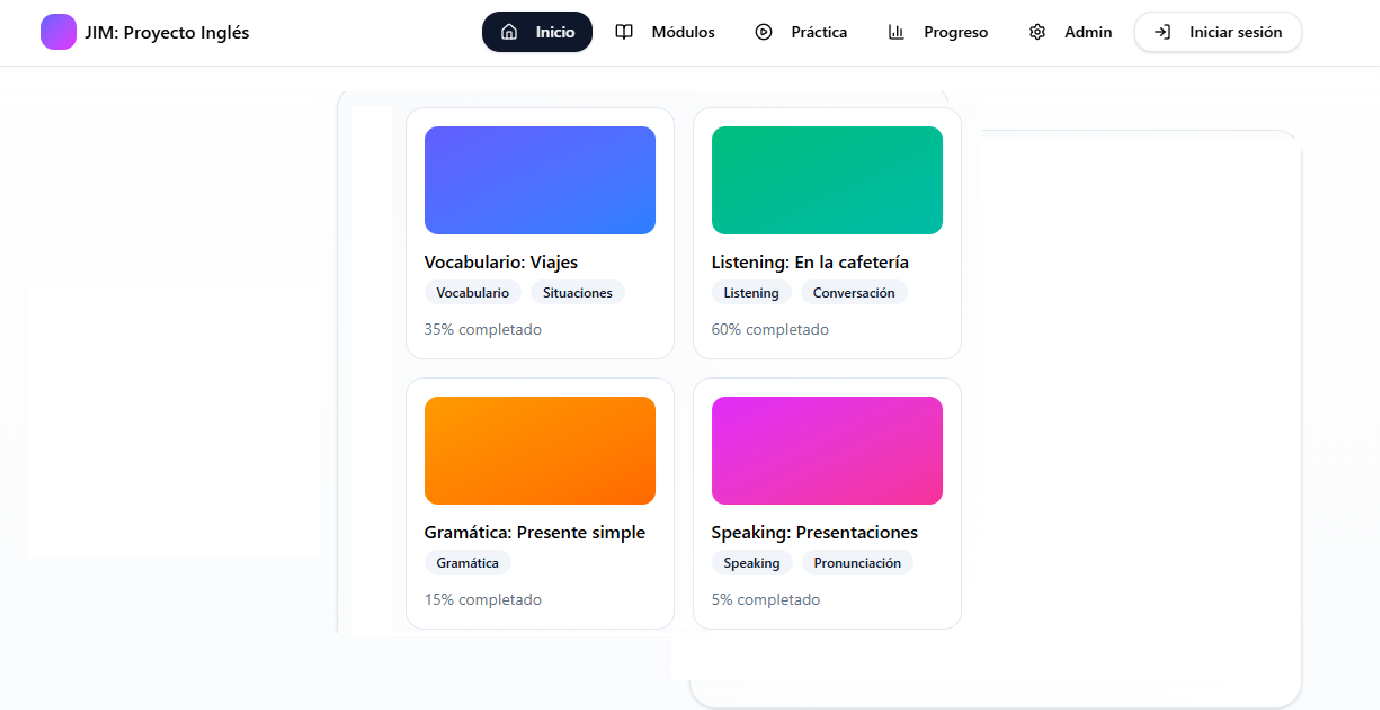
**UC-02 Revisión de resultados**

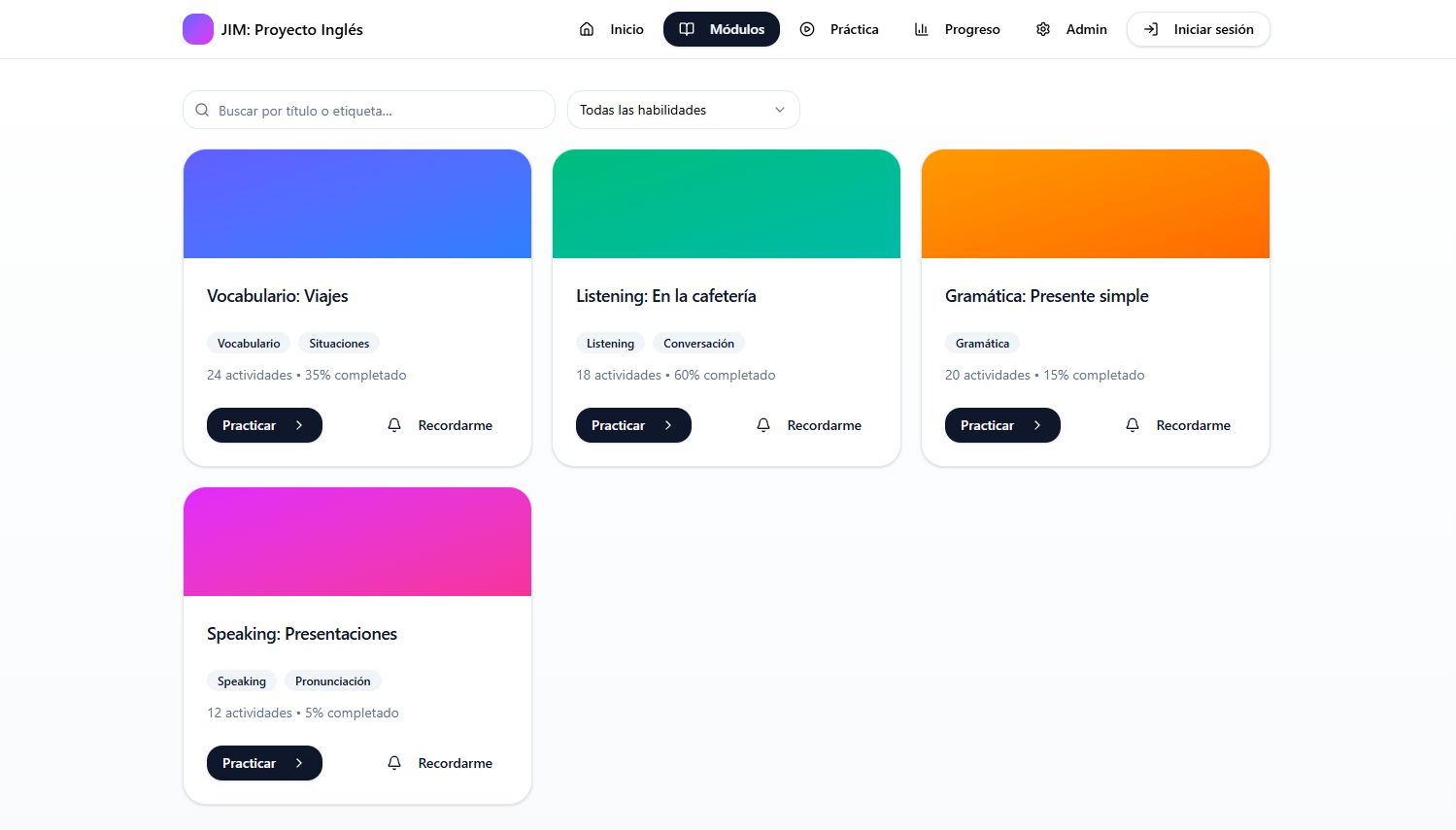
1. Usuario abre “Mi Progreso” → 2) Sistema agrega intentos por módulo → 3) Muestra % dominio, errores frecuentes y enlaces “reintentar”.

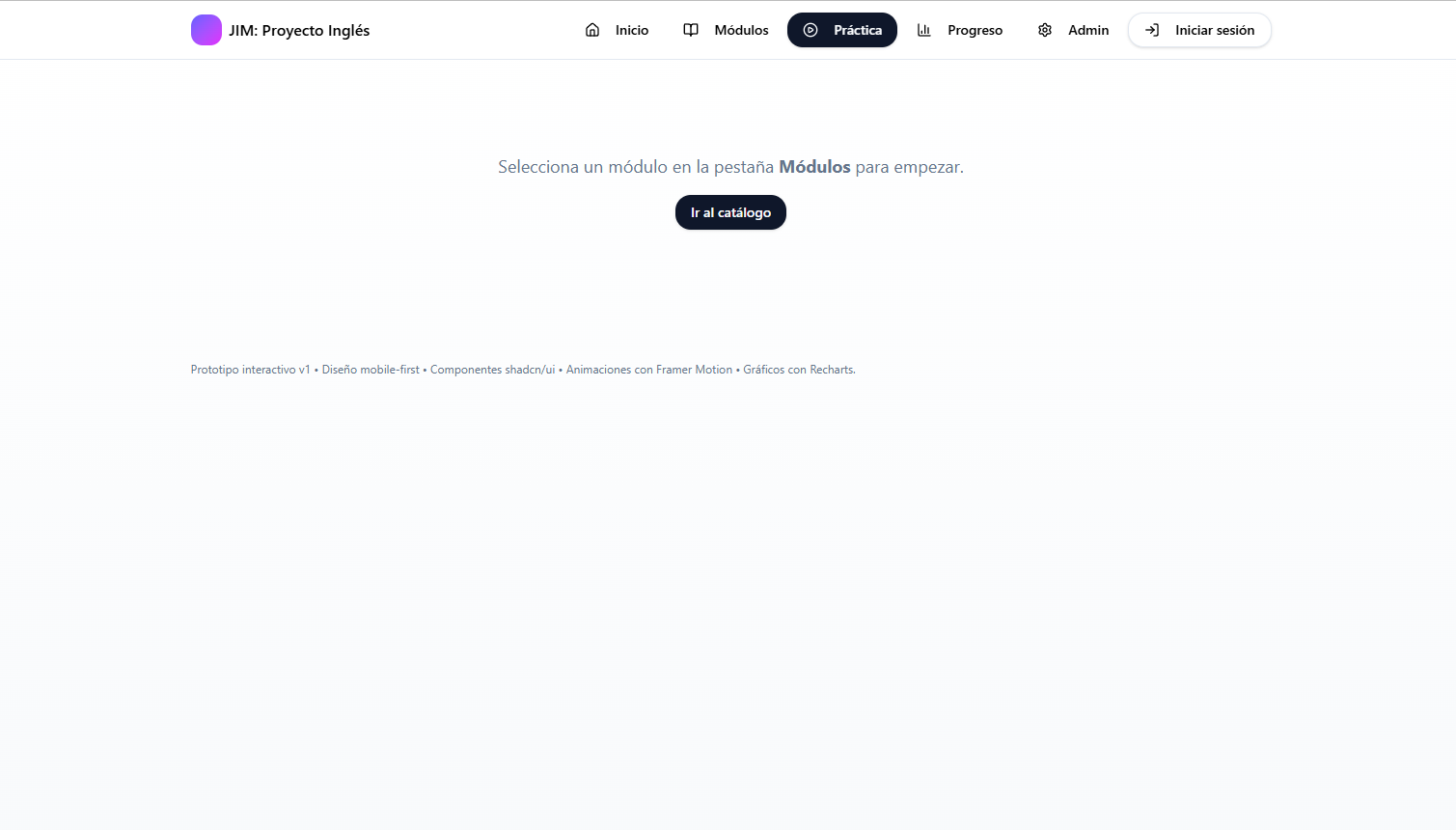
**UC-03 Admin publica contenido**

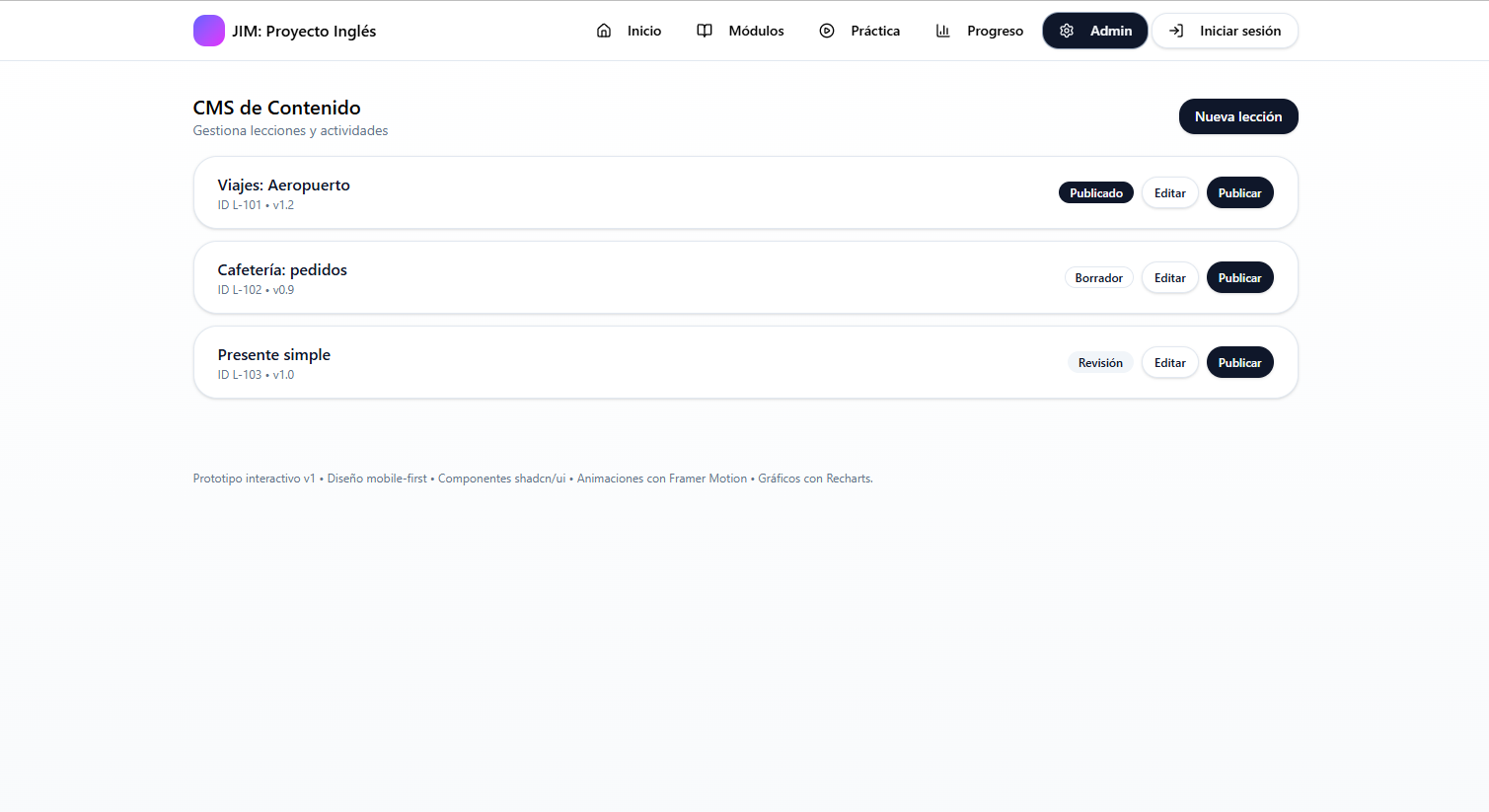
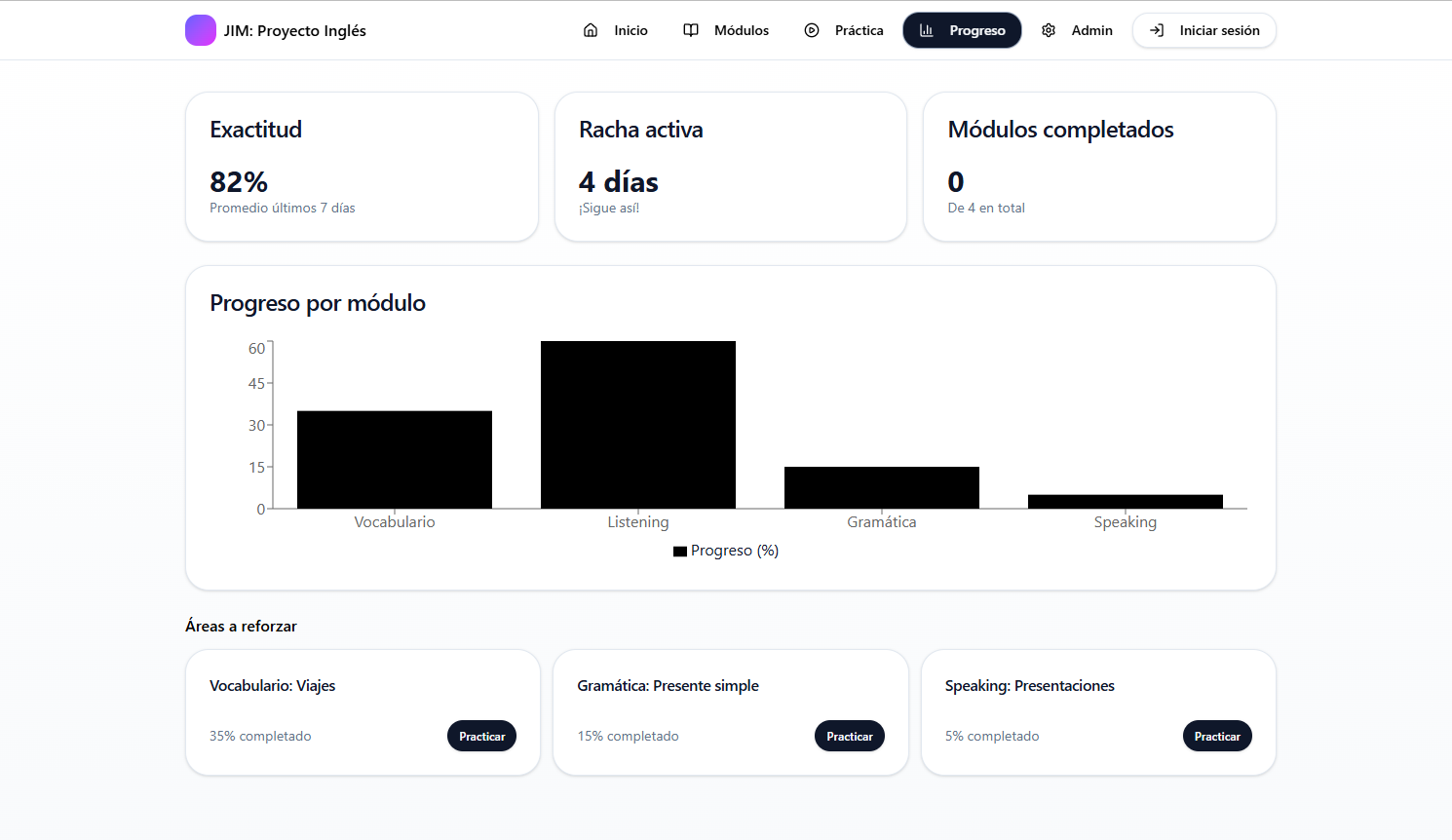
1. Admin crea lección (borrador) → 2) Valida ítems → 3) Previsualiza → 4) Publica versión → 5) Histórico conserva versiones.

1.9.3. Diseño de prototipos









1.10 Priorización de Casos de Uso.

| **ID** | **Caso de Uso** | **Actor(es)** | **Descripción breve** | **Prioridad** | **Justificación de valor** | **Riesgo / Complejidad** | **Versión objetivo** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| UC01 | Registro de usuario | Estudiante | Crear cuenta con email/Google/Apple | Must | Puerta de entrada al producto | Media (OAuth/validaciones) | MVP |
| UC02 | Inicio de sesión | Estudiante | Autenticarse y mantener sesión | Must | Acceso básico | Media (tokens, sesiones) | MVP |
| UC03 | Recuperación de contraseña | Estudiante | Restablecer por email | Should | Reduce fricción/soporte | Baja | v1.0 |
| UC04 | Gestionar perfil | Estudiante | Editar nombre, idioma, foto, objetivos | Should | Personalización y retención | Baja | v1.0 |
| UC05 | Seleccionar **categoría/módulo** | Estudiante | Explorar y elegir módulo de práctica | Must | Núcleo de navegación de contenidos | Baja | MVP |
| UC06 | Iniciar práctica | Estudiante | Crear sesión de ejercicios del módulo | Must | Núcleo de valor (practicar) | Media | MVP |
| UC07 | Resolver ejercicio con **feedback inmediato** | Estudiante | Pregunta→respuesta→corrección | Must | Aprendizaje efectivo | Media | MVP |
| UC08 | Ver resultados de la práctica | Estudiante | Puntaje, aciertos/errores, sugerencias | Must | Cierre del ciclo de aprendizaje | Baja | MVP |
| UC09 | Ver progreso/estadísticas | Estudiante | Historial, rachas, tiempo | Must | Retención y motivación | Media | MVP |
| UC10 | Continuar donde me quedé | Estudiante | Reanudar práctica pendiente | Must | Reduce abandono | Baja | MVP |
| UC11 | Búsqueda de contenidos | Estudiante | Buscar por palabra clave | Could | Útil con catálogo grande | Media | v1.x |
| UC12 | Notificaciones/recordatorios | Estudiante | Configurar y recibir recordatorios | Should | Mejora de engagement | Media (programación) | v1.0 |
| UC13 | Práctica de pronunciación (voz) | Estudiante | Grabar y obtener evaluación | Could | Diferenciador | Alta (STT/TTS) | v1.1 |
| UC14 | Administración de contenidos (CRUD) | Admin | Crear/editar módulos y ejercicios | Should | Escalabilidad operativa | Media | v1.0 |
| UC15 | Gestión de usuarios (bloq./roles) | Admin | Ver y ajustar cuentas/roles | Could | Soporte/seguridad | Media | v1.1 |
| UC16 | Reporte de errores/feedback | Estudiante | Enviar comentarios desde app | Should | Mejora continua | Baja | v1.0 |
| UC17 | Recomendaciones personalizadas | Estudiante | Sugerir siguiente módulo | Should | Aumenta NPS/retención | Alta (reglas/modelo) | v1.1 |
| UC18 | Gamificación (logros/insignias) | Estudiante | Desbloquear logros, rachas | Should | Engagement | Media | v1.1 |
| UC19 | Modo offline (limitado) | Estudiante | Practicar sin conexión | Could | Caso de uso móvil | Alta (sync) | v1.x |
| UC20 | Suscripción/pagos (si aplica) | Estudiante | Mejoras premium | Won’t (ahora) | No bloquea aprendizaje inicial | Alta (pasarela) | — |

1.10.1 Clasificación de los Casos de Uso

| **Categoría** | **Casos de Uso** |
| --- | --- |
| **Autenticación & Perfil** | UC01, UC02, UC03, UC04 |
| **Aprendizaje (core)** | UC05, UC06, UC07, UC08, UC09, UC10, UC13 |
| **Descubrimiento & Engagement** | UC11, UC12, UC17, UC18 |
| **Soporte & Calidad** | UC16 |
| **Administración** | UC14, UC15 |
| **Comercial** | UC20 |
| **Operación avanzada** | UC19 |

1.11 Flujo General de Navegación

**Mapa de alto nivel (app web/móvil):**

* **Onboarding/Splash** → **Autenticación** (Registrarse / Iniciar sesión / Omitir si ya hay sesión)
* **Inicio (Home)**
  + CTA “**Continuar donde quedaste**” (UC10)
  + Carruseles por **Categoría** y **Módulos** (UC05)
  + **Búsqueda** (UC11)
  + Atajos: **Recomendado para ti** (UC17), **Reanudar racha** (UC18)
* **Detalle de Módulo**
  + Descripción, objetivos, tiempo estimado
  + Botón **Iniciar práctica** (UC06)
  + Lista de subunidades/temas
* **Sesión de Práctica**
  + Secuencia de ejercicios → **Respuesta** → **Feedback inmediato** (UC07)
  + Pausar/guardar progreso → retorna a **Home** (UC10)
* **Resultados**
  + Puntaje, errores frecuentes, **Revisar explicación**
  + CTA **Reintentar** o **Siguiente recomendado** (UC08, UC17)
* **Progreso**
  + Estadísticas, racha, logros (UC09, UC18)
* **Perfil & Ajustes**
  + Objetivo diario, recordatorios (UC12), datos personales (UC04)
* **Admin (según rol)**
  + **Contenido**: módulos/ejercicios (UC14)
  + **Usuarios** (UC15)

**Flujos especiales**

* **Sesión expirada** → retorno a Login con preservación de intención.
* **Sin conexión** → sólo módulos descargados (UC19) + cola de sincronización.
* **Errores** → Snackbar/toast + opción **Reportar** (UC16).

1.12 Esquema de Seguridad

**Autenticación y sesiones**

* Web: sesiones con **cookies HttpOnly, Secure, SameSite**, + **CSRF token**.
* API: **JWT de acceso** (vida corta) + **Refresh tokens** rotativos con lista de revocación.
* Passwords: **Argon2id** o **bcrypt** con salt único. Política de complejidad y bloqueo por intentos.
* Opcional: **MFA** (TOTP/email) para Admin.

**Autorización (RBAC)**

* Roles: **Estudiante**, **Admin**.
* Regla por recurso y acción (ABAC ligero para propiedad del dato).

| **Recurso / Acción** | **Estudiante** | **Admin** |
| --- | --- | --- |
| Ver módulos/ejercicios | ✅ | ✅ |
| Resolver prácticas | ✅ | — |
| Ver/editar propio perfil | ✅ | — |
| Ver/editar progreso propio | ✅ | — |
| Crear/editar/eliminar contenidos | ❌ | ✅ |
| Ver/gestionar usuarios | ❌ | ✅ |
| Ver reportes agregados | ❌ | ✅ |

**Protecciones de capa aplicación**

* Validación/normalización de entradas, **limitación de tasa** (login, feedback, búsquedas).
* **CORS** restrictivo por dominio.
* **Cabeceras** de seguridad (CSP, HSTS, X-Content-Type-Options, etc.).
* **Escapes** y ORM para prevenir **XSS/SQLi**; subida de archivos con AV/Tipado MIME.
* **Logs de auditoría**: accesos admin, cambios de contenido, intentos fallidos.

**Datos y privacidad**

* Mínima recolección de PII.
* Encriptación **en tránsito (TLS 1.2+)** y **en reposo** (KMS) para PII y refresh tokens.
* Retención y borrado según política; exportación de datos del usuario.
* Backups cifrados, **RPO ≤ 24h / RTO ≤ 4h**.

**Infra & DevSecOps**

* Segregación de entornos (Dev/Stage/Prod).
* **Secret Manager**, rotación de credenciales.
* Análisis SCA/SAST en CI, escaneo de contenedores.
* **OWASP ASVS** como checklist de verificación.
* Monitoreo/alertas, detección de anomalías, WAF/CDN.

**Cumplimiento local**

* Considerar la **Ley N.° 29733 (Perú)** de protección de datos personales y su reglamento (consentimiento, finalidades, derechos ARCO).

1.13 Glosario

* **MVP (Producto Mínimo Viable)**: primera versión con funciones esenciales para validar valor.
* **Caso de Uso (UC)**: interacción típica entre un actor y el sistema para lograr un objetivo.
* **Actor**: rol que interactúa con el sistema (p. ej., Estudiante, Admin).
* **Categoría**: agrupación temática (p. ej., Vocabulario, Gramática).
* **Módulo**: unidad de aprendizaje seleccionable dentro de una categoría.
* **Sesión de práctica**: conjunto de ejercicios ejecutados de corrido.
* **Ejercicio/Ítem**: actividad puntual (opción múltiple, completar, pronunciación, etc.).
* **Feedback inmediato**: retroalimentación que aparece tras responder cada ejercicio.
* **Resultados**: resumen de desempeño al finalizar una sesión.
* **Progreso**: métricas históricas (puntaje, racha, tiempo, dominio por tema).
* **Racha**: días consecutivos cumpliendo objetivo.
* **Gamificación**: mecánicas de juego (insignias, logros) para motivar.
* **RBAC**: control de acceso basado en roles.
* **ABAC**: control basado en atributos (p. ej., “propietario del recurso”).
* **JWT**: token firmado para autenticar peticiones a la API.
* **Refresh token**: token de larga vida usado para obtener nuevos JWT.
* **CSRF**: ataque de falsificación de petición; se mitiga con tokens y cookies seguras.
* **CSP**: política de seguridad de contenidos para mitigar XSS.
* **PII**: información personal identificable.
* **KMS**: servicio de gestión de llaves de cifrado.
* **RPO/RTO**: objetivos de punto/tiempo de recuperación ante desastres.
* **SAST/SCA**: análisis estático de código / composición de software.
* **WAF**: firewall de aplicaciones web.
* **STT/TTS**: reconocimiento de voz / texto a voz para ejercicios de pronunciación.
* **NPS**: indicador de recomendación (satisfacción del usuario).
* **CDN**: red de distribución de contenido para estáticos y medios.
* **SameSite**: atributo de cookie que ayuda a mitigar CSRF.
* **HSTS**: obliga a usar HTTPS.

2. ANÁLISIS Y DISEÑO

2.1. Modelo de Análisis

**Objetivo.** Describir cómo se estructura lógicamente el sistema y cómo realiza sus casos de uso clave sin entrar aún en detalles tecnológicos específicos.

**Alcance y límites.**

* Dentro: autenticación, selección de categorías/módulos, práctica con feedback, resultados, progreso, recordatorios, administración de contenidos.
* Fuera: pasarela de pago (por ahora), analytics avanzadas de terceros, soporte telefónico.

**Actores**

* **Estudiante** (principal)
* **Admin** (gestión de contenidos/usuarios)
* **Servicios externos**: Proveedor OAuth, SMTP/email, (opcional) STT/TTS.

**Contexto del sistema**

flowchart LR

A[Estudiante] -->|web/móvil| S(Sistema JIM)

B[Admin] -->|web| S

S -->|OAuth| O[(Proveedor de Identidad)]

S -->|Email| M[(SMTP)]

S -->|Opcional STT/TTS| V[(Servicio de Voz)]

S -->|Persistencia| D[(Base de Datos)]

**Paquetes de análisis (vista lógica)**

* **Usuarios** (registro, login, perfil)
* **Catálogo** (categorías, módulos, ejercicios)
* **Práctica** (sesión, intentos, feedback)
* **Resultados & Progreso** (resúmenes, estadísticas, rachas, logros)
* **Notificaciones** (recordatorios)
* **Administración** (CRUD de contenidos, gestión básica de usuarios)
* **Integraciones** (OAuth, Email, STT/TTS)

2.1.1 Arquitectura del Sistema

**Objetivo.** Describir *dónde vive* la solución, *cómo se conecta* con otros sistemas y *cómo se despliega/opera*, sin entrar a la organización interna del código.

**1) Contexto y límites**

* **Clientes:** Web SPA (y móvil en futuro) usados por **Estudiante** y **Admin**.
* **Sistema JIM (núcleo):** API única + worker de tareas.
* **Sistemas externos:** OAuth/OIDC (Google/Apple), SMTP/email, (opcional) STT/TTS.
* **Almacenamiento:** BD transaccional, caché, objeto/medios.
* **Operación:** CDN/WAF, balanceador, monitoreo, backups.

flowchart LR

U[Estudiante/Admin] -->|HTTPS| CDN[CDN + WAF]

CDN --> LB[Load Balancer]

LB --> APP1[API JIM #1]

LB --> APP2[API JIM #2]

APP1 <-->|Pub/Sub| Q[Cola/Jobs]

APP2 <-->|Pub/Sub| Q

APP1 --> DB[(BD relacional)]

APP2 --> DB

APP1 --> C[(Caché)]

APP2 --> C

Q --> WK[Worker Notificaciones/Procesos]

APP1 --> STG[(Almacenamiento de objetos)]

APP1 --> OAUTH[(OAuth/OIDC)]

WK --> SMTP[(Servidor SMTP)]

APP1 -. métricas/logs .-> MON[(Monitoreo/Alertas)]

WK -. métricas/logs .-> MON

DB ==> BAK[(Backups cifrados)]

**2) Zonas de red**

* **Pública:** CDN/WAF, balanceador.
* **Privada (apps):** instancias de API y workers.
* **Datos:** BD, caché, almacenamiento de objetos (acceso restringido desde la zona privada).

**3) Integraciones y protocolos**

* Frontend ↔ API: **REST/JSON** sobre **HTTPS** (WebSocket opcional para tiempo real).
* Autenticación: **OAuth 2.1 / OIDC** (PKCE, scopes mínimos).
* Email: **SMTP** transaccional.
* Voz (opcional): API de **STT/TTS** sobre HTTPS.
* Estáticos y medios: **CDN** con invalidación.

**4) Disponibilidad, rendimiento y operación (SLOs sugeridos)**

* **Disponibilidad API:** 99.5% (MVP).
* **Latencia p95:** < 300 ms para endpoints de lectura.
* **Escalado:** horizontal en capa de API y workers.
* **Observabilidad:** métricas (CPU, latencia, tasa de errores), logs estructurados, trazas APM.
* **Resiliencia:** timeouts, retries con backoff, circuit breakers.
* **DR:** RPO ≤ 24h / RTO ≤ 4h con backups automáticos verificados.

**5) Seguridad (perimetral y de datos)**

* **WAF/CDN** (rate limit, bot protection).
* **TLS** extremo a extremo, secretos en **Secret Manager**.
* **RBAC a nivel de infraestructura** (IAM).
* **Encriptación en reposo** (KMS) para BD, backups y objetos.
* **Segmentación** de redes y least privilege entre servicios.

**6) Flujos sistémicos clave**

1. **Autenticación:** Cliente → API → OAuth (código+PKCE) → tokens → sesión.
2. **Práctica:** Cliente → API → BD/Cache → (si aplica) objetos (audio/imágenes).
3. **Recordatorios:** API agenda en **Cola** → **Worker** → **SMTP**.

*Nota:* Esta sección no define capas/módulos del código; eso va en **2.2 Arquitectura de Software**.

2.1.2 Realización de Casos de Uso – Análisis

**UC06 – Iniciar práctica**

sequenceDiagram

actor Estudiante

participant B as B:Vista Módulo

participant C as C:PrácticaService

participant E1 as E:Módulo

participant E2 as E:SesiónPráctica

participant R as Repo:Sesión

Estudiante->>B: Click "Iniciar práctica"

B->>C: iniciarSesion(moduloId, usuarioId)

C->>E1: validarDisponibilidad()

C->>E2: crearSesion(estado=EN\_CURSO)

C->>R: guardar(E2)

C-->>B: OK + sesiónId

B-->>Estudiante: navega a Sesión

**UC07 – Resolver ejercicio con feedback inmediato**

sequenceDiagram

actor Estudiante

participant B as B:Vista Sesión

participant C as C:PrácticaService

participant QE as E:Ejercicio

participant INT as E:Intento

participant RES as E:Resultado

participant R as Repo:Sesión/Intento

Estudiante->>B: Enviar respuesta

B->>C: responder(sesionId, ejercicioId, respuesta)

C->>QE: evaluar(respuesta) -> calificación, feedback

C->>INT: crearIntento(detalle, correcto?)

C->>R: guardar(INT)

C->>RES: actualizarParcial(sesionId, avance)

C-->>B: feedback + corrección + próximo ítem

B-->>Estudiante: mostrar feedback inmediato

**UC08 – Ver resultados de la práctica**

sequenceDiagram

actor Estudiante

participant B as B:Resultados

participant C as C:ResultadosService

participant S as E:SesiónPráctica

participant ST as E:Estadística

participant R as Repo

Estudiante->>B: Abrir resultados

B->>C: obtenerResumen(sesionId)

C->>R: cargar(S)

C->>ST: calcularKPIs(S)

C-->>B: resumen(puntaje, errores frecuentes, sugerencias)

**UC09 – Ver progreso/estadísticas**

sequenceDiagram

actor Estudiante

participant B as B:Progreso

participant C as C:ProgresoService

participant E as E:Estadística

participant R as Repo

Estudiante->>B: Abrir Progreso

B->>C: getDashboard(usuarioId)

C->>R: fetchHistorial(usuarioId)

C->>E: agregaciones(racha, tiempo, dominio por tema)

C-->>B: dataset para gráficas

**UC14 – Administración de contenidos (CRUD)**

sequenceDiagram

actor Admin

participant B as B:Admin UI

participant C as C:CatalogoService

participant G as Guard:RBAC

participant M as E:Módulo

participant R as Repo

Admin->>B: Crear/editar módulo

B->>G: checkPermiso(ADMIN)

G-->>B: OK

B->>C: upsertModulo(dto)

C->>M: validarReglas()

C->>R: guardar(M)

C-->>B: confirmación

Trazabilidad CU ↔ Clases de análisis (resumen)

| **CU** | **Boundary** | **Control** | **Entity principal** |
| --- | --- | --- | --- |
| UC05 Seleccionar módulo | Vista Catálogo | CatalogoService | Categoría, Módulo |
| UC06 Iniciar práctica | Vista Módulo | PrácticaService | SesiónPráctica |
| UC07 Resolver ejercicio | Vista Sesión | PrácticaService | Ejercicio, Intento, Resultado |
| UC08 Ver resultados | Vista Resultados | ResultadosService | SesiónPráctica, Estadística |
| UC09 Ver progreso | Vista Progreso | ProgresoService | Estadística, Logro |
| UC12 Recordatorios | Vista Ajustes | NotificacionesService | Recordatorio |
| UC14 Admin contenido | Admin UI | CatalogoService | Módulo, Ejercicio |

2.1.3 Modelo Conceptual

**es y relaciones (análisis de dominio)**

classDiagram

class Usuario{

+usuarioId: UUID

email: Email

nombre: string

rol: Rol

objetivoDiarioMin: int

}

class Categoria{

+categoriaId: UUID

nombre: string

descripcion: string

}

class Modulo{

+moduloId: UUID

titulo: string

descripcion: string

tiempoEstimadoMin: int

estado: EstadoPublicacion

}

class Ejercicio{

+ejercicioId: UUID

tipo: TipoEjercicio

enunciado: string

clave: any

retroBase: string

puntajeMax: int

}

class SesionPractica{

+sesionId: UUID

estado: EstadoSesion

inicio: datetime

fin: datetime

}

class Intento{

+intentoId: UUID

respuesta: any

correcto: bool

tiempoSeg: int

}

class Resultado{

+resultadoId: UUID

puntajeTotal: int

precision: float

recomendaciones: string

}

class Estadistica{

+estadisticaId: UUID

rachaDias: int

tiempoTotalMin: int

dominioPorCategoria: Map

}

class Recordatorio{

+recordatorioId: UUID

horaLocal: time

diasSemana: Set

activo: bool

}

class Logro{

+logroId: UUID

nombre: string

criterio: string

fechaDesbloqueo: date

}

class Feedback{

+feedbackId: UUID

mensaje: string

tipo: TipoFeedback

}

Usuario "1" --> "0..\*" SesionPractica

Usuario "1" --> "0..\*" Recordatorio

Usuario "1" --> "0..\*" Logro

Categoria "1" --> "1..\*" Modulo

Modulo "1" --> "1..\*" Ejercicio

SesionPractica "1" --> "1..\*" Intento

Intento "1" --> "0..1" Feedback

Intento "1" --> "1" Ejercicio

SesionPractica "1" --> "1" Resultado

Usuario "1" --> "1" Estadistica

**Enumeraciones (referencia)**

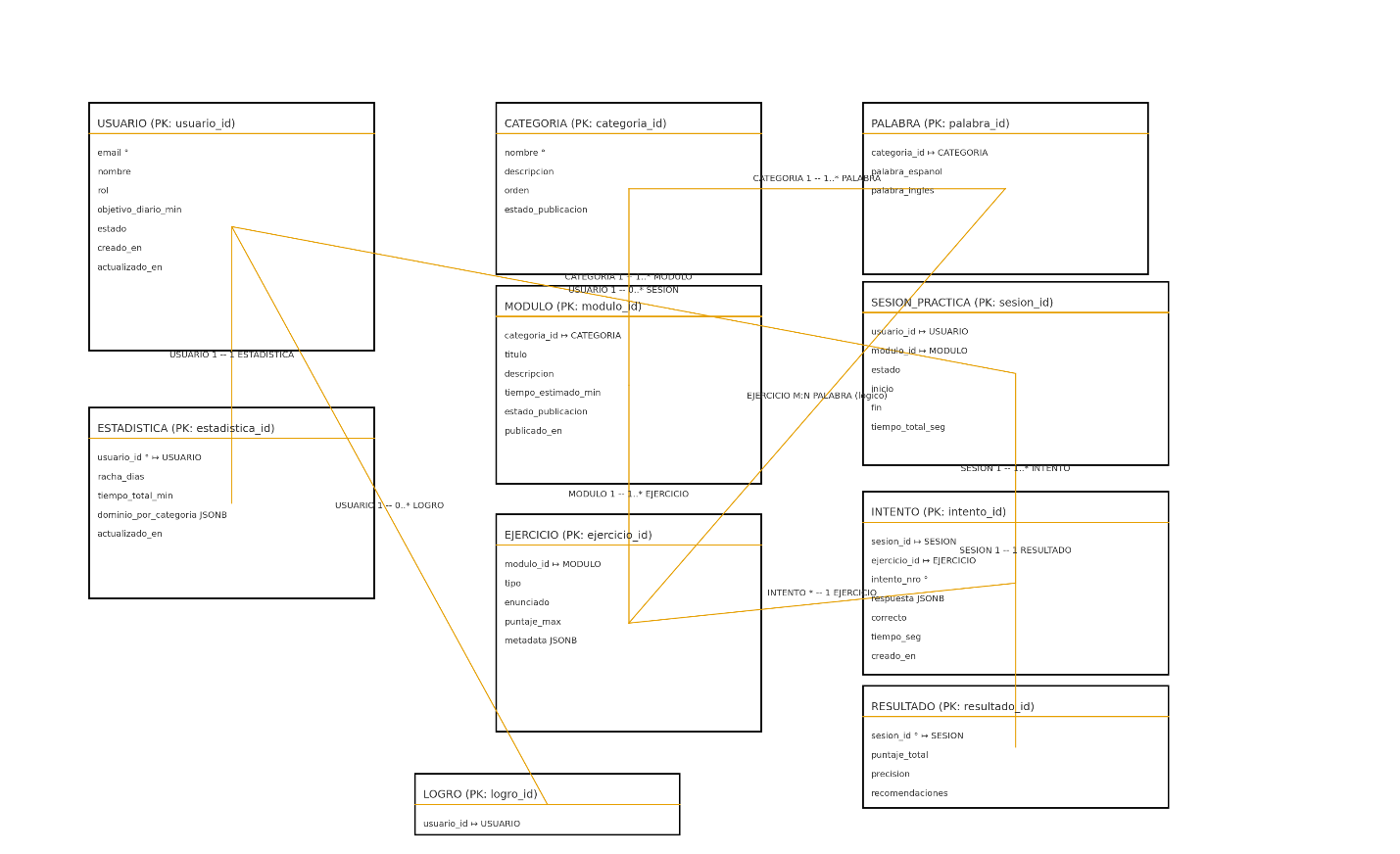
* Rol = {ESTUDIANTE, ADMIN}
* TipoEjercicio = {OPCION\_MULTIPLE, COMPLETAR, EMPAREJAR, REDACCION, PRONUNCIACION}
* EstadoSesion = {EN\_CURSO, FINALIZADA, CANCELADA}
* EstadoPublicacion = {BORRADOR, PUBLICADO, ARCHIVADO}
* TipoFeedback = {CORRECCION, PISTA, EXPLICACION}

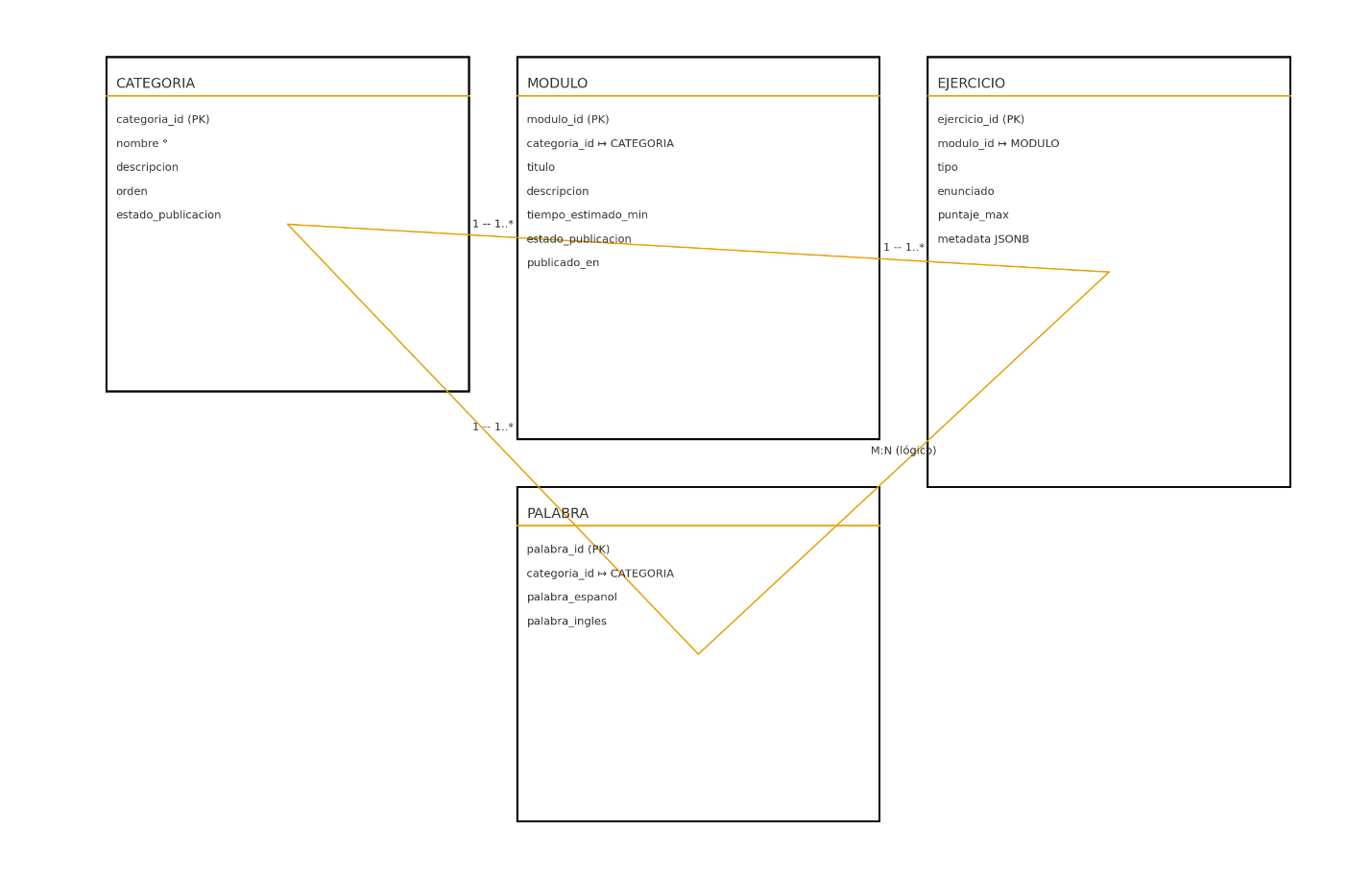
**Reglas/invariantes clave**

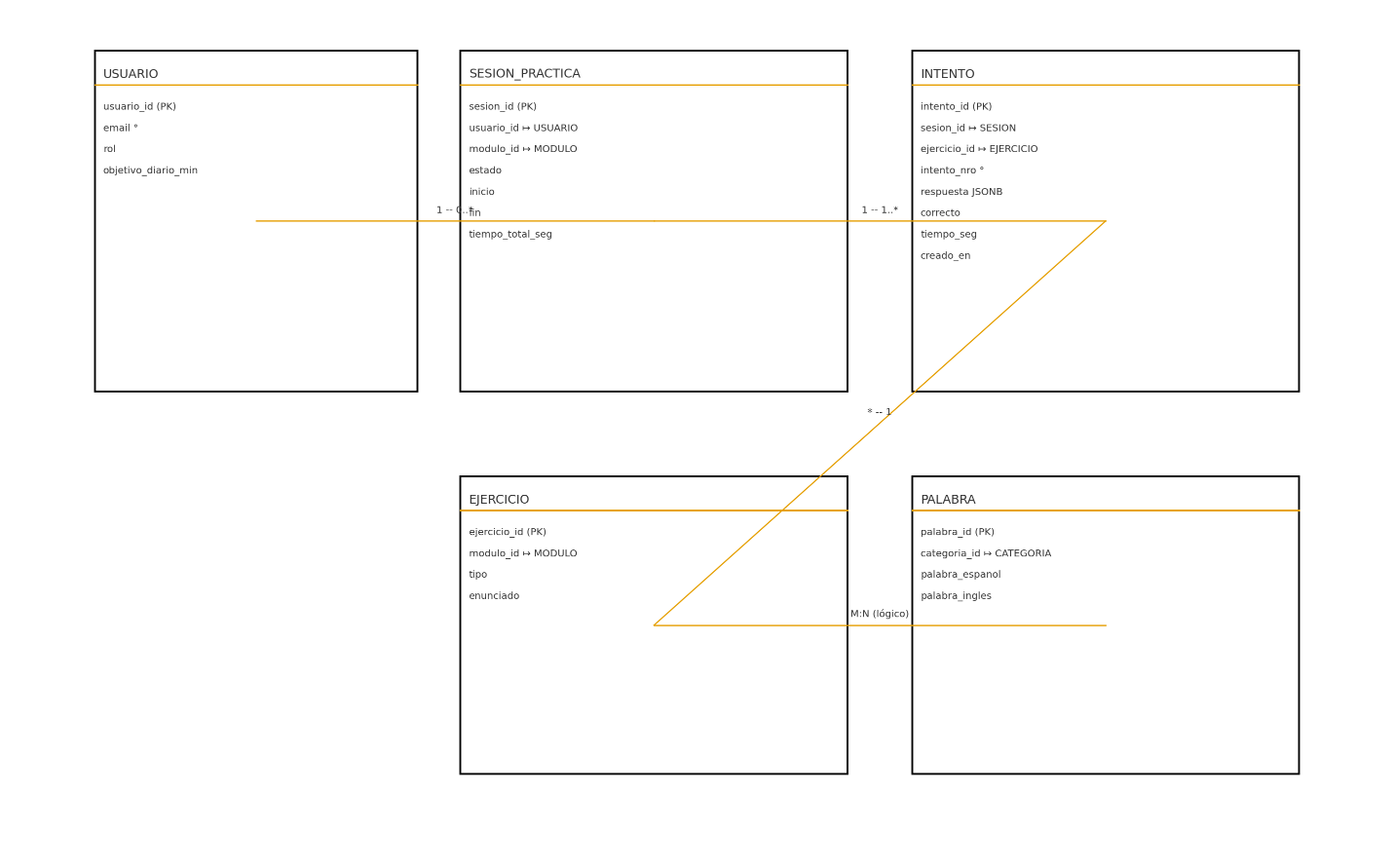
* Un **Módulo** *PUBLICADO* solo puede incluir **Ejercicios** activos.
* Una **SesiónPráctica** *FINALIZADA* es inmutable (solo lectura).
* El **Resultado** se recalcula a partir de los **Intentos**; no se edita manualmente.
* **Racha** se incrementa si tiempoEstudioDia >= objetivoDiarioMin.

2.1.4 Modelo Lógico

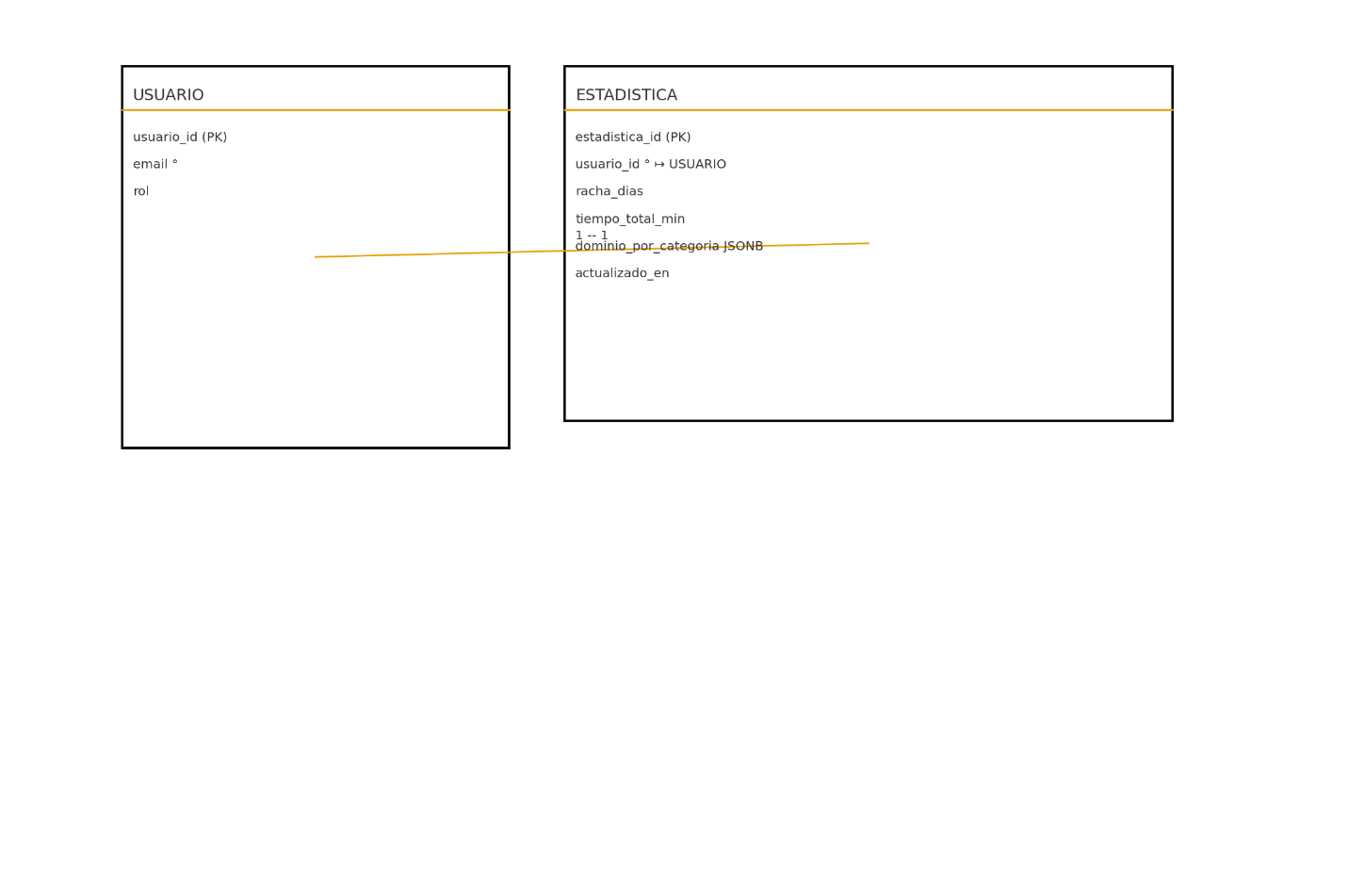
Visión general



Subdominio **Contenido** (Categoría–Módulo–Ejercicio–Palabra)  


**Práctica & Vocabulario** (Sesión–Intento–Ejercicio–Palabra, relación M:N)  


**Progreso** (sólo Usuario–Estadística)



Entidades (descripción y atributos clave)

| **Entidad** | **Descripción (propósito)** | **Atributos representativos** |
| --- | --- | --- |
| **Usuario** | Persona que usa la app (estudiante o admin). Fuente de sesiones, logros y estadísticas. | usuarioId, email (único), nombre, rol {ESTUDIANTE, ADMIN}, objetivoDiarioMin, estado, creadoEn, actualizadoEn |
| **Categoría** | Agrupa el contenido por tema (p. ej., Vocabulario, Gramática; o subtemas como Viajes, Trabajo). | categoriaId, nombre (único), descripcion, orden, estadoPublicacion {BORRADOR, PUBLICADO, ARCHIVADO} |
| **Palabra** | Banco de vocabulario reutilizable por tema; cada palabra pertenece a una categoría. | palabraId, categoriaId→Categoría, palabraEspanol, palabraIngles *(sugerencia: unicidad por categoría)* |
| **Módulo** | Unidad de aprendizaje seleccionable dentro de una categoría. | moduloId, categoriaId→Categoría, titulo, descripcion, tiempoEstimadoMin, estadoPublicacion, publicadoEn |
| **Ejercicio** | Ítem presentado durante una práctica (opción múltiple, completar, emparejar, redacción, pronunciación). | ejercicioId, moduloId→Módulo, tipo, enunciado, puntajeMax, *(campos dependientes del tipo, p. ej. opciones/clave)* |
| **SesiónPráctica** | Ejecución “de corrido” iniciada por un usuario sobre un módulo. | sesionId, usuarioId→Usuario, moduloId→Módulo, estado {EN\_CURSO, FINALIZADA, CANCELADA}, inicio, fin, tiempoTotalSeg |
| **Intento** | Respuesta del usuario a un ejercicio dentro de una sesión. Puede haber varios intentos por ejercicio. | intentoId, sesionId→SesiónPráctica, ejercicioId→Ejercicio, intentoNro, respuesta, correcto, tiempoSeg, creadoEn |
| **Resultado** | Resumen al cierre de la sesión (puntaje y precisión global). | resultadoId, sesionId→SesiónPráctica *(1:1)*, puntajeTotal, precision, recomendaciones |
| **Estadística** | Métricas acumuladas por usuario (racha, tiempo, dominio por tema). | estadisticaId, usuarioId→Usuario *(1:1)*, rachaDias, tiempoTotalMin, dominioPorCategoria, actualizadoEn |
| **Logro** | Insignias/retos desbloqueados por el usuario. | logroId, usuarioId→Usuario, nombre, criterio, fechaDesbloqueo |

**Relaciones (cardinalidades y reglas)**

**Usuario (1) — (0..\*) SesiónPráctica**

Un usuario puede tener muchas sesiones; cada sesión pertenece a un usuario.

*Reglas:* al finalizar, la sesión pasa a solo-lectura.

**Usuario (1) — (1) Estadística**

Cada usuario posee exactamente **una** ficha de estadísticas acumuladas.

*Reglas:* se actualiza tras completar sesiones; es derivada (no editable manualmente).

**Usuario (1) — (0..\*) Logro**

Un usuario puede desbloquear múltiples logros.

*Reglas:* los criterios de logro se evalúan a partir de la actividad (p. ej., rachas, puntajes).

**Categoría (1) — (1..\*) Módulo**

Una categoría contiene varios módulos; un módulo pertenece a una categoría.

*Reglas:* no se publica un módulo si su categoría está archivada.

**Categoría (1) — (1..\*) Palabra**

Una categoría agrupa múltiples palabras; cada palabra pertenece a una categoría.

*Reglas:* se recomienda unicidad por (categoría, palabraIngles) o por (categoría, palabraEspanol).

**Módulo (1) — (1..\*) Ejercicio**

Un módulo incluye muchos ejercicios; cada ejercicio pertenece a un módulo.

*Reglas:* un módulo **PUBLICADO** solo puede contener ejercicios válidos.

**SesiónPráctica (1) — (1..\*) Intento**

Una sesión registra múltiples intentos; cada intento pertenece a una sesión.

*Reglas:* intentoNro ordena los intentos de un mismo ejercicio dentro de la sesión.

**Intento (\*) — (1) Ejercicio**

Cada intento corresponde a un único ejercicio; un ejercicio puede tener intentos en muchas sesiones.

*Reglas:* la corrección/feedback se basa en la definición del ejercicio.

**SesiónPráctica (1) — (1) Resultado**

Cada sesión produce exactamente un resultado global.

*Reglas:* el resultado se **genera al finalizar** la sesión a partir de los intentos.

**Ejercicio (M) — (N) Palabra** *(asociación conceptual)*

Un ejercicio puede referenciar varias palabras; una palabra puede reutilizarse en muchos ejercicios.

*Motivo:* medir/entrenar vocabulario transversal.

*Implementación física típica:* tabla puente EjercicioPalabra(ejercicioId, palabraId) con unicidad compuesta.

 **Integridad referencial:** no se debe eliminar una Categoría si existen Módulos o Palabras activas que dependan de ella; similar para Módulo→Ejercicio y Sesión→Intento/Resultado.

 **Inmutabilidad post-cierre:** SesiónPráctica **FINALIZADA** y su **Resultado** no se editan.

 **Consistencia de publicación:** un Módulo **PUBLICADO** no puede contener Ejercicios en estado inválido.

 **Medición por vocabulario:** el vínculo Ejercicio–Palabra permite reportar dominio por tema y por palabra.

2.1.5 Modelo Físico

-- Recomendado (UUID y email case-insensitive)

CREATE EXTENSION IF NOT EXISTS pgcrypto;

CREATE EXTENSION IF NOT EXISTS citext;

-- =========================

-- Núcleo de usuarios

-- =========================

CREATE TABLE usuario (

usuario\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

email CITEXT UNIQUE NOT NULL,

nombre TEXT NOT NULL,

rol TEXT NOT NULL CHECK (rol IN ('ESTUDIANTE','ADMIN')),

objetivo\_diario\_min INT NOT NULL DEFAULT 10 CHECK (objetivo\_diario\_min >= 0),

estado TEXT NOT NULL DEFAULT 'ACTIVO' CHECK (estado IN ('ACTIVO','SUSPENDIDO')),

creado\_en TIMESTAMPTZ NOT NULL DEFAULT now(),

actualizado\_en TIMESTAMPTZ NOT NULL DEFAULT now()

);

-- =========================

-- Catálogo de contenido

-- =========================

CREATE TABLE categoria (

categoria\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

nombre TEXT UNIQUE NOT NULL,

descripcion TEXT,

orden INT NOT NULL DEFAULT 0 CHECK (orden >= 0),

estado\_publicacion TEXT NOT NULL CHECK (estado\_publicacion IN ('BORRADOR','PUBLICADO','ARCHIVADO'))

);

CREATE TABLE modulo (

modulo\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

categoria\_id UUID NOT NULL REFERENCES categoria(categoria\_id),

titulo TEXT NOT NULL,

descripcion TEXT,

tiempo\_estimado\_min INT CHECK (tiempo\_estimado\_min >= 0),

estado\_publicacion TEXT NOT NULL CHECK (estado\_publicacion IN ('BORRADOR','PUBLICADO','ARCHIVADO')),

publicado\_en TIMESTAMPTZ

);

CREATE INDEX idx\_modulo\_categoria ON modulo(categoria\_id);

CREATE TABLE ejercicio (

ejercicio\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

modulo\_id UUID NOT NULL REFERENCES modulo(modulo\_id) ON DELETE CASCADE,

tipo TEXT NOT NULL CHECK (tipo IN ('OPCION\_MULTIPLE','COMPLETAR','EMPAREJAR','REDACCION','PRONUNCIACION')),

enunciado TEXT NOT NULL,

puntaje\_max INT NOT NULL CHECK (puntaje\_max > 0),

metadata JSONB NOT NULL DEFAULT '{}'::jsonb

);

CREATE INDEX idx\_ejercicio\_modulo ON ejercicio(modulo\_id);

CREATE INDEX idx\_ejercicio\_metadata\_gin ON ejercicio USING GIN (metadata);

-- (Opcional) Búsqueda de texto en títulos/descripciones/enunciados:

-- CREATE INDEX idx\_modulo\_fts ON modulo USING GIN (to\_tsvector('simple', coalesce(titulo,'') || ' ' || coalesce(descripcion,'')));

-- CREATE INDEX idx\_ejercicio\_fts ON ejercicio USING GIN (to\_tsvector('simple', coalesce(enunciado,'')));

-- =========================

-- Vocabulario

-- =========================

CREATE TABLE palabra (

palabra\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

categoria\_id UUID NOT NULL REFERENCES categoria(categoria\_id),

palabra\_espanol TEXT NOT NULL,

palabra\_ingles TEXT NOT NULL,

-- evita duplicados por tema

UNIQUE (categoria\_id, palabra\_ingles),

UNIQUE (categoria\_id, palabra\_espanol)

);

CREATE INDEX idx\_palabra\_categoria ON palabra(categoria\_id);

-- Relación M:N Ejercicio–Palabra

CREATE TABLE ejercicio\_palabra (

ejercicio\_id UUID NOT NULL REFERENCES ejercicio(ejercicio\_id) ON DELETE CASCADE,

palabra\_id UUID NOT NULL REFERENCES palabra(palabra\_id) ON DELETE CASCADE,

PRIMARY KEY (ejercicio\_id, palabra\_id)

);

-- Índices de apoyo según consultas frecuentes

CREATE INDEX idx\_ejercicio\_palabra\_\_palabra ON ejercicio\_palabra(palabra\_id);

-- =========================

-- Práctica y resultados

-- =========================

CREATE TABLE sesion\_practica (

sesion\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

usuario\_id UUID NOT NULL REFERENCES usuario(usuario\_id),

modulo\_id UUID NOT NULL REFERENCES modulo(modulo\_id),

estado TEXT NOT NULL CHECK (estado IN ('EN\_CURSO','FINALIZADA','CANCELADA')),

inicio TIMESTAMPTZ NOT NULL DEFAULT now(),

fin TIMESTAMPTZ,

tiempo\_total\_seg INT NOT NULL DEFAULT 0 CHECK (tiempo\_total\_seg >= 0)

);

CREATE INDEX idx\_sesion\_usuario ON sesion\_practica(usuario\_id, inicio DESC);

CREATE INDEX idx\_sesion\_modulo ON sesion\_practica(modulo\_id);

CREATE TABLE intento (

intento\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

sesion\_id UUID NOT NULL REFERENCES sesion\_practica(sesion\_id) ON DELETE CASCADE,

ejercicio\_id UUID NOT NULL REFERENCES ejercicio(ejercicio\_id),

intento\_nro INT NOT NULL,

respuesta JSONB NOT NULL,

correcto BOOLEAN NOT NULL,

tiempo\_seg INT NOT NULL DEFAULT 0 CHECK (tiempo\_seg >= 0),

creado\_en TIMESTAMPTZ NOT NULL DEFAULT now(),

UNIQUE (sesion\_id, ejercicio\_id, intento\_nro)

);

CREATE INDEX idx\_intento\_sesion ON intento(sesion\_id, creado\_en DESC);

CREATE INDEX idx\_intento\_ejercicio ON intento(ejercicio\_id);

CREATE INDEX idx\_intento\_respuesta\_gin ON intento USING GIN (respuesta);

CREATE TABLE resultado (

resultado\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

sesion\_id UUID NOT NULL UNIQUE REFERENCES sesion\_practica(sesion\_id) ON DELETE CASCADE,

puntaje\_total INT NOT NULL CHECK (puntaje\_total >= 0),

precision NUMERIC(5,2) NOT NULL CHECK (precision >= 0 AND precision <= 100),

recomendaciones TEXT

);

-- =========================

-- Progreso y gamificación

-- =========================

CREATE TABLE estadistica (

estadistica\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

usuario\_id UUID NOT NULL UNIQUE REFERENCES usuario(usuario\_id) ON DELETE CASCADE,

racha\_dias INT NOT NULL DEFAULT 0 CHECK (racha\_dias >= 0),

tiempo\_total\_min INT NOT NULL DEFAULT 0 CHECK (tiempo\_total\_min >= 0),

dominio\_por\_categoria JSONB NOT NULL DEFAULT '{}'::jsonb,

actualizado\_en TIMESTAMPTZ NOT NULL DEFAULT now()

);

CREATE TABLE logro (

logro\_id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

usuario\_id UUID NOT NULL REFERENCES usuario(usuario\_id) ON DELETE CASCADE,

nombre TEXT NOT NULL,

criterio TEXT NOT NULL,

fecha\_desbloqueo DATE NOT NULL

);

CREATE INDEX idx\_logro\_usuario\_fecha ON logro(usuario\_id, fecha\_desbloqueo DESC);

-- =========================

-- Utilidades (opcional): trigger de updated\_at

-- =========================

CREATE OR REPLACE FUNCTION set\_updated\_at()

RETURNS TRIGGER AS $$

BEGIN

NEW.actualizado\_en := now();

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS trg\_usuario\_updated ON usuario;

CREATE TRIGGER trg\_usuario\_updated

BEFORE UPDATE ON usuario

FOR EACH ROW EXECUTE FUNCTION set\_updated\_at();

2.1.6 Metas y Restricciones de la Arquitectura

| **Eje** | **Meta (SLO/objetivo)** | **Métrica / Umbral** | **Medición/Alerta** |
| --- | --- | --- | --- |
| Disponibilidad | **API ≥ 99.5% mensual** | uptime mensual | Health checks + alertas por burn-rate SLO |
| Rendimiento (lecturas) | **p95 < 300 ms** en catálogo, progreso | latencia p95 por ruta | APM + budgets por endpoint |
| Rendimiento (práctica) | **p95 < 500 ms** al enviar respuesta | latencia p95 /intentos | APM + alerta si p95 > 500 ms 15 min |
| Concurrencia | **≥ 500 usuarios** simultáneos en práctica | throughput sostenido | Pruebas de carga (pico 2×) |
| Error rate | **< 1%** 5xx por día | tasa 5xx | Alertas si >1% por 10 min |
| Escalabilidad | **Escalado horizontal** API/workers sin cambios de código | replicas activas | HPA / orquestador |
| Seguridad | **100% TLS**, **datos cifrados en reposo** | verificación TLS/KMS | Scans + verificación en CI |
| Vulnerabilidades | **0 P1** abiertos > 7 días | backlog de vulns | SAST/SCA + alertas |
| Privacidad | **PII mínima** + derecho de eliminación | auditorías | Procesos y scripts verificados |
| DR/Backups | **RPO ≤ 24h / RTO ≤ 4h** | restauraciones exitosas | Prueba de restauración trimestral |
| Mantenibilidad | **Cobertura ≥ 60%** en servicios de aplicación | cobertura unitaria | CI con umbral de fallo |
| Observabilidad | **100%** rutas críticas con trazas | % de cobertura de trazas | APM + dashboards |
| Portabilidad | **Adapters** para proveedores (OAuth/STT/TTS) | dependencia directa = 0 | Revisión de dependencias |
| Coste | **Cost ceiling** definido por entorno | gasto mensual | Alertas por sobrepaso (FinOps) |

**Escenarios de calidad (ATAM-light)**

* **Pico de uso**: 18:00–20:00 → mantener p95 debajo de objetivos con autoescalado.
* **Fallo de zona/DB**: restaurar dentro de **RTO ≤ 4h**, pérdida de datos **≤ 24h** (RPO).
* **Ataques de fuerza bruta**: bloqueo y rate-limit en login/endpoints sensibles sin afectar UX legítima.
* **Cambio de esquema**: migraciones **sin downtime** perceptible (< 2 min) usando blue/green o mig. en dos pasos.
* **Ejercicio nuevo** (tipo distinto): incorporarlo sin refactor masivo gracias a metadata JSONB y adapters de corrección.

**Restricciones**

**Funcionales**

* No hay “niveles”; la práctica se organiza por **Categoría → Módulo → Ejercicio**.
* **PALABRA** es banco de vocabulario por categoría; **EJERCICIO–PALABRA** es **M:N**.
* **Pagos** y **recomendaciones ML** fuera del alcance inicial (reglas simples temporales).

**Técnicas**

* **Arquitectura de software**: monolito **modular** (puertos/adaptadores).
* **BD**: **PostgreSQL 15** como sistema de registro; **Redis** para caché/colas; *object storage* para medios.
* **API**: REST/JSON sobre HTTPS (WebSocket opcional).
* **Infra**: CDN/WAF, balanceador, despliegue contenedorizado, métricas/trazas/logs.
* **Eliminadas del modelo lógico**: *CuentaOAuth*, *Recordatorio*, *Feedback*. (La notificación/feedback inmediato se gestionará a nivel de aplicación/UI, sin tablas dedicadas en v1).

**Operativas**

* **RPO ≤ 24h / RTO ≤ 4h** con backups cifrados y prueba de restauración trimestral.
* Ventanas de mantenimiento programables y comunicadas.
* Monitoreo 24×7 de uptime/latencia/errores; on-call en horario pico.

**Seguridad y privacidad**

* TLS extremo a extremo, secretos en **Secret Manager**, mínimos privilegios (IAM/RBAC).
* Cifrado en reposo (KMS) para BD/backups/objetos.
* Cumplimiento **Ley N.º 29733 (Perú)**: consentimiento, finalidades, derechos ARCO, borrado.
* Logs de auditoría para acciones **admin** y cambios de contenido.

**Datos**

* Normalización 3FN; uso de **JSONB** para campos dependientes del tipo de ejercicio y respuestas.
* Retención: sesiones/intentos/resultados según política (p.ej., 12–24 meses) y anonimización para analítica histórica.

**UX**

* **Feedback inmediato** al responder; “**Continuar donde me quedé**”.
* Tiempos de carga inicial **< 2 s** en conexiones típicas.

**Equipo / Coste / Roadmap**

* Preferencia por **open-source** y servicios gestionados.
* Automatizar CI/CD, migraciones y pruebas.
* Fases: **MVP** (Must), **v1.0** (Should), **v1.1** (Could) según priorización acordada.

Riesgos y mitigaciones

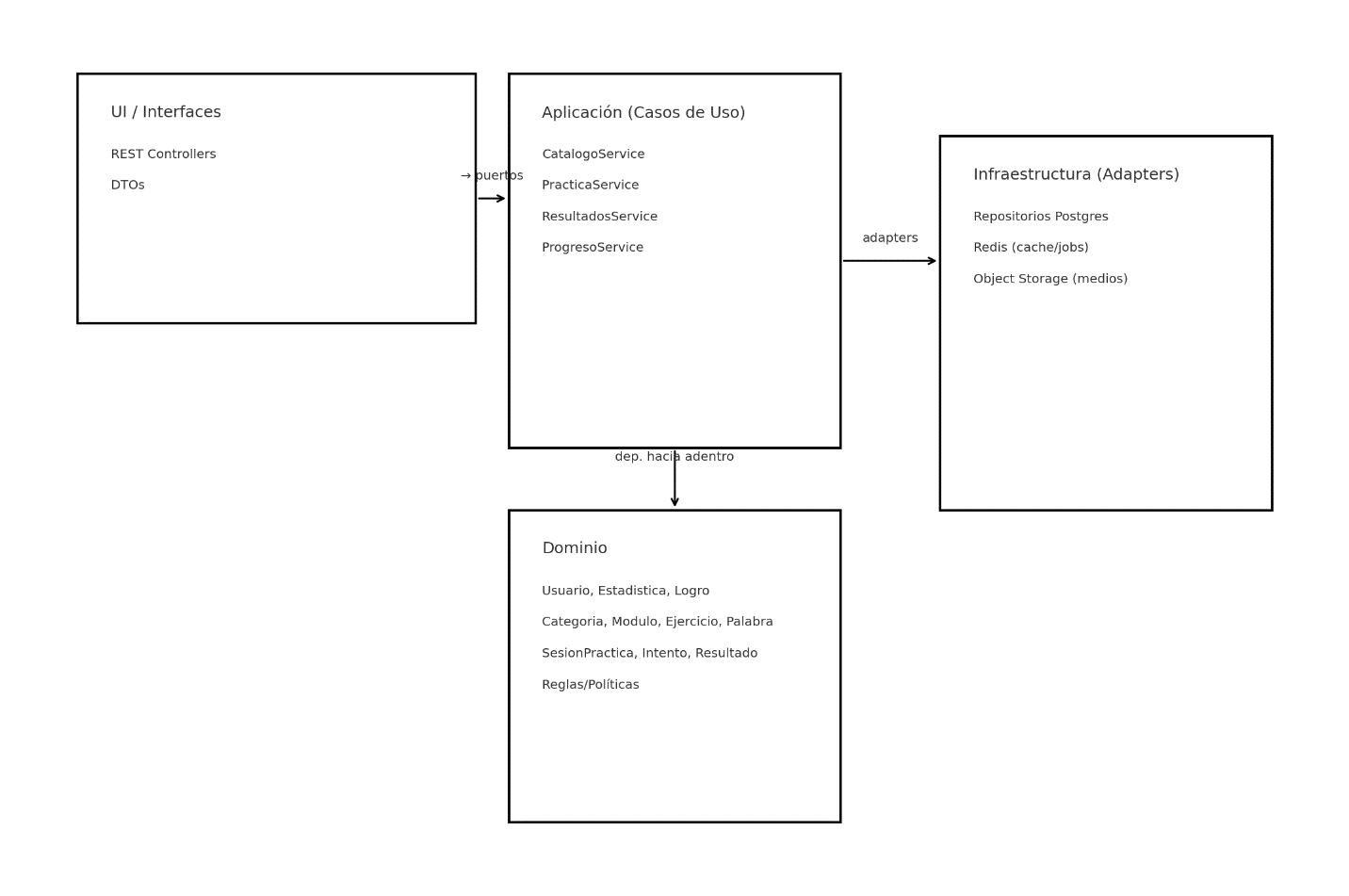
| **Riesgo** | **Impacto** | **Mitigación** |
| --- | --- | --- |
| Crecimiento de INTENTOS/SESIONES | Consultas lentas | Índices adecuados, **particionado** temporal, réplicas de lectura |
| Tipos de ejercicio heterogéneos | Complejidad de validación | Contratos por tipo + validadores y tests por adapter |
| Dependencia de terceros (STT/TTS) | Caída de feature | Fallback a texto, timeouts/circuit breakers |
| Costos en picos | Sobrepaso de presupuesto | HPA con límites, alertas FinOps, caché agresiva en lecturas |
| Cambios de esquema | Downtime | Migraciones en dos fases, feature flags |

**Criterios de aceptación (arquitectura)**

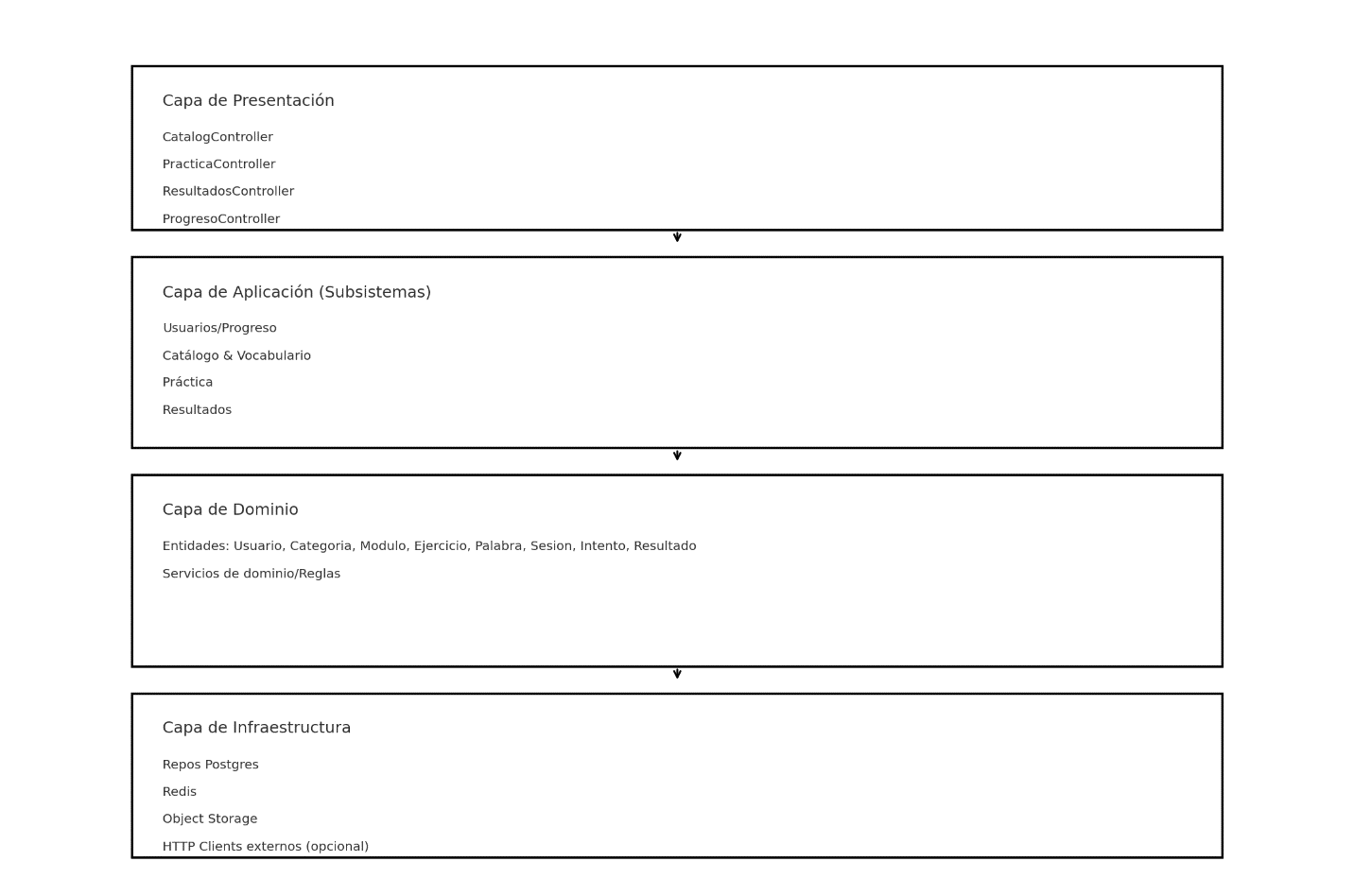
* Dashboards con **SLOs, latencia p95, error rate, uptime** por entorno.
* Prueba de **DR** documentada y repetible.
* **Pruebas de carga** con reporte ≥ 500 usuarios concurrentes cumpliendo SLO.
* **Amenazas** evaluadas (OWASP ASVS checklist) sin P1 abiertos.
* **Playbooks** de incidentes y de rollback listos.

2.2. Modelo de Diseño

2.2.1 Arquitectura del Software

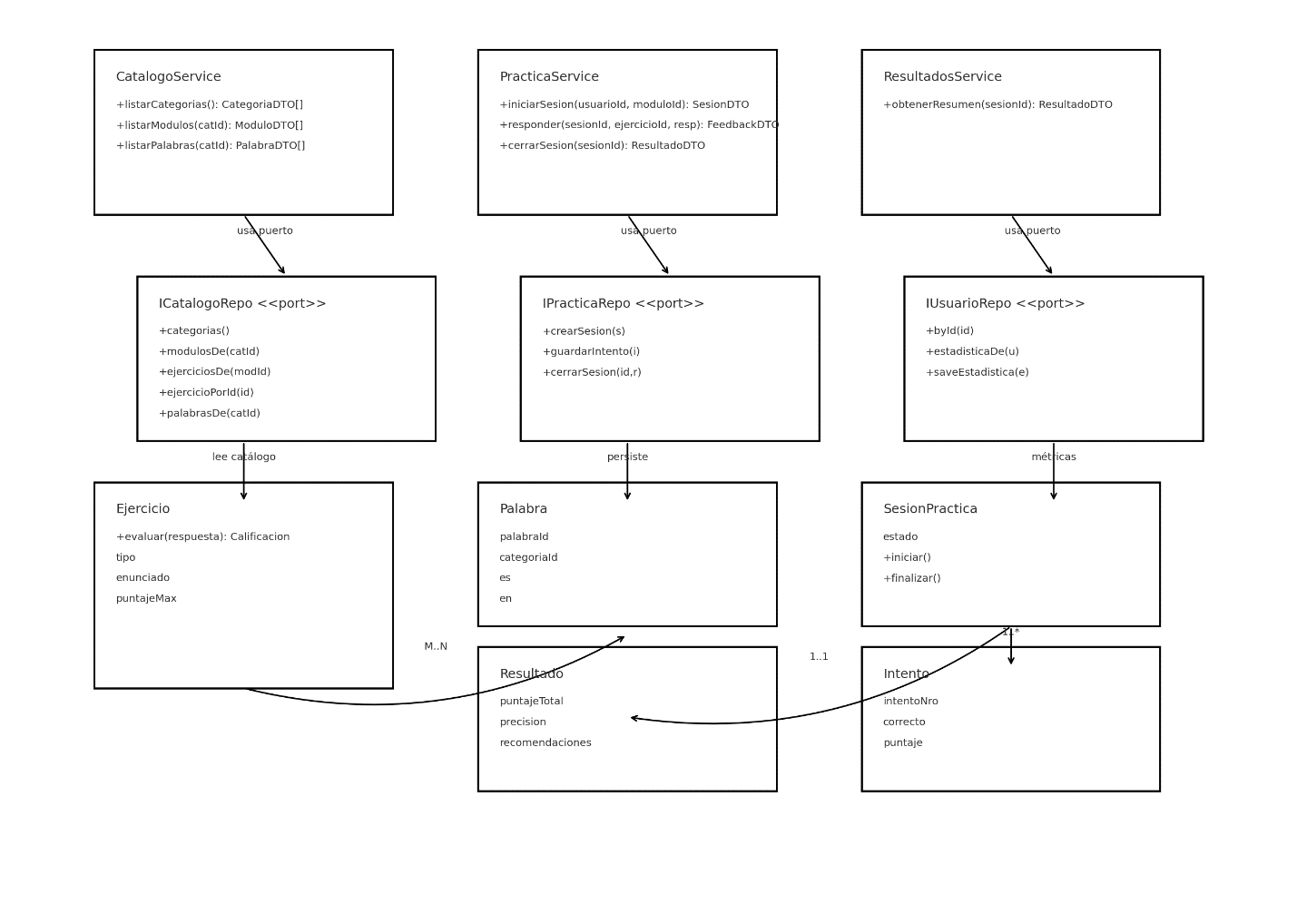


2.2.1. Vista de Capas y Subsistemas

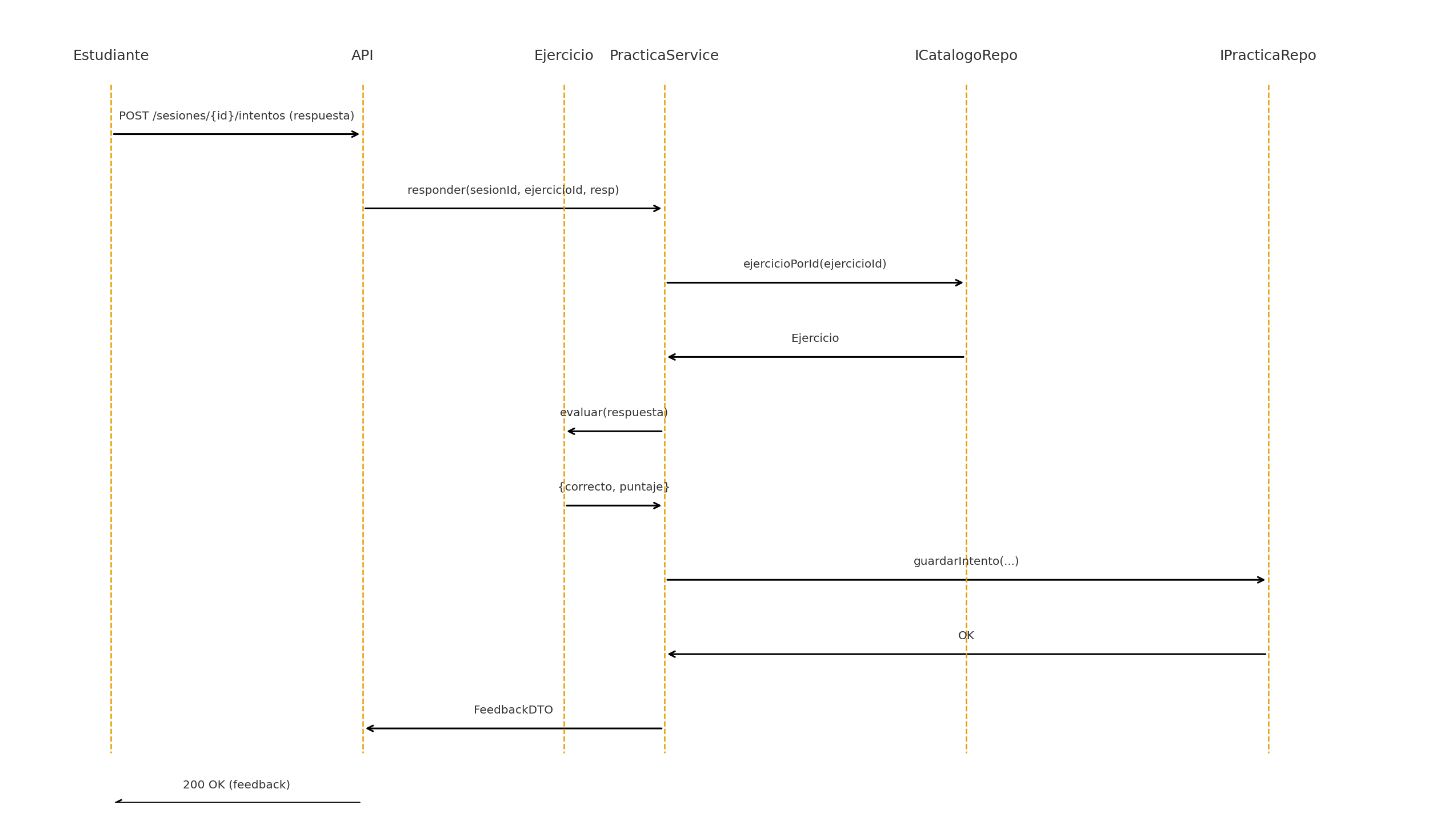


2.2.2 Realización de Casos de Uso – Modelo de Diseño

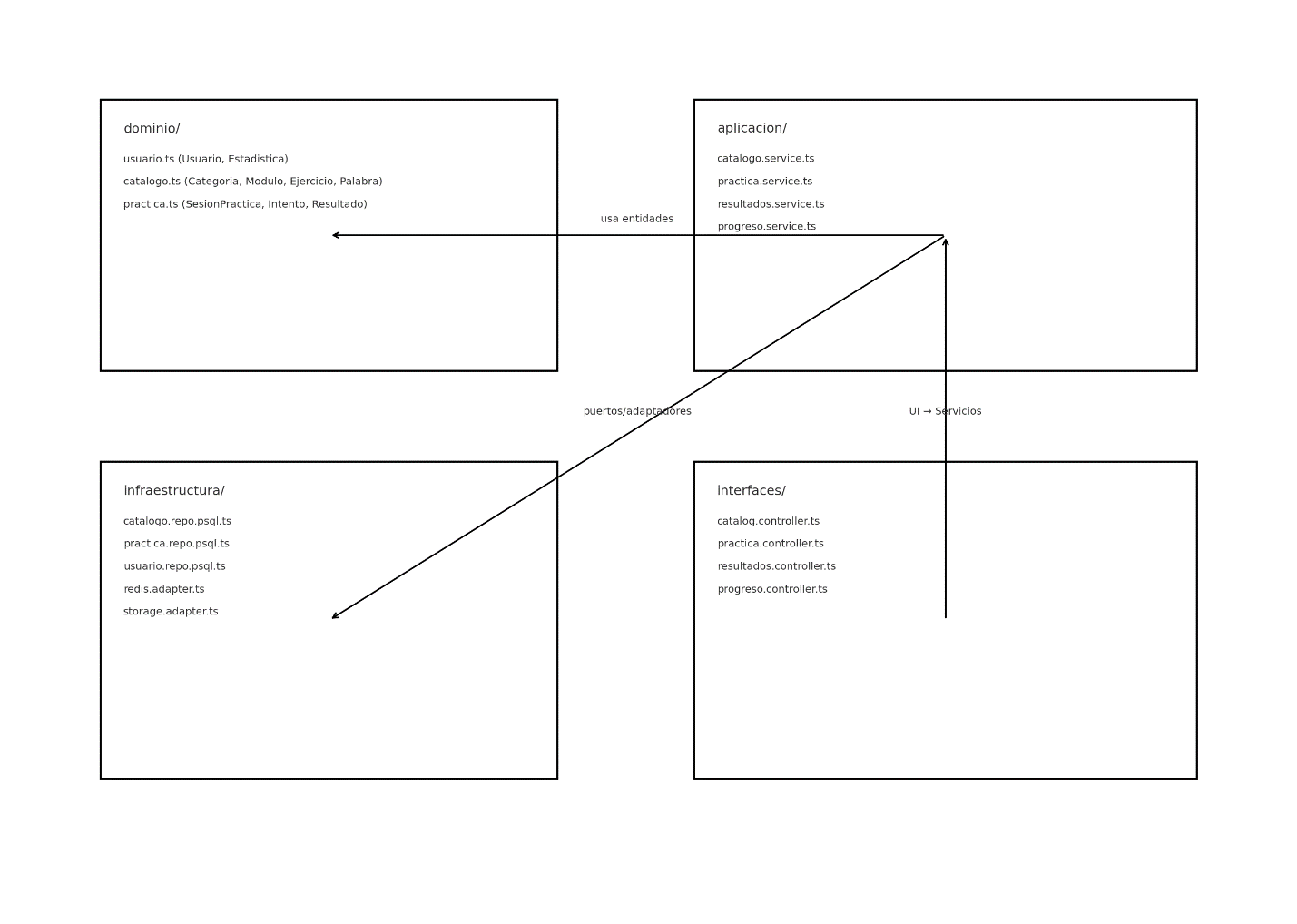
2.2.2.1 Diagrama de Clases



2.2.2.2 Diagrama de secuencia



2.2.3. Programación diagramas de clases



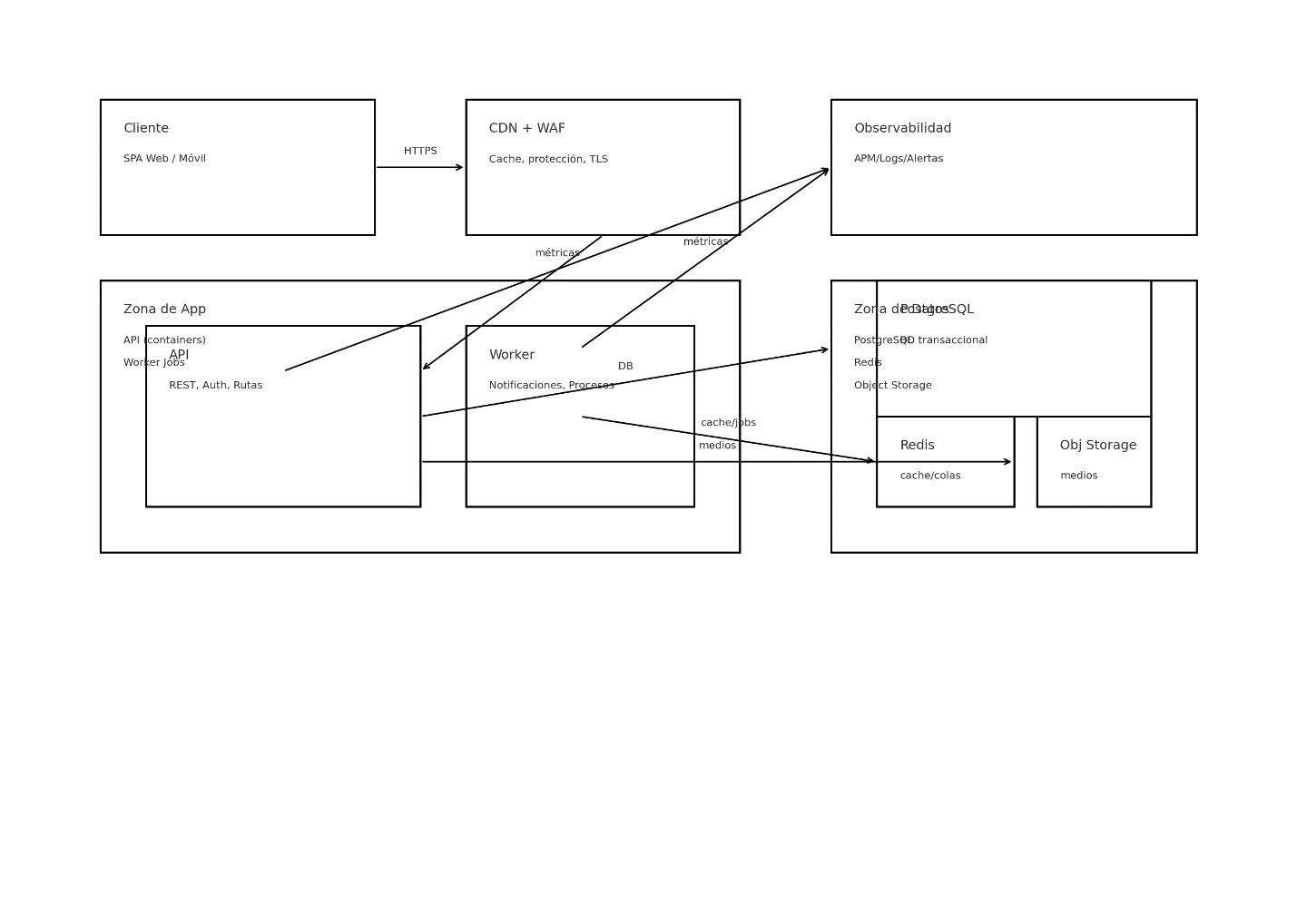
2.2.3.1. Código del DC – Nombre del DC 01

2.2.3.2. Código del DC – Nombre del DC 02

2.2.3.3. Código del DC – Nombre del DC 03

2.2.3.4. Código del DC – Nombre del DC 04

2.3. Diagrama de Despliegue



2.4. Diagrama de Componentes

