

Informe Interno de Producto

Análisis de Fortalezas, Debilidades
y Oportunidades de Mejora

Vallekas Basket

Público objetivo: clubes deportivos

Febrero 2026

Documento confidencial — uso interno exclusivo

Este informe está sujeto a revisión y actualización periódica.

Índice

- 1. Resumen ejecutivo
- 2. Fortalezas técnicas
- 3. Fortalezas de producto
- 4. Debilidades técnicas
- 5. Debilidades de producto
- 6. Recomendaciones prioritarias (A–L)
- 7. Ideas de diferenciación a largo plazo
- 8. Conclusión

1. Resumen Ejecutivo

VKBacademy es una plataforma LMS de dominio específico construida para el club Vallekas Basket. Su propuesta diferencial respecto a plataformas genéricas (Moodle, Google Classroom) es la integración nativa del contexto deportivo: niveles por año escolar (ESO/Bachillerato), reservas de clases con entrenadores, gamificación con mecánicas de racha, contenido generado con IA adaptado al deporte, y exámenes por curso y módulo descargables en PDF.

Dimensión	Estado	Nota
Arquitectura técnica	Sólida	Monorepo TypeScript estricto, corrección server-side, rate limiting
Cobertura de tests	Crítico	0% — cero archivos .spec.ts en toda la API
App móvil	Pendiente	Fase 9 no iniciada; audiencia consume principalmente en móvil
Engagement alumno	Parcial	Gamificación implementada pero sin leaderboard ni retos de equipo
Herramientas admin	Completo	Analytics, IA, CRUD completo, bancos de examen
Notificaciones	Email only	Resend activo; push notifications no implementadas
Documentación API	Manual	Sin Swagger; solo CLAUDE.md como referencia interna
Diferenciación vs Moodle	Alta	Reservas, Daily.co, IA contextualizada, lecciones interactivas

2. Fortalezas Técnicas

Monorepo tipado end-to-end	TypeScript estricto compartido entre API, web y mobile via packages/shared. Imposible desincronizar contratos entre capas.
Corrección server-side	Quiz y exámenes se corrigen en servidor desde questionsSnapshot en BD. El cliente nunca recibe isCorrect antes del submit.
Rate limiting global	ThrottlerModule activo: 100 req/min por IP. Protección básica ante bots, scraping y ataques de fuerza bruta.

URLs firmadas para vídeo	AWS S3 con expiración de 1 hora. El contenido multimedia del club no es público ni indexable externamente.
Gamificación event-driven	Los hooks checkAndAward se disparan con void — no bloquean el request HTTP. Decisión de arquitectura correcta.
Puntos denormalizados	User.totalPoints como campo calculado permite lecturas O(1) en la tienda, sin JOIN costoso en cada petición.
Contenido interactivo como JSON	Lesson.content Json? evita migraciones Prisma por cada nuevo tipo de actividad. Pragmático para la escala actual.
Fisher-Yates en exámenes	Selección aleatoria criptográficamente correcta de preguntas. Cada intento de examen es diferente.

3. Fortalezas de Producto

6 tipos de lección	VIDEO, QUIZ, EXERCISE, MATCH, SORT, FILL_BLANK cubre variedad pedagógica completa para un LMS deportivo.
Generación IA contextualizada	El generador incluye título del curso, nivel educativo y módulo en el prompt. Las preguntas son relevantes al contenido real.
Sistema de exámenes completo	Fisher-Yates, contador regresivo con auto-submit, respuesta única, correcciones con texto real, PDF descargable.
Analytics admin detallados	8 KPIs, gráfico SVG propio sin librerías externas, top cursos/alumnos, desglose de reservas por estado y modalidad.
Booking con videollamada	Daily.co integrado: sala creada automáticamente al confirmar reserva online, eliminada al cancelar.
Racha semanal ISO	Cálculo correcto de semana ISO con manejo del cambio de año (semana 1 del año nuevo). Credibilidad del sistema.
Rol TUTOR diferenciado	Los padres gestionan reservas sin que el alumno (menor de edad) tenga que hacerlo directamente.
Segmentación por curso escolar	Los alumnos solo ven cursos de su nivel (1ESO–2Bach). Evita sobrecarga de información irrelevante.
PDF de exámenes y quizzes	Informes descargables con jsPDF: diseño profesional con score, correcciones, fecha. Valor para padres y alumnos.

4. Debilidades Técnicas

Debilidad	Riesgo	Descripción
Sin tests automatizados	Alto	0 archivos .spec.ts en toda la API. Sin red de seguridad ante refactorizaciones o nuevas funcionalidades.
Sin Swagger/OpenAPI	Medio	Los endpoints no están documentados. Integración con app móvil (Fase 9) requerirá leer código fuente.
Rate limiting no granular	Medio	Límite de 100 req/min es global. Endpoints /generate (IA) y /auth/login necesitan límites propios más estrictos.
App móvil pendiente	Alto	Fase 9 no iniciada. El público objetivo (jugadores jóvenes) consume principalmente en smartphones.
Sin paginación en admin users	Bajo	GET /admin/users devuelve todos los usuarios sin paginación. Lento con >500 usuarios.

Sin caché Redis en servicios	Ø=ßá B a j o	Redis disponible pero sin usar en consultas pesadas como /admin/analytics. Queries costosas se repiten.
Tiempo de visionado no rastreado	Ø=ßà M e d i o	UserProgress.completed es boolean. Un alumno puede marcar una lección VIDEO sin verla. Datos de progreso poco fiables.
Sin notificaciones push	Ø=Ý4 A l t o	Email funciona para padres, no para adolescentes. Sin push, la plataforma es reactiva, nunca proactiva.

Ø=Üì 5. Debilidades de Producto

Sin ranking/clasificación	Los puntos existen pero no hay leaderboard. La mecánica competitiva — esencial en contexto deportivo — está desaprovechada.
Sin certificados de curso	No existe modelo Certificate ni generación de diplomas. Es un motivador clave y un recurso que los padres valoran.
Sin comentarios en lecciones	Los alumnos no pueden preguntar dudas sobre el contenido. Obliga al uso de canales externos (WhatsApp, email).
Tienda de merchandising estática	Los artículos están hardcoded en el frontend. Añadir/editar artículos requiere despliegue de código.
Sin prerequisites de curso	No se puede bloquear acceso a un módulo avanzado hasta completar uno básico. Pérdida de control pedagógico.
Sin calendario visual de reservas	El tutor ve una lista de reservas, no un calendario. Comparar disponibilidad y planificar es difícil.
Rol TEACHER pasivo	TEACHER confirma reservas y gestiona disponibilidad, pero no puede iniciar comunicación ni ver progreso agregado de alumnos.
Sin diario de entrenamiento	El alumno no puede registrar actividad fuera de la plataforma. Desconexión con la vida deportiva real.
Sin modo offline	Cuando la app móvil se implemente, no habrá soporte offline. Los desplazamientos a torneos impedirán estudiar sin datos.

Ø=p€ 6. Recomendaciones Prioritarias

Prioridad Alta — Retención y Engagement

A Tabla de clasificación (Leaderboard)

Por que: Los deportistas son intrínsecamente competitivos. Un ranking visible es el motivador más efectivo en este contexto. Los puntos ya existen pero no se muestran comparativamente.

Propuesta: GET /leaderboard?scope=global|course|schoolYear — top 10 por puntos. Filtrar por nivel educativo para evitar comparaciones injustas (2Bach vs 1ESO). Componente en DashboardPage. Implementación trivial: User.totalPoints ya existe.

B Notificaciones push (Expo)

Por que: Email funciona para padres/tutores, no para adolescentes. Sin push, los recordatorios de clases o nuevos contenidos no llegan a la audiencia principal.

Propuesta: User.expoPushToken String? en schema. POST /notifications/register-push guarda el token. Activar para: reserva confirmada, nueva lección, reto completado, recordatorio 1h antes de clase. Expo Notifications SDK ya está en el stack.

C

Certificados de finalización de curso (PDF)

Por que: Los padres los valoran para portafolios del alumno. Los alumnos los usan como motivación de fin de ciclo. Diferenciador frente a Moodle.

Propuesta: Modelo Certificate { id, userId, courseId, issuedAt, pdfKey }. Se genera automáticamente cuando CourseProgress.percentage = 100. PDF con jsPDF (ya instalado): nombre, curso, nivel, fecha, sello del club. Almacenar en S3, servir con URL firmada.

D

Comentarios y preguntas en lecciones

Por que: Sin canal de comunicación sobre el contenido, los alumnos abandonan cuando se bloquean en algún concepto.

Propuesta: Modelo LessonComment { id, lessonId, userId, text, parentId? } — árbol de 2 niveles. GET y POST /lessons/:id/comments. TEACHER ve comentarios de sus alumnos. Email/push al teacher cuando un alumno comenta.

Prioridad Media — Pedagogía y Operacional

E

Prerequisitos de módulo/curso

Por que: Permite diseñar rutas de aprendizaje progresivas (p.ej. "Táctica avanzada" requiere "Fundamentos básicos"). Sin esto, el alumno puede saltar contenido esencial.

Propuesta: Campo Course.prerequisiteCourseId String?. El endpoint GET /courses/:id devuelve isUnlocked: boolean calculado desde el progreso del alumno. Visual: cursos bloqueados con candado en CoursesPage.

F

Diario de entrenamiento personal

Por que: Conecta la plataforma con la vida deportiva real del alumno. Diferenciador clave respecto a Moodle y cualquier LMS genérico.

Propuesta: Modelo TrainingLog { userId, date, type (MATCH|TRAINING|GYM|REST), durationMinutes, notes? }. Vista de calendario en app móvil con heatmap de actividad. Nuevo reto TRAINING_LOGGED: registra N sesiones.

G

Gestión de artículos de tienda por admin

Por que: Los artículos de la tienda están hardcodedados en el frontend. Cambiar precios o añadir artículos requiere un despliegue de código.

Propuesta: Modelo StoreItem { name, description, cost, imageUrl?, isActive, stock? }. GET /store/items + CRUD en /admin/store/items. POST /challenges/redeem acepta itemId en lugar de { itemName, cost } hardcodedado.

H

Feedback del profesor al alumno

Por que: El TEACHER no tiene canal para comunicarse proactivamente. La tutoría se reduce a reservar horas.

Propuesta: Modelo Feedback { teacherId, studentId, courseId?, text, createdAt }. POST /feedback [TEACHER, ADMIN]. Visible en panel del alumno y del tutor como "Observaciones del profesor". Email/push de notificación.

I Tests automatizados (unit + e2e)

Por que: 0 archivos .spec.ts es el riesgo técnico más grave. Cualquier refactorización puede romper funcionalidades sin saberlo.

Propuesta: Jest ya configurado en el proyecto. Empezar por: exams.service.ts (corrección), challenges.service.ts (racha semanal), auth.service.ts. Tests e2e de flujos principales con supertest.

J Swagger/OpenAPI

Por que: La app móvil (Fase 9) necesitará documentación de la API. Sin ella, el desarrollo será lento y propenso a errores de contrato.

Propuesta: @nestjs/swagger ya disponible como dependencia transitiva. Decoradores @ApiOperation, @ApiResponse en controllers. Exponer en /api/docs solo en NODE_ENV !== production.

K Rastreo de tiempo de visionado de vídeo

Por que: Actualmente un alumno puede marcar una lección VIDEO como completada sin verla. Invalida los datos de progreso y el reto TOTAL_HOURS.

Propuesta: UserProgress.watchedSeconds Int @default(0). Frontend envía pings cada 10s de reproducción activa. POST /lessons/:id/complete requiere watchedSeconds >= duración * 0.8 para vídeos.

L Calendario visual de reservas

Por que: El flujo de reservas muestra una lista. Para tutores con varios hijos o varios profesores, la gestión visual es esencial.

Propuesta: Componente Calendar implementado con CSS Grid nativo (sin dependencias). Vista mensual/semanal. Slots disponibles del profesor en verde, reservas en morado.

7. Ideas de Diferenciación a Largo Plazo

Idea	Descripción
Retos de equipo	Agrupar alumnos reflejando el equipo deportivo real para retos colectivos. "El equipo sub-16 completa 50 lecciones esta semana."
Quiz en directo (WebSockets)	El profesor lanza un quiz en tiempo real durante un entrenamiento. Los alumnos responden desde el móvil y ven el ranking al instante en pantalla.
Integración calendario del club	Importar partidos y entrenamientos del club (vía iCal o API) para que aparezcan junto con las reservas en un mismo calendario.
Rutas de aprendizaje adaptativas	Basadas en resultados de exámenes: "Tu score en Defensa es 42% — te recomendamos repasar el módulo 2." Requiere IA de recomendación.
Modo torneos	Durante un torneo, desbloquear contenido especial (análisis táctico del rival) visible solo esa semana. Crear evento de club.

Rol COACH	Nuevo rol con acceso al progreso físico-táctico del equipo completo, sin gestionar reservas individuales. Distinto del TEACHER.
Análisis de vídeo propio	Reemplazar YouTube embebido por reproductor propio con anotaciones de timestamp por parte del profesor. Mayor control pedagógico.
Integración con wearables	Importar datos de frecuencia cardíaca o GPS de entrenamientos (Garmin, Apple Health) para enriquecer el diario de entrenamiento.

8. Conclusión y Próximos Pasos

La plataforma tiene una base técnica sólida y un conjunto de funcionalidades que supera a soluciones genéricas para el caso de uso de un club deportivo con alumnado menor de edad. Las tres brechas más críticas son:

- Ausencia de tests automatizados (riesgo de regresión al escalar el equipo)
- App móvil no implementada (canal principal del público objetivo)
- Falta de mecánicas competitivas visibles (leaderboard, retos de equipo)

Las tres apuestas de mayor retorno para la próxima fase serían:

1ª prioridad: App móvil + push notifications — llegar al alumno donde está

2ª prioridad: Tabla de clasificación por nivel educativo — activar la competencia inherente al deporte

3ª prioridad: Certificados de finalización de curso — motivación de cierre y valor para las familias

Este informe es un documento vivo. Se actualizará en cada sprint de producto con nuevos hallazgos, métricas de uso real y ajustes de prioridad según el feedback del club.