Plan Maestro
Nexus Boards v1

Aplicación web tipo Kanban con integración de calendarios externos, dashboard de métricas, slideshow y soporte offline básico, diseñada con foco en accesibilidad (WCAG 2.1 AA) y trazabilidad completa de requisitos → pruebas → despliegue.

Generado: 12 Sep 2025, 09:51 CST

Visión General

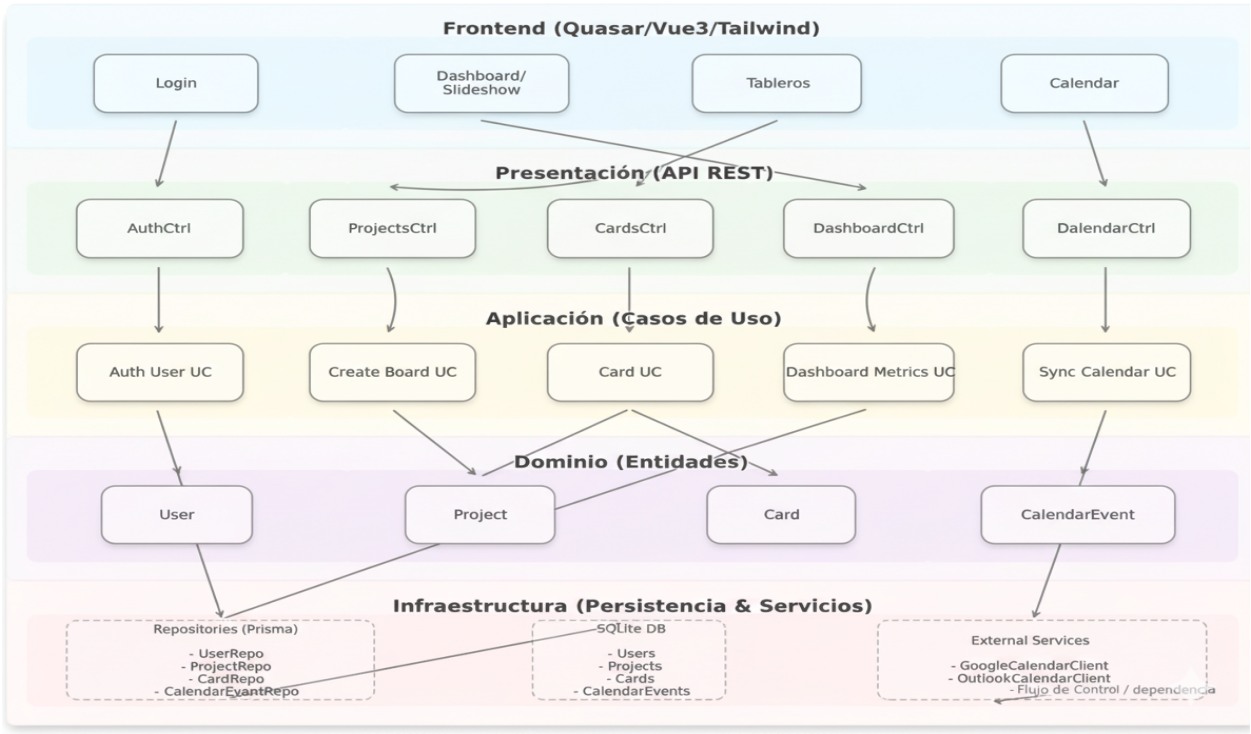
Entregar una aplicación web que permita gestionar proyectos y *cards* con tablero visual, integración con calendarios (Google/Outlook), dashboard con métricas y un slideshow de imágenes. La app será accesible, tendrá funcionamiento offline básico y trazabilidad completa.

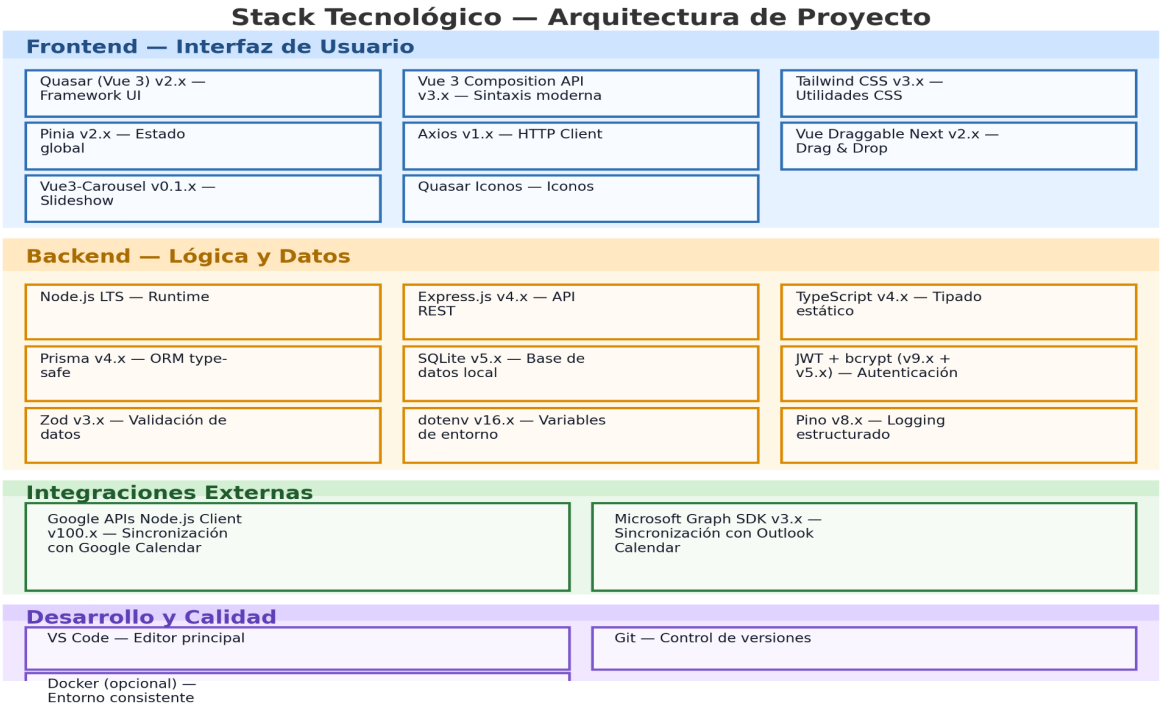
Principios Rectores

- Simplicidad operativa: priorizar soluciones claras sobre complejidad innecesaria.
- Idempotencia: operaciones seguras frente a reintentos.
- Transaccionalidad: consistencia en operaciones críticas (p. ej., reordenamiento).
- Consistencia eventual: aceptar retrasos en sincronización externa.
- Diseño accesible: cumplir WCAG 2.1 AA desde el inicio.

Alcance Técnico v1 (Incluye / Excluye)

Incluye	Excluye
• Autenticación JWT con rotación de refresh. • CRUD de Proyectos y Cards. • Drag■and■drop con persistencia transaccional. • Dashboard con KPIs (por estado y vencimientos). • Slideshow local. • Integración unidireccional con Google/Outlook. • Reintentos con backoff y cola de pendientes. • Offline básico y trazabilidad completa.	• Multi■tenancy avanzado y roles complejos. • Notificaciones push. • Sincronización bidireccional completa de calendarios. • Facturación.





1. Requisitos y Trazabilidad

- Historias de usuario en BDD (Given/When/Then) con identificador NB■###.
- Criterios de aceptación por historia.

- Matriz Requisito→Test (CSV/MD).
- Glosario de términos comunes (Card, Board, Position, etc.).

Artefactos: Matriz de trazabilidad, Glosario, Historias de usuario con criterios

2. Modelo de Dominio y Contratos

- Diagramas UML: clases, casos de uso, actividades.
- Entidades: User, Project, Card, CalendarEvent y enums (Status, Provider).
- Reglas: posición única por columna; límites y validaciones.
- Errores comunes: 409, 422, 423.
- Contratos de servicios (puertos/UC) con entradas/salidas.

Artefactos: Diagramas, Contratos y reglas de negocio

3. Persistencia y Migraciones

- Esquema SQLite con tablas, índices y FKs.
- Soft■delete para Cards; hard■delete para CalendarEvent.
- Migraciones iniciales y seeds (usuario/proyecto/12 cards).
- Transacciones para reorder y setDueDate.

Artefactos: Scripts de migración, Seeds, Esquema de BD

4. API Pública

- Endpoints: /api/v1/login, /password-reset, /projects, /cards, /cards/reorder, /calendar/events.
- Documentación OpenAPI v3 con códigos 200/201/400/401/403/409/422/429/500.
- Idempotencia con X■Idempotency■Key en calendario.

Artefactos: OpenAPI (YAML), Colección Postman/Bruno

5. Autenticación y Autorización

- Login con rate■limit y JWT (access+refresh).
- Reset de contraseña con token de 15 min.
- Autorización por ownerId.
- Rotación y blacklist de refresh tokens.

Artefactos: Módulo de auth, Tests de seguridad

6. Casos de Uso Núcleo

- UC: AuthUser, CreateProject, CRUDCard, ReorderCards, SetDueDate, DashboardMetrics, SyncCalendar.
- Reorder: transacción atómica con verificación de colisiones.
- Date chip: TZ y modo 'todo el día'.
- Métricas: contadores por estado y vencidos.

Artefactos: Implementación UC, Tests unitarios e integración

7. Integraciones Externas (Google/Outlook)

- OAuth para Google y Outlook.
- Servicio de sincronización con reintentos y backoff.
- Mapeo: summary, description, start/end, timezone.
- Estados: Synced, Pending, Error; limpieza al borrar fecha.

Artefactos: Adaptadores por proveedor, Tests con mocks

8. Frontend (Quasar/Vue/Tailwind)

- Páginas: Login, Dashboard, Boards (DnD), Calendar, Settings.
- Componentes: Kanban board, Cards, DatePicker, Slideshow.
- DnD accesible (teclado) y estados de UI (loading/empty/error).
- Offline: cambios locales y posterior sincronización.

Artefactos: Código frontend, Tests unitarios (Vue Test Utils) y E2E (Playwright)

9. Observabilidad y Seguridad

- Logs Pino con correlación por requestId.
- Helmet y CORS; validación con Zod.
- Auditoría de permisos y filtros de PII.

Artefactos: Config de logs, Middlewares de seguridad, Tests de sanitización

10. Calidad, CI y Entrega

- CI para lint, typecheck, unit, integration, E2E.
- Cobertura objetivo: ≥85% dominio, ≥70% UI.
- Build reproducible y Docker opcional.

- Git hooks (pre■commit: lint+typecheck; pre■push: unit).
- Publicar OpenAPI/colección y README de instalación/demo.

Artefactos: Pipeline CI, Scripts de build, Documentación

Matriz de Trazabilidad (Ejemplo)

ID	Descripción	Endpoint/UC	Test Principal
NB■001	Login	/login	E2E login (éxito/error)
NB■010	Crear Proyecto	/projects	Integración crear/listar
NB■020	CRUD Card	/cards	Unit + Integración CRUD
NB■030	Reordenar	/cards/reorder	E2E DnD + 409

Checklist de Aceptación (20 pasos)

1. Crear usuario demo, login y refresh token.
2. Crear proyecto y verificar en lista.
3. Crear 12 cards variadas.
4. Reordenar 3 cards y verificar persistencia tras refresh.
5. Forzar 409 con dos clientes; UI recarga orden del servidor.
6. Cambiar estado de card entre columnas.
7. Asignar fecha 'todo el día' y verificar chip.
8. Asignar fecha/hora y verificar duración por defecto.
9. Conectar Google y seleccionar calendario.
10. Guardar fecha y ver evento en proveedor.
11. Borrar fecha y verificar eliminación remota.
12. Simular 429/503 → reintentos y estado 'Pending'.
13. Desconectar cuenta y ver UI correcta.
14. Dashboard: KPIs y próximos 7 días.
15. Slideshow: imágenes locales.
16. Modo offline: cambios quedan 'Pending' y luego sincronizan.
17. Accesibilidad: navegación por teclado en DnD y chips.
18. Logs: sin PII y con requestId.
19. Lighthouse: Perf ≥80, A11y ≥90, Best Practices ≥90.
20. CI verde y cobertura cumplida.

Gestión de Riesgos (Top)

Riesgo	Prob.	Impacto	Mitigación
Límites de APIs externas	Alta	Medio	Backoff exponencial y cola de reintentos
Conflictos en reordenamiento	Media	Alto	Transacciones e inspección de consistencia
Time zones	Alta	Medio	Normalizar en backend (UTC)
Pérdida de conexión	Alta	Medio	Estado 'Pending' + reconciliación
Performance en tableros grandes	Media	Medio	Virtualización/Lazy loading

Cronograma (16 semanas)

- **Semanas 1–2:** Setup y arquitectura base; modelos de datos.
- **Semanas 3–4:** Autenticación (JWT/refresh), reset y rate■limit.
- **Semanas 5–6:** CRUD Proyectos/Cards; archivado.
- **Semanas 7–8:** Reordenamiento transaccional; conflictos.
- **Semanas 9–10:** Integración calendarios; OAuth; sync unidireccional.
- **Semanas 11–12:** Frontend Kanban/Calendario/Dashboard.
- **Semanas 13–14:** Offline básico y optimizaciones/a11y.
- **Semanas 15–16:** Testing integral; E2E; performance; preparación de despliegue.

Métricas de Calidad y Success Criteria

- Cobertura: ≥85% dominio, ≥70% UI; 0 errores de TypeScript.
- 0 vulnerabilidades críticas; Lighthouse Perf ≥80, A11y ≥90.
- Tiempo de carga inicial < 3s (3G); API p95 < 200 ms.
- Checklist de 20 pasos completado; documentación finalizada.

Artefactos Finales

- OpenAPI v1 (YAML) y colección Postman/Bruno.
- Matriz de trazabilidad (CSV/MD) y checklist QA.
- README + .env.example + scripts de demo.
- Diagramas UML en uml_out/.