

TaskFlow Pro — SCRUM Maestro (12 meses)

Plan profesional — 4 desarrolladores, 9h/día, sprints de 2 semanas

Resumen ejecutivo: Documento de planeación SCRUM integral para construir una aplicación web profesional de gestión de tareas (TaskFlow Pro). Incluye Sprint 0, épicas, backlog, historias, criterios de aceptación, roadmap de 12 meses, métricas, riesgos y Definition of Done.

1. Visión y Objetivos

- Gestor de tareas multiusuario con autenticación, múltiples listas, colaboración en tiempo real y UI moderna.
- API REST pública para integrar clientes (web, móvil, CLI).
- Calidad productiva: pruebas automáticas, seguridad OWASP, CI/CD y observabilidad.

2. Equipo, Roles y Ceremonias

Rol	Responsabilidades principales
Product Owner (tú)	Sprint 0 (documentación), priorización del backlog, aceptación de historias.
Arquitecto/Analista	Diseño de arquitectura, modelos, estándares de código, revisiones técnicas.
Backend Dev	Node.js/Express, BD, autenticación, websockets, pruebas de integración.
Frontend Dev	React + UI/UX, estado global, accesibilidad, e2e con Cypress.
QA/DevOps	CI/CD, cobertura, calidad de código, releases, monitoreo y alertas.

Ceremonias:

- Sprint Planning (4h): selección de historias, estimación y objetivos del sprint.
- Daily Scrum (15m): sincronización, bloqueadores, métricas de flujo (WIP, throughput).
- Refinement (1-2h/semana): descomposición, criterios de aceptación y DoR.
- Sprint Review (2h): demo del incremento, feedback de stakeholders.
- Retrospectiva (1.5h): acciones de mejora con responsables y fechas.

3. Sprint 0 — Documentación e Ingeniería (hecho por el PO)

Objetivo: que cualquier desarrollador pueda iniciar Sprint 1 solo con esta documentación.

- Documento de Visión y Alcance.
- SRS — Requisitos funcionales y no funcionales (rendimiento, seguridad, mantenibilidad, accesibilidad).
- Historias de Usuario (≥ 25), cada una con Criterios de Aceptación (Gherkin), dependencias y valor.
- Casos de Uso y UML: Caso de Uso, Clases preliminares, Secuencia (login, CRUD, compartir lista), Actividad (flujo).
- Arquitectura lógica: capas (presentación, dominio, persistencia), ADRs y modelos de datos.
- Plan de QA inicial: estrategia (unitarias, integración, e2e), riesgos de calidad y matriz de cobertura.
- Prototipo UI (Figma): navegación, estados vacíos, dark/light.
- Backlog inicial cargado en Trello/Jira con prioridades MoSCoW y estimaciones.
- Definiciones: DoR y DoD del equipo.

Criterios de aceptación Sprint 0:

- Documentos en /docs y versionados.
- Cada historia tiene criterios claros y trazabilidad a requisitos.
- Wireframes y diagramas exportados (PDF/PNG) y vinculados.

4. Épicas y Backlog de Producto

Epic A — Fundamentos/Infraestructura

- Monorepo (pnpm workspaces) con apps /api y /web.
- CI inicial: ESLint/Prettier, tests, build; husky + lint-staged.
- Base de datos (PostgreSQL) + ORM (Prisma). Seeds.
- Observabilidad base: logs estructurados, trazas (OpenTelemetry opcional).

Epic B — Gestión de Tareas

- Modelo Task: id, title, description, status, dueDate, tags, createdAt, updatedAt.
- CRUD API REST con validación (Zod/Joi).
- Filtros: pendientes/completadas, por etiqueta y por fechas.
- Orden: vencimiento, prioridad, creación.

Epic C — Multiusuario y Autenticación

- Registro/login, restablecer contraseña, verificación por email.
- JWT + refresh tokens; CORS/CSRF.
- Perfiles y preferencias; rate limiting y bloqueo por intentos.

Epic D — UI/UX Web (React)

- Design system (Tailwind + Radix UI).
- Dashboard con múltiples listas y panel de detalles.
- Accesibilidad (WCAG AA) e i18n.
- Dark/Light mode.

Epic E — Colaboración en tiempo real

- Compartir listas: owner, editor, lector.
- Socket.IO para actualizaciones instantáneas.
- Historial de cambios (audit log).
- Comentarios y menciones (@usuario).

Epic F — Integraciones y Notificaciones

- Auth social (Google).
- Notificaciones email (Resend/SES) y push web.
- Webhooks y export/import CSV/JSON.

Epic G — Reporting y Analítica

- Métricas personales y reportes semanales.
- Panel de productividad con gráficos.
- Endpoint de métricas para Prometheus.

Epic H — Seguridad/Compliance

- OWASP ASVS L1, SCA/SAST en CI.
- Política de contraseñas; cifrado en reposo/tránsito.
- Backups y retención.

Epic I — QA y E2E

- Unit tests (Jest) y cobertura mínima 80%.
- Integración (Supertest).
- E2E (Cypress), rendimiento (k6) y accesibilidad (axe).

Epic J — DevOps y Despliegue

- Dockerfiles multi-stage y docker-compose.
- Entornos dev/staging/prod y variables seguras.
- CD a nube (Fly.io/Render/Vercel) y observabilidad.

Epic K — Documentación & Publicación

- README pro, contribución y styleguide.
- Manual de usuario y FAQs.
- ADRs y diagramas finales; Docusaurus para docs.

5. Historias de Usuario (muestra ampliada con CA)

HU-01 Registro y Login

- Como visitante quiero registrarme con email para acceder a la app.
- CA: Email de verificación; sin verificar no se permite login.

HU-02 Crear tareas

- Como usuario quiero crear tareas con título y descripción para organizarlas.
- CA: Título obligatorio; feedback; aparece en la lista sin refrescar.

HU-03 Completar y filtrar

- Como usuario quiero marcar tareas como completadas y filtrarlas por estado.
- CA: Toggle en UI; filtros persistentes; accesos de teclado.

HU-04 Fechas de vencimiento y recordatorios

- Como usuario quiero establecer due date para priorizar.
- CA: Orden por fecha; notificación 24h antes; color de alerta.

HU-05 Compartir lista

- Como usuario quiero compartir mi lista con permisos.
- CA: Invitación por email; roles lector/editor; auditoría.

HU-06 Tiempo real

- Como colaborador quiero ver cambios al instante.
- CA: Latencia < 300ms entre clientes.

HU-07 Exportación

- Como usuario quiero exportar mis tareas a CSV/JSON.
- CA: Archivo descargable; respeta filtros.

HU-08 Accesibilidad

- Como usuario con lector de pantalla quiero navegar la app.
- CA: Roles ARIA, foco visible, contraste AA.

6. Roadmap de Sprints (12 meses, 26 sprints de 2 semanas)

# Sprint	Objetivo principal	Épicas foco
S0	Documentación completa (PO): SRS, HU, UML, arquitectura preliminar	Epic A
S1	Monorepo, CI básico, base de datos y modelos iniciales	Epic A
S2	API CRUD mínima de tareas + tests	Epic B, I
S3	Frontend React básico (listas y creación)	Epic D
S4	Filtros y orden; validación de API	Epic B
S5	Autenticación JWT + recuperación de contraseña	Epic C
S6	Perfiles y preferencias; seguridad	Epic C, H
S7	Multi-listas; búsqueda y paginación	Epic B, D
S8	Compartir listas (roles) y auditoría	Epic E
S9	Socket.IO tiempo real	Epic E
S10	Notificaciones email/push; export/import	Epic F
S11	Analítica y reportes	Epic G
S12	Accesibilidad e i18n; dark/light sólido	Epic D
S13	E2E Cypress para flujos críticos	Epic I
S14	Dockerización y entornos; CD staging	Epic J
S15	Monitoreo y alertas; métricas negocio	Epic J, G
S16	Optimización rendimiento API y UI	Epic B, D
S17	OAuth Google y webhooks	Epic F
S18	Seguridad CI (SAST/SCA) y políticas	Epic H
S19	Escalabilidad horizontal (caché/colas)	Epic J
S20	Refactor y deuda técnica	Epic A
S21	Colaboración avanzada (comentarios/menciones)	Epic E
S22	Panel de productividad extendido	Epic G
S23	Polish UX (atajos, estados vacíos, tour)	Epic D
S24	Hardening y performance final	Epic H, D
S25	Documentación final + lanzamiento v1.0.0	Epic K, J

7. Definition of Ready (DoR) y Definition of Done (DoD)

DoR — una historia está lista cuando:

- Descripción clara, valor de negocio, criterios de aceptación (Gherkin) y estimación (< 8 puntos).
- Dependencias identificadas; mockups o contratos de API listos.
- Datos de prueba disponibles; riesgo evaluado.

DoD — el incremento está hecho cuando:

- Código revisado y mergeado; cobertura $\geq 80\%$; ESLint sin errores.
- Documentación y changelog actualizados; rendimiento dentro de objetivos.
- Desplegado en entorno objetivo con monitoreo activo.

8. Requisitos No Funcionales y KPIs

NFR clave:

- P95 API < 300ms; disponibilidad 99.5% mensual.
- Seguridad: OWASP Top 10 mitigado; secretos rotados.
- Usabilidad: WCAG AA; responsive.
- Escalabilidad: 5k usuarios activos sin degradación notable.

KPIs del proceso:

- Velocidad del equipo (story points).
- Tiempo de ciclo y WIP.
- Defectos por release; tasa de escapes.
- Cobertura de pruebas; éxito de pipelines.

9. Riesgos y Mitigaciones

Riesgo	Impacto	Mitigación
Alcance cambiante	Retrabajo y atrasos	Refinement frecuente; criterios firmes.
Falta de pruebas	Bugs en producción	Cobertura mínima; gates en CI y PR templates.
Cuellos de botella en DB	Rendimiento pobre	Índices, caché, pruebas de carga.
Fuga de secretos	Brecha de seguridad	Vault/ENV seguros, rotación, escaneo SAST/SCA.
Rotación del equipo	Pérdida de contexto	Docs vivas, pairing y buen onboarding.

10. Artefactos y Herramientas

- Gestión: Trello/Jira, GitHub Projects.
- Repo: GitHub, Conventional Commits y Semantic Release.
- CI/CD: GitHub Actions, Docker, CD a Vercel/Fly.io.
- QA: Jest, Supertest, Cypress, k6, axe.
- Docs: Docusaurus, ADRs, PlantUML, Figma.