

Propuesta de Evaluación: Desarrollo Tradicional vs IA-Asistido

Evaluación del Impacto de Herramientas de Código Agéntico
en la Calidad y Productividad del Desarrollo

Desafío: Loan Tracker UI

Stack: Next.js 15.3, Tailwind CSS, shadcn/ui, SQLite + Prisma

Duración: 8 horas efectivas

Preparado por: Felipe Valenzuela Beck

Fecha: 3 de Junio de 2025

Versión: 1.0 Lean

Índice

1. Resumen Ejecutivo	2
2. Diseño del Experimento	2
2.1. Participantes	2
2.2. Desafío	2
3. Métricas Clave	2
3.1. Calidad de Código	2
3.2. Arquitectura	3
3.3. Performance & UX	3
3.4. Seguridad Básica	3
4. Herramientas de Medición	3
4.1. Automatizadas	3
4.2. Manuales (Code Review)	3

1 Resumen Ejecutivo

Propuesta para evaluar objetivamente el impacto de herramientas de código agéntico (Windsurf, Copilot, Cursor, Claude Code) mediante el desafío “Loan Tracker UI”. Buscamos métricas cuantificables que demuestren si el desarrollo asistido por IA mejora la calidad y productividad.

Objetivo: Obtener datos concretos sobre el ROI de adoptar herramientas IA en el equipo de desarrollo.

2 Diseño del Experimento

2.1 Participantes

- **Grupo A:** 1 desarrollador CON herramientas IA
- **Grupo B:** 2 desarrolladores SIN herramientas IA
- **Duración:** 8 horas efectivas (mismo desafío para todos)

2.2 Desafío

Componente	Descripción
Proyecto	Loan Tracker UI (gestión de préstamos)
Stack	Next.js 15.3, Tailwind CSS, shadcn/ui, SQLite + Prisma
Requisitos	5 historias de usuario definidas

3 Métricas Clave

3.1 Calidad de Código

- **ESLint Score:** Errores y warnings
- **SonarQube:** Code smells, technical debt, maintainability rating
- **Code Coverage:** Objetivo $\geq 80\%$
- **Complejidad:** Funciones ≤ 10 complejidad ciclomática
- **Duplicación:** $\leq 3\%$ código duplicado

3.2 Arquitectura

- **Separación Client/Server Components** - Rúbrica de puntuación numérica (0-25 puntos)
- **Modularidad:** Componentes reutilizables - Checklist con puntos (0-25 puntos)
- **Estructura de carpetas:** Organización lógica - Rúbrica de evaluación (0-25 puntos)
- **Patrones:** Uso correcto de Next.js 15 - Checklist con puntos (0-25 puntos)

3.3 Performance & UX

- **Lighthouse Score:** Performance, Accessibility, Best Practices (automatizado)
- **Bundle Size:** Tamaño del build (automatizado)
- **Core Web Vitals:** LCP, FID, CLS (automatizado)
- **Responsive Design:** Mobile-first - Rúbrica de puntuación numérica (0-100 puntos)

3.4 Seguridad Básica

- **npm audit:** Sin vulnerabilidades críticas (automatizado)
- **Autenticación:** Implementación correcta - Checklist con puntos (0-30 puntos)
- **Validación:** Input sanitization - Checklist con puntos (0-30 puntos)
- **HTTPS:** Configuración segura - Rúbrica de evaluación (0-40 puntos)

4 Herramientas de Medición

4.1 Automatizadas

```
- ESLint + Prettier (calidad de código)
- SonarQube Cloud (análisis estático integral)
- Jest + Coverage (testing y cobertura)
- Lighthouse CI (performance y accesibilidad)
- npm audit + Snyk (seguridad)
- Bundle Analyzer (optimización)
- CLOC (conteo de líneas de código lógicas)
```

4.2 Manuales (Code Review)

- Arquitectura y patrones (Rúbricas de puntuación numérica)
- Uso de mejores prácticas (Checklists con puntos)
- Documentación y claridad (Rúbrica 0-100)
- Innovación en soluciones (Escala cuantificada 0-100)