

# SISTEMA DE CONTROL DE ASISTENCIA SENA CAQUETÁ CTA

**Plan Agile Acelerado: MVP v1 (11 días) + Versión Completa (17 días)**

## INFORMACIÓN EJECUTIVA

### Fechas Críticas:

- **Inicio:** 10/11/2025 (Lunes)
- **MVP v1 Funcional:** 21/11/2025 (Viernes) - 11 días de desarrollo
- **Versión Completa:** 27/11/2025 (Jueves) - 17 días totales

**Equipo:** 4 Desarrolladores Full-Stack + Uso de IA (Cursor, GitHub Copilot)

**Jornada:** 9-11 horas diarias (6h mañana + 3-5h tarde)

**Enfoque:** MVP primero, features incrementales después

## 1. ESTRATEGIA ACELERADA

### 1.1 Principios Clave

**MVP (Minimum Viable Product):** Solo lo esencial para v1

- Autenticación de instructores
- Registro manual de asistencia (método más simple)
- Listado de aprendices
- Almacenamiento en BD

**Métodos de Captura (Post-MVP):** Agregados después del 21/11

- Código de barras (Fase 2)
- Código QR (Fase 2)
- Biometría (Fase 3)

### Stack Simplificado:

- PHP 8.2 nativo (sin frameworks de overhead)
- MySQL simple (sin Ratchet WebSocket aún)
- HTML5 + CSS3 + Vanilla JavaScript
- Cursor/Copilot para generar boilerplate

### 1.2 Uso de Herramientas IA

**GitHub Copilot / Cursor:** Acelerar código repetitivo

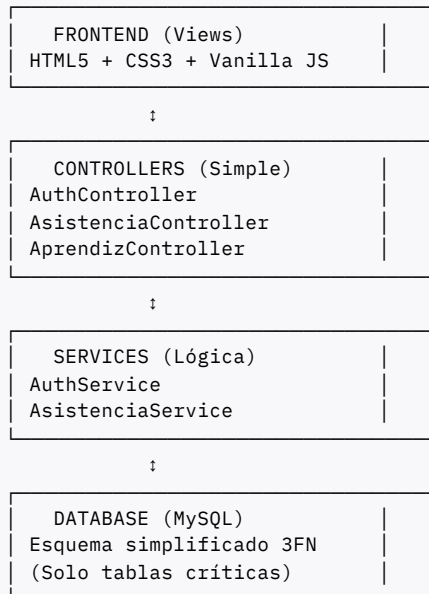
- Controllers CRUD automáticos
- Queries SQL comunes
- Validaciones boilerplate
- HTML/CSS templates

### Prompts Efectivos para Desarrollo Rápido:

"Genera un CRUD Controller en PHP para la entidad Asistencia con validaciones"  
"Crea las queries SQL necesarias para un registro de asistencia diario"  
"Genera la tabla HTML con paginación para mostrar aprendices"

**Estimación de Aceleración:** 30-50% más rápido en codificación boilerplate

## 2. ARQUITECTURA MVP SIMPLIFICADA



### 2.1 Tablas de BD MVP (Solo Esenciales)

```
-- Mínimo para MVP
CREATE TABLE usuarios (
  id INT AUTO_INCREMENT PRIMARY KEY,
  documento VARCHAR(20) UNIQUE,
  nombre VARCHAR(100),
  email VARCHAR(100) UNIQUE,
  password_hash VARCHAR(255),
  rol ENUM('instructor', 'coordinador', 'admin'),
  INDEX idx_documento (documento)
);

CREATE TABLE aprendices (
  id INT AUTO_INCREMENT PRIMARY KEY,
  documento VARCHAR(20) UNIQUE,
  nombre VARCHAR(100),
  apellido VARCHAR(100),
  codigo_carnet VARCHAR(50),
  estado ENUM('activo', 'retirado')
);

CREATE TABLE fichas (
  id INT AUTO_INCREMENT PRIMARY KEY,
  numero_ficha VARCHAR(20) UNIQUE,
  nombre VARCHAR(200),
  estado ENUM('activa', 'finalizada'),
  INDEX idx_numero (numero_ficha)
);

CREATE TABLE ficha_aprendiz (
  id_ficha INT,
  id_aprendiz INT,
  PRIMARY KEY (id_ficha, id_aprendiz),
  FOREIGN KEY (id_ficha) REFERENCES fichas(id),
  FOREIGN KEY (id_aprendiz) REFERENCES aprendices(id)
);

CREATE TABLE asistencias (
  id INT AUTO_INCREMENT PRIMARY KEY,
```

```

    id_aprendiz INT,
    id_ficha INT,
    fecha DATE,
    hora TIME,
    estado ENUM('presente', 'ausente', 'tardanza'),
    registrado_por INT,
    FOREIGN KEY (id_aprendiz) REFERENCES aprendices(id),
    FOREIGN KEY (id_ficha) REFERENCES fichas(id),
    FOREIGN KEY (registrado_por) REFERENCES usuarios(id),
    UNIQUE KEY unique_registro (id_aprendiz, id_ficha, fecha),
    INDEX idx_fecha (fecha)
);

```

### 3. SPRINT DIARIOS (11 DÍAS MVP)

#### SPRINT 1: DÍA 1-2 (Lunes 10 - Martes 11 Nov)

**Tema:** Setup, Base de Datos, Estructura MVC

##### Dev 1: Base de Datos (Líder)

- **Tarea 1.1:** Crear script SQL completo (5 tablas MVP)
  - Tiempo: 4 horas
  - Entregable: script.sql funcional
- **Tarea 1.2:** Poblar BD con datos iniciales (50 fichas, 500 aprendices)
  - Tiempo: 3 horas
  - Herramienta: Cursor generará queries INSERT batch

##### Dev 2: Backend Core (Líder)

- **Tarea 2.1:** Estructura carpetas MVC + Autoloader PSR-4
  - Tiempo: 2 horas
  - Archivos: /config, /src/Controllers, /src/Services, /src/Models
- **Tarea 2.2:** Clase Database (conexión PDO + Singleton)
  - Tiempo: 3 horas
  - Código generado con Copilot
- **Tarea 2.3:** AuthService (login/logout) + UserRepository
  - Tiempo: 4 horas

##### Dev 3: Frontend Setup (Líder)

- **Tarea 3.1:** Layout base + CSS framework (Bootstrap mínimo)
  - Tiempo: 3 horas
  - Archivos: base.html, style.css
- **Tarea 3.2:** Página de Login (HTML + Validación JS)
  - Tiempo: 3 horas
- **Tarea 3.3:** Dashboard template (esqueleto)
  - Tiempo: 2 horas

## Dev 4: Integración + DevOps

- **Tarea 4.1:** Configuración de servidor Apache/Nginx local
  - Tiempo: 2 horas
- **Tarea 4.2:** Router PHP básico (URL rewriting)
  - Tiempo: 3 horas
- **Tarea 4.3:** Middleware de autenticación
  - Tiempo: 4 horas

**Stand-up Diario:** 9:30 AM (15 min máximo)

- ¿Qué completé ayer?
- ¿Qué completaré hoy?
- ¿Qué me bloquea?

**Hito de Cierre:** BD creada, Backend estructura lista, Login funcional

## SPRINT 2: DÍA 3-4 (Miércoles 12 - Jueves 13 Nov)

**Tema:** Autenticación completa, RBAC

### Dev 1: BD - Queries Optimizadas

- **Tarea 1.1:** UserRepository CRUD + queries autenticación
  - Tiempo: 4 horas
- **Tarea 1.2:** Índices de BD + análisis EXPLAIN
  - Tiempo: 3 horas

### Dev 2: Backend - Autenticación Completa

- **Tarea 2.1:** Sistema de sesiones seguras (httpOnly cookies)
  - Tiempo: 4 horas
- **Tarea 2.2:** RBAC middleware (check permissions)
  - Tiempo: 4 horas
- **Tarea 2.3:** Password reset flow
  - Tiempo: 2 horas

### Dev 3: Frontend - Login + Dashboard

- **Tarea 3.1:** Integración Login ↔ Backend (Fetch API)
  - Tiempo: 3 horas
- **Tarea 3.2:** Dashboard instructor (menú dinámico por rol)
  - Tiempo: 4 horas
- **Tarea 3.3:** Estilos responsive (mobile-first)
  - Tiempo: 3 horas

### Dev 4: Controllers + Testing

- **Tarea 4.1:** AuthController (login, logout, register)
  - Tiempo: 4 horas
- **Tarea 4.2:** Pruebas manuales flujo autenticación
  - Tiempo: 3 horas
- **Tarea 4.3:** Error handling y logging

- Tiempo: 3 horas

**Hito de Cierre:** Sistema autenticación 100% funcional con 3 roles

### **SPRINT 3: DÍA 5-6 (Viernes 14 - Sábado 15 Nov)**

**Tema:** Gestión de Fichas y Aprendices

#### **Dev 1: Repositories Fichas/Aprendices**

- **Tarea 1.1:** FichaRepository CRUD (búsqueda, paginación)
  - Tiempo: 4 horas
- **Tarea 1.2:** AprendizRepository CRUD
  - Tiempo: 4 horas
- **Tarea 1.3:** Queries complejas (aprendices por ficha)
  - Tiempo: 2 horas

#### **Dev 2: Services Fichas/Aprendices**

- **Tarea 2.1:** FichaService (validaciones negocio)
  - Tiempo: 4 horas
- **Tarea 2.2:** AprendizService + importación CSV
  - Tiempo: 4 horas
- **Tarea 2.3:** Búsqueda y filtros
  - Tiempo: 2 horas

#### **Dev 3: Interfaces CRUD Fichas/Aprendices**

- **Tarea 3.1:** Tabla de fichas con búsqueda
  - Tiempo: 4 horas
- **Tarea 3.2:** CRUD aprendices (modal crear/editar)
  - Tiempo: 4 horas
- **Tarea 3.3:** Upload CSV aprendices
  - Tiempo: 2 horas

#### **Dev 4: Controllers + Integración**

- **Tarea 4.1:** FichaController (list, show, create, update)
  - Tiempo: 4 horas
- **Tarea 4.2:** AprendizController
  - Tiempo: 4 horas
- **Tarea 4.3:** Validación frontend-backend
  - Tiempo: 2 horas

**Hito de Cierre:** CRUD fichas y aprendices operativo

### **SPRINT 4: DÍA 7-8 (Domingo 16 - Lunes 17 Nov)**

**Tema:** Registro de Asistencia Manual

## **Dev 1: AsistenciaRepository Optimizado**

- **Tarea 1.1:** Queries para listado diario de aprendices
  - Tiempo: 4 horas
- **Tarea 1.2:** Query para marcar asistencia + validaciones
  - Tiempo: 3 horas
- **Tarea 1.3:** Índices para queries rápidas (fecha, aprendiz, ficha)
  - Tiempo: 3 horas

## **Dev 2: AsistenciaService (Líder Fase)**

- **Tarea 2.1:** Crear registro asistencia (validar duplicados)
  - Tiempo: 4 horas
- **Tarea 2.2:** Modificar estado asistencia
  - Tiempo: 3 horas
- **Tarea 2.3:** Lógica de cambios auditados
  - Tiempo: 3 horas

## **Dev 3: Interfaz Registro Manual**

- **Tarea 3.1:** Selector de ficha + carga aprendices dinámicamente
  - Tiempo: 3 horas
- **Tarea 3.2:** Tabla de aprendices con checkboxes (presente/ausente)
  - Tiempo: 4 horas
- **Tarea 3.3:** Guardar asistencia (Fetch POST)
  - Tiempo: 3 horas

## **Dev 4: AsistenciaController**

- **Tarea 4.1:** Endpoints POST/PUT asistencia
  - Tiempo: 4 horas
- **Tarea 4.2:** Manejo de errores y validaciones
  - Tiempo: 3 horas
- **Tarea 4.3:** Testing flujo completo
  - Tiempo: 3 horas

**Hito de Cierre:** Registro manual de asistencia funcional al 100%

## **SPRINT 5: DÍA 9-10 (Martes 18 - Miércoles 19 Nov)**

**Tema:** Modificación de Estados + Reportes Básicos

### **Dev 1: Queries Reportes**

- **Tarea 1.1:** Query asistencia diaria por ficha
  - Tiempo: 3 horas
- **Tarea 1.2:** Query por rango de fechas (semana, mes)
  - Tiempo: 3 horas
- **Tarea 1.3:** Optimización con GROUP BY y JOIN
  - Tiempo: 4 horas

## Dev 2: Servicios Reportes

- **Tarea 2.1:** ReporteService (generar datos)
  - Tiempo: 4 horas
- **Tarea 2.2:** Generación de PDF (TCPDF)
  - Tiempo: 3 horas
- **Tarea 2.3:** Export a Excel (PhpSpreadsheet)
  - Tiempo: 3 horas

## Dev 3: Interfaz Reportes + Cambios Estado

- **Tarea 3.1:** Modal para cambiar estado asistencia
  - Tiempo: 3 horas
- **Tarea 3.2:** Vista de reportes diarios (tabla)
  - Tiempo: 3 horas
- **Tarea 3.3:** Botones generar PDF/Excel
  - Tiempo: 4 horas

## Dev 4: Controllers Reportes

- **Tarea 4.1:** ReporteController (endpoints GET)
  - Tiempo: 3 horas
- **Tarea 4.2:** Descargas PDF/Excel
  - Tiempo: 4 horas
- **Tarea 4.3:** Testing reportes con datos reales
  - Tiempo: 3 horas

**Hito de Cierre:** Reportes básicos generados correctamente

## SPRINT 6: DÍA 11 (Jueves 20 Nov)

**Tema:** QA, Bug Fixes, Documentación

### Todos los Devs: Testing Exhaustivo (9-11 horas)

- **Tarea General 1:** Flujo usuario instructor completo
  - Login → Seleccionar ficha → Registrar asistencia → Ver reportes
  - Tiempo: 5 horas
- **Tarea General 2:** Testing de casos erróneos
  - Intentos duplicados, datos inválidos, etc.
  - Tiempo: 3 horas
- **Tarea General 3:** Bug fixes críticos identificados
  - Tiempo: 2 horas
- **Tarea General 4:** Documentación README básica
  - Instalación, uso
  - Tiempo: 1 hora

**Presentación MVP v1:** Viernes 21 Nov a las 3 PM

## 4. SPRINTS DÍAS 12-17 (Versión Completa)

### SPRINT 7-8: DÍAS 12-14 (Viernes 21 - Domingo 23 Nov)

**Tema:** Módulos de Captura Avanzada

#### Dev 4: Módulo Código de Barras (Líder)

- **Tarea 4.1:** JavaScript para captura barcode
  - Tiempo: 6 horas
- **Tarea 4.2:** Testing con scanners reales
  - Tiempo: 4 horas

#### Dev 4: Módulo Código QR

- **Tarea 4.3:** Integración html5-qrcode library
  - Tiempo: 8 horas

#### Dev 1-2-3: Soporte

- Generación de códigos en BD
- API endpoints soporte
- Interfaces cámara web

### SPRINT 9: DÍAS 15-16 (Lunes 24 - Martes 25 Nov)

**Tema:** WebSocket y Tiempo Real

#### Dev 4: Servidor WebSocket (Líder)

- Ratchet PHP WebSocket
- Broadcasting de eventos
- Actualización automática

### SPRINT 10: DÍA 17 (Miércoles 26 - Jueves 27 Nov)

**Tema:** Ajustes Finales

- Optimización de performance
- Biometría (si tiempo permite)
- Testing de carga
- Documentación final

## 5. ASIGNACIÓN DE TAREAS RESUMIDA

### DESARROLLADOR 1: Especialista Base de Datos

#### Semana 1:

- Creación script SQL MVP
- Poblamiento datos iniciales
- Índices y optimización

#### Semana 2:

- Repositories CRUD
- Queries complejas



- Queries para reportes

**Post-MVP:**

- Indexación para módulos avanzados
- Optimización de performance

**Total Estimado: 65-75 horas**

## **DESARROLLADOR 2: Especialista Backend/Servicios**

**Semana 1:**

- AuthService + SessionManagement
- Estructura MVC + autoloading
- UserRepository

**Semana 2:**

- FichaService + AprendizService
- AsistenciaService (lógica principal)
- Validaciones de negocio

**Post-MVP:**

- ReporteService
- Servicios para módulos avanzados

**Total Estimado: 75-85 horas**

## **DESARROLLADOR 3: Especialista Frontend/Análisis**

**Semana 1:**

- Layout base + CSS
- Página login
- Dashboard template

**Semana 2:**

- CRUD fichas y aprendices
- Interfaz registro manual
- Interfaz cambio de estados
- Reportes visualización

**Post-MVP:**

- Interfaces para módulos avanzados
- Optimización UX

**Total Estimado: 70-80 horas**

## **DESARROLLADOR 4: Especialista Integración**

**Semana 1:**

- Router PHP
- Middleware autenticación
- Configuración servidor

**Semana 2:**

- Controllers CRUD (Auth, Ficha, Aprendiz, Asistencia)
- AsistenciaController (complejidad alta)
- ReporteController

#### Post-MVP:

- Módulo código de barras (Líder)
- Módulo código QR (Líder)
- WebSocket Ratchet (Líder)
- Módulo biometría (si aplica)

**Total Estimado: 80-95 horas**

## 6. CHECKLIST DIARIO POR DEVELOPER

### Inicio de Jornada (9:00 AM)

- ☐ Stand-up 15 min
- ☐ Revisar tareas del día
- ☐ Actualizar status en board
- ☐ Pull últimos cambios Git

### Durante la Jornada

- ☐ Commits frecuentes (máximo 2 horas de trabajo)
- ☐ Mensajes de commit descriptivos
- ☐ Código documentado (PHPDoc, JSDoc)
- ☐ Testing mientras desarrollo

### Fin de Jornada (6:00 PM)

- ☐ Push a rama personal
- ☐ Pull request para review
- ☐ Update de status en Jira/board
- ☐ Documentar bloqueadores

### Antes de Dormir (Asíncrono)

- ☐ Review de 1 PR de otro dev
- ☐ Actualizar documentación
- ☐ Preparar tareas para mañana

## 7. HERRAMIENTAS Y STACK RECOMENDADO

### Control de Versiones

- **GitHub:** Repositorio principal
- **Branching:** Feature branches + main
- **PRs:** Requerido antes de merge

## Comunicación

- **Slack:** Daily stand-ups + mensajes rápidos
- **Reuniones:** Stand-up 9:30 AM (15 min sync)
- **Documentación:** GitHub Wiki o Notion

## Desarrollo

- **Cursor IDE:** Para ambos (frontend + backend con PHP support)
- **GitHub Copilot:** Alternativa o complemento
- **Postman:** Testing de APIs
- **TablePlus:** Visualización y testing BD

## Testing

- **PHPUnit:** Pruebas unitarias (opcional en MVP)
- **Manual Testing:** Flujos críticos
- **Checklist de QA:** Antes de cada release

## 8. CRITERIOS DE ACEPTACIÓN MVP v1

### Login & Autenticación

- ☐ Instructor puede logearse con email/contraseña
- ☐ Sistema de sesiones funciona (no expira accidentalmente)
- ☐ Logout funciona
- ☐ No puede acceder sin autenticarse

### Gestión Fichas

- ☐ Instructor ve sus fichas asignadas
- ☐ Puede seleccionar ficha
- ☐ Carga aprendices de ficha correctamente

### Registro Manual Asistencia

- ☐ Instructor marca presentes/ausentes en tabla
- ☐ Checkbox funciona
- ☐ Botón guardar registra en BD

### Visualización

- ☐ Tabla de asistencias muestra registros del día
- ☐ Datos están correctos en BD

### Performance

- ☐ Cargar tabla 30 aprendices: < 500ms
- ☐ Marcar asistencia: < 200ms
- ☐ Sin errores JavaScript en consola

Seguridad

- [ ] Contraseñas hasheadas (no plain text)
- [ ] CSRF protection en formularios
- [ ] Input validation en backend

9. ENTREGABLES

MVP v1 (21 Noviembre)

- ✔ Código fuente en GitHub (rama main)
- ✔ Script SQL completo
- ✔ README con instalación
- ✔ Demo funcional en vivo (o video)
- ✔ Documento técnico básico

Versión Completa (27 Noviembre)

- ✔ Todos los anteriores +
- ✔ Módulos código de barras y QR
- ✔ WebSocket tiempo real
- ✔ Reportes completos (PDF, Excel)
- ✔ Documentación usuario final
- ✔ Video tutorial (opcional)

10. RIESGOS Y MITIGACIONES

Riesgo	Probabilidad	Impacto	Mitigación
Retraso en BD	Media	Alto	Dev 1 comienza day 1, otros esperan min
Bugs críticos semana 2	Alta	Medio	Testing daily, QA checklist
Mal entendido requisitos	Media	Alto	Validar con stakeholder cada 3 días
Desarrollador enfermo	Baja	Alto	Documentación en tiempo real, pair programming ocasional
Scope creep	Alta	Alto	MVP estricto, features adicionales en roadmap

11. PLAN DE CONTINGENCIA

Si algo falla en MVP (Día 11)

Opción A: Demorar release 2-3 días

- Priorizar: Auth + Asistencia Manual
- Postergar: Reportes, cambio de estados

Opción B: MVP sin reportes (solo registro)

- Funcional al 100%
- Reportes en v1.1 (siguiente semana)

Opción C: MVP sin interfaz web (CLI MVP)

- Sistema funciona pero sin UI

- Frontend agregado rápidamente después

## 12. ROADMAP FUTURO

### Fase 2 (Después de 27 Nov)

#### Semana 4:

- Módulo biometría dactilar
- API Node.js middleware
- Testing exhaustivo

#### Semana 5:

- Dashboard coordinador con analytics
- Exportación masiva
- Sistema de notificaciones

#### Semana 6-7:

- Mobile app (opcional)
- Integración SOFIA Plus SENA
- Backup automático

## 13. REFERENCIAS TÉCNICAS RÁPIDAS

### Crear tabla asistencias (Copilot prompt)

```
"Crea la tabla asistencias con campos id, id_aprendiz, id_ficha, fecha, hora, estado, registrado_por y las validaciones necesarias en MySQL con índices de performance"
```

### Generar Controller CRUD

```
"Genera un Controller PHP para CRUD de asistencias con métodos list, show, create, update, delete que valide entrada y retorne JSON"
```

### Query con Copilot

```
"Escribe una query SQL que retorne todos los aprendices de una ficha con sus asistencias del día actual"
```

## 14. CONCLUSIÓN

Este plan **es ejecutable en 17 días** con:

- ✓ **MVP v1 (21 Nov):** Autenticación + registro manual de asistencia
- ✓ **Versión Completa (27 Nov):** + Códigos, WebSocket, reportes
- ✓ **Uso inteligente de IA:** 30-50% más rápido en boilerplate
- ✓ **Equipo especializado:** Cada dev enfocado en su área
- ✓ **Testing continuo:** Identificar bugs daily, no al final
- ✓ **Documentación viva:** Actualizada con el código

Éxito depende de:

1. Disciplina en sprints diarios
2. No agregar features adicionales (scope creep)
3. Comunicación clara y rápida
4. Debugging ágil

¡A ejecutar! ☐

**Documento Preparado:** 10 Nov 2025

**Versión:** 2.0 - ACELERADO

**Estado:** LISTO PARA EJECUCIÓN

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41]  
[42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56]

✱

1. <https://ieeexplore.ieee.org/document/8859476/>
2. <https://ojs.wiserpub.com/index.php/CCDS/article/view/7583>
3. <https://arxiv.org/ftp/arxiv/papers/1808/1808.07650.pdf>
4. <https://arxiv.org/pdf/1408.6142.pdf>
5. <https://arxiv.org/pdf/2406.11912.pdf>
6. [https://zenodo.org/records/11522834/files/variables\\_delay\\_risk.pdf](https://zenodo.org/records/11522834/files/variables_delay_risk.pdf)
7. <https://arxiv.org/ftp/arxiv/papers/2103/2103.15268.pdf>
8. <http://arxiv.org/pdf/2105.12439.pdf>
9. <https://downloads.hindawi.com/journals/sp/2022/3556809.pdf>
10. <https://arxiv.org/pdf/1307.3356.pdf>
11. [https://en.wikipedia.org/wiki/Agile\\_software\\_development](https://en.wikipedia.org/wiki/Agile_software_development)
12. <https://journalwjarr.com/node/399>
13. <https://www.shortcut.com/guides/minimum-viable-product>
14. <https://zencoder.ai/blog/best-ai-for-php>
15. <https://techtipsolutions.com/what-is-a-sprint-in-agile/>
16. [https://en.wikipedia.org/wiki/Minimum\\_viable\\_product](https://en.wikipedia.org/wiki/Minimum_viable_product)
17. <https://www.jetbrains.com/guide/ai/links/ai-for-php-how-to-make-ai-assistant-generate-test-implementations/>
18. <https://axify.io/blog/agile-software-development>
19. <https://www.productplan.com/glossary/minimum-viable-product/>
20. <https://workik.com/php-code-generator>
21. <https://business.adobe.com/blog/basics/daily-stand-up>
22. <https://ieeexplore.ieee.org/document/11133629/>
23. <https://jurnal.unimed.ac.id/2012/index.php/cess/article/view/51173>
24. <https://www.ijcesen.com/index.php/ijcesen/article/view/1846>
25. <https://ieeexplore.ieee.org/document/10812696/>
26. <https://ieeexplore.ieee.org/document/10741797/>
27. <https://arxiv.org/abs/2409.17434>
28. <https://ieeexplore.ieee.org/document/11202778/>
29. <https://www.ijlrp.com/research-paper.php?id=1673>
30. <https://ieeexplore.ieee.org/document/11050654/>
31. <https://www.mdpi.com/2073-431X/14/3/94>
32. <https://arxiv.org/abs/2507.00788>
33. <https://arxiv.org/pdf/2406.17910.pdf>
34. <https://fepbl.com/index.php/estj/article/view/1080>
35. <https://arxiv.org/pdf/2302.06590.pdf>
36. <https://arxiv.org/pdf/2403.08299.pdf>
37. <http://arxiv.org/pdf/2412.10953.pdf>

38. <https://arxiv.org/pdf/2502.13199.pdf>
39. <https://arxiv.org/pdf/2410.12944.pdf>
40. <http://arxiv.org/pdf/2402.09022.pdf>
41. <https://arxiv.org/pdf/2410.02091.pdf>
42. <https://www.sidetool.co/post/cursor-vs-github-copilot-which-ai-tool-suits-your-workflow/>
43. <https://www.trigyn.com/insights/10-best-practices-rapid-software-delivery>
44. <https://www.atlassian.com/agile/project-management/sprint-planning-tools>
45. <https://www.multidisciplinaryfrontiers.com/search?q=FMR-2025-2-072&search=search>
46. <https://zencoder.ai/blog/cursor-vs-copilot>
47. <https://www.hatica.io/blog/delivering-software-projects-on-time/>
48. <https://www.workstatus.io/project-management/agile-sprint>
49. <https://www.zignuts.com/blog/cursorai-vs-vscode>
50. <https://daedtech.com/deliver-software-projects-time/>
51. <https://ones.com/blog/top-5-tools-for-agile-project-management-sprint-planning-in-2025-a-comprehensive-comparison/>
52. [https://www.reddit.com/r/ChatGPTCoding/comments/1cft751/my\\_experience\\_with\\_github\\_copilot\\_vs\\_cursor/](https://www.reddit.com/r/ChatGPTCoding/comments/1cft751/my_experience_with_github_copilot_vs_cursor/)
53. <https://jitt.polman-babel.ac.id/index.php/jitt/article/view/125>
54. <https://hdl.handle.net/2117/402144>
55. <https://pepadun.fmipa.unila.ac.id/index.php/jurnal/article/view/255>
56. <https://journal.lontaradigitech.com/Digitech/article/view/1201>