

**Centro de Enseñanza Técnica Industrial**

**Desarrollo de Software**

**Desarrollo y gestión de software**

**Jesús Alberto Aréchiga Carrillo**

**22310439 6N**

**Profesor**

**José Francisco Pérez Reyes**

**Mayo 2025**

**Guadalajara, Jalisco**

## Objetivo

Establecer un proceso cíclico de revisión, evaluación y mejora continua del software que asegure su evolución constante.

|  |  |
| --- | --- |
| Fase | Descripción |
| Plan (Planificar) | Recopilar requerimientos, evaluar desempeño actual, definir mejoras |
| Do (Hacer) | Desarrollar y probar los cambios o mejoras |
| Check (Verificar) | Validar resultados mediante pruebas, métricas y feedback |
| Act (Actuar) | Aplicar ajustes, documentar y estandarizar si son efectivos |

|  |  |
| --- | --- |
| Herramientas de Soporte | |
| Área | Herramienta sugerida |
| Gesión del código | Git + GitHub / GitLab |
| Issues y mejoras | Kira / Trello / GitHub Projects |
| Monitoreo de errores | Sentry / Firebase Crashlytics |
| Mérticas de uso | Mixpanel / Google Analytics |
| Control de versiones | SemVer (Versionado Semántico) |
| Integración continua | GitHub Actions / CircleCl |
| Documentación técnica | Notion / Confluence / Docusaurus |

## Componentes clave del procedimiento

### Revición técnica (Mensual)

* Revisión del rendimiento del backend y la app móvil/web.
* Análisis de fallos reportados (logs, errores).
* Evaluación del estado de integraciones con APIs de marcas externas (por ejemplo, cambios en la API de Tuya o Philips Hue).
* Actualización de dependencias y frameworks

### Revisión Funcional y de Negocio (Trimestral)

* Evaluación de la experiencia del usuario (UX/UI).
* Revisión del backlog de funcionalidades solicitadas.
* Análisis de datos de uso para identificar funciones poco usadas o mejoras potenciales.
* Validación de que la app sigue alineada con la estrategia (por ejemplo, integración con nuevas categorías de dispositivos: cámaras, termostatos, etc.).

### Captura de Feedback

* Sistema integrado en la app para recibir sugerencias y reportes.
* Encuestas semestrales a usuarios clave.
* Revisión de calificaciones/comentarios en tiendas (App Store / Play Store).

### Evaluación y Pruebas

* Pruebas automatizadas: unitarias, integración, UI.
* Pruebas manuales: por QA interno o beta testers.
* Despliegue escalonado: blue/green deployment o feature flags.
* KPIs de evaluación: Tiempo promedio entre fallos (MTBF)
* Tasa de errores por versión
* Tiempo medio de recuperación (MTTR)
* Satisfacción del usuario (CSAT)

### Documentación y Retroalimentación

* + Tener un registro asociado (issue, commit, changelog).
  + Estar documentada en el repositorio de conocimiento del proyecto.
  + Ser parte de una nueva versión con número asignado y notas de versión.
  + Retroalimentación analizada y respondida al final de cada ciclo.

### Cronograma Sugerido

|  |  |
| --- | --- |
| Actividad | Frecuencia |
| Revición técnica de código | Mensual |
| Análisis de rendimiento / errores | Semanal |
| Feedback y encuestas | Trimestal / Semestral |
| Evaluación estratégica del roadmap | Trimestral |
| Pruebas automatizadas | En cada commit |
| Despliegues | Semanal / quincenal |

### Resultados Esperados

* La app se mantiene actualizada y libre de errores críticos.
* Mejora continua de la experiencia del usuario.
* Rápida adaptación ante cambios de terceros (APIs).
* Alineación constante con los objetivos del negocio.