

MÉTRICAS DEL PROYECTO

Al medir indicadores específicos, se busca:

Evaluar la eficacia de las funcionalidades críticas (validación en tiempo real, alertas de seguridad, reportes).

Determinar si el sistema cumple con tiempos de respuesta adecuados en horarios de alta afluencia.

Hay que asegurar que la calidad percibida por los usuarios (vigilantes, estudiantes y docentes) sea positiva.

Monitorear el progreso del desarrollo en relación con los objetivos planteados en lalicitación.

1. Visión general

- 1.1. Progreso y esfuerzo: medir el avance en cronograma, productividad del equipo y utilización de recursos humanos.
- 1.2. Calidad: identificar defectos, medir eficiencia en su corrección y satisfacción del usuario.
- 1.3. Aspectos técnicos: complejidad del código, cobertura de pruebas y frecuencia de despliegue.
- 1.4. Rendimiento: tiempo de respuesta en validaciones, rendimiento en horas pico y consumo de recursos.

2. Metodología de medición

Definición de métricas

Desviación del cronograma: % de diferencia entre fechas planificadas y reales de entregables.

Productividad: funcionalidades implementadas por sprint, calculadas en puntos de función (PF).

Tasa de defectos: número de defectos encontrados por módulo en pruebas de integración.

Eficiencia en la eliminación de defectos (EED):

$$EED = \frac{E}{D}$$

donde E = errores antes de entrega, D = defectos tras entrega.

Tiempo de respuesta: tiempo promedio en validar un vehículo mediante reconocimiento de placas.

Rendimiento: cantidad de accesos procesados por minuto.

Satisfacción del usuario: medida en encuestas con estudiantes, vigilantes y administrativos.

Fuentes de datos

- Registros del sistema: logs de accesos vehiculares, base de datos de eventos y alertas.
- Repositorio de código: métricas de complejidad y cobertura de pruebas.
- Encuestas y entrevistas: percepción de satisfacción y usabilidad.
- Actas y cronogramas de proyecto: para medir desviaciones y progreso.

Herramientas

- GitHub/GitLab (seguimiento de commits y versiones).
- PostgreSQL (logs de accesos, auditoría y trazabilidad).
- Google Forms/Encuestas internas (satisfacción de usuarios).
- Jira (asignación de tareas y monitoreo de progresos)

Frecuencia de medición

- Semanal: defectos encontrados, avances de cronograma, tiempo de respuesta.
- Mensual: satisfacción de usuarios, análisis de alertas, rendimiento en horas pico.
- Por sprint: productividad en puntos de función, cobertura de pruebas y estabilidad del sistema.

3. Métricas clave del proyecto

a) Métricas de progreso y esfuerzo

Desviación del cronograma: asegura que los hitos definidos (análisis, diseño, desarrollo, pruebas) se cumplen en los tiempos acordados.

Productividad: calculada como PF/personames. Ejemplo: registro de vehículos (2 PF), validación de accesos (3 PF), alertas de seguridad (2 PF).

Utilización de recursos humanos: mide la carga de trabajo distribuida entre vigilantes (usuarios clave en pruebas), desarrolladores y administradores.

b) Métricas de calidad

Tasa de defectos: número de defectos encontrados en pruebas funcionales del módulo de validación en tiempo real y en la interfaz de reportes.

Eficiencia en eliminación de defectos (EED): meta ≥ 0.9 (90% de errores corregidos antes de la entrega).

Tiempo de corrección de errores: tiempo promedio desde la detección del error hasta su solución registrada en la base de datos de incidencias.

Satisfacción del usuario: resultados de encuestas aplicadas a estudiantes y vigilantes después de pruebas piloto.

c) Métricas técnicas

Cobertura de pruebas: % de código cubierto con pruebas automatizadas (meta: $\geq 80\%$).

Tasa de despliegue: frecuencia con que los módulos se integran en entornos de prueba o producción (meta: al menos 1 despliegue por sprint).

d) Métricas de rendimiento

Tiempo de respuesta: meta de ≤ 2 segundos por validación vehicular en horas pico.

Rendimiento: número de accesos vehiculares procesados por minuto (meta: 20 vehículos/minuto en picos).

Utilización de recursos: monitoreo del % de CPU, memoria y red consumidos (meta: mantener $\leq 70\%$ de uso promedio).

4. Análisis y visualización de datos

Tendencias

El seguimiento semanal permitirá observar si los tiempos de validación mejoran tras optimizaciones, si el número de defectos disminuye en cada sprint, o si la satisfacción del usuario aumenta después de pruebas piloto.

Comparaciones

Con proyectos similares en universidades (ejemplo: Seamless Parking en Unicentro Cúcuta).

Con valores de referencia de la industria en métricas de usabilidad y rendimiento.

Gráficos y tablas

Gráficas de barras para defectos detectados por módulo.

Diagramas de línea para medir el tiempo de respuesta semana a semana.

Tablas comparativas de rendimiento en distintos horarios (mañana, tarde, pico).

Interpretación

Ejemplo: si el tiempo de validación sube de 2 a 4 segundos en horas pico, se analizará la causa (consultas lentas en BD, limitación de hardware o saturación de red).

5. Conclusiones y recomendaciones

Resumen de hallazgos

El control automatizado permitirá reducir la informalidad en el parqueadero.

Las métricas de rendimiento y calidad garantizan seguridad, confiabilidad y satisfacción del usuario.

Los indicadores iniciales permiten ajustar el sistema antes de su despliegue final.

Acciones a tomar

Reforzar las pruebas automatizadas si la cobertura baja del 80%.

Mejorar índices de satisfacción con capacitaciones a vigilantes sobre el uso del sistema.

Optimizar la base de datos si el rendimiento no cumple los estándares esperados.

Lecciones aprendidas

La definición anticipada de métricas fortalece la trazabilidad del proyecto.

La participación de los usuarios en las pruebas es clave para validar la usabilidad.

El monitoreo constante evita que los problemas de rendimiento se conviertan en incidentes críticos.