



ESTUDIANTE: Luis Quishpe

CARRERA: Software

PARALELO: "A"

ASIGNATURA: Gestión de calidad del software **NIVEL:** 7 **FECHA:** 13/10/2025

DOCENTE: Ing. José Caiza, Mg.

TEMA: Talleres realizados en clase

TALLER 1

Taller 1

lunes, 13 de octubre de 2025 21:43

Grupo 1. Describa el peor proyecto de desarrollo de software del que ha tenido conocimiento. ¿Que problemas se presentaron y que consecuencias tuvo?

Toyota

A mediados de la década de 2000', muchos conductores de Toyota informaban que su coche aceleraba sin tocar el pedal. Después de una serie de accidentes que dieron lugar a investigaciones, los investigadores descubrieron que errores de software eran la causa de la aceleración involuntaria.

Luis Quishpe

TALLER 2



Taller 2

Monday, April 18, 2022 12:29 PM

1. El modelo en espiral es iterativo incremental?
2. Dada la siguiente situación, elegir el modelo de ciclo de vida más apropiado y construir el mapa de actividades para el proyecto.

1. El modelo en espiral es iterativo incremental?

Si el modelo espiral avanza en vueltas iterativas y en cada vuelta se construye y valida un incremento proyecto o sistema. Lo que lo caracteriza es que las iteraciones están seguidas por análisis de riesgos por lo que primero se identifican y mitigan riesgos entonces después se desarrolla el incremento de esa vuelta iterativa.

2. Dada la siguiente situación, elegir el modelo de ciclo de vida más apropiado y construir el mapa de actividades para el proyecto.

El ciclo de vida más apropiado para el desarrollo de este proyecto es en "Espiral" ya que es un proyecto novedoso poco tratado con software lo que implica muchos riesgos al momento de realizarlo entonces necesita la identificación de los mismo en cada avance.

Mapa de actividades por iteración

Q1 — Definir objetivos y alternativas

Entradas:

- Necesidades del usuario, backlog priorizado, restricciones

Actividades:

- Aterrizar objetivos y criterios de salida del ciclo
- Analizar alternativas técnicas: arquitecturas, librerías, etc.
- Ajustar alcance del incremento y criterios de aceptación

Responsables:

- Jefe de Proyecto o Product Manager, Analista/Arquitecto Senior, expertos del dominio

Entregables:

- Objetivos del ciclo, alcance del incremento, criterios de aceptación

Q2 — Identificación y mitigación de riesgos

Entradas:

- Matriz de riesgos del proyecto, supuestos y restricciones

Actividades:

- Actualizar registro de riesgos
- Plan de mitigación
- Definir pruebas tempranas para riesgos críticos como: (latencia, precisión, seguridad, interoperabilidad).

Responsables:

- Arquitecto, QA Lead y PM

Entregables:

- Matriz de riesgos actualizada, evidencias de mitigación.

Responsables:

- Jefe de Proyecto o Product Manager, Analista/Arquitecto Senior, expertos del dominio

Entregables:

- Objetivos del ciclo, alcance del incremento, criterios de aceptación

Q2 — Identificación y mitigación de riesgos

Entradas:

- Matriz de riesgos del proyecto, supuestos y restricciones

Actividades:

- Actualizar registro de riesgos
- Plan de mitigación
- Definir pruebas tempranas para riesgos críticos como: (latencia, precisión, seguridad, interoperabilidad).

Responsables:

- Arquitecto, QA Lead y PM

Entregables:

- Matriz de riesgos actualizada, evidencias de mitigación.





TALLER 3

Taller 3

Jueves, 11 de septiembre de 2025 8:21

Cuestionario

1. Describa cómo evaluaría la calidad de una universidad antes de aplicar. Qué factores serían importantes. Cuáles serían críticos?

Para evaluar la calidad de una universidad antes de aplicar, primero defino mis objetivos y los traduzco a preguntas y métricas concretas (enfoque GQM) para comparar opciones con evidencia: qué empleabilidad muestran, qué nivel y trayectoria tienen sus docentes, que tal es la metodología de trabajo, qué recursos académicos y retroalimentaciones ofrece, que tal se desenvuelven los estudiantes egresados de esa universidad en la sociedad o en las distintas áreas o empresas en las que trabajen, que tan acogidos son al estar graduado de tal universidad.

2. Usando el modelo de Garvín. Dé un ejemplo de cada dimensión usando uno o más productos electrónicos con los cuales usted está familiarizado.

- **Rendimiento:** una laptop que funciona de manera óptima y rápida en aplicaciones pesadas.
- **Características:** un smartwatch que tiene muchas funcionalidades como verificar tu salud (ritmo cardíaco, temperatura, nivel de oxígeno en la sangre, etc.) no solamente el de dar la hora.
- **Confiabilidad:** un router que se mantiene funcionando bien todos los días sin caldas ni bajones de señal de internet o wifi.
- **Conformidad:** un teclado que respeta estándares de ergonomía y convenciones de diseño.
- **Durabilidad:** unos auriculares que soportan años de uso sin degradarse.
- **Facilidad de servicio:** un teléfono con repuestos que en caso de dañarse se pueda reparar de manera rápida.
- **Estética:** un monitor con marcos finitos, UI elegante y velocidad de imagen.

3. Adicionar dos preguntas adicionales para dimensión del modelo de Garvín.

Confiabilidad: ¿Cuál es la tasa de fallos por 1.000 horas de uso en condiciones reales?

Facilidad de servicio: ¿Cuál es el tiempo promedio de reparación y la disponibilidad de repuestos en tu país?

4. Los factores de calidad de McCall fueron desarrollados durante los años 70. Cada aspecto de la computación ha cambiado dramáticamente hasta la actualidad. Los factores de calidad de McCall continúan aplicándose al software moderno. Se puede sacar alguna conclusión basada en este hecho?

Si los 11 factores de McCall siguen describiendo objetivos de calidad que importan hoy tanto como antes. Son principios centrados en el usuario y el producto, no en una tecnología puntual, por eso perduran, de hecho estándares posteriores por ejemplo ISO 9126/25010 evolucionan, pero no los niegan.



TALLER 4

Taller 4

miércoles, 24 de septiembre de 2025 10:41

Seleccione 4 sub características de calidad

Desarrolle un set de preguntas para explorar si está presentes cada una de las 4 sub características del estándar de calidad del producto.

Eficiencia de desempeño

- Comportamiento temporal
- Utilización de recursos

Adecuación funcional

- Completitud funcional
- Correctitud funcional

Desarrolle un set de preguntas para explorar si está presentes cada una de las 4 sub características del estándar de calidad del producto

1. COMPORTAMIENTO TEMPORAL (Eficiencia de Desempeño)

Preguntas:

1. ¿Los tiempos de respuesta del sistema cumplen con los requisitos establecidos para cada operación?
2. ¿El procesamiento de transacciones se completa dentro de los límites de tiempo esperados?
3. ¿Existe degradación significativa del rendimiento bajo carga de trabajo normal?
4. ¿Los tiempos de respuesta son consistentes across diferentes horarios y volúmenes de datos?

Métricas clave:

- Tiempo promedio de respuesta por transacción
- Percentil 95 de tiempos de respuesta
- Tiempos máximos aceptables por operación

2 UTILIZACIÓN DE RECURSOS (Eficiencia de Desempeño)

Preguntas:

1. ¿El consumo de memoria RAM se mantiene dentro de los límites esperados durante operación normal?
2. ¿La utilización de CPU es eficiente sin picos innecesarios o periodos de inactividad prolongados?
3. ¿El uso de almacenamiento en disco es optimizado y no crece desproporcionadamente?
4. ¿Existe monitorización del consumo de recursos por componente/función específica?

Métricas clave:

- % de utilización de CPU promedio/máxima
- Consumo de memoria (MB/GB) por usuario/transacción
- Uso de almacenamiento y tasa de crecimiento

3. COMPLETITUD FUNCIONAL (Adecuación Funcional)

Preguntas:

1. ¿El producto implementa todas las funciones especificadas en los requisitos?
2. ¿Existen funcionalidades críticas pendientes de implementar que afecten la operación?
3. ¿La cobertura funcional satisface las necesidades completas del usuario final?
4. ¿Se han identificado gaps funcionales entre lo esperado y lo entregado?

Métricas clave:

- % de requisitos funcionales implementados
- Número de funciones críticas no disponibles
- Cobertura de casos de uso implementados

4. CORRECTITUD FUNCIONAL (Adecuación Funcional)

Preguntas:

1. ¿Las funciones del sistema producen los resultados exactos esperados?
2. ¿Existen errores conocidos que afecten la exactitud de los cálculos o procesamientos?
3. ¿Los datos de salida son consistentes y válidos según las reglas de negocio?
4. ¿Se han validado los algoritmos y lógica de negocio contra casos de prueba exhaustivos?

Métricas clave:

- Tasa de errores funcionales por módulo
- Porcentaje de cálculos correctos



TALLER 5

Taller 5

lunes, 13 de octubre de 2025 21:48

Lectura de artículos

Comparative Study of Software Quality Models (Boukouchi Y., et. al.)

A Comparative Study of Software Quality Models (Suman et al.)

Realizar una comparativa entre las dos comparativas

Interpretar la tabla comparativa de atributos

Aspecto	Comparative Study of Software Quality Models (Boukouchi et al.)	A Comparative Study of Software Quality Models (Suman et al.)	Comparación general
Objetivo principal	Analizar distintos modelos de calidad del software (ISO 9126, McCall, Boehm, Dromey, FURPS, etc.) para identificar sus similitudes, diferencias y aplicabilidad práctica.	Evaluar los principales modelos de calidad de software con un enfoque en sus características, métricas y utilidad en diferentes contextos de desarrollo.	Ambos buscan comparar modelos de calidad, pero Boukouchi lo hace desde una visión técnica y de ingeniería , mientras que Suman enfatiza la aplicación práctica y la facilidad de uso .
Metodología	Presenta una comparación sistemática basada en atributos (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad, etc.).	Utiliza una revisión bibliográfica para analizar cómo cada modelo aborda los distintos factores de calidad y su impacto en el ciclo de vida del software.	Boukouchi emplea un enfoque estructurado y tabular , Suman un enfoque más descriptivo y analítico .
Modelos analizados	McCall, Boehm, Dromey, ISO 9126, FURPS, y algunos modelos extendidos.	McCall, Boehm, Dromey, ISO 9126, ISO 25010 y FURPS.	Suman incluye ISO 25010 , que es la versión más actualizada de ISO 9126.
Enfoque temporal	Compara los modelos clásicos y destaca su evolución.	Compara los modelos clásicos y modernos, mostrando su relevancia actual .	Suman ofrece una visión más contemporánea al incluir estándares actuales.
Conclusiones	Ningún modelo cubre completamente todos los aspectos de la calidad del software; se recomienda combinarlos según el contexto del proyecto.	Los modelos más recientes como ISO 25010 son más completos y adaptables a la actualidad del desarrollo de software.	Ambos coinciden en que no hay un modelo perfecto , pero Suman señala la importancia de la actualización y adaptación .

Interpretación

- En conjunto, la tabla comparativa de atributos muestra la evolución del concepto de calidad del software:
- Los modelos clásicos (McCall, Boehm, Dromey) se centraban más en la calidad interna del producto (código, estructura, corrección).
- Los modelos modernos (ISO 9126, ISO 25010) amplían la visión hacia la calidad externa y en uso, considerando factores como usabilidad, seguridad, compatibilidad y experiencia del usuario.
- Se observa una tendencia hacia modelos más integrales y adaptativos, reflejando las necesidades actuales del desarrollo ágil, la interoperabilidad y la seguridad digital.



TALLER 6

Alineación de párrafo

Taller 6

miércoles, 1 de octubre de 2025 10:30

Seleccione 4 sub características de calidad.

Desarrolle un set de preguntas para explorar si está presentes cada una de las 4 sub características del estándar de calidad del producto.

1. Funcionalidad — Precisión de resultados

- **Propósito:** ¿Qué porcentaje de resultados producidos por el sistema es correcto?
- **Método de aplicación:** Ejecutar un conjunto de casos de aceptación con oráculos de verdad y registrar aciertos/errores.
- **Fórmula:** $P = \left(\frac{\# \text{ resultados correctos}}{\# \text{ resultados evaluados}} \right) \times 100\%$
- **Interpretación del valor:** $0\% \leq P \leq 100\%$. Mientras más alto, mejor.
- **Tipo de escala:** Porcentaje.
- **Tipo de medida:** Razón.
- **Entrada para la medición:** Reporte de pruebas de aceptación, registros de ejecución, oráculo de referencia.
- **Referencia ISO 12207:** Integración, Pruebas, Operación.
- **Audiencia objetivo:** Usuario, Desarrollador, Responsable de mantenimiento, Responsable de ACS.

2. Eficiencia de rendimiento — Tiempo de respuesta P95

- **Propósito:** ¿En cuánto tiempo responde el sistema para una operación crítica en el 95% de los casos?
- **Método de aplicación:** Generar carga representativa, medir tiempos de respuesta y calcular percentil 95.
- **Fórmula:** $TP95_{\text{percentil95}} (\{\text{tiempos_respuesta}\})$
- **Interpretación del valor:** $TP95 > 0$.
Mientras más bajo, mejor.
- **Tipo de escala:** Número.
- **Tipo de medida:** Tiempo.

• **Entrada para la medición:** APM/monitoreo, logs de servidor, reporte de prueba de desempeño.

• **Referencia ISO 12207:** Pruebas, Operación, Mantenimiento.

• **Audiencia objetivo:** Usuario, Desarrollador, Responsable de mantenimiento, Responsable de ACS.

3. Compatibilidad — Tasa de éxito de interoperabilidad (API/Mensajería)

- **Propósito:** ¿Qué porcentaje de transacciones con sistemas externos se ejecuta sin error?
- **Método de aplicación:** Ejecutar escenarios de integración con los sistemas objetivo y contar transacciones exitosas/fallidas.
- **Fórmula:** $I = \left(\frac{\# \text{ transacciones totales}}{\# \text{ transacciones exitosas}} \right) \times 100\%$
- **Interpretación del valor:** $0\% \leq I \leq 100\%$. Mientras más alto, mejor.
- **Tipo de escala:** Porcentaje.
- **Tipo de medida:** Razón.
- **Entrada para la medición:** Reportes de pruebas de integración, trazas de API, colas/mensajería.
- **Referencia ISO 12207:** Integración, Pruebas, Operación.
- **Audiencia objetivo:** Usuario, Desarrollador, Responsable de mantenimiento, Responsable de ACS.

4. Usabilidad — Tasa de éxito de la tarea

- **Propósito:** ¿Qué porcentaje de usuarios completa correctamente una tarea sin ayuda?
- **Método de aplicación:** Pruebas con usuarios representativos, definir criterios de éxito y observar finalización sin asistencia.
- **Fórmula:** $TS = \left(\frac{\# \text{ usuarios que completan la tarea}}{\# \text{ usuarios que intentan la tarea}} \right) \times 100\%$
- **Interpretación del valor:** $0\% \leq TS \leq 100\%$. Mientras más alto, mejor.
- **Tipo de escala:** Porcentaje.
- **Tipo de medida:** Razón.
- **Entrada para la medición:** Reportes de pruebas de usabilidad, grabaciones de sesión, notas de evaluación.
- **Referencia ISO 12207:** Pruebas, Validación con usuarios, Operación.
- **Audiencia objetivo:** Usuario, Desarrollador, Responsable de mantenimiento, Responsable de ACS.



TALLER 7

Taller 7

jueves, 25 de septiembre de 2025 8:15

Seleccionar 6 atributos de calidad e investigar acerca de las métricas relacionadas y dependencias con otros atributos

1) Adecuación funcional

Subcaracterísticas: completitud, exactitud, adecuación.

Métricas:

- Cobertura de requisitos (%) = $\frac{\text{req. implementados}}{\text{req. especificados}} \times 100$.
- Exactitud de resultados (%) = $\frac{\text{salidas correctas}}{\text{total de casos}} \times 100$.
- Éxito de casos de prueba funcionales (%) (pasados/ejecutados).
- Densidad de fallos funcionales (defectos funcionales / KLOC o / función).

Dependencias:

↑ con Usabilidad (operabilidad, reconocibilidad); ↑ con Fiabilidad (menos fallos); trade-off con Seguridad si controles impiden completar tareas.

2) Eficiencia de desempeño

Subcaracterísticas: comportamiento temporal, utilización de recursos, capacidad.

Métricas:

- Latencia p95/p99 (ms) por operación.
- Throughput (transacciones/seg).
- Uso de CPU/RAM/IO promedio y pico por transacción.
- Usuarios concurrentes máximos con p95 < umbral.
- Escalabilidad = $\frac{\text{throughput}(\text{nodos } N)}{\text{throughput}(1)}$.

Dependencias:

Trade-off con Seguridad (cifrado, controles); ↑ con Fiabilidad (menos fallos por sobrecarga); impacta Usabilidad (percepción de velocidad).

3) Usabilidad

Subcaracterísticas: reconocibilidad de adecuación, aprendibilidad, operabilidad, protección de errores, estética UI, accesibilidad.

Métricas:

- SUS (0–100) o UMUX-Lite.
- Tasa de éxito de tareas (%) y tiempo por tarea (s/min).
- Tasa de errores de usuario (%) y pasos innecesarios por tarea.
- Conformidad WCAG (A/AA/AAA) y % de criterios cumplidos.

Dependencias:

↑ con Adecuación funcional (tareas relevantes existen); depende de Eficiencia de desempeño (latencia); trade-off con Seguridad (fricción de autenticación).

4) Fiabilidad

Subcaracterísticas: madurez, disponibilidad, tolerancia a fallos, recuperabilidad.

Métricas:

- Disponibilidad (%) = $\frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} \times 100$.
- MTBF (tiempo medio entre fallos) y MTTR (tiempo medio de recuperación).
- Tasa de fallos por 1k h o por 1k transacciones.
- Éxito de recuperación (%): restauraciones/rollbacks válidos.

Dependencias:

↑ con Mantenibilidad (diagnóstico y cambios rápidos); afectada por Eficiencia de desempeño (sobrecarga eleva fallos); influye en Usabilidad (interrupciones).



5) Seguridad

Subcaracterísticas: confidencialidad, integridad, no repudio, responsabilidad, autenticidad.

Métricas:

- Cobertura de autenticación/autorización (%) (endpoints protegidos/total).
- Cobertura de cifrado (%) en tránsito y en reposo.
- Vulnerabilidades abiertas (por severidad) y tiempo de remediación.
- ▣ Tasa de incidentes de seguridad y % de eventos auditables con identidad.

Dependencias:

Trade-off con Usabilidad (fricción); impacta Eficiencia (coste de cifrado); ↑ con Mantenibilidad (parches rápidos reducen riesgo); relación con Fiabilidad (DoS, integridad).

6) Mantenibilidad

Subcaracterísticas: modularidad, reusabilidad, analizabilidad, modificabilidad, comprobabilidad.

Métricas:

- Complejidad ciclomática (promedio y por módulo).
- Cobertura de pruebas (%) (unitarias/integración).
- Índice de mantenibilidad / code smells por KLOC (estático).
- Lead time de cambio y Change Failure Rate (CFR).
- Acoplamiento/cohesión (fan-in/fan-out, LCOM).

Dependencias:

↑ Fiabilidad (menos defectos) y Seguridad (menos vulnerabilidades); puede mejorar Eficiencia tras optimizaciones; facilita Portabilidad (no incluida) y Compatibilidad.