



## INGENIERÍA DE SOFTWARE 3

Modalidad Virtual

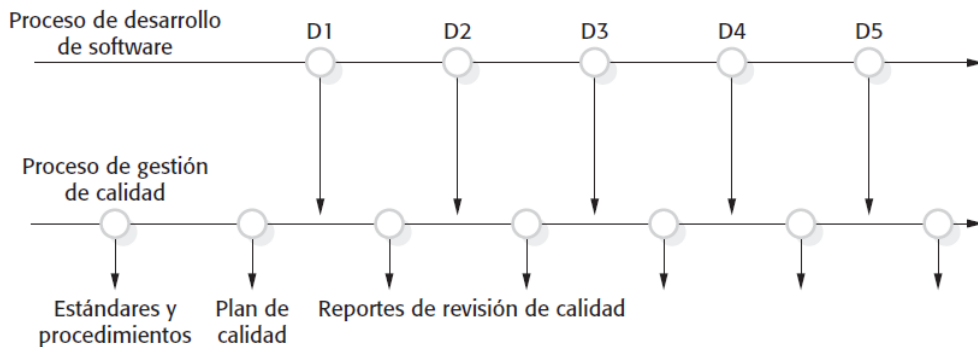
## GESTIÓN DE LA CALIDAD

LOS PROBLEMAS DE CALIDAD DEL SOFTWARE SE DESCUBRIERON INICIALMENTE EN LA DÉCADA DE 1960 CON EL DESARROLLO DE LOS PRIMEROS GRANDES SISTEMAS DE SOFTWARE, Y HAN CONTINUADO INVADIENDO LA INGENIERÍA DE SOFTWARE A PARTIR DE ESA DÉCADA. EL SOFTWARE ENTREGADO ERA LENTO Y POCO FIABLE, DIFÍCIL DE MANTENER Y DE REUTILIZAR. EL DESCONTENTO CON ESTA SITUACIÓN CONDUJO A LA ADOPCIÓN DE TÉCNICAS FORMALES DE GESTIÓN DE CALIDAD DEL SOFTWARE, DESARROLLADAS A PARTIR DE MÉTODOS USADOS EN LA INDUSTRIA MANUFACTURERA. ESTAS TÉCNICAS DE GESTIÓN DE CALIDAD, EN CONJUNTO CON NUEVAS TECNOLOGÍAS Y MEJORES PRUEBAS DE SOFTWARE, LLEVARON A PROGRESOS SIGNIFICATIVOS EN EL NIVEL GENERAL DE CALIDAD DEL SOFTWARE.

EL **ASEGURAMIENTO DE CALIDAD** ES LA DEFINICIÓN DE PROCESOS Y ESTÁNDARES QUE DEBEN CONDUCIR A LA OBTENCIÓN DE PRODUCTOS DE ALTA CALIDAD Y, EN EL PROCESO DE FABRICACIÓN, A LA INTRODUCCIÓN DE PROCESOS DE CALIDAD. INCLUYE TAMBIÉN TODAS LAS ACTIVIDADES DE GESTIÓN DE CONFIGURACIÓN, VERIFICACIÓN Y VALIDACIÓN

# GESTIÓN DE LA CALIDAD

LA GESTIÓN DE CALIDAD PROPORCIONA UNA COMPROBACIÓN INDEPENDIENTE SOBRE EL PROCESO DE DESARROLLO DE SOFTWARE. EL PROCESO DE GESTIÓN DE CALIDAD VERIFICA LOS ENTREGABLES DEL PROYECTO PARA GARANTIZAR QUE SEAN CONSISTENTES CON LOS ESTÁNDARES Y LAS METAS DE LA ORGANIZACIÓN



# PLAN DE CALIDAD

LA PLANEACIÓN DE CALIDAD ES EL PROCESO DE DESARROLLAR UN PLAN DE CALIDAD PARA UN PROYECTO. EL PLAN DE CALIDAD DEBE ESTABLECER LAS CUALIDADES DESEADAS DE SOFTWARE Y DESCRIBIR CÓMO SE VALORARÁN.

UN PLAN DE CALIDAD INCLUYE:

1. INTRODUCCIÓN DEL PRODUCTO UNA DESCRIPCIÓN DEL PRODUCTO, LA PRETENSIÓN DE SU MERCADO Y LAS EXPECTATIVAS DE CALIDAD PARA EL PRODUCTO.
2. PLANES DEL PRODUCTO INDICAN LAS FECHAS DE ENTREGA CRÍTICAS Y LAS RESPONSABILIDADES PARA EL PRODUCTO, JUNTO CON PLANES PARA DISTRIBUCIÓN Y SERVICIO AL PRODUCTO.
3. DESCRIPCIONES DE PROCESOS DESCRIBEN LOS PROCESOS Y ESTÁNDARES DE DESARROLLO Y SERVICIO QUE DEBEN USARSE PARA DISEÑO Y GESTIÓN DEL PRODUCTO.
4. METAS DE CALIDAD LAS METAS Y LOS PLANES DE CALIDAD PARA EL PRODUCTO, INCLUYENDO UNA IDENTIFICACIÓN Y JUSTIFICACIÓN DE LOS ATRIBUTOS ESENCIALES DE CALIDAD DEL PRODUCTO.
5. RIESGOS Y GESTIÓN DEL RIESGO LOS RIESGOS CLAVE QUE PUEDEN AFECTAR LA CALIDAD DEL PRODUCTO Y LAS ACCIONES A TOMAR PARA ENFRENTAR DICHOS RIESGOS.

# PLAN DE CALIDAD

GUÍA PARA ESTABLECER UN PLAN DE CALIDAD (Y MINIMIZAR LAS SUBJETIVIDAD):

1. ¿EN EL PROCESO DE DESARROLLO SE SIGUIERON LOS ESTÁNDARES DE PROGRAMACIÓN Y DOCUMENTACIÓN?
2. ¿EL SOFTWARE SE VERIFICÓ DE MANERA ADECUADA?
3. ¿EL SOFTWARE ES SUFICIENTEMENTE CONFIABLE PARA UTILIZARSE?
4. ¿EL RENDIMIENTO DEL SOFTWARE ES ACEPTABLE PARA USO NORMAL?
5. ¿EL SOFTWARE ES UTILIZABLE?
6. ¿EL SOFTWARE ESTÁ BIEN ESTRUCTURADO Y ES COMPRENSIBLE?

# PROCESO DE SOFTWARE

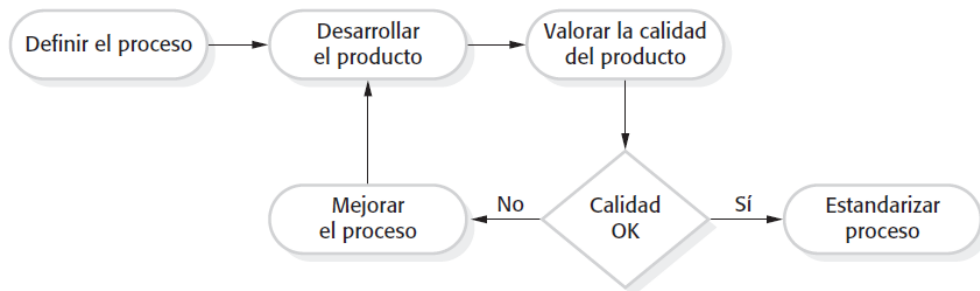
LA CALIDAD DEL SOFTWARE NO SÓLO SE TRATA DE SI LA FUNCIONALIDAD DE ÉSTE SE IMPLEMENTÓ CORRECTAMENTE, SINO TAMBIÉN DEPENDE DE LOS ATRIBUTOS NO FUNCIONALES DEL SISTEMA.

Protección	Comprensibilidad	Portabilidad
Seguridad	Comprobabilidad	Usabilidad
Fiabilidad	Adaptabilidad	Reusabilidad
Flexibilidad	Modularidad	Eficiencia
Robustez	Complejidad	Facilidad para que el usuario aprenda a utilizarlo

NO ES POSIBLE QUE ALGÚN SISTEMA SE OPTIMICE PARA TODOS ESOS ATRIBUTOS; POR EJEMPLO, MEJORAR LA ROBUSTEZ PUEDE CONDUCIR A PÉRDIDA DE RENDIMIENTO. EN CONSECUENCIA, EL PLAN DE CALIDAD DEBE DEFINIR LOS ATRIBUTOS DE CALIDAD MÁS IMPORTANTES PARA EL SOFTWARE QUE SE DESARROLLARÁ.

# PROCESO DE SOFTWARE

UNA SUPOSICIÓN QUE SUBYACE EN LA GESTIÓN DE LA CALIDAD DEL SOFTWARE ES QUE LA CALIDAD DEL SOFTWARE SE RELACIONA DIRECTAMENTE CON LA CALIDAD DEL PROCESO DE DESARROLLO. ESTO PROVIENE DE NUEVO DE LOS SISTEMAS FABRILES.



# ESTÁNDARES DE CALIDAD

EL PROCESO DE DESARROLLO UTILIZADO TIENE UNA INFLUENCIA IMPORTANTE SOBRE LA CALIDAD DEL SOFTWARE. SIN EMBARGO, ES DIFÍCIL VALORAR LOS ATRIBUTOS DE CALIDAD DEL SOFTWARE, COMO LA MANTENIBILIDAD, SIN USAR EL SOFTWARE DURANTE UN LARGO PERIODO.

UN ASPECTO IMPORTANTE DEL ASEGURAMIENTO DE CALIDAD ES LA DEFINICIÓN O SELECCIÓN DE ESTÁNDARES QUE DEBEN APLICARSE AL PROCESO DE DESARROLLO DE SOFTWARE O AL PRODUCTO DE SOFTWARE.

**ESTÁNDARES DEL PRODUCTO** SE APLICAN AL PRODUCTO DE SOFTWARE A DESARROLLAR. INCLUYEN ESTÁNDARES DE DOCUMENTOS (COMO LA ESTRUCTURA DE LOS DOCUMENTOS DE REQUERIMIENTOS), ESTÁNDARES DE DOCUMENTACIÓN (COMO EL ENCABEZADO DE UN COMENTARIO ESTÁNDAR PARA UNA DEFINICIÓN DE CLASE DE OBJETO) Y ESTÁNDARES DE CODIFICACIÓN, LOS CUALES DEFINEN CÓMO DEBE USARSE UN LENGUAJE DE PROGRAMACIÓN.

**ESTÁNDARES DE PROCESO** ESTABLECEN LOS PROCESOS QUE DEBEN SEGUIRSE DURANTE EL DESARROLLO DEL SOFTWARE. DEBEN ESPECIFICAR CÓMO ES UNA BUENA PRÁCTICA DE DESARROLLO. LOS ESTÁNDARES DE PROCESO PUEDEN INCLUIR DEFINICIONES DE ESPECIFICACIÓN, PROCESOS DE DISEÑO Y VALIDACIÓN, HERRAMIENTAS DE SOPORTE DE PROCESO Y UNA DESCRIPCIÓN DE LOS DOCUMENTOS QUE DEBEN ESCRIBIRSE DURANTE DICHOS PROCESOS.

# ESTÁNDARES DE CALIDAD

Estándares de producto	Estándares de proceso
Formato de revisión de diseño	Realizar revisión de diseño
Estructura de documento de requerimientos	Enviar nuevo código para construcción de sistema
Formato de encabezado por método	Proceso de liberación de versión
Estilo de programación Java	Proceso de aprobación del plan del proyecto
Formato de plan de proyecto	Proceso de control de cambio
Formato de solicitud de cambio	Proceso de registro de prueba

## ISO 9001

EXISTE UN CONJUNTO INTERNACIONAL DE ESTÁNDARES QUE PUEDEN UTILIZARSE EN EL DESARROLLO DE LOS SISTEMAS DE ADMINISTRACIÓN DE CALIDAD EN TODAS LAS INDUSTRIAS, LLAMADO ISO 9000. LOS ESTÁNDARES ISO 9000 PUEDEN APLICARSE A VARIAS ORGANIZACIONES, DESDE LAS INDUSTRIAS MANUFACTURERAS HASTA LAS DE SERVICIOS. ISO 9001, EL MÁS GENERAL DE DICHOS ESTÁNDARES, SE APLICA A ORGANIZACIONES QUE DISEÑAN, DESARROLLAN Y MANTIENEN PRODUCTOS, INCLUIDO SOFTWARE.

EL ESTÁNDAR ISO 9001 NO ES EN SÍ MISMO UN ESTÁNDAR PARA EL DESARROLLO DE SOFTWARE, SINO UN MARCO PARA ELABORAR ESTÁNDARES DE SOFTWARE. ESTABLECE PRINCIPIOS DE CALIDAD TOTAL, DESCRIBE EN GENERAL EL PROCESO DE CALIDAD, Y EXPLICA LOS ESTÁNDARES Y PROCEDIMIENTOS ORGANIZACIONALES QUE DEBEN DETERMINARSE. ÉSTOS TIENEN QUE DOCUMENTARSE EN UN MANUAL DE CALIDAD DE LA ORGANIZACIÓN.

EL ESTÁNDAR ISO 9001 NO DEFINE NI PRESCRIBE LOS PROCESOS DE CALIDAD ESPECÍFICOS QUE DEBEN USARSE EN UNA COMPAÑÍA. PARA ESTAR DE CONFORMIDAD CON ISO 9001, UNA COMPAÑÍA DEBE ESPECIFICAR LOS TIPOS DE PROCESO QUE ESTABLECE EL ESTÁNDAR Y TENER PROCEDIMIENTOS QUE DEMUESTREN QUE SE SIGUEN SUS PROCESOS DE CALIDAD.

# ISO 9001

PROCESOS CENTRALES ISO 9001:

## Procesos de entrega de producto

Adquisición  
empresarial

Diseño y  
desarrollo

Prueba

Producción y  
entrega

Servicio y  
soporte

## Procesos de soporte

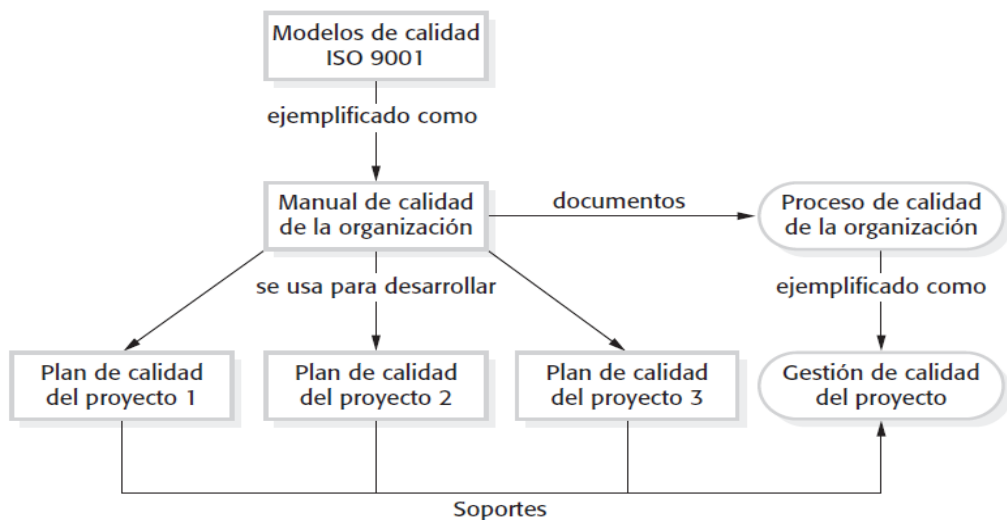
Administración  
empresarial

Administración  
del proveedor

Administración  
del inventario

Administración  
de la configuración

# ISO 9001



# REVISIONES E INSPECCIONES

LAS **REVISIONES E INSPECCIONES** SON ACTIVIDADES QUE COMPRUEBAN LA CALIDAD DE LOS ENTREGABLES DEL PROYECTO. ESTO INCLUYE EXAMINAR EL SOFTWARE, SU DOCUMENTACIÓN Y LOS REGISTROS DEL PROCESO PARA DESCUBRIR ERRORES Y OMISIONES, ASÍ COMO OBSERVAR QUE SE SIGUIERON LOS ESTÁNDARES DE CALIDAD.

REVISIONES E INSPECCIONES SE USAN JUNTO CON LAS PRUEBAS DEL PROGRAMA COMO PARTE DEL PROCESO GENERAL DE VERIFICACIÓN Y VALIDACIÓN DEL SOFTWARE.

DURANTE UNA REVISIÓN, UN GRUPO DE PERSONAS EXAMINAN EL SOFTWARE Y SU DOCUMENTACIÓN ASOCIADA EN BUSCA DE PROBLEMAS POTENCIALES Y LA FALTA DE CONFORMIDAD CON LOS ESTÁNDARES. EL EQUIPO DE REVISIÓN REALIZA JUICIOS INFORMADOS SOBRE EL NIVEL DE CALIDAD DE UN ENTREGABLE DE SISTEMA O DE PROYECTO.

## REVISIÓN DE SOFTWARE

EL PROCESO DE REVISIÓN SE ESTRUCTURA POR LO GENERAL EN TRES FASES:

1. **ACTIVIDADES PREVIAS A LA REVISIÓN:** SE OCUPAN DE LA PLANEACIÓN Y PREPARACIÓN DE LA REVISIÓN. INCLUYE ESTABLECER UN EQUIPO DE REVISIÓN, ORGANIZAR UN TIEMPO, DESTINAR UN LUGAR PARA LA REVISIÓN Y DISTRIBUIR LOS DOCUMENTOS A REVISAR.
2. **LA REUNIÓN DE REVISIÓN:** LA REVISIÓN EN SÍ DEBE SER RELATIVAMENTE CORTA, DOS HORAS A LO SUMO. UN MIEMBRO DEL EQUIPO DEBE DIRIGIR LA REVISIÓN Y OTRO REGISTRAR FORMALMENTE TODAS LAS DECISIONES Y ACCIONES DE REVISIÓN A TOMAR.
3. **ACTIVIDADES POSTERIORES A LA REVISIÓN:** DEBEN TRATARSE LOS CONFLICTOS Y PROBLEMAS SURGIDOS DURANTE LA REVISIÓN. ESTO PUEDE IMPLICAR CORREGIR BUGS DE SOFTWARE, REFACTORIZAR EL SOFTWARE, O REESCRIBIR LOS DOCUMENTOS. DESPUÉS DE EFECTUAR LOS CAMBIOS, LA DIRECCIÓN DE LA REVISIÓN DEBERÁ COMPROBAR QUE SE HAYAN CONSIDERADO TODOS LOS COMENTARIOS DE LA REVISIÓN. EN OCASIONES SE REQUERIRÁ UNA REVISIÓN ULTERIOR PARA COMPROBAR QUE LOS CAMBIOS REALIZADOS COMPRENDEN TODOS LOS COMENTARIOS DE REVISIÓN ANTERIORES.

# INSPECCIONES DE SOFTWARE

LAS INSPECCIONES DEL PROGRAMA SON “REVISIONES DE PARES” EN LAS QUE LOS MIEMBROS DEL EQUIPO COLABORAN PARA ENCONTRAR BUGS EN EL PROGRAMA EN DESARROLLO.

LAS INSPECCIONES DEL PROGRAMA INCLUYEN A MIEMBROS DEL EQUIPO CON DIFERENTES ANTECEDENTES QUE REALIZAN UNA CUIDADOSA REVISIÓN, LÍNEA POR LÍNEA, DEL CÓDIGO FUENTE DEL PROGRAMA. BUSCAN DEFECTOS Y PROBLEMAS, Y LOS INFORMAN EN UNA REUNIÓN DE INSPECCIÓN. LOS DEFECTOS PUEDEN SER ERRORES LÓGICOS, ANOMALÍAS EN EL CÓDIGO QUE INDICAN UNA CONDICIÓN ERRÓNEA O CIERTAS CARACTERÍSTICAS QUE SE HAYAN OMITIDO DEL CÓDIGO.

DURANTE UNA INSPECCIÓN, CON FRECUENCIA SE USA UNA LISTA DE VERIFICACIÓN DE ERRORES COMUNES DE PROGRAMACIÓN PARA ENFOCAR LA BÚSQUEDA DE BUGS. ESTA LISTA DE VERIFICACIÓN SE BASA EN EJEMPLOS DE LIBROS, O BIEN, EN EL CONOCIMIENTO DE DEFECTOS NORMALES EN UN DOMINIO DE APLICACIÓN COMÚN. PARA DIFERENTES LENGUAJES DE PROGRAMACIÓN SE USAN DISTINTAS LISTAS DE VERIFICACIÓN, PUESTO QUE CADA LENGUAJE TIENE SUS ERRORES CARACTERÍSTICOS.

# INSPECCIONES DE SOFTWARE

Clase de falla	Comprobación de inspección
Fallas de datos	<ul style="list-style-type: none"> <li>• ¿Todas las variables del programa se inician antes de usar sus valores?</li> <li>• ¿Todas las constantes tienen nombre?</li> <li>• ¿La cota superior de los arreglos es igual al tamaño del arreglo o Valor – 1?</li> <li>• Si se usan cadenas de caracteres, ¿se asigna explícitamente un delimitador?</li> <li>• ¿Existe alguna posibilidad de desbordamiento de buffer?</li> </ul>
Fallas de control	<ul style="list-style-type: none"> <li>• Para cada enunciado condicional, ¿la condición es correcta?</li> <li>• ¿Hay certeza de que termine cada ciclo?</li> <li>• ¿Los enunciados compuestos están correctamente colocados entre paréntesis?</li> <li>• En caso de enunciados, ¿se justifican todos los casos posibles?</li> <li>• Si después de cada caso en los enunciados se requiere un paréntesis, ¿éste se incluyó?</li> </ul>
Fallas de entrada/salida	<ul style="list-style-type: none"> <li>• ¿Se usan todas las variables de entrada?</li> <li>• ¿A todas las variables de salida se les asigna un valor antes de que se produzcan?</li> <li>• ¿Entradas inesperadas pueden causar corrupción?</li> </ul>
Fallas de interfaz	<ul style="list-style-type: none"> <li>• ¿Todas las llamadas a función y método tienen el número correcto de parámetros?</li> <li>• ¿Los tipos de parámetro formal y real coinciden?</li> <li>• ¿Los parámetros están en el orden correcto?</li> <li>• Si los componentes acceden a memoria compartida, ¿tienen el mismo modelo de estructura de memoria compartida?</li> </ul>
Fallas de gestión de almacenamiento	<ul style="list-style-type: none"> <li>• Si se modifica una estructura vinculada, ¿todos los vínculos se reasignan correctamente?</li> <li>• Si se usa almacenamiento dinámico, ¿el espacio se asignó correctamente?</li> <li>• ¿El espacio se cancela explícitamente después de que ya no se requiere?</li> </ul>
Fallas de gestión de excepción	<ul style="list-style-type: none"> <li>• ¿Se tomaron en cuenta todas las posibles condiciones de error?</li> </ul>



# MEDICIONES Y MÉTRICAS

LA **MEDICIÓN DEL SOFTWARE** SE OCUPA DE DERIVAR UN VALOR NUMÉRICO O PERFIL PARA UN ATRIBUTO DE UN COMPONENTE, SISTEMA O PROCESO DE SOFTWARE. AL COMPARAR DICHOS VALORES UNOS CON OTROS, Y CON LOS ESTÁNDARES QUE SE APLICAN A TRAVÉS DE UNA ORGANIZACIÓN, ES POSIBLE EXTRAER CONCLUSIONES SOBRE LA CALIDAD DEL SOFTWARE, O VALORAR LA EFECTIVIDAD DE LOS PROCESOS, LAS HERRAMIENTAS Y LOS MÉTODOS DE SOFTWARE.

UNA **MÉTRICA DE SOFTWARE** ES UNA CARACTERÍSTICA DE UN SISTEMA DE SOFTWARE, DOCUMENTACIÓN DE SISTEMA O PROCESO DE DESARROLLO QUE PUEDE MEDIRSE DE MANERA OBJETIVA. LOS EJEMPLOS DE MÉTRICAS INCLUYEN EL TAMAÑO DE UN PRODUCTO EN LÍNEAS DE CÓDIGO; EL NÚMERO DE FALLAS REPORTADAS EN UN PRODUCTO DE SOFTWARE ENTREGADO, Y EL NÚMERO DE DÍAS-HOMBRE REQUERIDO PARA DESARROLLAR UN COMPONENTE DE SISTEMA.

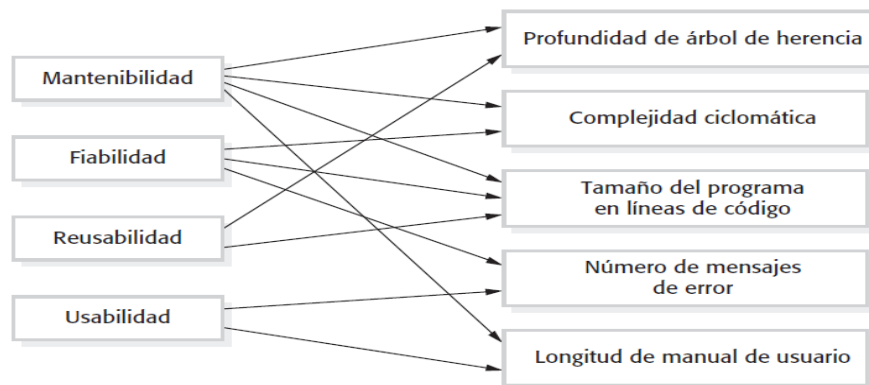
LAS MÉTRICAS DE SOFTWARE PUEDEN SER MÉTRICAS DE CONTROL O DE PREDICCIÓN. LAS MÉTRICAS DE CONTROL APOYAN LA GESTIÓN DEL PROCESO, Y LAS MÉTRICAS DE PREDICCIÓN AYUDAN A PREDECIR LAS CARACTERÍSTICAS DEL SOFTWARE. LAS MÉTRICAS DE CONTROL SE ASOCIAN POR LO GENERAL CON PROCESOS DE SOFTWARE. LAS MÉTRICAS DE PREDICCIÓN SE ASOCIAN CON EL SOFTWARE EN SÍ Y A VECES SE CONOCEN COMO MÉTRICAS DE PRODUCTO.

# MEDICIONES Y MÉTRICAS

ES DIFÍCIL HACER MEDICIONES DIRECTAS DE MUCHOS DE LOS ATRIBUTOS DE CALIDAD DEL SOFTWARE. PARA HACER UN JUICIO SOBRE ESTOS ATRIBUTOS, HAY QUE MEDIR ALGUNOS ATRIBUTOS INTERNOS DEL SOFTWARE (COMO TAMAÑO, COMPLEJIDAD, ETCÉTERA) Y SUPONER QUE ÉSTOS SE RELACIONAN CON LAS CARACTERÍSTICAS DE CALIDAD POR LAS QUE UNO SE INTERESA.

**Atributos de calidad externos**

**Atributos internos**



# MODELO CMM

EL **MODELO CMM** (MODELO DE MADUREZ DE CAPACIDADES), TAMBIÉN DENOMINADO CMM-SW, FUE DESARROLLADO POR EL SEI (INSTITUTO DE INGENIERÍA DE SOFTWARE) COMO MARCO DE REFERENCIA PARA LA EVALUACIÓN Y MEJORA DE PROCESOS SOFTWARE.

EL MODELO CMM ORGANIZA LA MADUREZ DE UN PROCESO SOFTWARE EN 5 NIVELES EN CADA NIVEL, ADEMÁS DE ESTABLECER UNA ESCALA DE MEDIDA DE LA CAPACIDAD DE LOS PROCESOS, SE FIJAN UNOS OBJETIVOS QUE AYUDAN A LA ORGANIZACIÓN A PRIORIZAR LOS ESFUERZOS DEDICADOS A LA MEJORA DE ESTOS PROCESOS.

EN EL **NIVEL 1 O INICIAL**, NO EXISTEN ÁREAS DE PROCESO, LOS PROCESOS NO ESTÁN DEFINIDOS AUNQUE ALGUNOS DE ELLOS SE REALICEN DE FORMA CONTROLADA. EL TRABAJO SE REALIZA DE MANERA IMPREDECIBLE. EL ÉXITO DEPENDE DE LAS CAPACIDADES Y ESFUERZOS INDIVIDUALES REALIZADOS POR LAS PERSONAS.

# MODELO CMM

EN EL **NIVEL 2 O REPETIBLE**, SE ENCUENTRAN ESTABLECIDAS LAS ACTIVIDADES BÁSICAS DE GESTIÓN DE PROYECTOS, DE FORMA QUE PUEDEN OBSERVARSE Y REPRODUCIRSE LAS PRÁCTICAS QUE HAN TENIDO ÉXITO EN PROYECTOS ANTERIORES.

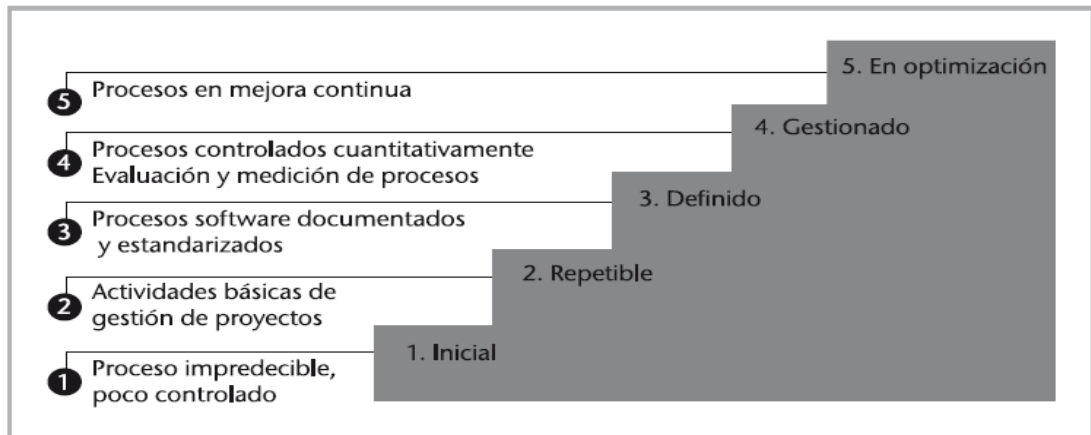
EN EL **NIVEL 3 O DEFINIDO**, SE DOCUMENTAN Y ESTANDARIZAN TANTO LOS PROCESOS DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE, COMO LOS DE GESTIÓN. TODOS LOS PROYECTOS USAN UNA VERSIÓN DE LOS PROCESOS INTEGRADA EN LA ORGANIZACIÓN.

EN EL **NIVEL 4 O GESTIONADO**, SE ESTABLECE UN PROGRAMA AMPLIO Y DETALLADO DE MEDIDAS, TANTO PARA EL PROCESO COMO PARA EL PRODUCTO SOFTWARE. SE RECOGEN Y ANALIZAN LOS DATOS DE TODOS LOS PROYECTOS, FORMANDO UNA BASE DE DATOS CUANTITATIVA QUE SERÁ DE GRAN AYUDA EN LA EVALUACIÓN Y ESTIMACIÓN DE PROYECTOS.

EN EL **NIVEL 5 O EN OPTIMIZACIÓN**, LA ORGANIZACIÓN ESTÁ INMERSA EN UN PROCESO DE MEJORA CONTINUA DE TODOS SUS PROCESOS, RECOGIENDO DATOS DE TODOS SUS PROYECTOS Y UTILIZÁNDOLOS PARA APORTAR NUEVAS IDEAS Y PARA PROPORCIONAR INNOVACIONES QUE UTILIZAN NUEVOS MÉTODOS Y TECNOLOGÍAS.

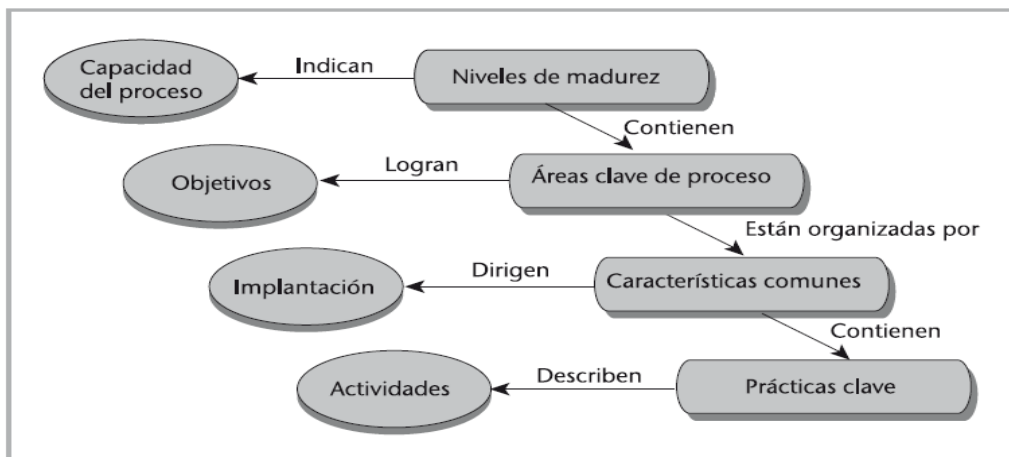
# MODELO CMM

ESCALA DE MADUREZ DEL MODELO CMM:



# MODELO CMM

ARQUITECTURA DEL MODELO CMM:



# MODELO CMM

CADA NIVEL DE MADUREZ ESTÁ ORGANIZADO EN ÁREAS CLAVE DE PROCESO KPA (KEY PROCESS AREA).

Nivel	Áreas clave de proceso
1. Inicial	No hay áreas clave de proceso establecidas.
2. Repetible	Gestión de la configuración del software. Aseguramiento de la calidad del software. Gestión de acuerdos y contratos con proveedores. Planificación, seguimiento y control de proyectos. Gestión de requisitos.
3. Definido	Revisión detallada de los procesos. Coordinación dentro del grupo de trabajo. Ingeniería del producto software. Gestión integrada del proyecto. Programa de formación. Definición del proceso organizativo. Enfoque hacia los procesos organizativos.
4. Gestionado	Gestión de la calidad de software. Gestión cuantitativa de procesos.
5. En optimización	Gestión del cambio en los procesos. Gestión de los cambios tecnológicos. Prevención de defectos.

# MODELO CMM

LAS CINCO CARACTERÍSTICAS COMUNES SON:

- **COMPROMISO.** ES EL CONJUNTO DE ACCIONES QUE LA ORGANIZACIÓN DEBE REALIZAR PARA PODER ASEGURAR QUE EL PROCESO ES REPETIBLE Y DURADERO. NORMALMENTE ESTÁ RELACIONADO CON LAS POLÍTICAS DE LA ORGANIZACIÓN Y EL LIDERAZGO DE LA DIRECCIÓN.
- **CAPACIDAD.** DESCRIBE LAS PRECONDICIONES QUE DEBEN DARSE EN UN PROYECTO O EN LA ORGANIZACIÓN PARA IMPLANTAR DE FORMA EFECTIVA LOS PROCESOS DE SOFTWARE. HABITUALMENTE AFECTA A LOS RECURSOS, A LA ESTRUCTURA Y A LA FORMACIÓN.
- **ACTIVIDADES.** DESCRIBEN LOS ROLES Y LOS PROCEDIMIENTOS NECESARIOS PARA IMPLANTAR UNA KPA. HABITUALMENTE INCLUYEN PROCEDIMIENTOS RELACIONADOS CON LA PLANIFICACIÓN Y EL SEGUIMIENTO DEL TRABAJO, ASÍ COMO LAS ACCIONES CORRECTIVAS NECESARIAS.
- **MEDIDAS Y ANÁLISIS.** DESCRIBEN LA NECESIDAD DE REALIZAR MEDICIONES DE LOS PROCESOS Y ANALIZAN DICHAS MEDIDAS. SUELEN INCLUIR EJEMPLOS DE LAS MEDIDAS TOMADAS PARA DETERMINAR EL ESTADO Y LA EFICACIA DE LAS ACTIVIDADES REALIZADAS.
- **VERIFICACIÓN.** DESCRIBE LOS PASOS QUE DEBEN LLEVARSE A CABO PARA ASEGURAR QUE LAS ACTIVIDADES SE REALIZAN SEGÚN LOS PROCESOS ESTABLECIDOS. HABITUALMENTE INCLUYE REVISIONES Y AUDITORÍAS POR PARTE DE LA DIRECCIÓN Y DEL GRUPO DE ASEGURAMIENTO DE LA CALIDAD.

# MODELO CMMI

LAS CARACTERÍSTICAS COMUNES CONTIENEN **PRÁCTICAS CLAVE** KP (KEY PRACTICES). LAS PRÁCTICAS CLAVE DESCRIBEN LA INFRAESTRUCTURA Y LAS ACTIVIDADES QUE CONTRIBUYEN PRINCIPALMENTE A LA IMPLANTACIÓN E INSTITUCIONALIZACIÓN EFECTIVA DE LAS KPA.

EL MODELO CMM SE FUE ADAPTANDO A MÚLTIPLES DISCIPLINAS: INGENIERÍA DE SISTEMAS, INGENIERÍA DEL SOFTWARE, COMPRAS, DESARROLLO DE PROCESOS Y PRODUCTOS INTEGRADOS, ETC., DERIVANDO MODELOS DIFERENTES.

EL **CMMI** SURTIÓ COMO SOLUCIÓN A TODOS ESTOS PROBLEMAS DE FALTA DE INTEGRACIÓN Y USO DE MÚLTIPLES MODELOS CMM.

SE PUEDEN ENCONTRAR LAS SIGUIENTES ÁREAS DE PROCESO EN EL MODELO CMMI QUE NO SE ENCUENTRAN EN EL MODELO CMM:

**NIVEL 2.** MEDICIÓN Y ANÁLISIS. HAN SIDO AISLADAS DE CMM TODAS LAS PRÁCTICAS RELACIONADAS CON ESTE OBJETIVO Y HAN SIDO AGRUPADAS DENTRO DE ESTA NUEVA ÁREA DE PROCESO.

# MODELO CMMI

**NIVEL 3.** EL ÁREA INGENIERÍA DEL PRODUCTO SOFTWARE DE CMM HA SIDO REEMPLAZADA EN CMMI POR MÚLTIPLES Y MÁS DETALLADAS ÁREAS DE PROCESO: DESARROLLO DE REQUISITOS, SOLUCIONES TÉCNICAS, INTEGRACIÓN DEL PRODUCTO, VERIFICACIÓN Y VALIDACIÓN. EN EL ÁREA DE GESTIÓN INTEGRADA DEL PROYECTO DE CMM SE CONTEMPLABA LA GESTIÓN DE RIESGOS, PERO AHORA HA SIDO CONSIDERADA COMO UN ÁREA DE PROCESO INDEPENDIENTE. FINALMENTE, A ESTE NIVEL SE HA AÑADIDO UNA NUEVA ÁREA DENOMINADA ANÁLISIS DE DECISIONES Y RESOLUCIÓN, QUE NO SE ENCONTRABA EN CMM.

**NIVEL 4.** ESTE NIVEL HA SUFRIDO UNA REESTRUCTURACIÓN Y LAS ÁREAS DE GESTIÓN CUANTITATIVA DE PROCESOS Y GESTIÓN DE LA CALIDAD DE SOFTWARE HAN SIDO CONVERTIDAS A GESTIÓN CUANTITATIVA DEL PROYECTO Y RENDIMIENTO O REALIZACIÓN DEL PROCESO ORGANIZACIONAL, RESPECTIVAMENTE.

**NIVEL 5.** TAMPOCO HA HABIDO GRANDES CAMBIOS EN ESTE NIVEL, SIMPLEMENTE UNA FUSIÓN DE LAS ÁREAS GESTIÓN DE LOS CAMBIOS TECNOLÓGICOS Y GESTIÓN DEL CAMBIO EN LOS PROCESOS EN UNA ÚNICA ÁREA DE PROCESO: INNOVACIÓN ORGANIZACIONAL Y DESPLIEGUE. EL ÁREA DE PREVENCIÓN DE DEFECTOS HA SIDO REESTRUCTURADA Y RENOMBRADA A ANÁLISIS CAUSAL Y RESOLUCIÓN.

# TRABAJO PRÁCTICO

## TRABAJO PRÁCTICO:

IMAGINE SER EL GERENTE GENERAL /DUEÑO DE UN ORGANIZACIÓN DE DESARROLLO DE SOFTWARE QUE SE ENCUENTRA EN EL NIVEL 3 DE MADUREZ DEL MODELO CMM:

- 1) DESCRIBA LOS ESTÁNDARES DE PROCESOS DE DESARROLLO QUE POSEE SU ORGANIZACIÓN.
- 2) DESCRIBA LOS ESTÁNDARES DE PROCESOS DE GESTIÓN QUE POSEE SU ORGANIZACIÓN.
- 3) DESCRIBA UN PLAN DE ACCIÓN PARA ALCANZAR EL NIVEL 4 DE MADUREZ.
- 4) ALCANZADO EL NIVEL 4, DESCRIBA LOS PROCEDIMIENTOS DE MEDICIÓN INCORPORADOS Y ESTÁNDARES.
- 5) ¿QUÉ PLAN DE ACCIÓN SEGUIRÍA PARA ALCANZAR EL NIVEL 5 DE MADUREZ?