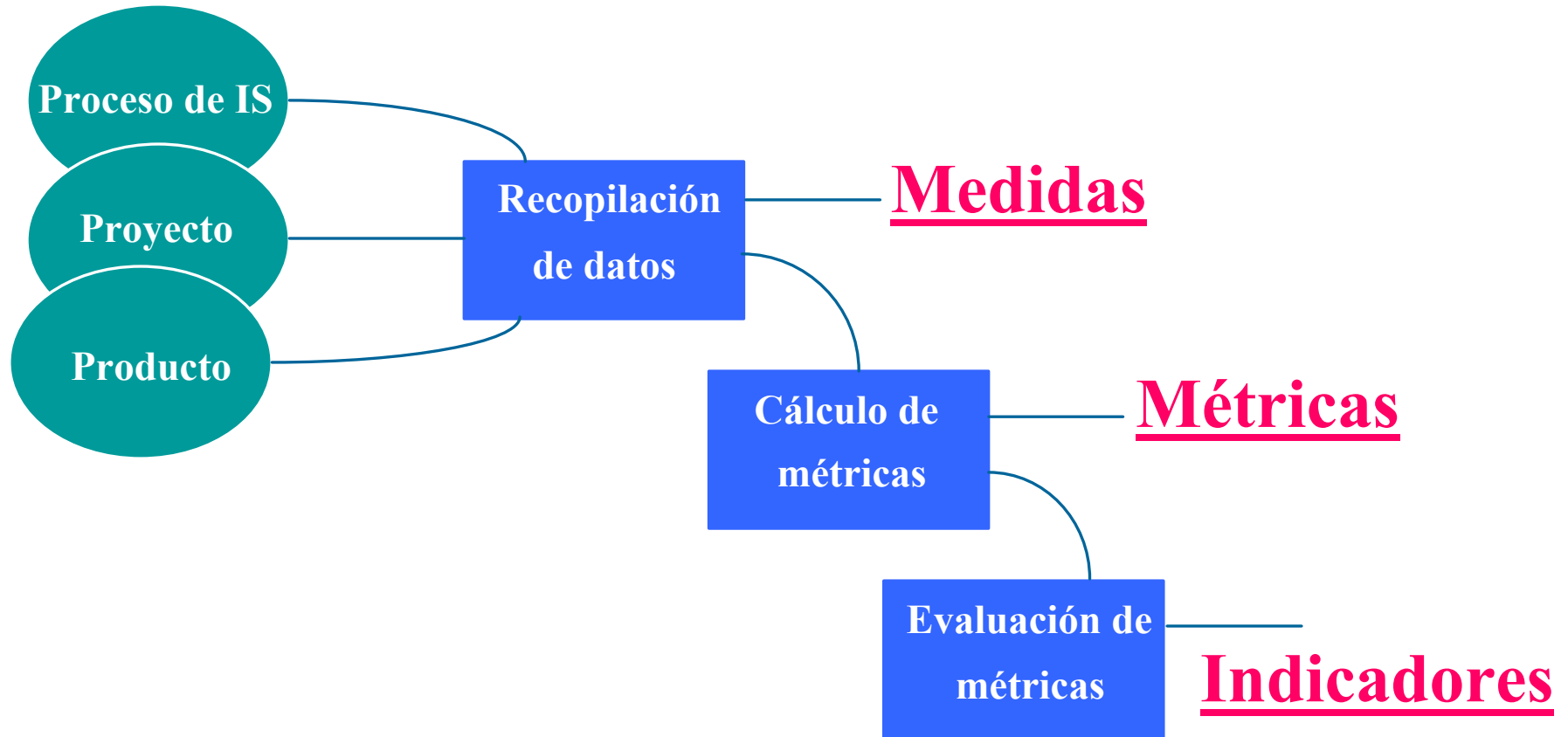




E77 - Gestión de Recursos de la Información

***Tema 1 - Métricas
del Proyecto de Software***

Medición y Métricas



Para qué medir el software?

- **Para indicar la calidad del producto.**
- **Para evaluar la productividad de las personas.**
- **Para evaluar los beneficios derivados del uso de nuevos métodos y herramientas.**
- **Para establecer una línea de base para la estimación.**
- **Para justificar el uso de nuevas herramientas y la necesidad de formación.**

Indicadores del Proyecto

- **Evaluar el estado del proyecto.**
- **Hacer un seguimiento de los riesgos.**
- **Detectar las áreas problemáticas.**
- **Ajustar el flujo y las tareas del trabajo para evitar retrasos.**
- **Evaluar la habilidad del personal.**
- **Evaluar la calidad del producto.**

Métricas del software

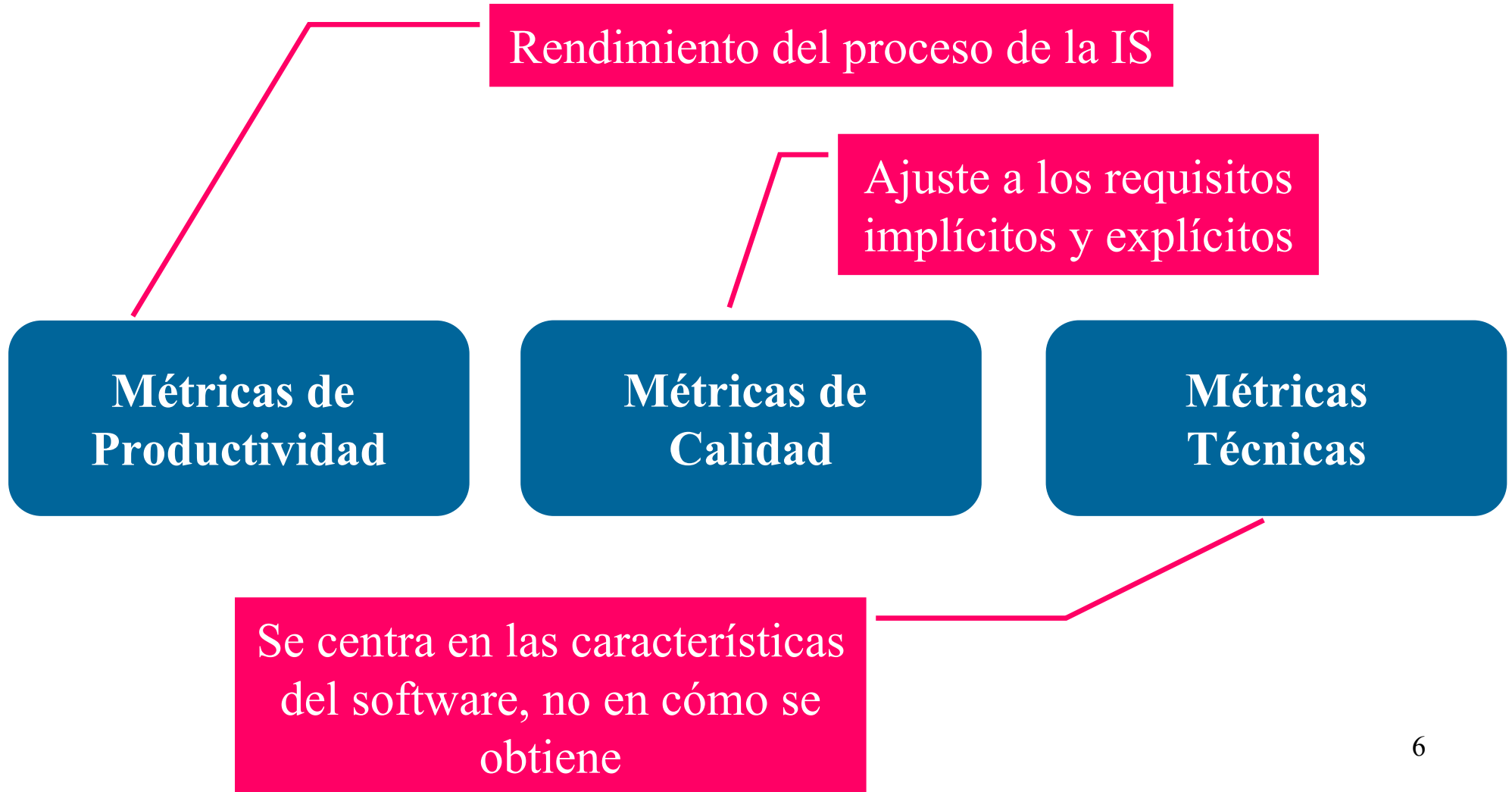
Medidas Directas

- Coste.
- Esfuerzo humano.
- Líneas de código.
- Velocidad de ejecución.
- Tamaño de memoria.
- Número de defectos.
- Etc.

Medidas Indirectas

- Funcionalidad.
- Calidad.
- Complejidad.
- Eficiencia.
- Fiabilidad.
- Facilidad de uso.
- Etc.

Métricas del software



Métricas del software

**Métricas orientadas
al Tamaño**

**Medidas directas del resultado
y del proceso.**

**Métricas orientadas
a la Función**

**Medidas indirectas del
software y del proceso.**

**Medida de la efectividad
de los métodos y las
herramientas.**

**Métricas orientadas
a la Persona**

Características de las métricas del software

- **Simple y fácil de calcular.**
- **Empírica e intuitiva.**
- **Sin ambigüedades y objetiva.**
- **Consistente en el empleo de unidades y tamaños.**
- **Independiente del lenguaje de programación.**
- **Eficaz para aumentar la calidad del software.**

Métricas orientadas al tamaño



$$\text{Productividad} = \text{KLDC} / \text{personas-mes}$$

$$\text{Calidad} = \text{Nº errores (defectos)} / \text{KLDC}$$

$$\text{Coste medio} = \text{Pts} / \text{KLDC}$$

$$\text{Documentación} = \text{Páginas de documentación} / \text{KLDC}$$

Métricas orientadas a la función

Característica	Cantidad	Simple	Medio	Complejo	Cuenta
Entradas de usuario		3	4	6	
Salidas de usuario		4	5	7	
Peticiones de usuario		3	4	6	
Archivos		7	10	15	
Interfaces externas		5	7	10	
CUENTA TOTAL					

Métricas orientadas a la función

$$PF = \text{cuenta}_{\text{total}} \times [0'65 + 0'01 \times \text{SUM}(F_i)]$$

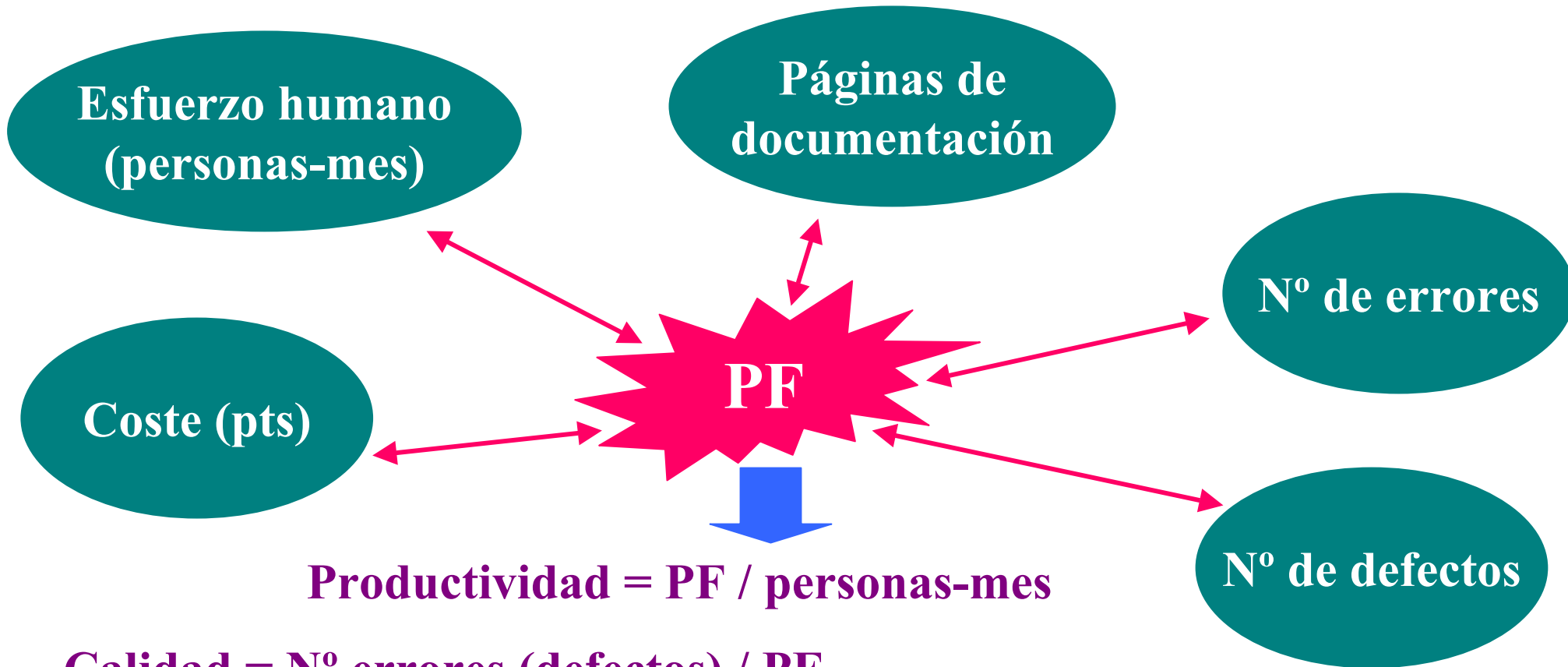
Punto de
Función

Suma de las entradas
de la tabla anterior

En función de las respuestas
a un cuestionario (0, ..., 5)

Valores de
ajuste de
complejidad

Métricas orientadas a la función



$$\text{Productividad} = \text{PF} / \text{personas-mes}$$

$$\text{Calidad} = \text{Nº errores (defectos)} / \text{PF}$$

$$\text{Coste medio} = \text{Pts} / \text{PF}$$

$$\text{Documentación} = \text{Páginas de documentación} / \text{PF}$$

Puntos de característica

Parámetro	Cantidad	Peso	Cuenta
Entradas de usuario		4	
Salidas de usuario		5	
Peticiones de usuario		4	
Archivos		7	
Interfaces externas		7	
Algoritmos		3	
CUENTA TOTAL			

Calidad del software

Concordancia con:

- los requisitos explícitos: funcionales y de rendimiento.
- los estándares de desarrollo.
- las características implícitas que debe reunir cualquier software.

Cuándo y cómo medir la calidad del software?

- Complejidad del programa
- Modularidad efectiva
- Tamaño global
- Etc.

Base cuantitativa para tomar decisiones en diseño y prueba

Entrega del Software

- No. de defectos
- Facilidad de mantenimiento
- Facilidad de uso
- Etc.

Concordancia con los requisitos explícitos y técnicos

Factores que influyen en la calidad del software

- **Operación del producto: su uso.**
- **Revisión del producto: su modificación.**
- **Transición del producto: su portabilidad.**

(McCall y Cavano, 1978)

Medidas de calidad

- **Corrección:** adecuación del software a la función requerida.

nº de defectos por KLDC

- **Facilidad de mantenimiento:** facilidad para corregir un error, adaptar un programa a cambios en los requisitos, y mejorarlo.

TMEC (tiempo medio entre cambios)

Medidas de calidad

- **Integridad:** capacidad para resistir ataques, provocados o no, contra su seguridad.
 - **Amenaza:** probabilidad de que un cierto tipo de ataque ocurra en un tiempo.
 - **Seguridad:** probabilidad de que se pueda contrarrestar un cierto tipo de ataque.

$$\text{Integridad} = \sum[1 - \text{amenaza} \times (1 - \text{seguridad})]$$

Medidas de calidad

- **Facilidad de uso: “amistad con el usuario”.**
 - **Habilidad intelectual y/o física requerida para aprender a utilizar el sistema.**
 - **Tiempo necesario para llegar a dominar su uso.**
 - **Aumento neto en productividad.**
 - **Valoración subjetiva de la predisposición de los usuarios hacia el sistema.**

Medidas de calidad

- **Eficiencia:** recursos y código necesarios para que un programa realice su función.
- **Reusabilidad:** facilidad para volver a utilizar partes de un programa en otras aplicaciones.

Modularidad, independencia del hardware y del sistema, generalidad, ...

- **Interoperatividad:** esfuerzo necesario para acoplar un sistema con otros.

Métricas de fiabilidad

- **Probabilidad de fallo en demanda:** probabilidad de que el sistema se comporte de forma rara ante una petición.
- **Tasa de fallos:** frecuencia de comportamientos inesperados.

Métricas de fiabilidad

- **Tiempo medio entre fallos:** tiempo de operatividad del sistema antes de que aparezcan fallos.

$$\text{TMEF} = \text{TMDF} + \text{TMDR}$$

- **Disponibilidad:** probabilidad de que el sistema se encuentre disponible para su uso.

$$\text{Disponibilidad} = \text{TMDF} / (\text{TMDF} + \text{TMDR}) \times 100$$