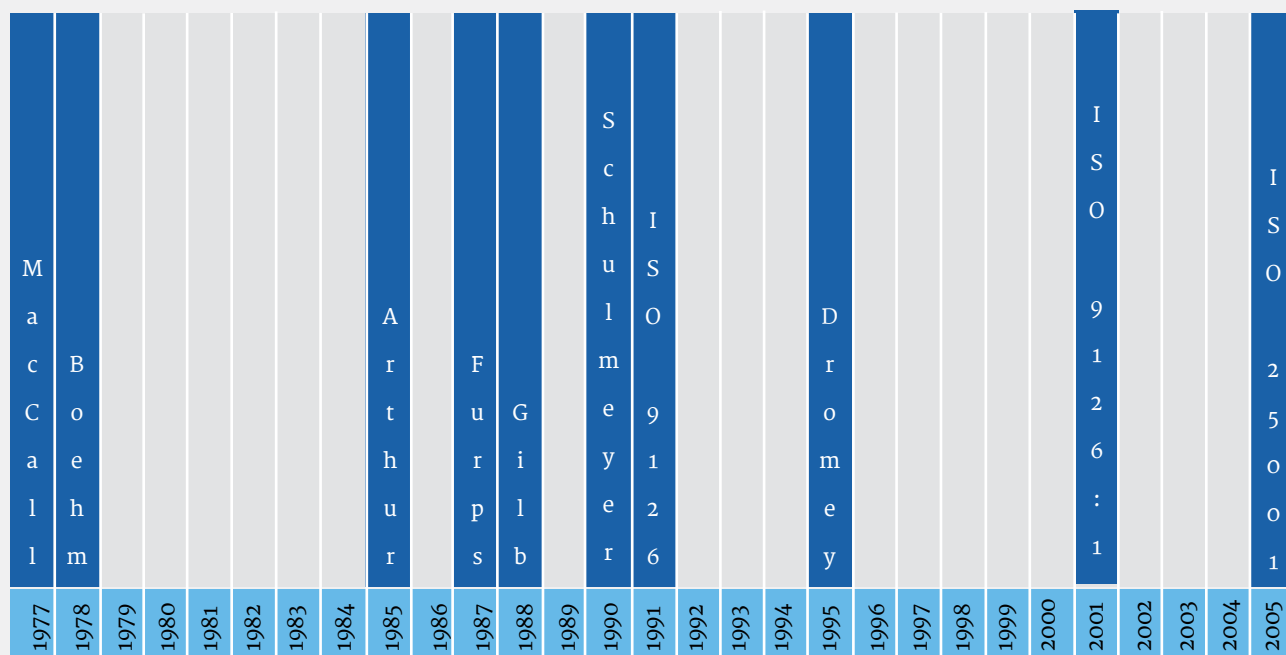




## 2.2. ALGUNOS MODELOS DE CALIDAD. ■

La calidad del software debe ser evaluada con objetividad y no con subjetividad, es por esto que desde hace mucho tiempo para que se pudiera medir la calidad con objetividad, se empezaron a definir modelos que incorporaban lo que se conoce como factores de calidad.

Existen varias propuestas de modelos que definen factores cualitativos que afectan la calidad del software, la más conocida es quizá la de los factores de calidad de McCall y esta será una de las que se presente junto con otras propuestas como la de Boehm, o FURPS (Functionality, Usability, Reliability, Performance, Supportability) de Hewlett-Packard, que son también conocidas y realizaron aportes interesantes. En la [figura 5](#) se presenta una línea de tiempo que muestra la aparición de las propuestas más conocidas para modelos de calidad.



**Figura 5.** Línea de Tiempo Modelos de Calidad.

Antes de presentar los modelos de calidad es conveniente resaltar que de una propuesta a otra, se maneja una notación diferente para referirse a los mismos conceptos, por ejemplo: McCall define factores, criterios, y métricas; Boehm define características, primitivas y métricas; y FURPS, define factores y atributos. Lo importante para evitar confusiones es recordar que los modelos de calidad presentan estructuras jerárquicas, donde los elementos de nivel superior son mucho más abstractos que los del nivel inferior que son más específicos y deben poder ser medidos con facilidad.

## 2.2.1. MODELO DE CALIDAD DE MACCALL.

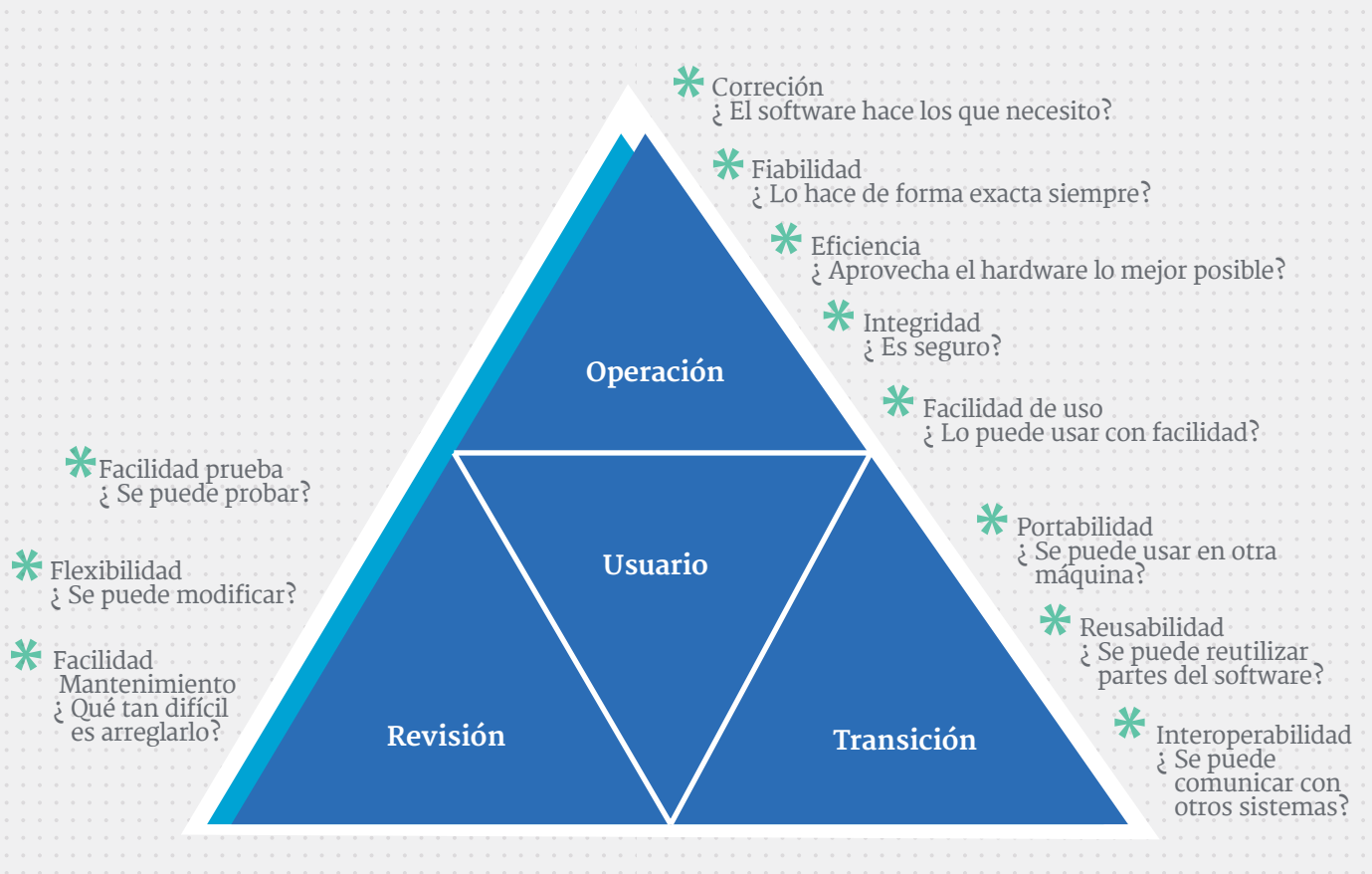
Este modelo de calidad fue presentado en 1977 y propone una serie de factores de calidad conocidos como factores de McCall, Richards, & Walters (1977), la idea del modelo es la descomposición del concepto genérico de calidad en tres capacidades importantes para un producto software, todo desde la mirada del usuario. A su vez cada capacidad se descompone en un conjunto de factores y finalmente se definen criterios para evaluar el factor a través de métricas que indican en qué medida el sistema posee una característica dada. En la [Tabla 3](#) se presentan detallan las capacidades y factores propuestos por McCall.

**Tabla 3.** Capacidades y Factores del Producto según McCall.

Capacidad	Factor	Significado
Operación	Corrección	Grado de cumplimiento de las especificaciones y objetivos del usuario.
	Confiabilidad	Grado en el sistema está disponible para usarse.
	Usabilidad	Grado de esfuerzo necesario que se requiere para aprender a utilizarlo.
	Integridad o Seguridad	Grado en el que se controla el acceso al programa o los datos por usuarios no autorizados.
	Eficiencia o Performance	Cantidad de recursos y código requeridos por un programa para realizar una función.
Transición	Portabilidad	Grado que mide el esfuerzo para migrar un programa de un entorno de operación a otro.
	Reusabilidad	Grado de esfuerzo requerido para que el programa o una de sus partes pueda ser utilizado en otro proyecto.
	Interoperabilidad	Grado de esfuerzo dedicado para que un sistema o programa pueda operar conjuntamente con otro.
Revisión	Facilidad Mantenimiento	Esfuerzo requerido para localizar y corregir un error en un programa en funcionamiento.
	Flexibilidad	Esfuerzo requerido para modificar un software en funcionamiento.
	Facilidad de Prueba	Grado de esfuerzo requerido para probar un programa verificando que realice adecuadamente sus funciones.

**Nota Fuente.** Adaptado de McCall & Cavano (1978)

En la figura 6 se representa como el modelo está planteado desde la perspectiva del usuario.



**Figura 6.** Modelo de Calidad McCall.  
**Nota Fuente.** Adaptado de Scalone (2006).

Las métricas propuestas por McCall para indicar el grado en que un sistema posee una determinada característica que impacte la calidad se presentan en la [Tabla 4](#).

**Tabla 4.** Métricas Propuestas por el Modelo de McCall

Métrica	Significado
Auto documentación	Grado en que el código fuente brinda información de documentación importante.
Capacidad de expansión	Grado permitido de ampliación del diseño de la arquitectura de datos o procedural.
Compleción de las funciones	Grado en que se pudieron implementar las funciones requeridas.
Complejidad	Complejidad del sistema
Concisión	Densidad del programa en relación a las líneas de código.
Consistencia	Diseño uniforme del programa empleando técnicas de documentación.
Eficiencia de ejecución	Rendimiento en tiempo de ejecución
Estandarización de comunicaciones	Grado de uso de estándares y protocolos.
Estandarización de datos y estructuras	Manejo de tipos de datos y estructuras uniformes a lo largo del programa.
Exactitud de cálculo y de control	Precisión obtenida en los cálculos.
Facilidad de auditoria	Facilidad de comprobación.
Independencia del hardware	Grado de desacople del software en relación al hardware donde opera.
Independencia del software	Grado de independencia del software en relación al sistema operativo, y otras limitaciones del entorno.
Instrumentación	Grado de auto-vigilancia en el funcionamiento e identificaciones de errores.
Modularidad	Independencia funcional de los componentes.

Operatividad	Facilidad de operación.
Seguridad	Disponibilidad de elementos de protección del programa y la información.
Simplicidad	Grado de la dificultad para entender el software.
Tolerancia a errores	Grado de afectación causado por un error.
Trazabilidad	Capacidad de seguimiento y asociación de los requisitos con los elementos de diseño.

**Nota Fuente:** Adaptado de McCall & Cavano (1978)

Las métricas representan medidas indirectas, es decir, permiten medir la calidad pero no de manera directa sino a través de una manifestación visible de ella. La [Tabla 5](#) presenta la relación entre algunos de los factores de calidad que propone McCall y algunas métricas comúnmente utilizadas para evaluar la calidad del software.

**Tabla 5.** Relación entre Factores de Calidad y Métricas de la Calidad del Software

Métricas de la Calidad del Software	Factores de Calidad										
	Corrección	Confiabilidad	Eficiencia	Integridad	Mantenimiento	Flexibilidad	Facilidad Prueba	Portabilidad	Reusabilidad	Interoperabilidad	Usabilidad
Auto documentación					X	X	X	X	X		
Capacidad de expansión						X					
Compleción	X										
Complejidad		X				X	X				
Concisión			X		X	X					
Consistencia	X	X			X	X					
Eficiencia de ejecución			X								
Estd. Comunicaciones										X	
Estandarización de datos										X	
Exactitud		X									
Facilidad de auditoria				X			X				
Facilidad de formación											X
Generalidad						X		X	X	X	
Independencia hardware									X	X	
Independencia del sistema									X	X	
Instrumentación				X	X		X				
Modularidad		X			X	X	X	X	X	X	
Operatividad			X								X
Seguridad				X							
Simplicidad		X			X	X	X				
Tolerancia a errores		X									
Trazabilidad	X										

**Nota Fuente.** Tomado de Fenton (1991)

Este modelo es un modelo de calidad fijo y para realizar la evaluación de la calidad de un producto utilizándolo, se selecciona un subconjunto de factores y métricas aceptando que solo están disponibles las que propone el modelo.

## 2.2.2. MODELOS DE CALIDAD BOEHM.

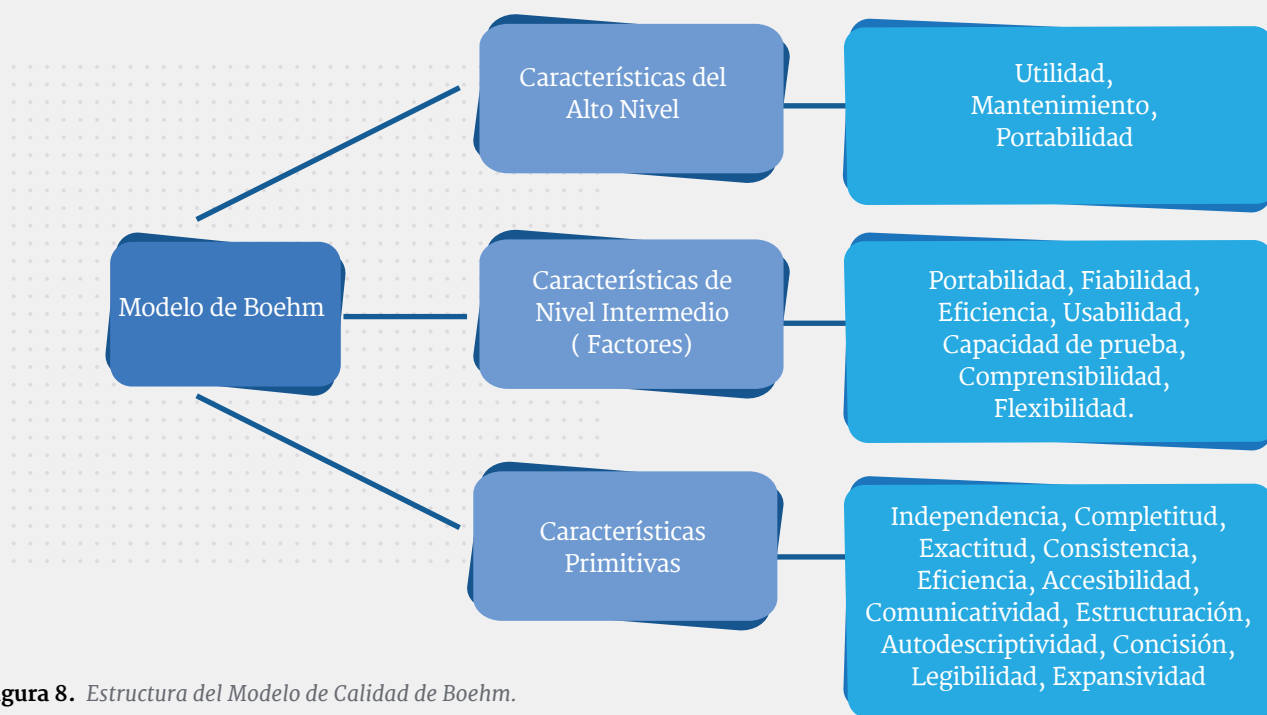
Este modelo de calidad fue propuesto por Barry Boehm en el año de 1978 y es similar al modelo de McCall definiendo la calidad en términos de atributos cualitativos y métricas para realizar las medidas, ver figura 7.



**Figura 7.** Relación entre Factores y Métricas en el Modelo de Calidad de Boehm.

**Nota Fuente.** Elaborado a partir de Scalone (2006)

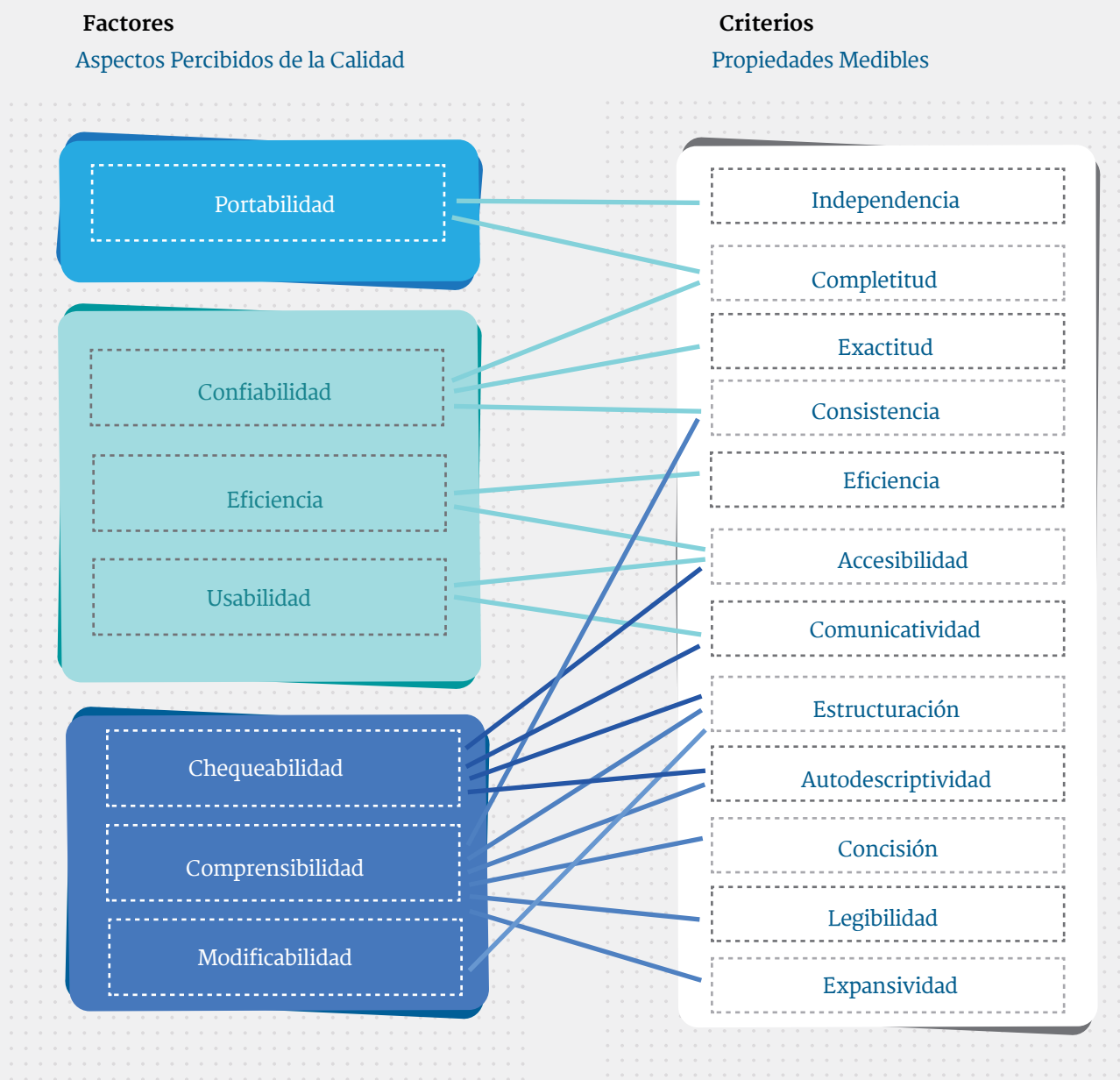
Dentro de los factores que se describen en el modelo se toman muchos de los que propone McCall. La estructura jerárquica del modelo se presenta en la figura 8 y plantea 3 niveles para las características: de alto nivel, de nivel intermedio y nivel primitivo.



**Figura 8.** Estructura del Modelo de Calidad de Boehm.

**Nota Fuente.** Elaborado a partir de Scalone (2006)

La relación entre factores y características/ propiedades/criterios medibles se presenta en la figura 9.

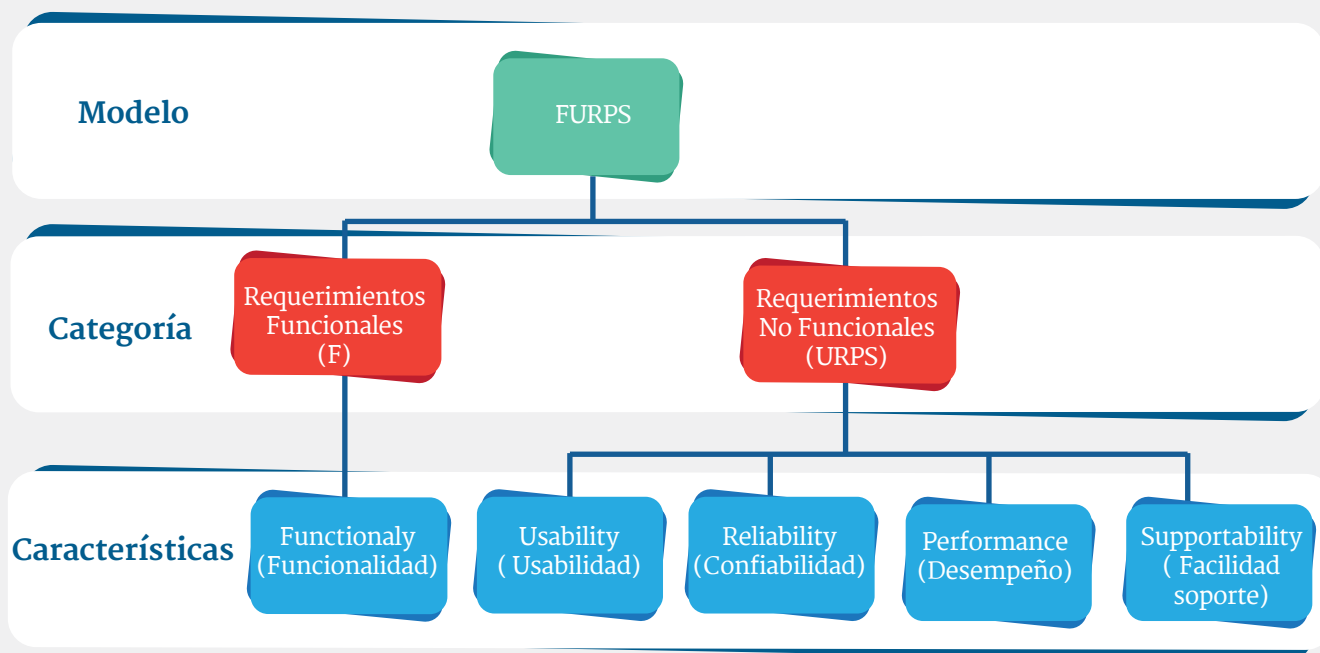


**Figura 9.** Relación entre Factores y Criterios en el Modelo de Boehm.  
**Nota Fuente.** Elaborado a partir de Scalone(2006)



### 2.2.3. FURPS.

Modelo de calidad propuesto por Robert Grady y Hewlett Packard Co (HP) en 1987. Esta propuesta contempla, por un lado 5 características de las cuales se deriva su nombre (Funcionalidad, Facilidad de Uso, Confiabilidad, Performance y Facilidad de Soporte), y por otro, que los requisitos se clasifiquen en dos categorías: requisitos funcionales (F), que son los que especifican funciones que el sistema debe ser capaz de realizar sin tener en cuenta las restricciones físicas; y requerimientos no funcionales (URPS), que puntualizan atributos del sistema o del medio ambiente del sistema. En la figura 10 se presenta la representación de este modelo.



**Figura 10.** Modelo de Calidad FURPS.

Este modelo es también un modelo de calidad fijo y para realizar la evaluación de la calidad de un producto, primero se asignan prioridades y después se definen los atributos de calidad que pueden ser medidos.

## 2.2.4 ESTÁNDARES Y MODELOS DE CALIDAD.

Se puede dar el caso de que un estándar proponga un modelo de calidad, como es el caso del estándar ISO 9126 que define un modelo de calidad mixto, basándose en elementos definidos en modelos ya existentes como McCall & Boehm (1976).

La norma ISO 9126 presenta dos partes, la primera es el modelo de calidad para tratar la calidad externa e interna, y la segunda es el modelo de calidad uso para tratar la calidad en uso; esta norma será presentada en el siguiente capítulo.

