

# Proyecto Integrador

BMJIvan

August 2021

# 1. Definiciones

Mecatronics journal	1991 M, E	Balance óptimo de estructuras mecánicas y control
IEEE/ASME	1996 M, E, C	Integración sinérgica, diseño y manufactura de producto y procesos
Bishop	2002 M, E, I	Proceso en evolución, integración, sinérgica
Amerogen	2003 M, E	Diseño como un todo
Iserman	2005 M, E, I, C	Integrar sistemas, diseño simultaneo, balance óptimo entre estructuras, actuadores, sensores, información y control
Bolton	2008 M, E, C	Completa integración
De Silva	2008 M, E, I, C	Sinergia y diseño integrado, sistemas electromecánicos, grado de inteligencia, más preciso, exacto, seguro, flexible, funcional y mecánicamente menos complejo
Sheltty y Kolk	2011 M, E, I	Diseño óptimo, productos electromecánicos, concurrencia y sinergia
Merzouki	2013 M, E, C	Mejorar sistemas mecánicos con control inteligente. Reemplazar componentes mecánicos con electrónica
Cetinkowt	2015 M, E, I	Integración sinérgica, nivel de sistemas. Diseño simultaneo para obtener un diseño óptimo

Tabla 1: Definiciones

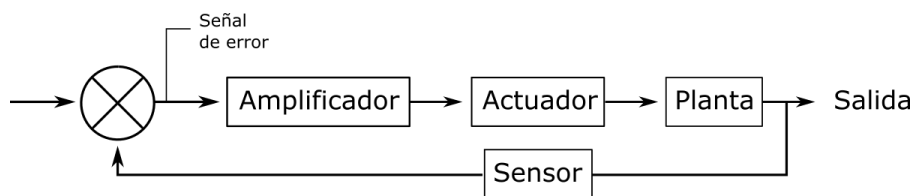


Figura 1: Ogata: Sistema control industrial

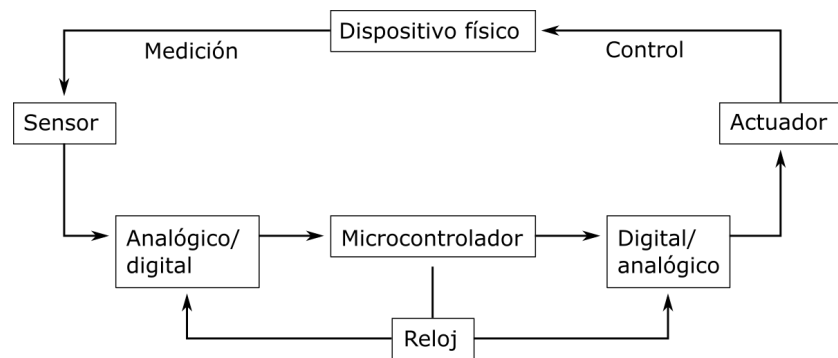


Figura 2: Bishop: Sistema mecatrónico

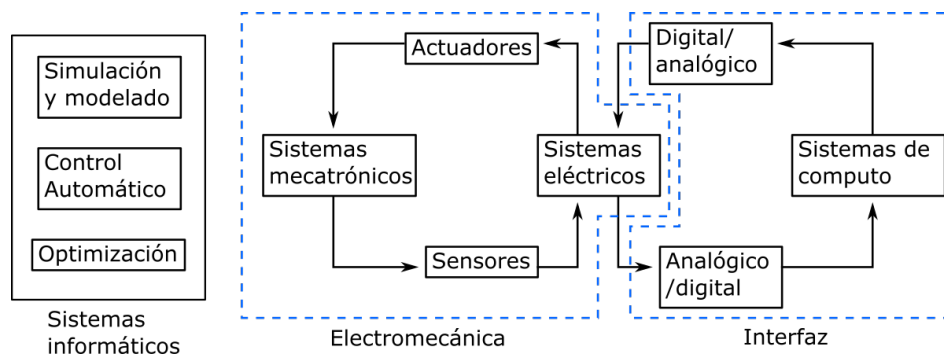


Figura 3: Shetty: Sistema mecatrónico

## 2. Proyectos

- Proyecto: Conjunto de actividades temporales para obtener un producto o un servicio.
- PMBOK: Project Management Body of Knowledge (PMP)
- Hito: Resultado medible atemporal

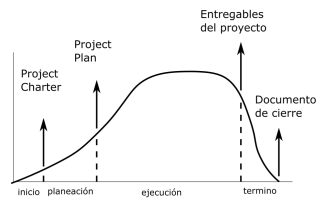


Figura 4: Ciclo de vida de proyecto

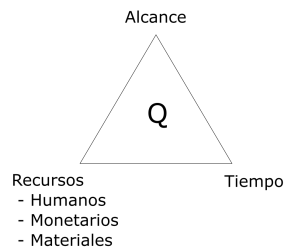


Figura 5: Triángulo de hierro

### 3. Ingeniería concurrente y de sistemas

1. Ingeniería concurrente (CE): Es un enfoque sistematizado para el diseño concurrente de productos y procesos, considerando desde el inicio todos los elementos que componen el ciclo de vida del producto.

Principales objetivos Incrementar la calidad del producto, la reducción general del gasto y la reducción del tiempo de desarrollo. Principales características

- Mejor definición del problema de diseño
  - Implementación de herramientas de manufactura y ensamble en la etapa de diseño
  - Mejora de la estimación del costo
  - Reducción de problemas entre los procesos de diseño y manufactura
2. Ingeniería de sistemas: Consiste en el desarrollo de sistemas que sean capaces de satisfacer requerimientos dentro de un conjunto de restricciones bien definidas.

Sistema: Es un conjunto o colección de diferentes elementos que en conjunto producen resultados que no se podrían obtener de forma independiente.

Modelo TTDSE (Traditional Top-Down System Engineering) Se compone de dos etapas principales

- S1: Análisis de necesidades, definición del sistema, definición de subsistemas, definición de componentes y definición de configuración de componentes.
- S2: Verificación desde la configuración de componentes al sistema final, validación y aprobando los resultados obtenidos. (modelo V)

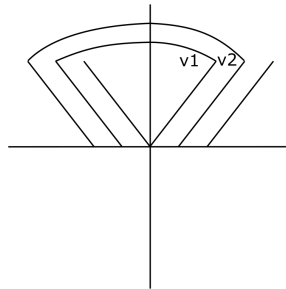


Figura 6: Modelo V

Modelo espiral: Consiste de cuatro procesos

- Planificación: investigación
- Análisis y evaluación de riesgos: diseño y prototipo
- Desarrollo y pruebas
- Aprobación: mantenimiento, retroalimentación

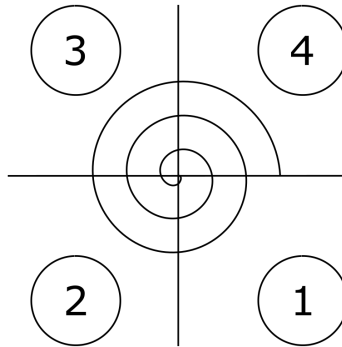


Figura 7: Modelo en espiral

Existen tres tipos de arquitecturas para sistemas

- A1: Arquitectura funcional-lógica  
Define que es lo que el sistema debe hacer, las acciones o funciones del sistema. Incluye modelos de información, procesos y el comportamiento inicial. (Modos de operación)
- A2: Arquitectura física Define los componentes, ensambles y elementos físicos que se requieren para el cumplimiento de las funciones, también representa las conexiones físicas entre componentes, sistemas, subsistemas y elementos.
- A3: Arquitectura de asignación Es el mapeo o relación de funciones y recursos necesarios. Se define el modelo final de comportamiento del sistema.

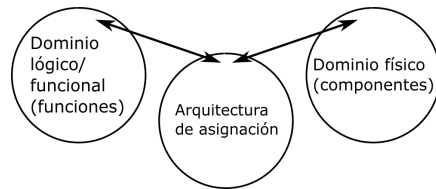


Figura 8: Arquitectura de asignación



Figura 9: Sistema de funciones

## 4. Diseño del sistema

### 4.1. Conceptos

- Necesidad: Es el resultado de una transformación de uno a más conceptos en un expectativa para el cumplimiento de una función con cierto o dentro de cierto rendimiento. Indicadores de calidad (KPI).
- Requerimiento: Es el resultado de una transformación de una o más necesidades en una obligación para cumplir cierta función dentro de las cotas establecidas.

### 4.2. Clasificación general de requerimientos

1. Requerimientos funcionales: Algo que el sistema debe hacer o proveer.
2. Requerimientos no funcionales: Alguna propiedad o atributo que el sistema debe tener pero que no se modifica el comportamiento o cumplimiento de las funciones que desempeña (deben cumplirse, pero no afecta la funcionalidad).
3. Restricción: Son las fronteras en las que el sistema debe operar. Comúnmente se enlistan dentro de los requerimientos funcionales y no-funcionales.

Un requerimiento debe definir qué es lo que debe hacer, no como hacerlo.  
¿Por qué se necesitan?

- Para definir el alcance del proyecto.
- Para asegurar el cumplimiento de las expectativas.
- Para poder reportar un progreso.
- Para medir el avance en el proceso del diseño.

Principales características

1. Necesario
2. Singular
3. Correcto
4. inequívoco
5. Realizable
6. Completo: Tenga un rango
7. Ajustable: Que se pueda medir



### 4.3. Términos

- Necesidad: Expectativa no tangible
- Requerimiento: Algo acotado
- Especificación: Valor final medible
- Diseñar: Cumplir requerimientos
- Necesidad no funcional: Son aquellas que no afectan la función principal

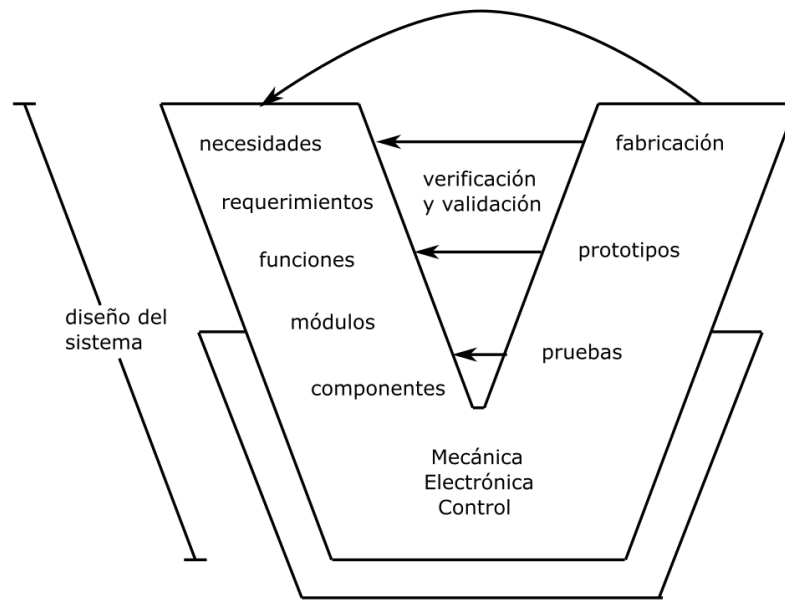


Figura 10: VDI-2206

## 5. Dominios en el desarrollo de sistemas mecatrónicos

### 5.1. Dominio lógico o funcional: funciones, relación, comportamiento

Es el dominio donde se lleva a cabo la descomposición de funciones, la función principal que el sistema debe realizar es dividida en funciones que busquen definir y describir el comportamiento del sistema. Así mismo, deben cumplir los objetivos de diseño, los requerimientos y las necesidades.

El dominio puede dividirse en  $m$  espacios, de acuerdo a los niveles jerárquicos de funciones

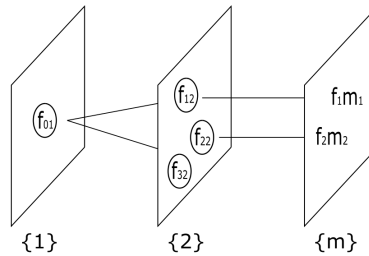


Figura 11: Dominio funcional

Una función puede definirse de manera general como la transformación de una entrada en una salida. Esta transformación debe mantener el índice de desempeño deseado. Las entradas y salidas de la transformación deben ser del tipo

1. Materia
2. Energía
3. Información

### 5.2. Clasificación de funciones

Las funciones se puede clasificar en los siguientes tipos

1. Función principal: Es una parte fundamental del sistema, es lo que cumple el comportamiento del sistema. Es predominante independiente. A partir de esta función, se puede definir las funciones propias del sistema.
2. Funciones secundarias: Son las funciones requeridas para que el sistema cumpla con la función principal.
3. Funciones básicas: Son funciones que un sistemas mecatrónico debe desempeñar, sin importar los objetivos de diseño, necesidades y requerimientos.

4. Funciones irrealizables: Son funciones que debido a fenómenos físicos o tecnología actual no pueden realizarse.

La descomposición de funciones se debe realizar de una forma jerárquica y sistematizada. Se recomienda que cada nivel (espacio) contenga las funciones que permitan describir el comportamiento del sistema. El proceso de descomposición es iterativo.

### 5.3. Relación entre funciones

La interrelación entre funciones está definida por la conexión y la comunicación de las entradas y salidas, describiendo en conjunto el comportamiento del sistema.

Se puede definir una función de manera general como:

$$\mathcal{F} : A(E, M, I) \rightarrow B(E, M, I)$$

Donde  $\mathcal{F}$  es la función que transforma las entradas de  $A$  en las salidas de  $B$ .

Cuando una función está compuesta de dos funciones ( $f_1$ ,  $f_2$ ), la relación entre ellas se puede definir como:

1. Si la trayectoria de  $f_1$  está contenida en  $f_2$  se considera que las funciones son secuenciales. Se puede definir la relación como la composición de funciones ( $f_1 \circ f_2$ ). Las salidas de  $f_1$  se convierten en las entradas de  $f_2$ .



Figura 12: Ciclo de vida de proyecto

2. Si la trayectoria de cada función es independiente, entonces las funciones son paralelas y su transformación puede ser simultanea. El operador „AND” se emplea para describir esta operación.

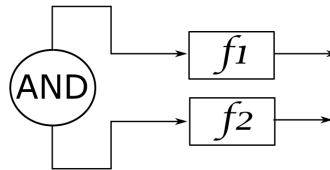


Figura 13: Ciclo de vida de proyecto

3. Si la trayectoria de cada función es independiente, son paralelas, pero no pueden realizar la transformación simultánea. Se emplea el operador "XOR".

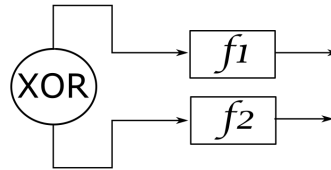


Figura 14: Ciclo de vida de proyecto

## **6. Arquitectura funcional**

### **6.1. consideraciones para generar la arquitectura funcional**

1. Las funciones deben ser claras.
2. Entender la conexión y comunicación entre funciones.

### **6.2. Esta compuesta por los siguientes tres modelos**

1. Jerarquía de funciones (FBS: Function, Behaviour, Structure)
2. Relación entre funciones (IDEF-0: Integration Definition for Function Modeling)
3. Comportamiento entre funciones (EFFBD: Enhanced Function Flow Block Diagram)

Considerando los atributos de sistemas mecatrónicos, y las funciones generales del sistema propuesto. Las funciones genéricas del sistema mecatrónico son:

- F10: Medición de parámetros del sistema y entorno.
- F20: Detección de errores del sistema.
- F30: Interacción con el usuario.
- F40: Administración de decisiones, proceso de aprendizaje, manejo del conocimiento.
- F50: Acondicionamiento de entradas y salidas.
- F60: Almacenamiento de información.
- F70: Comunicación interna y externa.
- F80: Recuperación de la funcionalidad.
- F90: Desempeñar funciones propias del sistema.
- F100: Soporte y protección.
- F110: Administración energética (alimentación, conversión, distribución, almacenamiento)

### 6.3. Principales errores al generar la arquitectura funcional

- Mejorar dominios, incluir funciones que describen el comportamiento, sistemas o componentes físicos en la misma arquitectura. Por ejemplo: HMI (Human Machine Interface), cuando la función que desempeña es la interacción con el usuario.
- Inconsistencias en la salida de las funciones, deben representar la transformación.
- Elegir nombres confusos e incorrectos para las funciones, entradas o salidas.
- Que la función represente o desarrolle más de una función.
- La arquitectura (modos de operación) no representa el comportamiento completo del sistema.
- Las salidas del sistema y las funciones no representan las expectativas y necesidades del sistema.

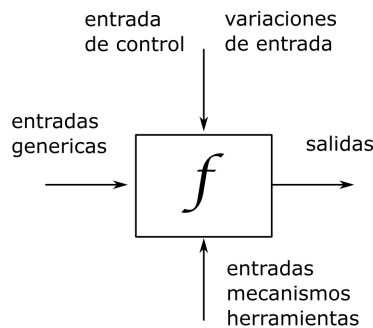


Figura 15: IDEF-0

### 6.4. Modelo EFFBD

Permite en la arquitectura funcional determinar el comportamiento que tiene un sistema a través del flujo de funciones. Este flujo está definido como el trayecto que siguen las entradas y salidas a lo largo de transformaciones.



Figura 16: Función

#### 6.4.1. Generalidades

Las funciones se representan con bloques, estos tienen un indicador (el mismo que se obtiene de la estructura funcional).

Existen dos tipos de entradas y salidas, las entradas "pasivas" las entradas "activas".

Las activas se refieren a que mantienen su flujo activo mientras se realiza la transformación. Las pasivas ya no transmiten flujo hasta que termine la transformación.



Figura 17: Activas y pasivas

Para describir el comportamiento se tiene cuatro tipos de operadores lógicos.

1. AND
2. OR (paralelo), XOR (no se puede en paralelo)
3. LOOP
4. ITERACION



Figura 18: Operadores

Estos operadores se representan con un círculo y su símbolo se emplean deben aparecer al inicio y al fin de su uso. En el caso de los operadores "LP", se deben incluir la condición a la que están sujetas.

Al inicio y al término del modelo se deben incluir funciones o acciones de referencias: se expresan como:

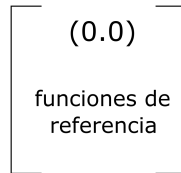


Figura 19: Funciones de referencia

Existen otro tipo de funciones que son tentativas, es decir, no siempre existen. Estas se representan con bloques punteados.

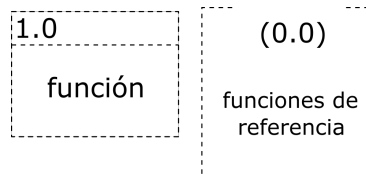


Figura 20: Funciones tentativas

Si existieran elementos actuadores del comportamiento, estos se modelan con rectángulos con aristas redondeadas.

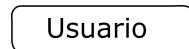


Figura 21: Elementos actuadores

Para evaluación del diseño se emplea el coeficiente de diseño mecatrónico (MQD). Con los siguientes atributos.

1. Inteligencia
2. Robusto
3. Flexible
4. Adaptable
5. Eficiencia del ensamble



## 7. Dominio físico

El dominio físico es la transformación de las funciones que el sistema debe desempeñar en componentes físicos. Estos pueden ser partes, ensambles, módulos, dispositivos, sistemas.

La representación del dominio físico es mediante un diagrama a bloques, denominado arquitectura funcional, con base en las funciones generadas de un sistema mecatrónico, se propone la siguiente arquitectura.

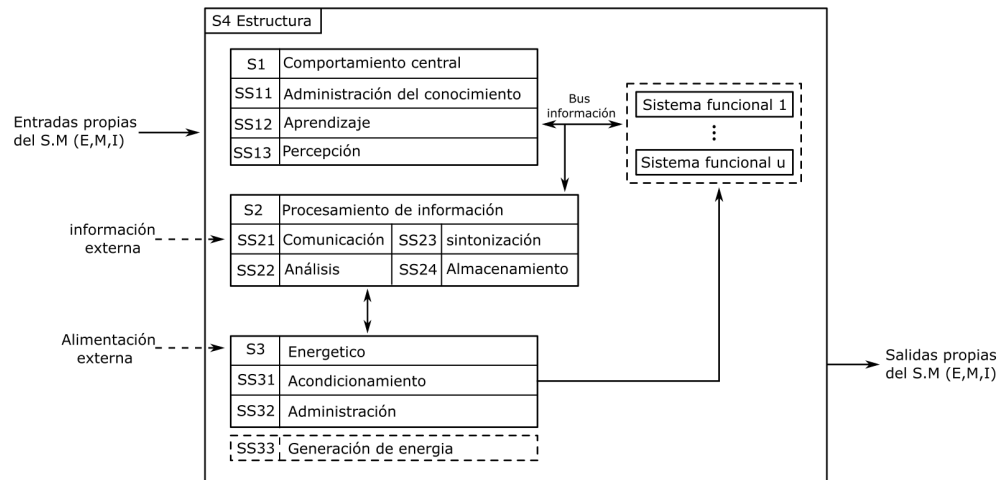


Figura 22: Dominio físico

Con base en teoría de sistemas, un sistema mecatrónico se puede considerar como un "sistema de sistemas"(SoS), y a su vez el sistema que tiene mayor impacto en el sistema mecatrónico se denomina como "sistema de interés"(SoI).

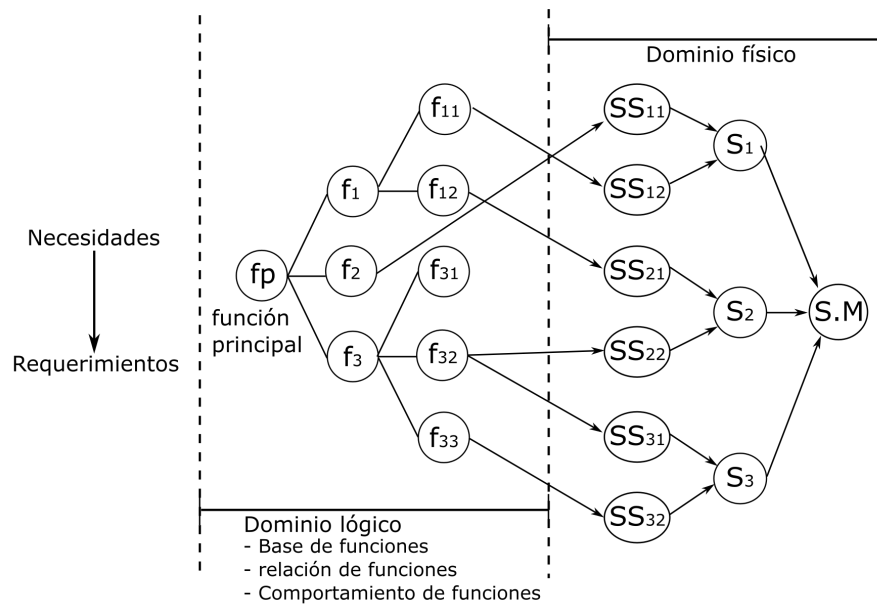


Figura 23: Dominio lógico y físico

## 8. Diseño conceptual

Con base en los sistemas que conforman la arquitectura física se requiere comenzar a buscar posibles soluciones. Estas soluciones deben cumplir las funciones, los requerimientos y las necesidades.

La integración de las soluciones por cada uno de los módulos se denomina diseño conceptual.

### 1. Análisis morfológico, caja morfológica

- De cada característica se requiere al menos una posible solución.
- Posibles soluciones, no importa si parecen "absurdas".
- Mayor cantidad de posibles soluciones.

$C_1$	$Ps_{11}$	$\cdots$	$Ps_{1j}$
$C_2$	$Ps_{21}$	$\cdots$	$Ps_{2j}$
$\vdots$	$\vdots$		$\vdots$
$C_i$	$Ps_{i1}$	$\cdots$	$Ps_{ij}$

Tabla 2: Caja morfológica

No

- Componentes
- Materiales
- Procesos de manufactura

Si

- Mecanismos
- Controladores
- Formas
- Métodos
- Ubicación componentes

### 2. Generar combinaciones

- Deben ser congruentes
- De 3 a 5 combinaciones
- Deben incluir todas las características
- Deben buscar el mejor desempeño de las funciones

Cada combinación se denomina "Diseño conceptual"

### 3. Seleccionar el mejor diseño conceptual

- Generar los criterios de selección
  - Determinar una herramienta de selección multicriterio
  - Resolver el problema de selección
4. Analizar resultados del método de selección
  5. Comprobar