



# Unidad 1

## Autómatas, Sistemas Discretos y Sistemas Híbridos

---



# AUTÓMATAS y CONTROL DISCRETO

## Unidades Temáticas:

### 1. Autómatas, Sistemas Discretos y Sistemas Híbridos

Aplicaciones y Fundamentos. FSM. STATECHARTS. PETRI Nets. GRAFCET. Sistemas Híbridos.

### 2. Autómatas Programables de Control – IEC 61131: Arquitecturas y Lenguajes

PLCs, arquitecturas. Seguridad Funcional. Entornos de desarrollo. Configuración. Lenguajes ST/ SFC.

### 3. Diseño de Sistemas de Control Secuencial con Autómatas Programables

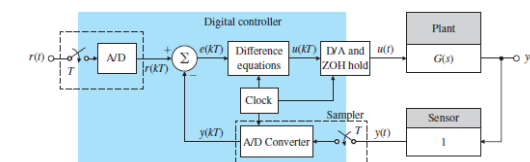
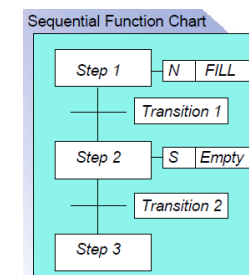
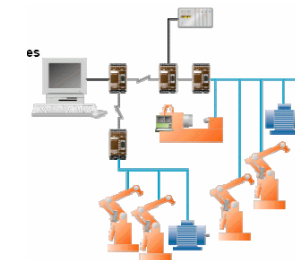
Métodos clásicos. Métodos sistemáticos y modernos para sistemas complejos. Aplicación.

### 4. Arquitecturas Distribuidas de Control y Comunicaciones Industriales

Arquitecturas distribuidas. Buses de Campo, Redes y Protocolos industriales de comunicación.

### 5. Control Discreto de Sistemas Continuos, Control Híbrido

Sistemas muestreados. Simulación y Control. Control y Simulación Híbridos. Aplicación.

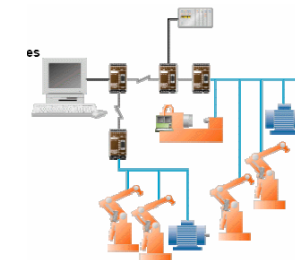


# AUTÓMATAS y CONTROL DISCRETO

## 1. *Autómatas, Sistemas Discretos y Sistemas Híbridos*

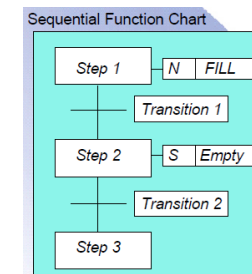
### A. Concepto y Aplicaciones

- Automatización y Control Industrial: Concepto y esquema general. Componentes y estructura.
- Aplicaciones Mecatrónicas → Control automático secuencial + control de tiempo discreto.



### B. Fundamentos de Autómatas Secuenciales de Control

- Sistemas secuenciales: estados discretos activados por eventos. Conceptos.
- Autómatas finitos. Modelado y representación. FSM. STATECHARTS.
- Simulación. STATEFLOW.

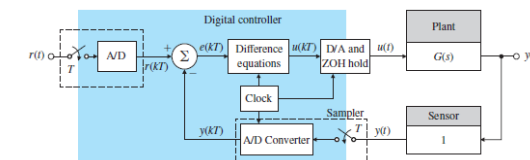


### C. Petri Nets (ISO/IEC 15909). GRAFCET (IEC 60848)

- Metodologías de representación y diseño. Redes de Petri.
- Lenguaje GRAFCET. Estructuras Jerarquizadas.

### D. Sistemas Híbridos. Práctica de Modelado y Simulación

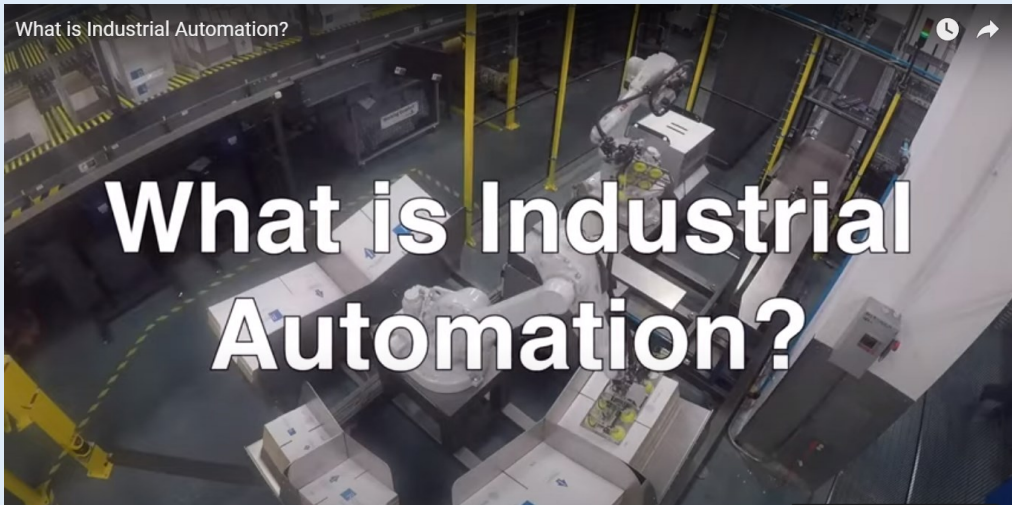
- Modelado y Simulación en STATEFLOW. Representación en GRAFCET.



# Sistemas de Automatización y Control Industrial

## What is INDUSTRIAL AUTOMATION? (Manufacturing & Factory Automation)

Video (2:35 min), RADWELL International: <https://www.youtube.com/watch?v=tw-79FiRYKA>

- **Maquinaria Automatizada** → Operación Optimizada y Confiable.
  - Interacción con **pocas Personas** en el Proceso.
  - Sistemas de **Control Discreto**: PLCs, HMI, ... Lógica y Programación (**Información**).
  - Dos tipos diferentes de **Automatización**:
    - **FIJA: Hardware/Firmware** (procesos repetitivos fijos);
    - **PROGRAMABLE: Software** (procesos x lotes, flexible).
- 
- Sistemas con Estados **Continuos** + Estados **Discretos** **activados por eventos asincrónicos**.
  - Control **Regulatorio (Retroalimentación)** + Control **Digital Secuencial (Autómatas)**  
→ Sistemas Dinámicos **Híbridos** (Estados **continuos** y **discretos** con **acoplamiento explícito**).

Aplicaciones Industriales en Fabricación Discreta o Procesos Continuos.  
Dominio de la **Alta Complejidad** que implican los procesos tecnológicos actuales.

# Sistemas de Automatización y Control Industrial

## *Nueva realidad y Tendencias: Presente → Futuro* *Industry 4.0 (Digital Industry) Overview*

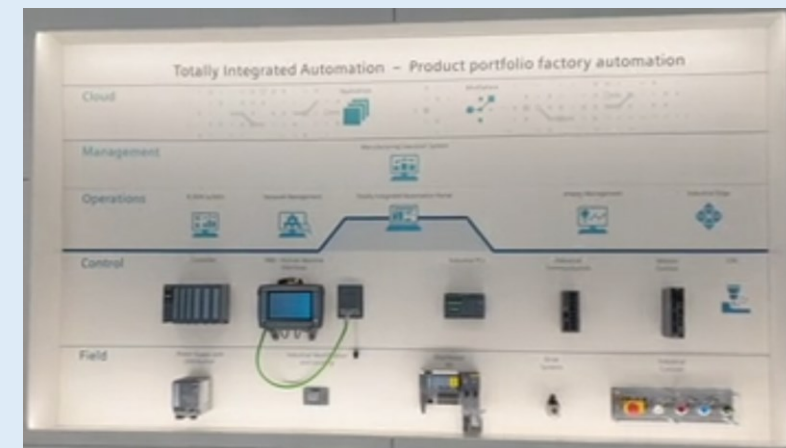
### *SIEMENS Virtual Tour Digital Industries 2020*

Panorama por Fernando Sanchez Gamboa,  
SIEMENS Nürnberg, Alemania (29/9/2020)

Video (15 min) <https://vimeo.com/463129690> (← ya no disponible aquí → BUSCAR)

### *Industria 4.0 (4ta. Revolución Industrial)*

- *Convergencia de Nuevas Tecnologías en Automatización Industrial:*
- *Digitalización e integración ...*
- *Digital Twin (Gemelo Digital o Virtual) ...*
- *Big Data → Smart Data ...*
- *Cloud Computing / Edge Computing ...*
- *AI: Inteligencia Artificial...*
- *Realidad Virtual / Realidad Aumentada ...*
- *IoT industrial (Internet of Things)...*
- ...



# Sistemas de Automatización y Control Industrial

## *Nueva realidad y Tendencias: Presente → Futuro* *Industry 4.0 (Digital Industry) Overview*

### **Industrie 4.0 -The Fourth Industrial Revolution**

**Video** (5:34 min, 2013) <https://www.youtube.com/watch?v=HPRURtORnis>

#### *Industria 4.0 (4ta. Revolución Industrial)*

- *Convergencia de Nuevas Tecnologías en Automatización Industrial:*
- *Digitalización e integración ...*
- *Digital Twin (Gemelo Digital o Virtual) ...*
- *Big Data → Smart Data ...*
- *Cloud Computing / Edge Computing ...*
- *AI: Inteligencia Artificial...*
- *Realidad Virtual / Realidad Aumentada ...*
- *IoT industrial (Internet of Things)...*
- ...



# Sistemas de Automatización y Control Industrial

## *Nueva realidad y Tendencias: Presente → Futuro* *Industry 4.0 (Digital Industry) Overview*

**Industry 4.0** - Germany's 4th industrial revolution

Video (9:31 min, 2016) <https://www.youtube.com/watch?v=Y990kaGbJD0>

### *Industria 4.0* (4ta. Revolución Industrial)

- *Convergencia de Nuevas Tecnologías en Automatización Industrial:*
- *Digitalización e integración ...*
- *Digital Twin (Gemelo Digital o Virtual) ...*
- *Big Data → Smart Data ...*
- *Cloud Computing / Edge Computing ...*
- *AI: Inteligencia Artificial...*
- *Realidad Virtual / Realidad Aumentada ...*
- *IoT industrial (Internet of Things)...*
- ...



# Sistemas de Automatización y Control Industrial

*Nueva realidad y Tendencias: Presente → Futuro*  
*Industry 4.0 (Digital Industry) Overview*

**SIEMENS Virtual Tour Digital Industries 2020**

Virtual Guided Tour "Digital Enterprise SPS 2020 Dialog" – **Motion Control Topics**

[https://www.youtube.com/watch?v=im5tOLtuMzg&ab\\_channel=Siemens](https://www.youtube.com/watch?v=im5tOLtuMzg&ab_channel=Siemens)  
Video (12:42 min, 2020)



# Sistemas de Automatización y Control Industrial

**SISTEMA**: Conjunto de **Elementos interrelacionados**, considerado como **un Todo** en contexto definido, y **separado** de su **Entorno o Ambiente**, intercambio de variables o señales.

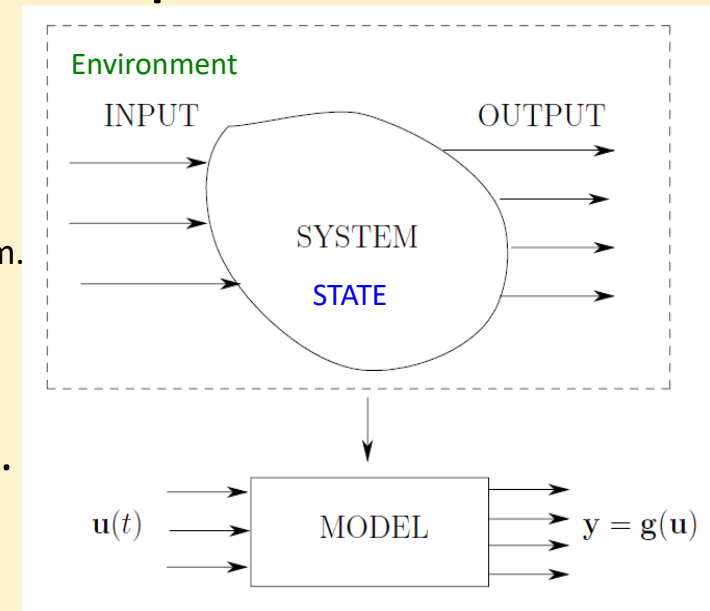
**SYSTEM**: Set of **interrelated Elements** considered as a **Whole** in a defined context and **separated** from their **Environment**.

- A system is generally defined with the view of **achieving a given objective**, for example by **performing a definite function**.
- Elements of a system may be natural or man-made material objects, as well as modes of thinking and the results thereof (for example forms of organization, mathematical methods, programming languages).
- The system is considered to be **separated** from the **environment** and the other external systems **by an imaginary surface**, through which pass the links (i.e. signals or variables) between them and the considered system.
- The term "system" should be qualified when it is not clear from the context to what it refers, for example control system, calorimetric system, system of units, transmission system.

**MODEL**: mathematical or physical **Representation** of a **System** or a process, based with **sufficient precision** upon **known laws**, **identification** or specified suppositions.

Fuente: **IEC 60050 Electropedia: The World's Online Electrotechnical Vocabulary**

<https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=351-42-08/>  
<https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=351-42-26>



**Ejemplo: SISTEMA DINÁMICO FÍSICO Continuo (CDS)**: Elementos **físicos** (naturales o contruidos) que **almacenan** e intercambian **ENERGÍA**. Las variables involucradas: entradas, salidas, estado, son **CONTINUAS** (Reales).

**MODELO Dinámico**: Representación (matemática) **Aproximada** del SD que permite **analizar y simular** los aspectos de interés del **comportamiento** del SD en su **evolución temporal**.

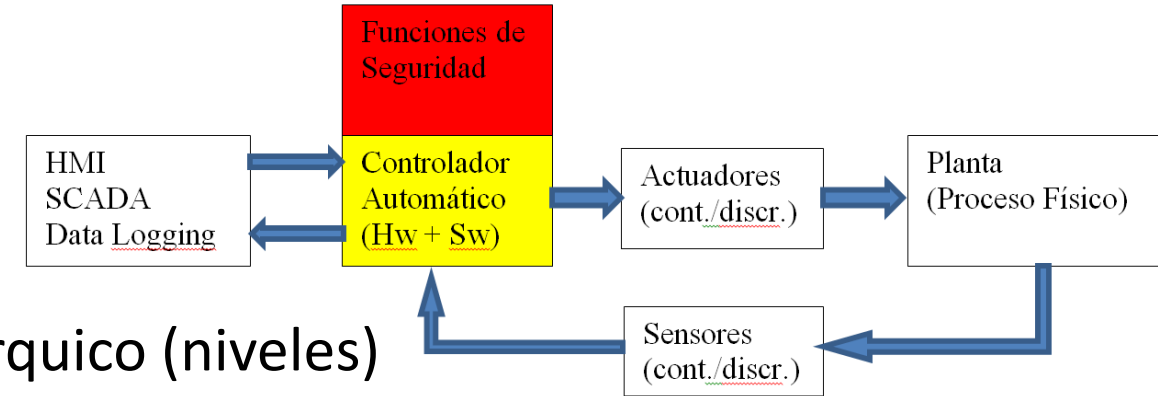
**Ejemplo: SISTEMA COMPUTACIONAL O SIMBÓLICO (DEDS)**: Elementos **lógicos o simbólicos**, **almacenan** e intercambian **INFORMACIÓN**.

# Sistemas de Automatización y Control Industrial

## Estructura Conceptual

### Planta o Proceso Físico a Controlar:

- Sistema Dinámico **Multivariable** (MV o MIMO)
- **Híbrido**: Estados Continuos + Estados Discretos
- **Complejo**: Descentralizado vs. Distribuido y Jerárquico (niveles)



### Funciones de **CONTROL**: (Regulatorio Continuo + Supervisor Discreto)

- Comandos, Monitoreo y Supervisión, Registro de datos y alarmas.
- Salidas Medidas de Planta (+Estimación) → **Señales de Retroalimentación (Feedback)**.
- **Señales de Control** → Actuación sobre Entradas Manipuladas de Planta.

### Funciones de **SEGURIDAD**: (**PROTECCIÓN** x Fallas: **Safety** / Sabotaje: **Security**)

- Vigilancia interna (Watchdog) / Redundancia I/O.
- Sistema separado de Emergencia cableado (relé maestro de Seguridad) ó Sistema de Seguridad Intrínseca (Hardware + Software).

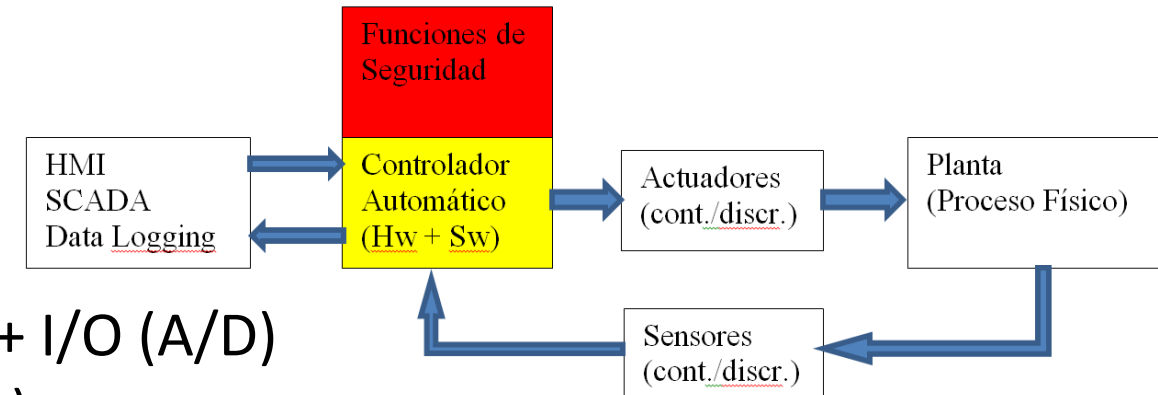
# Sistemas de Automatización y Control Industrial

## Estructura Conceptual

**Controlador Automático:** (Hardware + Software)

**Autómata Programable (PLC, PAC)**

- Procesador digital + Memoria + Comunicaciones + I/O (A/D)
- **Arquitectura:** Centralizada o Distribuida (Remoto)



Funciones de CONTROL → Objetivo: **Desempeño**, dentro de Condiciones Normales

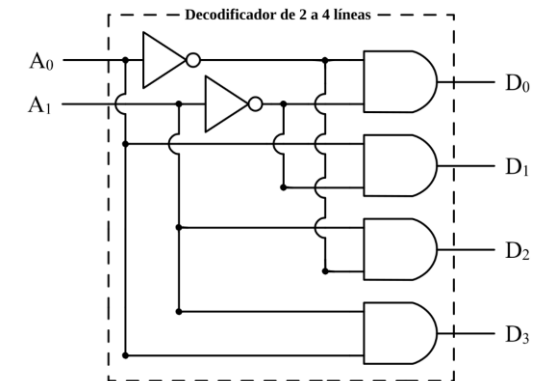
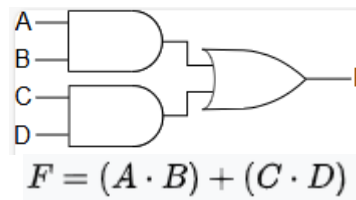
- **Control Regulatorio a Lazo Cerrado (Continuo)**: Retroalimentado (FBK) / en avance (FFW) (implementación digital en **tiempo discretizado**, datos muestreados)
- +
- **Control Supervisor (Secuencial)**, Autómata **Discreto** activado por **Eventos** (FSM): Modos de Operación, Diagnóstico de Estado y Monitoreo de Fallas, etc.

Funciones de SEGURIDAD → Objetivo: **Protección**, hacia Estado Seguro ante Fallas (cadena de Emergencia cableada vs. sistema integrado de seguridad)

# Sistemas (Electrónicos) Digitales [Hw/Sw]: Lógica Booleana

**Sistema o Dispositivo DIGITAL:** genera, procesa, transmite y/o almacena señales, datos, o información *codificada* en Valores o Estados Discretos (Discontinuos), representables como combinación de **bits** (binary digits: {0,1}) → **Algebra de Boole** (operadores lógicos **AND, OR, NOT**) para **Análisis y Síntesis** de Sistemas Digitales.

1. Sistemas Digitales **COMBINACIONALES** (inmediatos o **Estáticos, SIN memoria**): Salidas Digitales solo dependen del valor **instantáneo actual** de sus Entradas Digitales.



| Tabla de verdad |   | A <sub>1</sub> | A <sub>0</sub> | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |
|-----------------|---|----------------|----------------|----------------|----------------|----------------|----------------|
| 0               | 0 | 0              | 0              | 0              | 0              | 1              |                |
| 0               | 1 | 0              | 0              | 1              | 0              |                |                |
| 1               | 0 | 0              | 1              | 0              | 0              |                |                |
| 1               | 1 | 1              | 0              | 0              | 0              |                |                |

Minitérminos

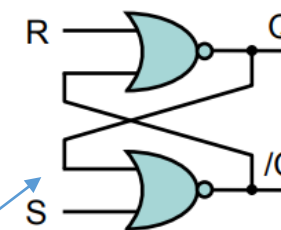
$$D_0 = \bar{A}_1 \cdot \bar{A}_0$$

$$D_1 = \bar{A}_1 \cdot A_0$$

$$D_2 = A_1 \cdot \bar{A}_0$$

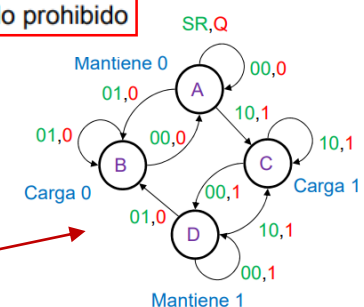
$$D_3 = A_1 \cdot A_0$$

2. Sistemas Digitales **SECUENCIALES = AUTÓMATAS** (demorados o **Dinámicos, Con MEMORIA**): Salidas Digitales dependen del valor instantáneo **actual** de sus Entradas Digitales, Y de la **Secuencia o Evolución (dinámica)** de los valores anteriores, o **Estado** Discreto Interno (Memoria).



| S | R | Q | /Q |                    |
|---|---|---|----|--------------------|
| 0 | 0 | Q | /Q | → Mantener estado  |
| 1 | 0 | 1 | 0  | → Set              |
| 0 | 1 | 0 | 1  | → Reset            |
| 1 | 1 | 0 | 0  | → Estado prohibido |

Diagrama de Estados y Transiciones



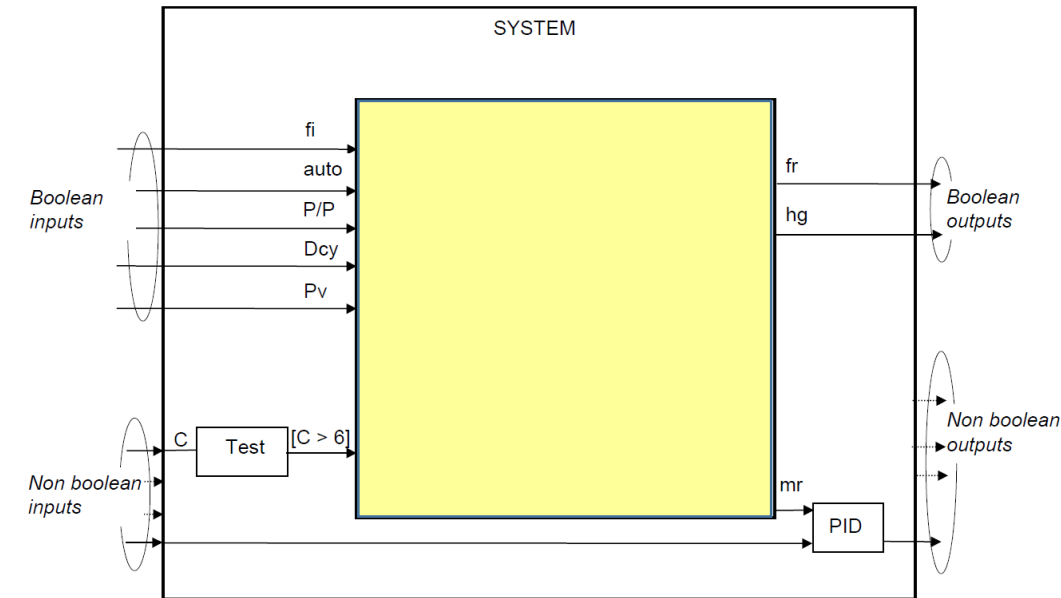
Implementación Física: **Hardware** Electrónico, **Fijo** o **Programable + Software**.  
Modelo: **Máquina Abstracta Secuencial (FSM) = Autómata Finito determinístico.**

# AUTÓMATA de Control: Control Lógico Secuencial

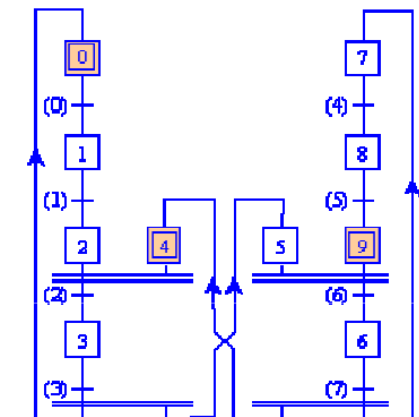
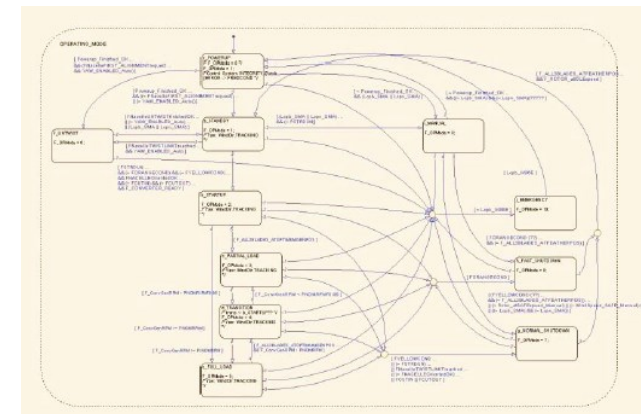
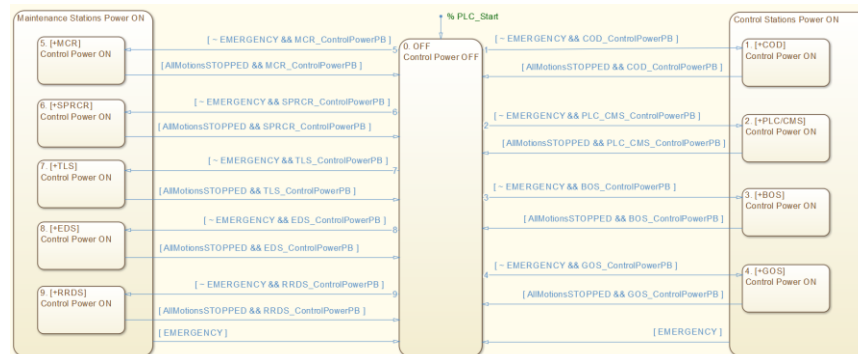
Subsistema de **Control Lógico Secuencial**  
(ej. Nivel de **Control Supervisor**)

Representación Gráfica (Diagrama de Bloques)

- Lógica Secuencial** (Memoria → **Estados** Discretos  
Transiciones entre Estados: Dinámica=evolución)



Especificación de comportamiento lógico secuencial (Estados Discretos activados por Eventos):  
Autómata Finito (FSM) → STATE CHART, GRAFCET



# APLICACIONES: *Automatización y Control de Movimientos*

## Máquinas-Herramienta y Centros de Mecanizado CNC

- **Automatización y Control Secuencial:**
  - Generación de Trayectorias de Referencia.
  - Coordinación de Movimientos entre distintos ejes.
- Integración en **Celdas Flexibles de Fabricación** (FMS).



# APLICACIONES: *Automatización y Control de Movimientos*

## ROBÓTICA Industrial / Fabricación Flexible

- **Automatización y Control Secuencial:**
  - Generación de Trayectorias de Referencia.
  - Coordinación de Movimientos entre distintos ejes.
- Integración en **Celdas Flexibles de Fabricación** (FMS).

**Fully automated battery module production** (KUKA Video 6:18 min, 2020) <https://www.youtube.com/watch?v=JESmMRO5CGY>



# APLICACIONES: *Automatización y Control de Movimientos*

## ROBÓTICA Farmacéutica

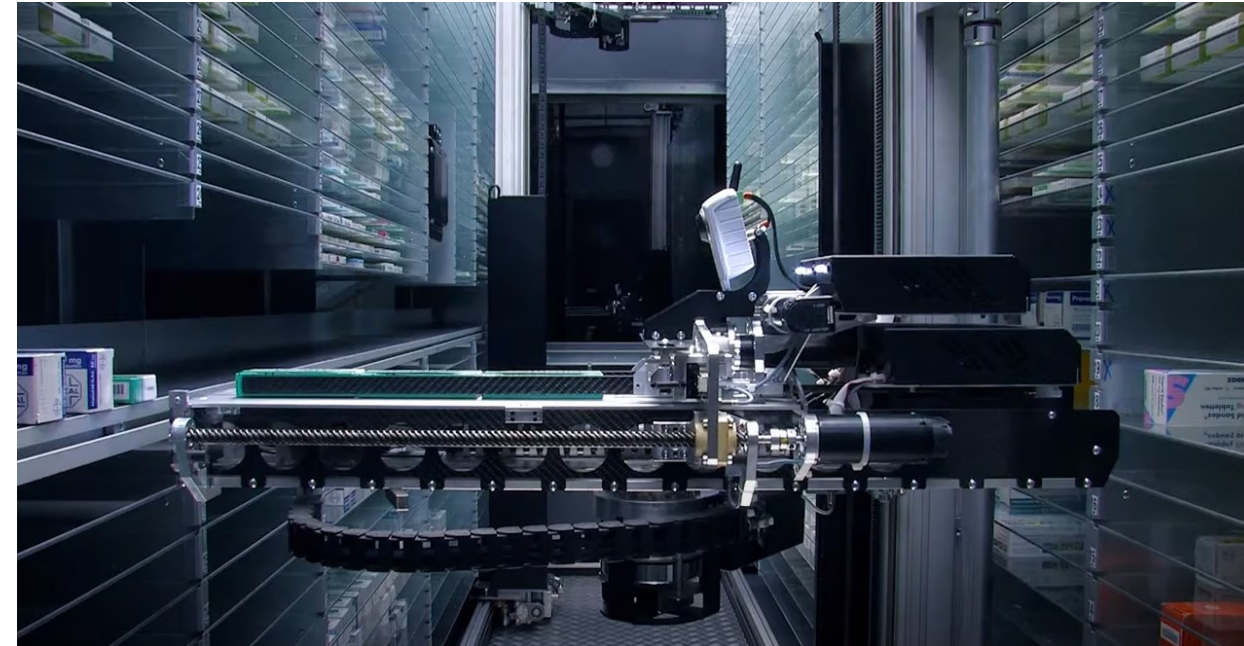
- Automatización, Control Secuencial y Gestión de Stock:

Ej. **BD Rowa Automation Systems** – Vmax Robot dispensador de medicamentos

(ver en **Farmacia Chester**: S.Martín e H.Irigoyen, G.Cruz)

<https://www.youtube.com/watch?v=RwfrZGrtdxA>

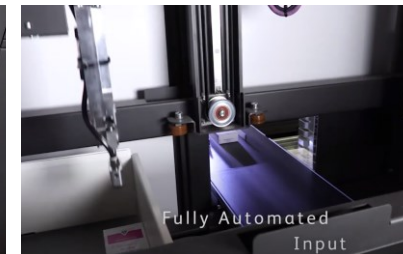
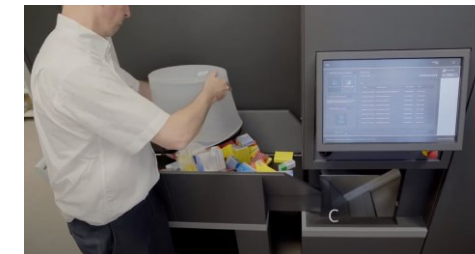
(Video 2:33 ' , 2010-04-22)



Ej. **GOLLMAN Rapid 2022** – Almacén automatizado y dispensador de medicamentos

<https://www.youtube.com/watch?v=mzXhf40UHD8>

(Video 2:33 ' , 2021-12-21)



# APLICACIONES: *Fabricación de Microcircuitos Electrónicos*

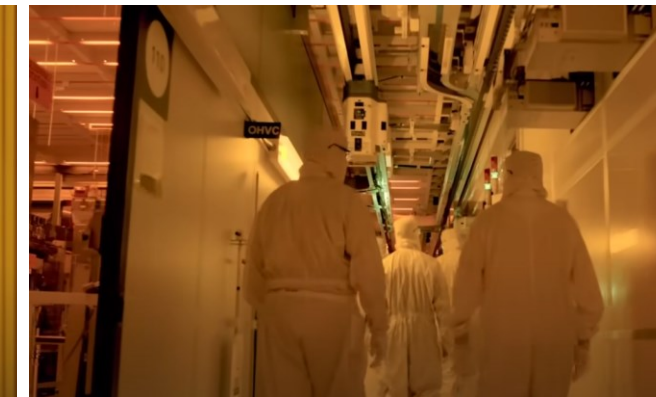
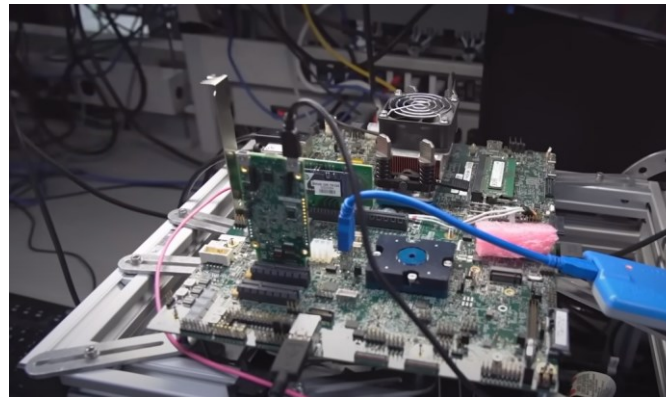
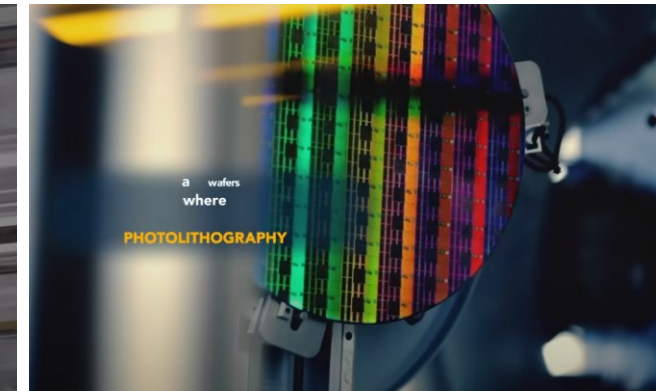
## Dentro de la Fábrica Futurista de Intel en EEUU (Chips or ICs)

Explore the cutting-edge engineering of Intel microelectronics, where every circuit is a testament to the relentless pursuit of innovation and excellence. From conceptualizing innovative chip architectures to the **precision of the semiconductor manufacturing process**, this behind-the-scenes tour offers a glimpse into the evolution of computing power.



## Automatización, Control Secuencial:

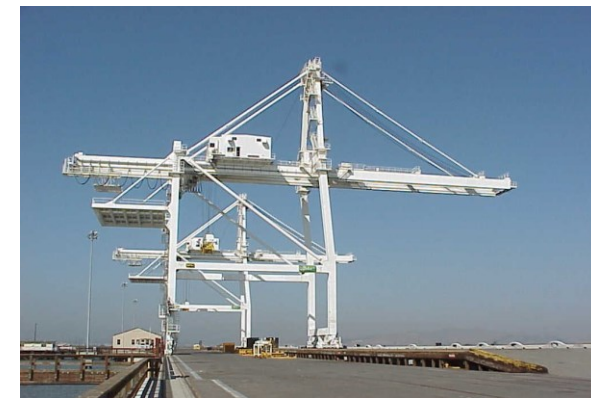
[https://www.youtube.com/watch?v=SrckOCFGBrs&ab\\_channel=FRAME](https://www.youtube.com/watch?v=SrckOCFGBrs&ab_channel=FRAME)  
(Video 17:01 ' , 2024-02-20)



# APLICACIONES: *Automatización y Control de Movimientos*

## Sistemas de Izaje y Manipulación de Cargas (GRÚAS)

- Operación Semi-Automática con control de Movimientos coordinados.
- Control Manual c/comandos del operador procesados por PLC.

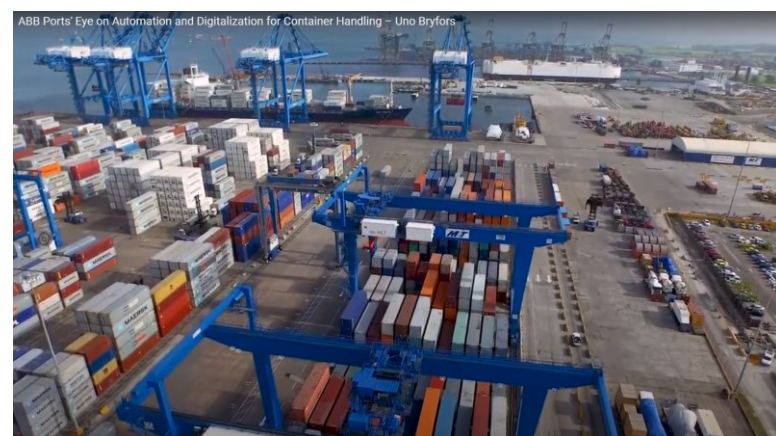
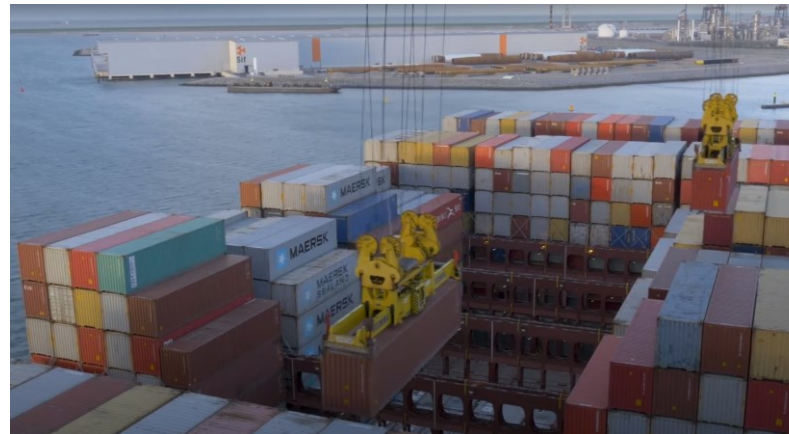


# APLICACIONES: *Automatización y Control de Movimientos*

## ABB: Automation & Digitalization for Container Handling

(Video ABB 2018 6:34 min.) **Operación Automática con Supervisión remota - AGVs**

[https://www.youtube.com/watch?v=sKtae0UA1Ok&ab\\_channel=PortTechnology](https://www.youtube.com/watch?v=sKtae0UA1Ok&ab_channel=PortTechnology)



# APLICACIONES: *Automatización y Control de Transporte*

## Trenes de alta velocidad, Aviones, Aeroespacial

- Sistemas de control **embebidos y distribuidos**
- **Drive-by-wire**
- **Comunicaciones digitales**
- etc.



# APLICACIONES: *Automatización y Control Distribuido*

## Industrias de PROCESOS Continuos:

Petróleo y gas, petroquímica, siderurgia y metales, pulpa y papel, alimentos, farmacéuticas, etc.):

- **Procesos por Lotes**
- Coordinación de operaciones
- **Supervisión (SCADA)**
- **Control Distribuido y Jerárquico**
- Etc.



# APLICACIONES: Automatización Agropecuaria

## Ordeñe Automatizado voluntario de Vacas



Video DeLaval Robot de Ordeñe DeLaval VMS V300 (5,25 min.)

[https://www.youtube.com/watch?v=MoNMF\\_o\\_hmg&ab\\_channel=DeLavalEspa%C3%B1ayPortugal](https://www.youtube.com/watch?v=MoNMF_o_hmg&ab_channel=DeLavalEspa%C3%B1ayPortugal)

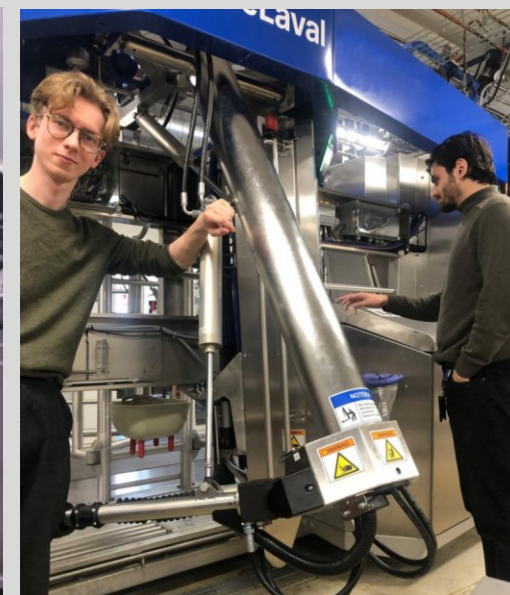
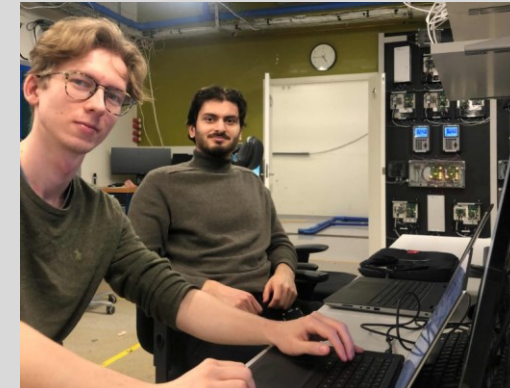
<https://corporate.delaval.com/>

<https://www.delaval.com/es-ar/explore/ordeno/delaval-vms-series/>

**Aplicación de Estudiantes de Ingeniería Mecatrónica de KTH (Suecia)**

**Ordeñe robotizado**

<https://corporate.delaval.com/2024/07/students-from-kth-helping-to-shape-the-future-of-robotic-milking/>



# APLICACIONES: Automatización y Control de Generación

## GENERACIÓN de Energía Eléctrica de Fuentes Renovables

### Turbinas Eólicas

- Generación a **velocidad/frecuencia variable** → Conversión a frecuencia de red
- **Control** por **torque** (modo frenado regenerativo en Generador).
- Control por **pitch** (ángulo de palas), **yaw** (orientación de góndola), etc..



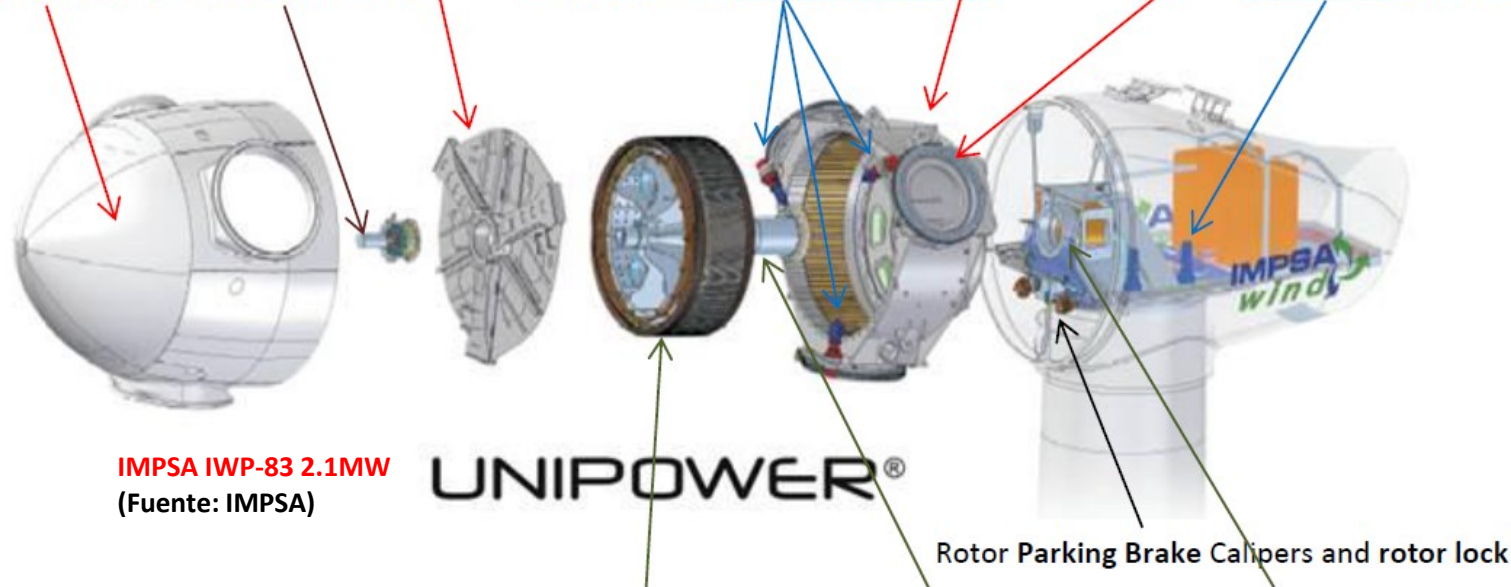
# APLICACIONES: Automatización y Control de Generación

## GENERACIÓN de Energía Eléctrica de Fuentes Renovables

### Turbinas Eólicas

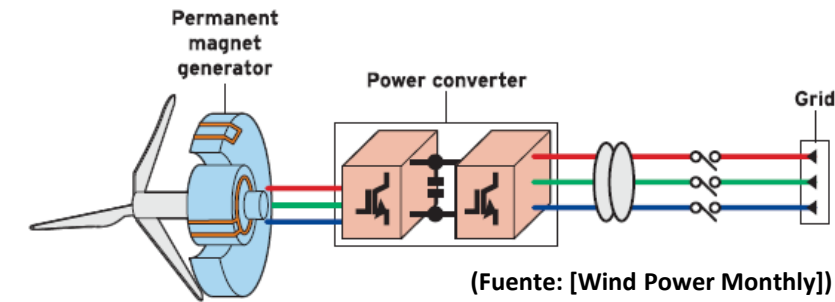
Video IMPSA IWP-100 (7 min.) ➔

ROTATING Parts: ROTOR Cover w/ bearing PMSG Generator Outer ROTOR with BLADE Bearings attached directly  
Rotor Fairing Encoder/Slip Ring 3 independent PITCH servo-drives YAW drive (4 motors)



STATIONARY Parts: PMSG Generator Inner STATOR w/ Stationary SHAFT, fixed to Nacelle MAINFRAME

**IMPSA Unipower® direct-drive concept and main components**



# APLICACIÓN: *Control Híbrido Multinivel de Turbina Eólica*

## Turbina Eólica: Descripción general

### Componentes (HAWT)

- **ROTOR:** (Palas / Hub)
  - **PALAS:**
    - perfil aerodinámico, mat. compuestos ej. fibra vidrio
    - 2 palas (ej. NREL-CART) vs. **3 palas** (uso gralizado.)
- **ESTRUCTURA**
  - **GÓNDOLA:** soporta rotor, se orienta en dir. viento
  - **TORRE:** Tubular de acero, hormigón, reticulada, etc.
  - **FUNDACIÓN:** Hormigón
- **MECANISMOS / ACCIONAMIENTOS**
  - YAW
  - PITCH
  - TREN TRANSM. – GENERADOR
- **SISTEMA ELÉCTRICO**
  - CONVERTIDOR
  - TRANSFORMADOR / SWITCHGEAR (aparatos de Maniobra)
  - Servicios Auxiliares y Alimentación de Respaldo p/Emergencias
- **SISTEMA DE CONTROL Y PROTECCIÓN**

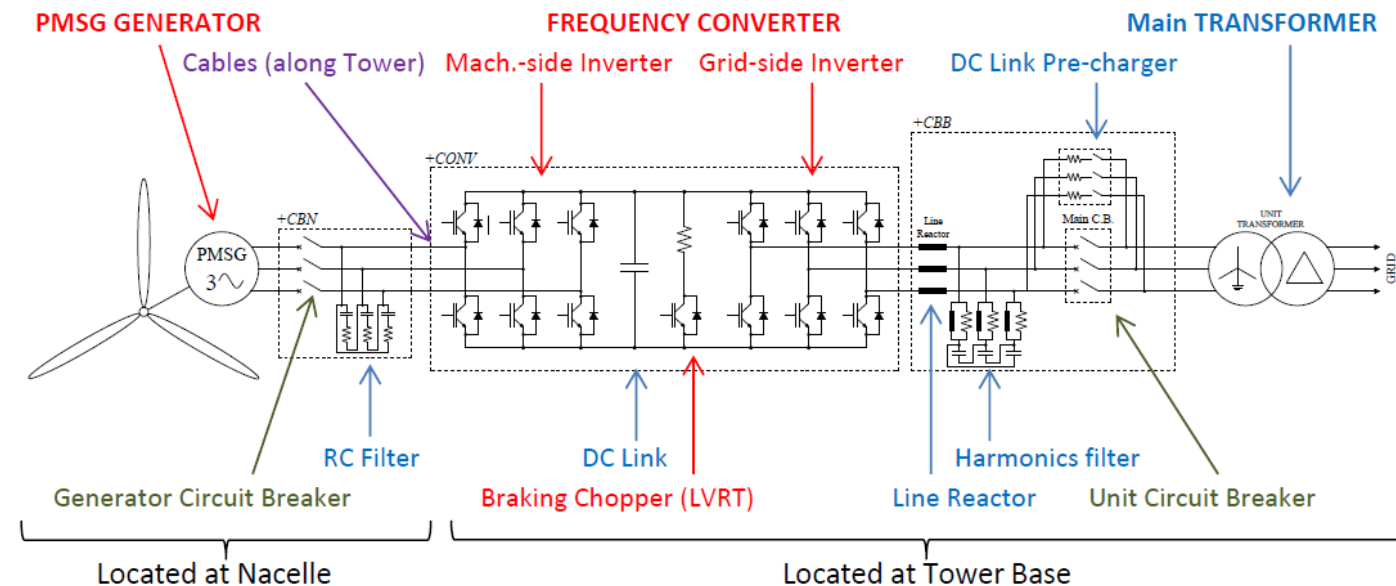


(Fuente: IMPSA)

# APLICACIÓN: *Control Híbrido Multinivel de Turbina Eólica*

## Turbina Eólica:

**Subsistema de Potencia:**  
Generación y Conversión de Energía Eléctrica



## Subsistema de Control y Accionamiento:

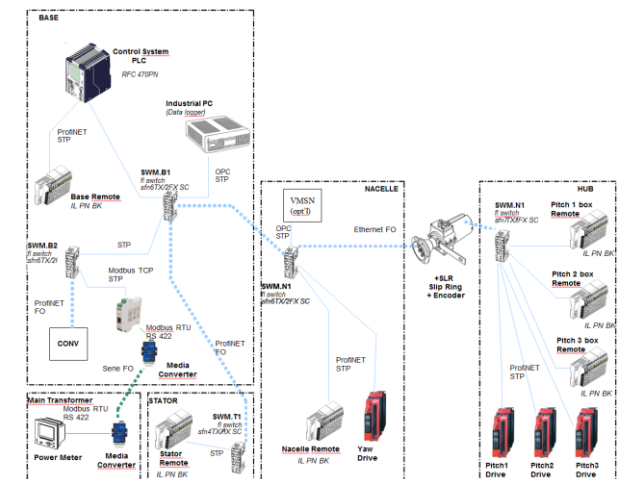
**Control Híbrido Multinivel, Hardware + Software**

**Accionamientos:** Pitch, Yaw, Generación y Conversión de Energía Eléctrica.

**Control:** Regulatorio y Supervisor. Diagnóstico de Fallas. Optimización.

**Protección:** Seguridad ante Emergencias (Safety) y ante Sabotajes (Security)

(Fuente: IMPSA)



# APLICACIÓN: *Control Híbrido Multinivel de Turbina Eólica*

## Turbina Eólica:

### Subsistema de Control y Accionamiento:

- Arquitectura de HARDWARE
- Drives
- Controladores
- Red de Comunicación industrial

### Control Híbrido Multinivel, Hardware + Software

#### Accionamientos:

Pitch, Yaw, Generación y Conversión de Energía Eléctrica.

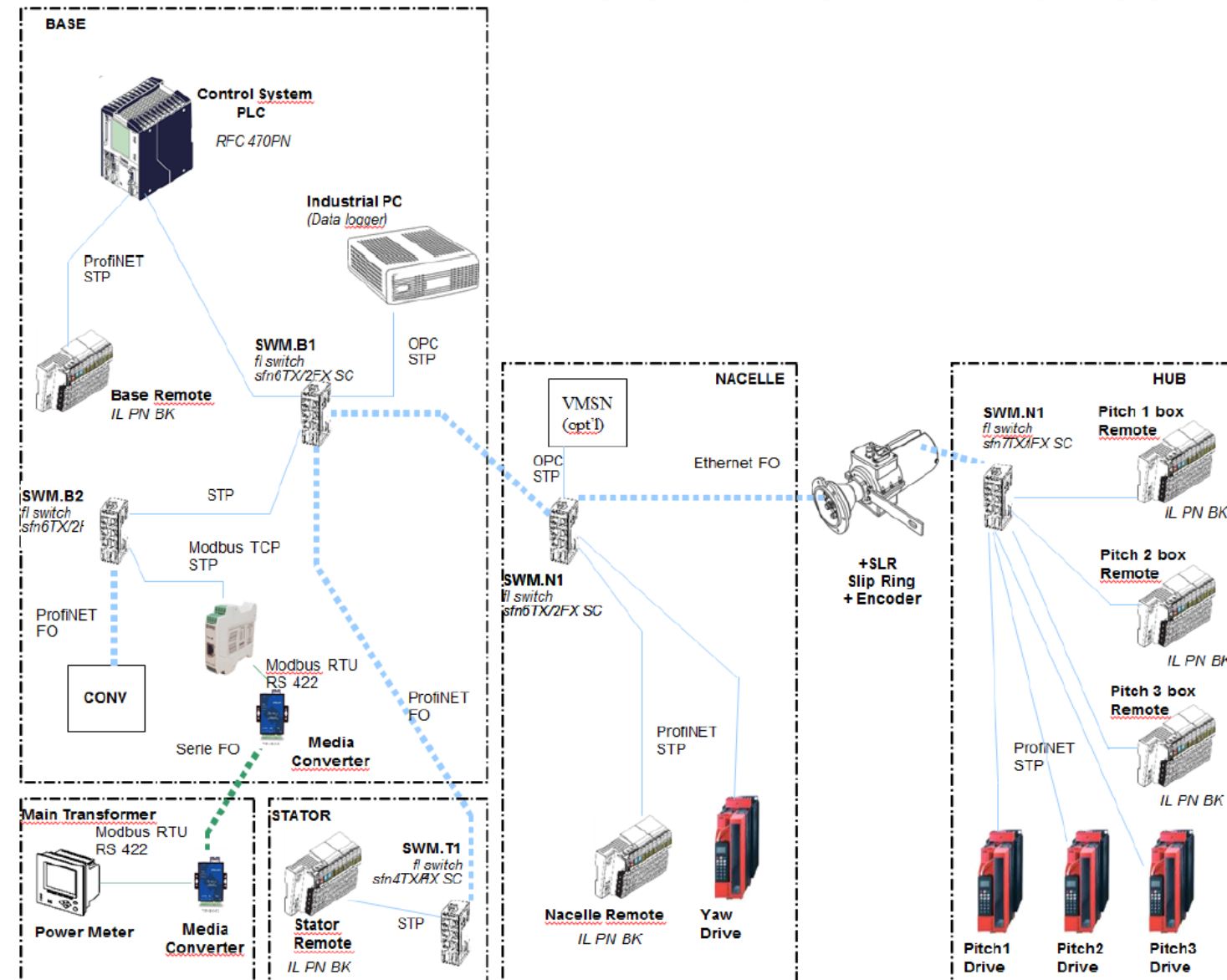
#### Control:

Regulatorio y Supervisor. Diagnóstico de Fallas. Optimización.

#### Protección:

Seguridad ante Emergencias (Safety) y ante Sabotajes (Security)

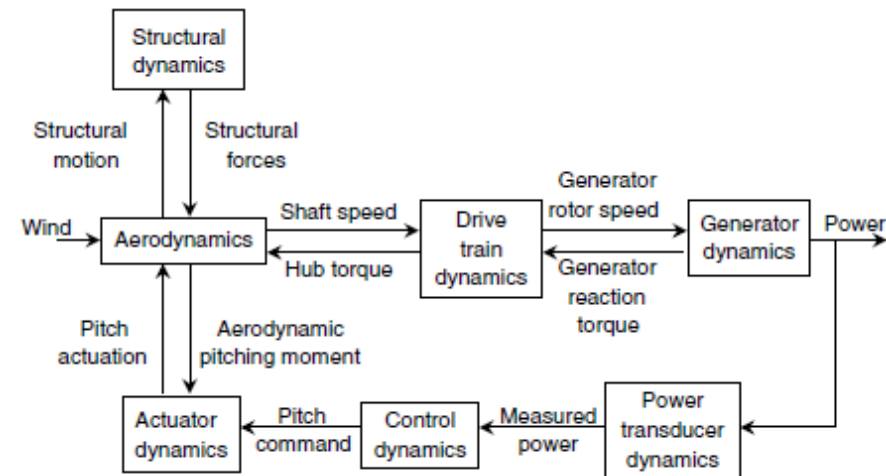
(Fuente: IMPSA)



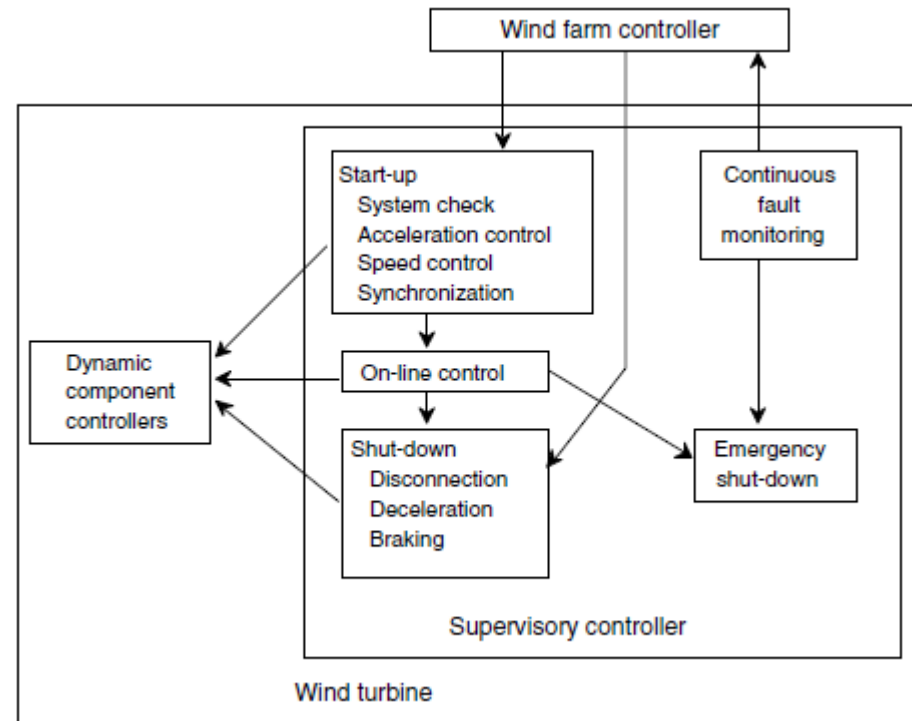
# APLICACIÓN: *Control Híbrido Multinivel de Turbina Eólica*

## Turbinas Eólicas: Conceptos de Modelado y Control

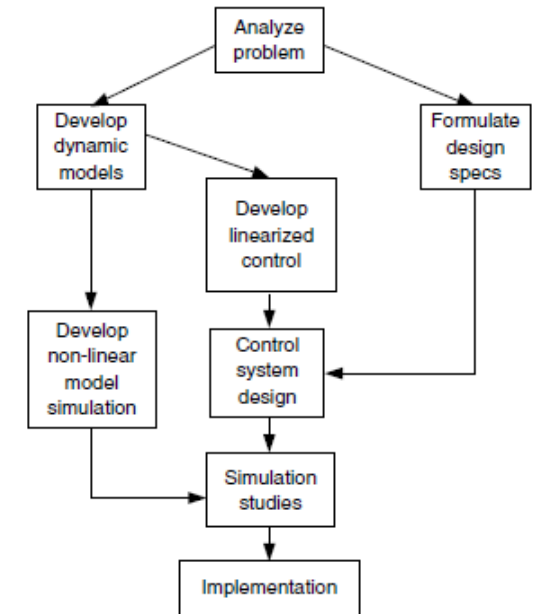
### Bloques funcionales



### Control Supervisor y Regulatorio



### Etapas de Análisis y Diseño



Fuente: [1] Manwell et al.

### Referencias:

- [1] Manwell et al., **Wind Energy Explained – Theory, Design and Application**. 2<sup>nd</sup> Ed., Wiley 2009.
- [2] Burton et al., **Wind Energy Handbook**. 2<sup>nd</sup> Ed., Wiley 2011.

# APLICACIÓN: *Control Híbrido Multinivel de Turbina Eólica*

## Sistema de Control y Protección de TE

### Arquitectura básica y Componentes

#### Sensores: (→ información de **Status**, **Feedback**)

- viento (anemómetro/veleta),
- velocidad de rotor,
- potencia eléctrica,
- posición de pitch de palas, llaves de fin de carrera (limit switches),
- vibración, temperaturas, nivel de aceite, presión,
- llaves y pulsadores de operador, etc..

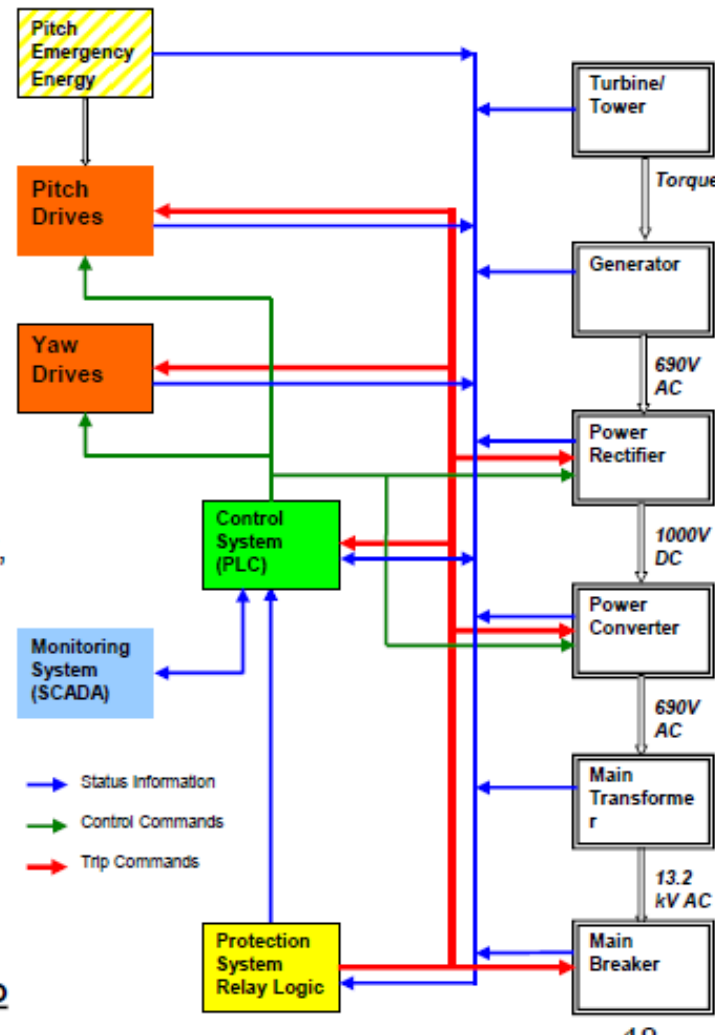
#### Actuadores: (→ acción de **Control**, **Comando**)

- pitch (motores eléctricos/hidráulicos), motores de yaw,
- controlador de torque de generador,
- contactores y equipos de maniobra, frenos, etc.

Controlador: (procesamiento: **Lógica y Cálculo**)  
(hardware + software, alta performance)  
procesa **señales de entrada** (fbk ← **sensores**) y genera **señales de salida** (comandos → **actuadores**)  
→ **Funciones de control** (necesarias para operar la turbina en condiciones normales).

#### Sistema de Seguridad o Protección:

(cableado, alta confiabilidad)  
reemplaza al controlador normal en caso de **falla o problema serio**, para llevar la TE a un estado **seguro**  
→ **Funciones de protección o seguridad**.



# APLICACIÓN: *Control Híbrido Multinivel de Turbina Eólica*

## Funciones de CONTROL

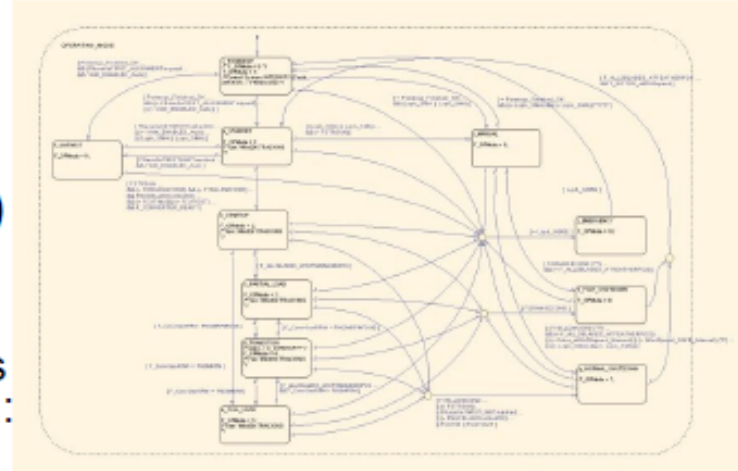
### Nivel 1. Control SUPERVISOR:

**Sist. Digital Secuencial:** Autómata o Máq. Estados Finitos (Eventos Discretos)

Control de Secuencia de Operación global de TE  
→ conmutación (transiciones) y control de etapas internas para distintos **MODOS DE OPERACIÓN**:

- **Espera (Stand-by):** TE disponible para arrancar (si lo permiten las condiciones externas);
- **Arranque (Start-up);**
- **Producción de energía** (Carga Parcial, Transición, Carga Plena);
- **Frenado o Detención (Shutdown):** normal, rápida, de emergencia (sist. protección), desde varios estados (ej. por viento extremo, fallas puntuales detectadas, etc.);
- **Emergencia:** detenido con Fallas graves, desde cualquier estado;
- **Manual / Mantenimiento**, etc.

Detección y Diagnóstico de **FALLAS** no críticas.  
Monitoreo de EVENTOS externos y STATUS.



# APLICACIÓN: *Control Híbrido Multinivel de Turbina Eólica*

## Funciones de CONTROL

### Nivel 2. Control REGULATORIO:

Sist. Continuo [digital]:  
*Lazo Cerrado o Realimentado*

Ajuste automático del estado operacional de TE para cada Modo de Operación → mantenerla en Curva Característica de Operación (**Estrategia**) predefinida.

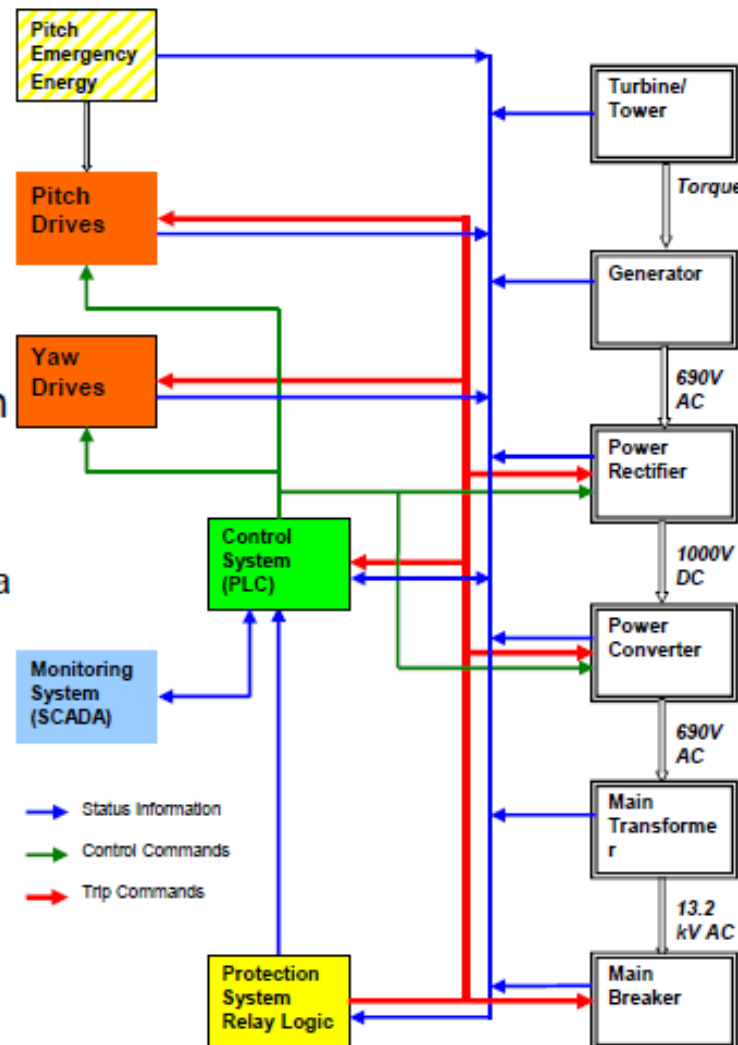
#### Lazos de Control:

- **Pitch de Palas** → regular potencia de salida (potencia nominal) para vientos superiores al nominal; seguir rampa de aceleración/frenado durante arranque o paradas;
- **Torque del Generador** → regular la velocidad de rotación (TE de velocidad variable);

**Nota:** requieren respuestas dinámicas rápidas para buen desempeño (seguim. de consignas y rechazo de perturbaciones) → diseñar cuidadosamente.

- **Orientación de Góndola (Yaw)** → minimizar el error de seguimiento de dirección de viento.

**Nota:** respuesta lenta (para reducir cargas giroscópicas) → diseño menos crítico.



# APLICACIÓN: *Control Híbrido Multinivel de Turbina Eólica*

## Funciones de **SEGURIDAD**

### Nivel 0. Protección / Seguridad

#### Sist. de *Respaldo*

(distinto e independiente del controlador normal)

TE → **condición segura** (detención y desenergización) ante evento de **FALLA** o **problema potencialmente serio**:

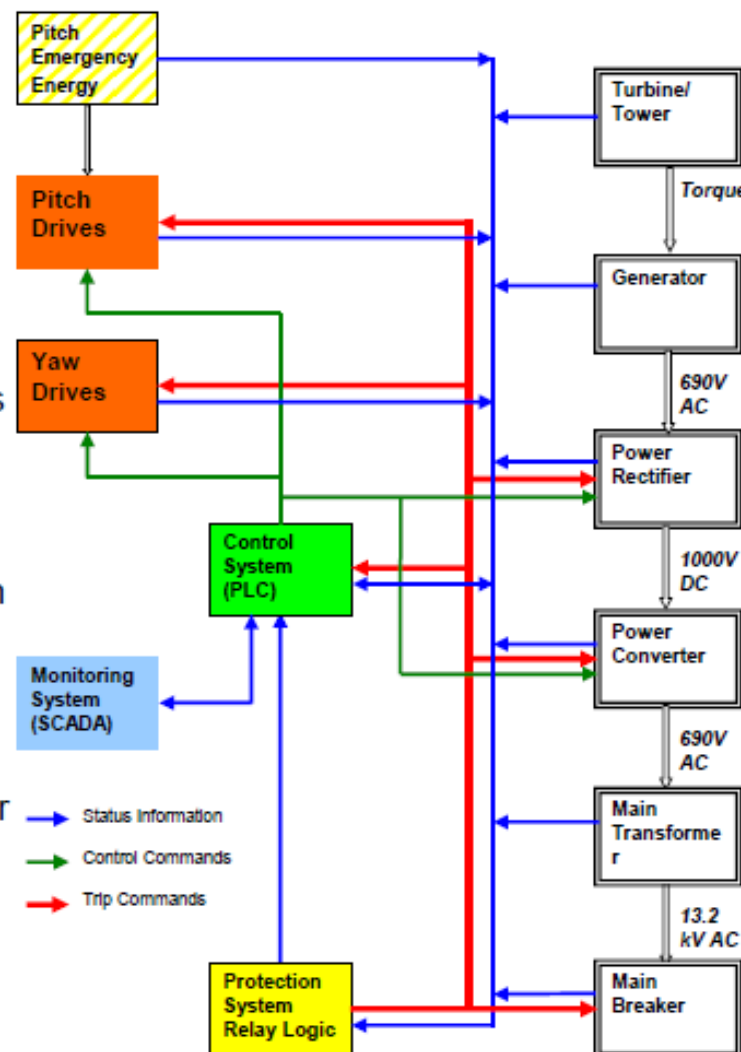
- Sist. de Control no puede hacerse cargo (se superan los límites de operación normal), ó
- Activación por pulsador de emergencia.

Altamente confiable y con falla segura (**Fail-Safe**).  
Generalmente es circuito **cableado** c/contactos a relé **N.C.** en serie (o usando sistemas instrumentados de seguridad).

**Activación (apertura)** de cualquier contacto: dispara la cadena → acciones de falla segura (palas a bandera, desenergización, etc.).

#### Causas de disparo:

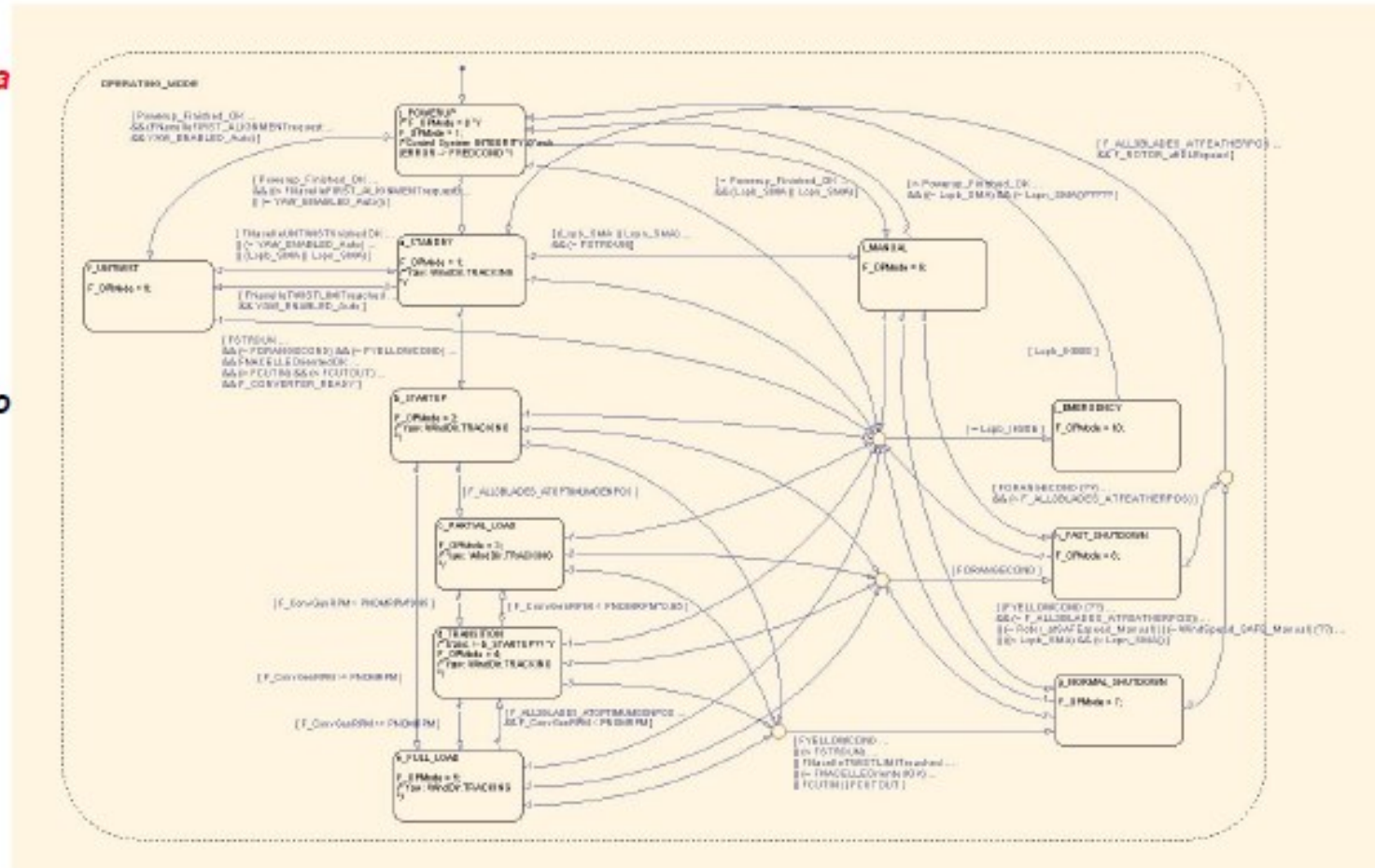
- **SOBREVELOCIDAD** de rotor: límite de emergencia superior al límite de sobrevelocidad por software;
- **VIBRACIÓN** excesiva: ej. fallas estructurales, etc.
- Temporizador **WATCHDOG** del Controlador activado: anuncia problemas en el controlador.
- **PULSADOR DE EMERGENCIA** (Parada de emergencia);
- Otras fallas (Convertidor / Eléctricas, etc.).



# APLICACIÓN: *Control Híbrido Multinivel de Turbina Eólica*

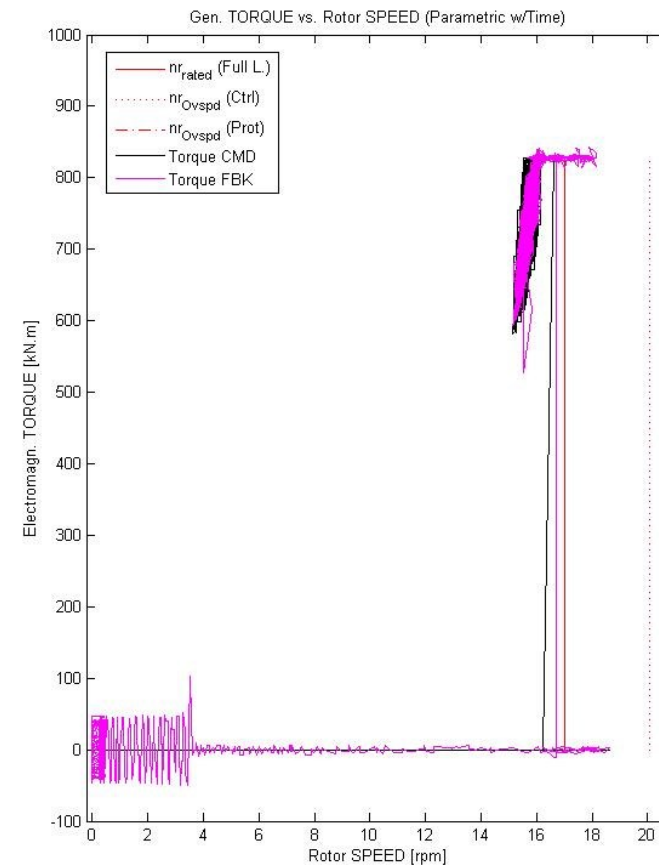
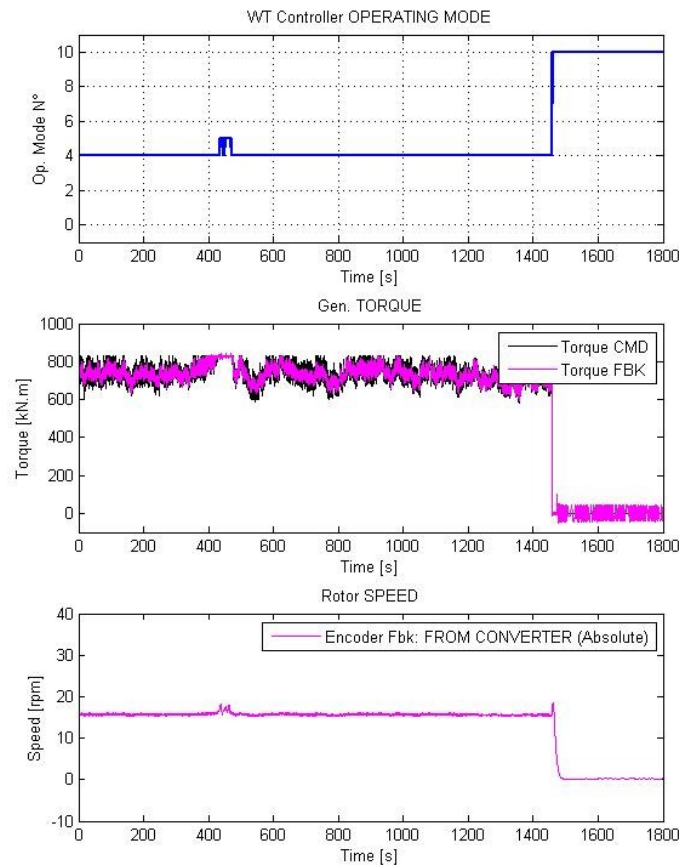
## MODOS de Operación del Sist. Control

- **Espera**
- **Arranque**
- **Producción de energía**
  1. **Carga Parcial**
  2. **Plena Carga**
  3. **Transición**
- **Paradas**
  - **Normal**
  - **Emergencia**
- **Manual/Mantenimiento**
- **Destorzado.**



# APLICACIÓN: *Control Híbrido Multinivel de Turbina Eólica*

## Ejemplo de Registración de Datos durante Falla



# Control Discreto (ON/OFF) Marcha/Parada de Motor ACIM

**Accionamiento DOL (= Direct On Line):** Conexión Directa a Línea trifásica 380 Vac 50 Hz mediante **1 CONTACTOR**. Operación en **1° y 2° Cuadrantes**: **FORWARD Motoring/Braking** (Motorización/Frenado en Sentido giro DIRECTO).

**[Frenado Regenerativo: SOLO SI LA CARGA LO ARRASTRA, RESBALAMIENTO NEGATIVO]**

**ESPECIFICACIÓN de Requisitos de Operación:** ([NC] = Normally Closed / [NO] = Normally Open)

- Condición **SEGURA**: Motor **Detenido (desconectar)** ante **Alarma, Sobrecarga, Falla, o Emergencia**.
- Relé **TÉRMICO** OL=**Overload** [NC] se activa (OFF) → **Desconecta** Motor [Prioridad 1: **PROTECCIÓN**].
- Pulsador **PARADA Normal** [NC] se activa (OFF) → **Desconecta** Motor [Prioridad 2: **MANIOBRA**].
- Pulsador **MARCHA** [NO] se activa (ON) → **Conecta** Motor [Prioridad 3: **MANIOBRA**], si se cumple que (**Lógica Booleana**): Motor desconectado **Y NO** activados (Relé **Térmico** **O** Pulsador de **Parada**).

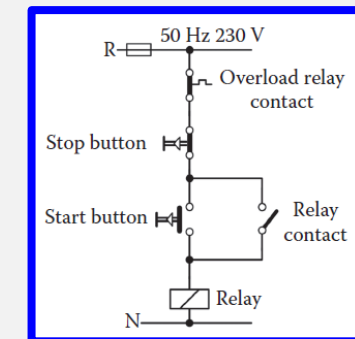
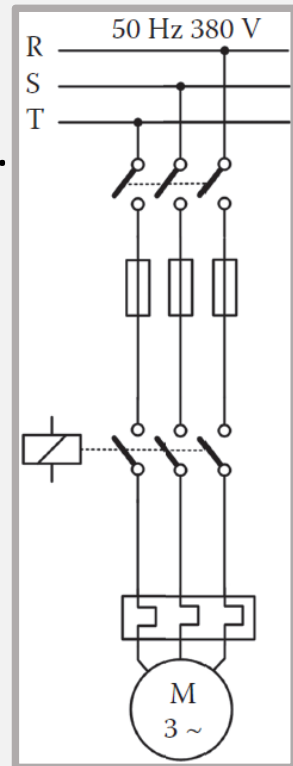
**Entradas/Salidas (I/O= Inputs/Outputs) del Controlador Discreto (Lógico Secuencial):**

- 1 Salida Discreta (**DO**): **CONTACTOR** de Conexión (**ON**) / Desconexión (**OFF**) de Línea al Motor.
- 3 ó 4 Entradas Discretas (**DI**): **Relé TÉRMICO** / Pulsador **MARCHA** / Pulsador **PARADA Normal** / Contacto Auxiliar de **CONTACTOR** (Feedback discreto externo) vs Bit Interno para **enclavamiento**.

→ **Hacer:**

1. Implementar en papel, en formas convencionales conocidas:
  - **Lógica cableada (Hardwired)** → Lógica de contactos (Ladder Diagram)
  - Bloques de Funciones Lógicas (Compuertas Booleanas)
  - Pseudocódigo informático de alto nivel (similar a Pascal, C, etc.)
2. **Tabla de Verdad** y **Diagrama Lógico Temporal**
3. **Diagrama de Estados Discretos y Transiciones**

**Modelo DISCRETO:** NO considera Dinámica Continua de Motor



# Control Discreto Marcha FWD-REV/Parada de Motor ACIM

**Accionamiento DOL (= Direct On Line) con INVERSIÓN:** Conexión Directa/Inversa a Línea con **2 CONTACTORES**.  
**Operación en 4 Cuadrantes:** FORWARD/REVERSE Motoring/Braking (Motorización/Frenado en ambos Sentidos).  
**[Frenado Regenerativo: SOLO SI LA CARGA LO ARRASTRA, RESBALAMIENTO NEGATIVO]**

## ESPECIFICACIÓN de Requisitos de Operación:

- Condición **SEGURA**: Motor **Detenido** (desconectar ante **Falla, Emerg.**).
- Relé **TÉRMICO** [NC] se activa (OFF) → **Desconecta** Motor [Prioridad 1].
- Pulsador **PARADA** [NC] se activa (OFF) → **Desconecta** Motor [Prioridad 2].
- Pulsadores **FWD/REV** [NO] se activan (ON) → **Conectan** Motor FWD **Ó** REV **mutuamente excluyentes! (XOR)** [Prioridad 3], si se cumple que (**Lógica Booleana**): Motor desconectado **Y NO** activados (Relé Térmico **O** Pulsador de Parada).

## Entradas/Salidas (I/O) del Controlador Discreto (Lógico Secuencial):

- 2 Salidas Discretas (**DO**): **CONTACTORES** FWD/REV de Conexión al Motor.
- 4 **ó** 6 Entradas Discretas (**DI**): **Relé TÉRMICO** / **Pulsadores FWD/REV** / **Pulsador PARADA Normal** / **Contactos Auxiliares de CONTACTORES** (Feedback discretos externos) vs Bits Internos para **enclavamientos**.

## → Hacer:

- **Lógica cableada (Hardwired)** → Lógica de contactos (Ladder Diagram)
- Bloques de Funciones Lógicas (Compuertas Booleanas)
- Pseudocódigo informático de alto nivel (similar a Pascal, C, etc.)

## 2. Tabla de Verdad y Diagrama Lógico Temporal

## 3. Diagrama de Estados Discretos y Transiciones

**Modelo DISCRETO:** NO considera Dinámica Continua de Motor

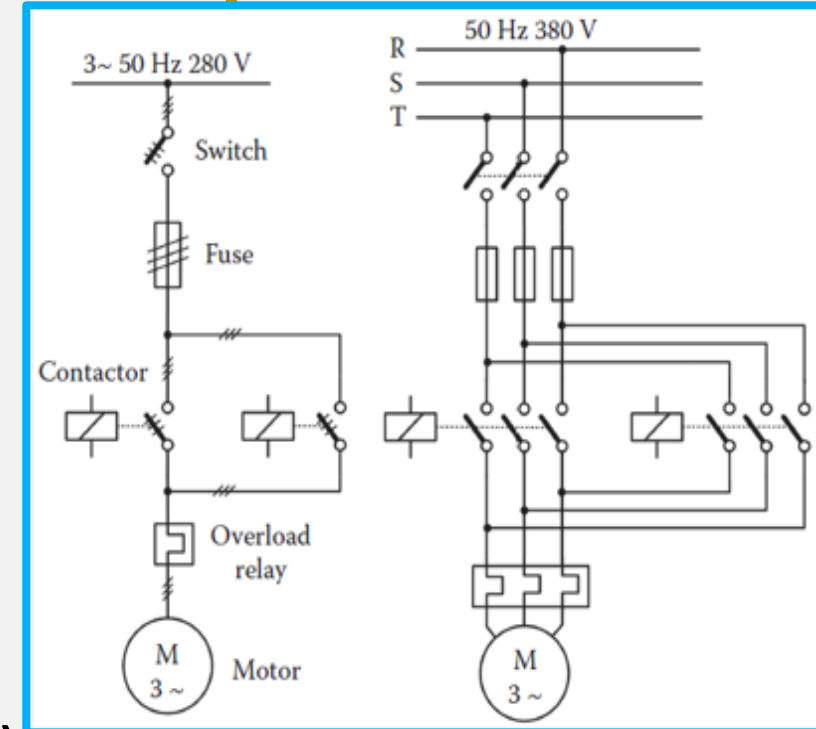


Diagrama Unifilar → Trifilar



# Otra APLICACIÓN Interesante:

## *Beer Bartender Robot!*

**Coordinación y sincronización de movimientos**

**Trabajo colaborativo: Robot – Mesa giratoria**

**Ingeniería bien aplicada! 😊**

Video 1 min. aprox.

LA INGENIERÍA BIEN  
APLICADA





# Unidad 1

## Autómatas, Sistemas Discretos y Sistemas Híbridos

---

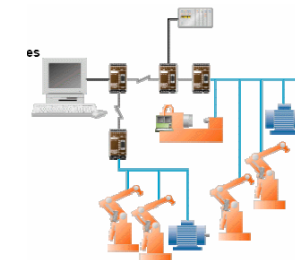


# AUTÓMATAS y CONTROL DISCRETO

## 1. *Autómatas, Sistemas Discretos y Sistemas Híbridos*

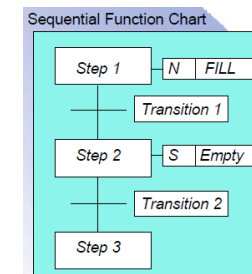
### A. Concepto y Aplicaciones

- Automatización y Control Industrial: Concepto y esquema general. Componentes y estructura.
- Aplicaciones Mecatrónicas → Control automático secuencial + control de tiempo discreto.



### B. Fundamentos de Autómatas Secuenciales de Control

- Sistemas secuenciales: estados discretos activados por eventos. Conceptos.
- Autómatas finitos. Modelado y representación. FSM. STATECHARTS.
- Simulación. STATEFLOW.

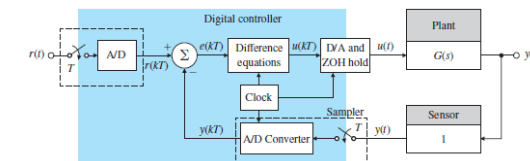


### C. Petri Nets (ISO/IEC 15909). GRAFCET (IEC 60848)

- Metodologías de representación y diseño. Redes de Petri.
- Lenguaje GRAFCET. Estructuras Jerarquizadas.

### D. Sistemas Híbridos. Práctica de Modelado y Simulación

- Modelado y Simulación en STATEFLOW. Representación en GRAFCET.

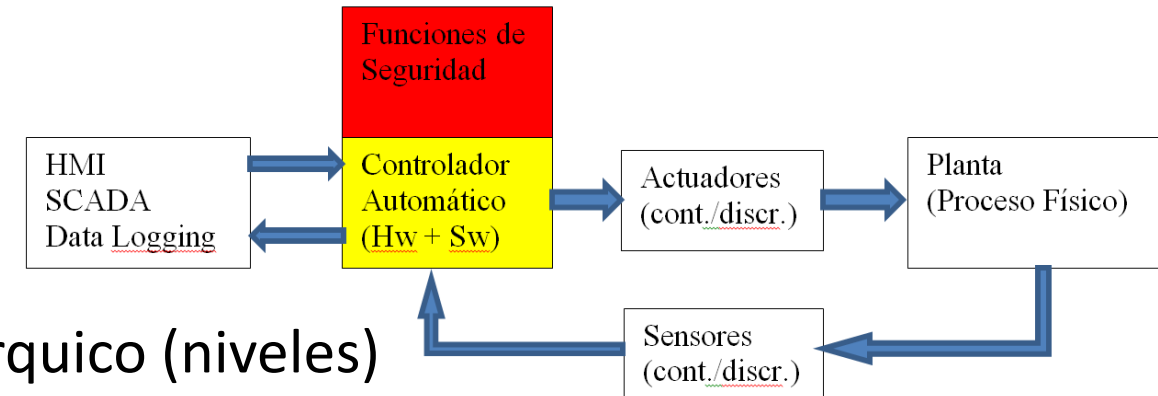


# Sistemas de Automatización y Control Industrial

## Estructura Conceptual

### Planta o Proceso Físico a Controlar:

- Sistema Dinámico **Multivariable** (MV o MIMO)
- **Híbrido**: Estados Continuos + Estados Discretos
- **Complejo**: Descentralizado vs. Distribuido y Jerárquico (niveles)



### Funciones de **CONTROL**: (Regulatorio Continuo + Supervisor Discreto)

- Comandos, Monitoreo y Supervisión, Registro de datos y alarmas.
- Salidas Medidas de Planta (+Estimación) → **Señales de Retroalimentación (Feedback)**.
- **Señales de Control** → Actuación sobre Entradas Manipuladas de Planta.

### Funciones de **SEGURIDAD**: (**PROTECCIÓN** x Fallas: **Safety** / Sabotaje: **Security**)

- Vigilancia interna (Watchdog) / Redundancia I/O.
- Sistema separado de Emergencia cableado (relé maestro de Seguridad) ó Sistema de Seguridad Intrínseca (Hardware + Software).

# Sistemas Dinámicos (SD): Conceptos

**SISTEMA**: Conjunto de **Elementos interrelacionados**, considerado como **un Todo** en contexto definido, y **separado** de su **Entorno o Ambiente**, intercambio de variables o señales.

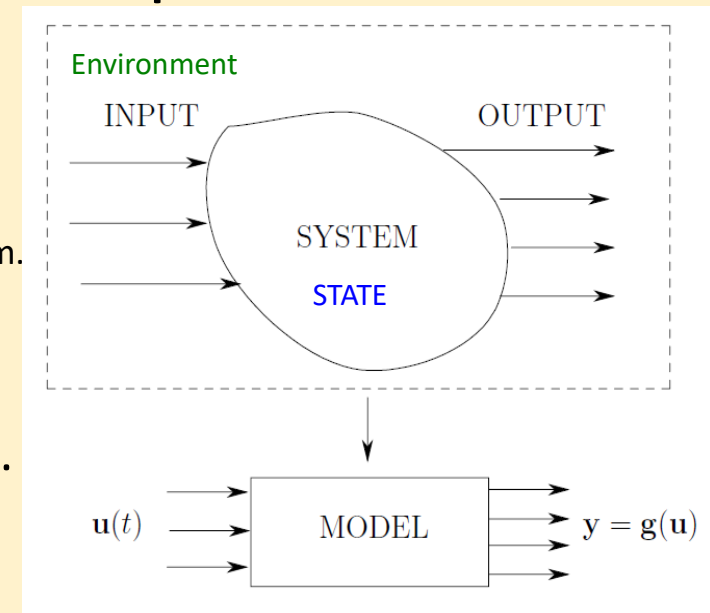
**SYSTEM**: Set of **interrelated Elements** considered as a **Whole** in a defined context and **separated** from their **Environment**.

- A system is generally defined with the view of **achieving a given objective**, for example by **performing a definite function**.
- Elements of a system may be natural or man-made material objects, as well as modes of thinking and the results thereof (for example forms of organization, mathematical methods, programming languages).
- The system is considered to be **separated** from the **environment** and the other external systems **by an imaginary surface**, through which pass the links (i.e. signals or variables) between them and the considered system.
- The term "system" should be qualified when it is not clear from the context to what it refers, for example control system, calorimetric system, system of units, transmission system.

**MODEL**: mathematical or physical **Representation** of a **System** or a process, based with **sufficient precision** upon **known laws**, **identification** or specified suppositions.

Fuente: **IEC 60050 Electropedia: The World's Online Electrotechnical Vocabulary**

<https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=351-42-08/>  
<https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=351-42-26/>



**SISTEMA DINÁMICO FÍSICO Continuo (CDS)**: Elementos **físicos** (naturales o contruidos) que **almacenan**, disipan e intercambian **ENERGÍA**.

- Las **variables** involucradas: entradas, salidas, estado, son **CONTINUAS** (Reales,  $\mathbb{R}$ ). Evolucionan en tiempo (Time-Driven).

**SISTEMA COMPUTACIONAL, LÓGICO O SIMBÓLICO (DEDS)**: Elementos **lógicos o simbólicos**, **almacenan** e intercambian **INFORMACIÓN**.

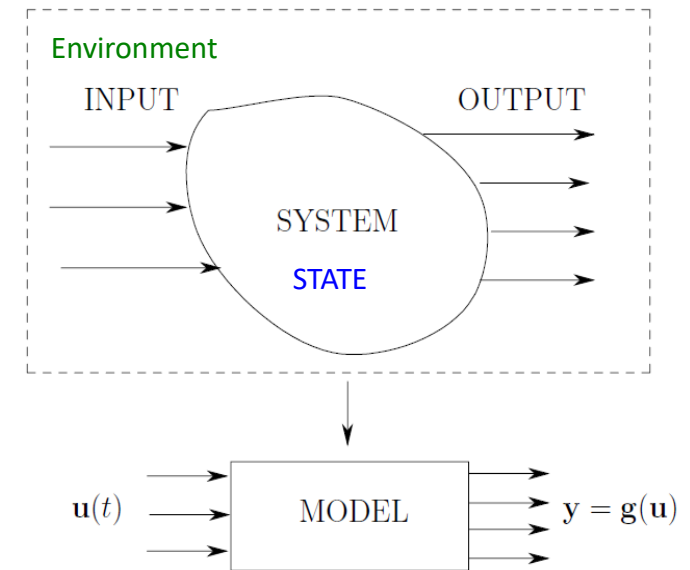
- Las **variables** involucradas: entradas, salidas, estado, son **DISCRETAS** (Naturales,  $\mathbb{N}$ ). Evolucionan activadas por Eventos (Event-Driven).

**SISTEMA DINÁMICO HÍBRIDO (HDS)**: Conjunción de Elementos **físicos y lógicos** que **almacenan** e intercambian **ENERGÍA + INFORMACIÓN**.

# Sistemas Dinámicos (SD): Modelos

## *Sistema DINÁMICO:*

Proceso cuyas variables de **estado** evolucionan en el tiempo en forma demorada = retardos (almacenamiento de energía, **memoria**) ante **excitaciones** o **entradas** dadas.



## *Modelado Dinámico:*

Incluir todos los aspectos del sistema que son **relevantes (dominantes)** para el análisis deseado, pero no incluir aspectos innecesarios.

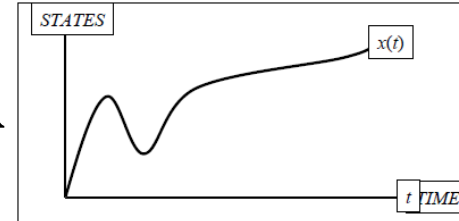
→ **MODELO Dinámico**: Representación (matemática) **Aproximada** del SD que permite **analizar y simular** los aspectos de interés del **comportamiento** del SD en su **evolución temporal**.

# Sistemas Dinámicos (SD): Clasificación

**SD FÍSICOS: Estados CONTINUOS, activados x TIEMPO (Time-driven)** “**FLOW Variables**”

Leyes físicas, fuerzas de la naturaleza (macroscópico).

$$\dot{x} = f(x, t)$$



**SDCTC** – Modelos de Estados Continuos en Tiempo Continuo  $t \in \mathbb{R}$

- Almacenamientos e intercambios de **Energía** (ej: circuito eléctrico RLC, motor,..);
- Variables (vectores) de **Estado**  $x(t)$ , **Entrada**  $u(t)$  y **Salida**  $y(t)$ : “**FLOW Variables**” i.e. toman valores en conj. **Continuos**, finitos o infinitos ( $\mathbb{R}^n, \mathbb{R}^m, \mathbb{R}^p$ ), evolucionando (**FLUYENDO**) en Tiempo **Continuo**  $t (t \in \mathbb{R})$ .

Ej.  $S_{LTI} \begin{cases} \dot{x}(t) = A \cdot x(t) + B \cdot u(t); & x_0 = x(t_0) \\ y(t) = C \cdot x(t) + D \cdot u(t) \end{cases}$ , donde:  $u(t)_{m \times 1}, x(t)_{n \times 1}, y(t)_{p \times 1}$   
 $A_{n \times n}, B_{n \times m}, C_{p \times n}, D_{p \times m}$

- Subclases más generales: LTI, LPV, LTV, NLI, NLTV, etc.

**SDCTD** – Modelos de Estados Continuos en Tiempo Discreto  $k \in \mathbb{Z}$

- Representación **muestreada** de SDCTC (**Zero-Order Hold**);
- Var. **Estado**, **Entrada** y **Salida**: “**Staircase FLOW Variables**”, i.e. valores en conj. **Continuos** (o **cuantizados**), finitos o infinitos ( $\mathbb{R}^n, \mathbb{R}^m, \mathbb{R}^p$ ), evolucionando en Tiempo **Discreto** (o Muestreado)  $t \rightarrow k.T (t, T \in \mathbb{R} \rightarrow k \in \mathbb{Z})$ .

Ej.  $S_{LTI} \begin{cases} x((k+1).T) = [\Phi \equiv e^{A.T}] \cdot x(k.T) + [\Gamma \equiv \int_0^T e^{A.\eta} . d\eta . B] \cdot u(k.T); & x_0 = x(0) \\ y(k.T) = C \cdot x(k.T) + D \cdot u(k.T) \end{cases}$



# Sistemas Dinámicos (SD): Clasificación

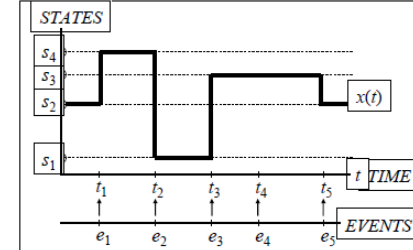
## SD SIMBÓLICOS: Estados **DISCRETOS**, **activados x EVENTOS** (DEDS)

**Estados** (Discretos): Modos de operación, Etapas, Situación;

**Eventos** (Asincrónicos): Comandos de usuario (arranque/parada), eventos externos, Fallas, cambio en entradas discretas, comparaciones, etc..

$$x' = f(x, e)$$

### "JUMP Variables"



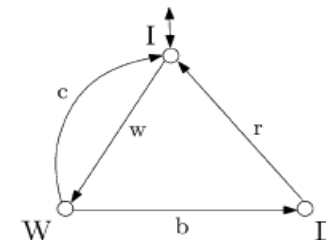
## DEDS – Modelos de Estados **Discretos** activados por **Eventos** = **Autómatas**

- Almacenamientos [**Memoria**] e intercambios [**Comunicación**] de **Información (= Datos)**;
- "**Discrete – Events DS**": DEDS = Discrete-States, Events-Driven DS
- Vectores de Estado, de Entrada y Salida: "JUMP Variables", i.e. toman valores **Discretos** en conj. **Finitos** o **Contables** ( $\mathbb{N}$ ), evolucionando (**SALTANDO O CONMUTANDO INSTANTÁNEAMENTE**) en instantes de Tiempo dados, activados por **EVENTOS asincrónicos**.
- (ej. redes de comunicaciones digitales, **sistemas digitales secuenciales de control**, **autómatas** = máquinas abstractas (computacionales) de estados finitos FSM, etc.) →

$$X(n+1) = f(X(n), u)$$

### → Automatización:

- Control **Supervisor**
- Secuenciamiento de Tareas o Modos
- Gestión de Fallas.



# Control Discreto (ON/OFF) Marcha/Parada de Motor ACIM

**Accionamiento DOL (= Direct On Line):** Conexión Directa a Línea trifásica 380 Vac 50 Hz mediante **1 CONTACTOR**. Operación en **1° y 2° Cuadrantes**: **FORWARD Motoring/Braking** (Motorización/Frenado en Sentido giro DIRECTO).

**[Frenado Regenerativo: SOLO SI LA CARGA LO ARRASTRA, RESBALAMIENTO NEGATIVO]**

**ESPECIFICACIÓN de Requisitos de Operación:** ([NC] = Normally Closed / [NO] = Normally Open)

- Condición **SEGURA**: Motor **Detenido (desconectar)** ante **Alarma, Sobrecarga, Falla, o Emergencia**.
- Relé **TÉRMICO** OL=**Overload** [NC] se activa (OFF) → **Desconecta** Motor [Prioridad 1: **PROTECCIÓN**].
- Pulsador **PARADA Normal** [NC] se activa (OFF) → **Desconecta** Motor [Prioridad 2: **MANIOBRA**].
- Pulsador **MARCHA** [NO] se activa (ON) → **Conecta** Motor [Prioridad 3: **MANIOBRA**], si se cumple que (**Lógica Booleana**): Motor desconectado **Y NO** activados (Relé **Térmico** **O** Pulsador de **Parada**).

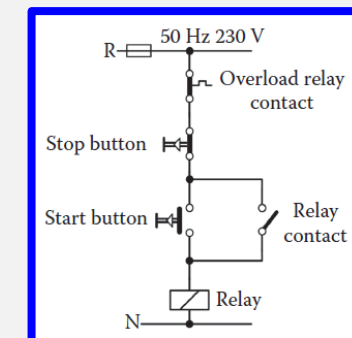
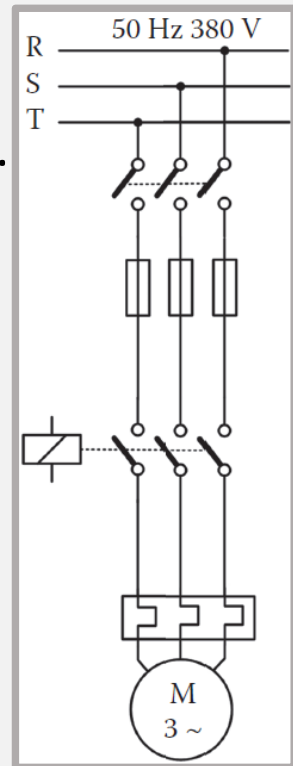
**Entradas/Salidas (I/O= Inputs/Outputs) del Controlador Discreto (Lógico Secuencial):**

- 1 Salida Discreta (**DO**): **CONTACTOR** de Conexión (**ON**) / Desconexión (**OFF**) de Línea al Motor.
- 3 ó 4 Entradas Discretas (**DI**): **Relé TÉRMICO** / Pulsador **MARCHA** / Pulsador **PARADA Normal** / Contacto Auxiliar de **CONTACTOR** (Feedback discreto externo) vs Bit Interno para **enclavamiento**.

→ **Hacer:**

1. Implementar en papel, en formas convencionales conocidas:
  - **Lógica cableada (Hardwired)** → Lógica de contactos (Ladder Diagram)
  - Bloques de Funciones Lógicas (Compuertas Booleanas)
  - Pseudocódigo informático de alto nivel (similar a Pascal, C, etc.)
2. **Tabla de Verdad** y **Diagrama Lógico Temporal**
3. **Diagrama de Estados Discretos y Transiciones**

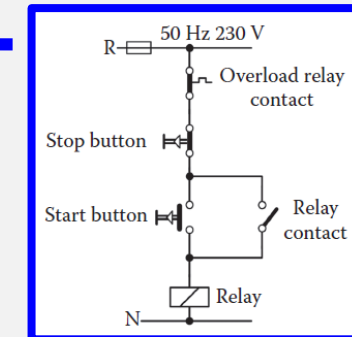
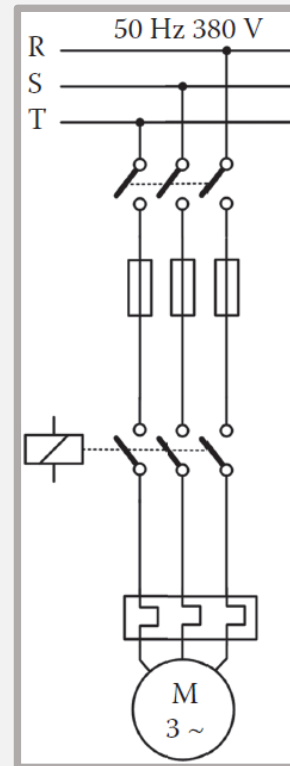
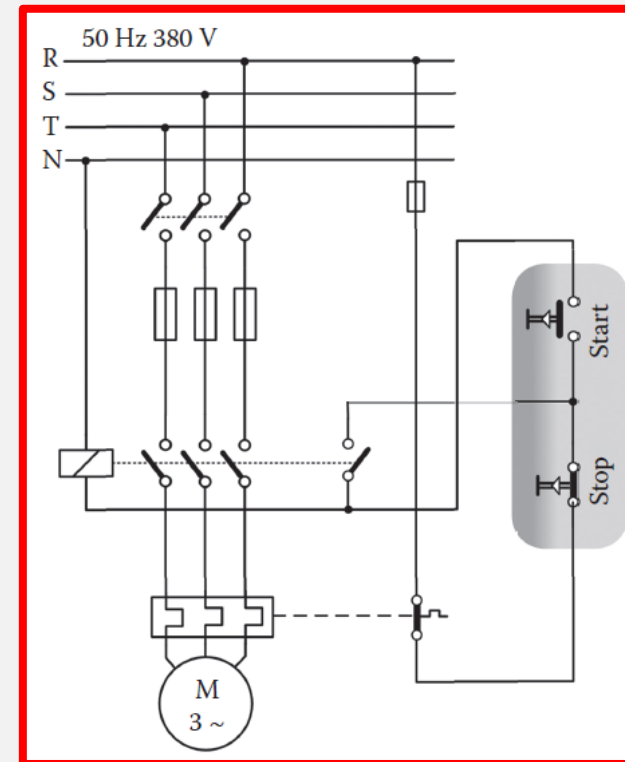
**Modelo DISCRETO:** NO considera Dinámica Continua de Motor



# Control Discreto (ON/OFF) Marcha/Parada de Motor ACIM

**Accionamiento DOL (= Direct On Line):** Conexión Directa a Línea trifásica 380 Vac 50 Hz mediante **1 CONTACTOR**.  
**Operación en 1° y 2° Cuadrantes:** **FORWARD Motoring/Braking** (Motorización/Frenado en Sentido giro DIRECTO).  
[Frenado Regenerativo: SOLO SI LA CARGA LO ARRASTRA, RESBALAMIENTO NEGATIVO]

**Solución con Lógica Cableada (Hardwired):**



**Modelo DISCRETO:** NO considera Dinámica Continua de Motor

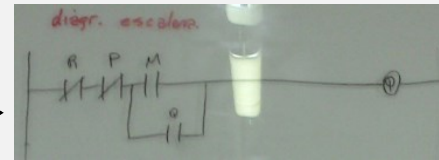
# Control Discreto (ON/OFF) Marcha/Parada de Motor ACIM

**Accionamiento DOL (= Direct On Line):** Conexión Directa a Línea trifásica 380 Vac 50 Hz mediante **1 CONTACTOR**.  
**Operación en 1° y 2° Cuadrantes:** **FORWARD Motoring/Braking** (Motorización/**Frenado** en Sentido giro DIRECTO).  
**[Frenado Regenerativo: SOLO SI LA CARGA LO ARRASTRA, RESBALAMIENTO NEGATIVO]**

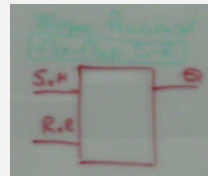
**Solución con Lógica Cableada (Hardwired):**

**Formas de Representación (equivalentes):**

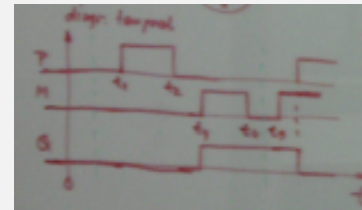
- Diagrama de lógica de Contactos (red escalera o "ladder" con enclavamiento):



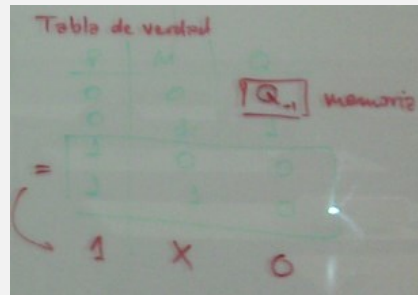
- Diagrama de Bloques Funcionales: (Flip-Flop S-R)



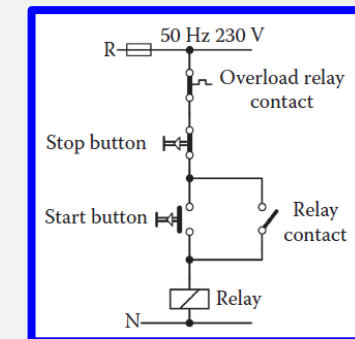
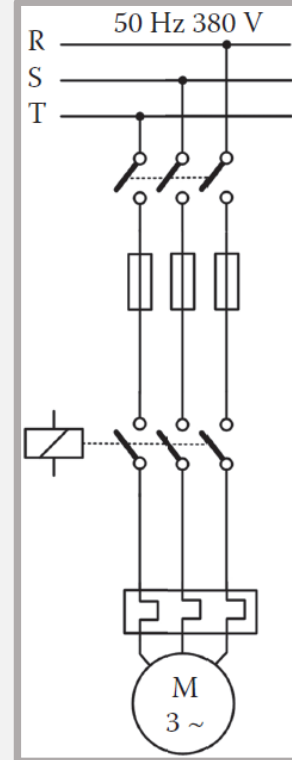
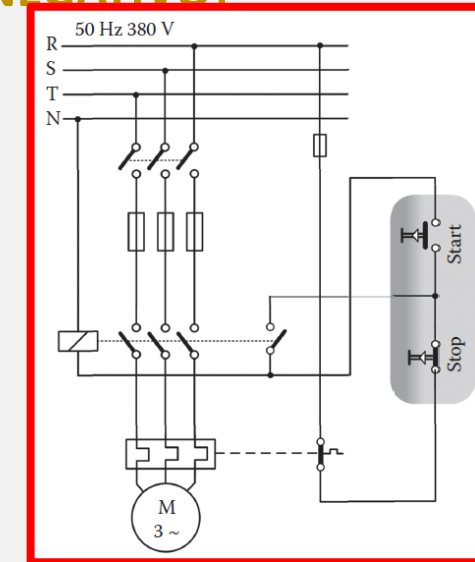
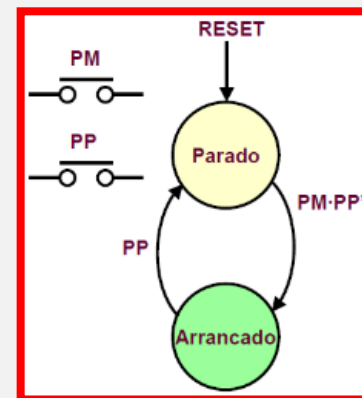
- Diagrama de Evolución Temporal:



- Tabla de Verdad:



- Diagrama de Transiciones de Estados Discretos



**Modelo DISCRETO:** NO considera Dinámica Continua de Motor

# Control Discreto Marcha FWD-REV/Parada de Motor ACIM

**Accionamiento DOL (= Direct On Line) con INVERSIÓN:** Conexión Directa/Inversa a Línea con **2 CONTACTORES**.  
**Operación en 4 Cuadrantes:** FORWARD/REVERSE Motoring/Braking (Motorización/Frenado en ambos Sentidos).  
**[Frenado Regenerativo: SOLO SI LA CARGA LO ARRASTRA, RESBALAMIENTO NEGATIVO]**

## ESPECIFICACIÓN de Requisitos de Operación:

- Condición **SEGURA**: Motor **Detenido** (desconectar ante **Falla, Emerg.**).
- Relé **TÉRMICO** [NC] se activa (OFF) → **Desconecta** Motor [Prioridad 1].
- Pulsador **PARADA** [NC] se activa (OFF) → **Desconecta** Motor [Prioridad 2].
- Pulsadores **FWD/REV** [NO] se activan (ON) → **Conectan** Motor FWD **Ó** REV **mutuamente excluyentes! (XOR)** [Prioridad 3], si se cumple que (**Lógica Booleana**): Motor desconectado **Y NO** activados (Relé Térmico **O** Pulsador de Parada).

## Entradas/Salidas (I/O) del Controlador Discreto (Lógico Secuencial):

- 2 Salidas Discretas (**DO**): **CONTACTORES** FWD/REV de Conexión al Motor.
- 4 **ó** 6 Entradas Discretas (**DI**): **Relé TÉRMICO** / **Pulsadores FWD/REV** / **Pulsador PARADA Normal** / **Contactos Auxiliares de CONTACTORES** (Feedback discretos externos) vs Bits Internos para **enclavamientos**.

## → Hacer:

- **Lógica cableada (Hardwired)** → Lógica de contactos (Ladder Diagram)
- Bloques de Funciones Lógicas (Compuertas Booleanas)
- Pseudocódigo informático de alto nivel (similar a Pascal, C, etc.)

## 2. Tabla de Verdad y Diagrama Lógico Temporal

## 3. Diagrama de Estados Discretos y Transiciones

**Modelo DISCRETO:** NO considera Dinámica Continua de Motor

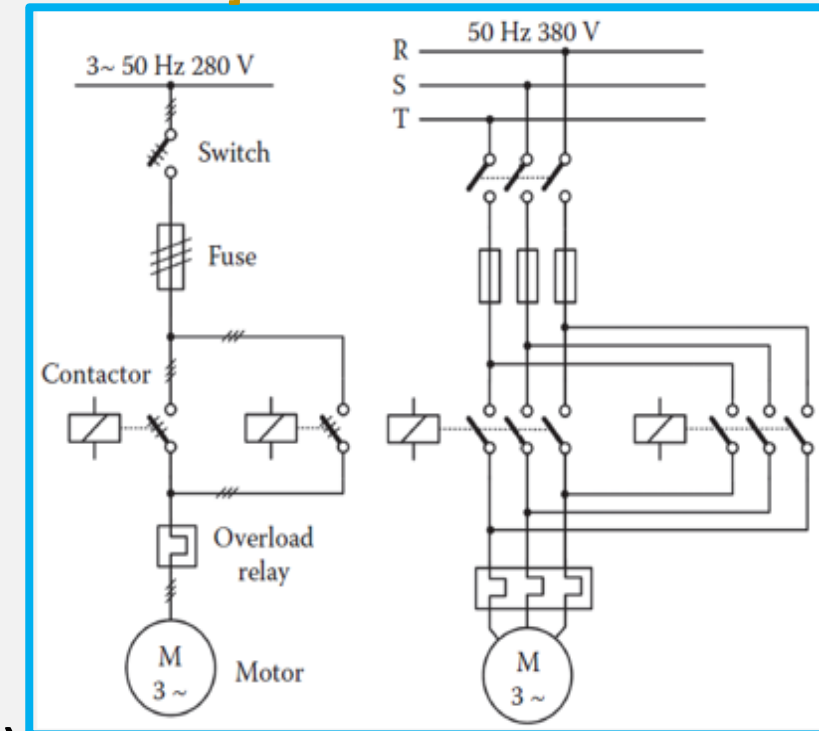


Diagrama Unifilar → Trifilar



# Sistemas Dinámicos (SD): Clasificación

## SD HÍBRIDOS, activados por TIEMPO + EVENTOS:

Combinación e interacción de ambos aspectos:  
Estados **Continuos** (Físicos) + **Discretos** (Simbólicos).

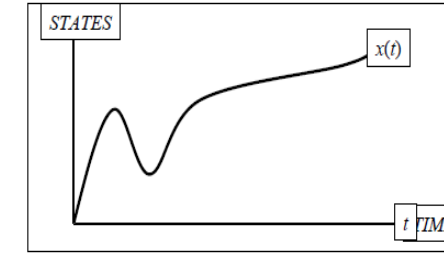
## SDH – Modelos “Híbridos”

Almacenamiento e intercambio de **Energía** + **Información**;

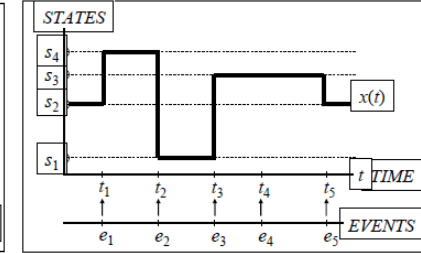
Acoplamiento o interacción EXPLÍCITA entre Estados y Entradas

Continuos (subsistema **SDCTC+SDCTD**) + Discretos (subsistema **SDDE**)  
evolucionando en conjunto, parte “fluyendo” en Tiempo continuo  
y parte “saltando” activado por Eventos (asincrónicos).

## “FLOW Variables” + “JUMP Variables”



$$\dot{x} = f(x, t)$$



$$x' = f(x, e)$$



## Mayoría de SD de Ingeniería & Tecnología: SD HÍBRIDOS

- **Sistemas “Mecatrónicos”**: autos, computadoras, aviones, electrodomésticos, máquinas industriales y control de procesos industriales.
- Estabilización y optimización de SDNL mediante **control conmutado** (on/off, etc.).
- Control Digital **Secuencial** de sistemas físicos en ingeniería (e.g. PLC, PAC, DCS, etc.).
- Verificación de programas digitales interactuando con ambientes continuos.
- Control Jerárquico, interacción de paquetes de datos con procesos físicos.

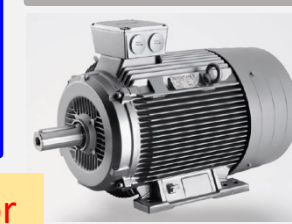
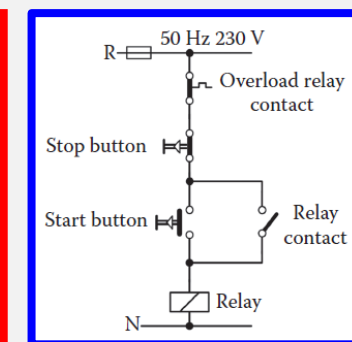
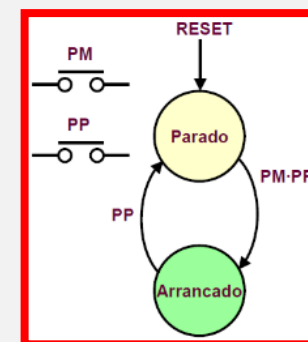
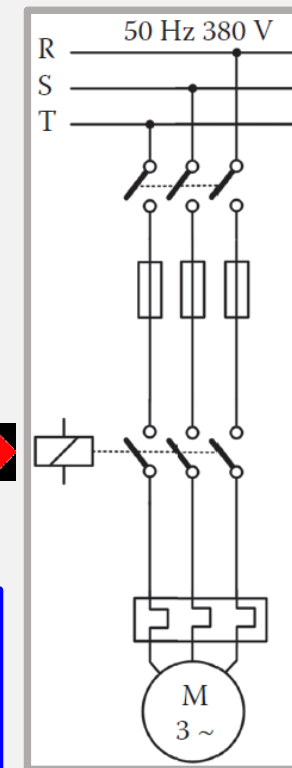
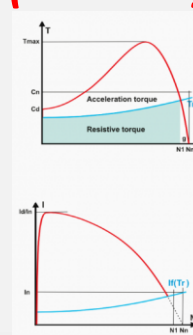
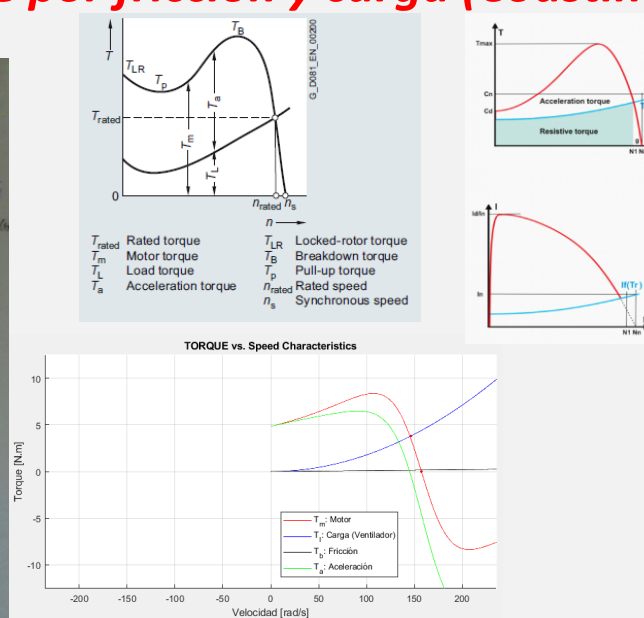
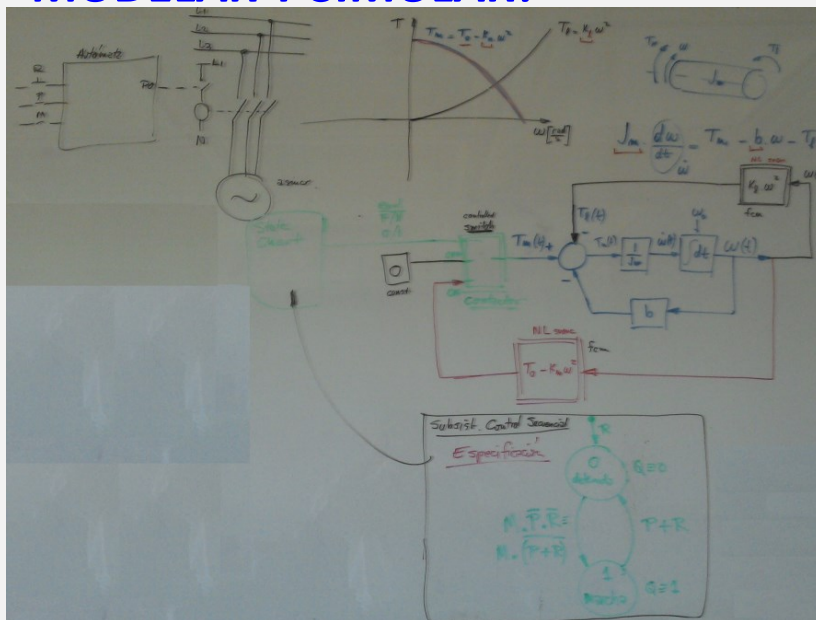
**Modelo HÍBRIDO: SÍ considera Dinámica Continua de Motor**

# Control Discreto (ON/OFF) Marcha/Parada de Motor ACIM

**Accionamiento DOL (= Direct On Line):** Conexión Directa a Línea trifásica 380 Vac 50 Hz mediante **1 CONTACTOR**.  
**Operación en 1° y 2° Cuadrantes:** **FORWARD Motoring/Braking** (Motorización/Frenado en Sentido giro DIRECTO).  
**[Frenado Regenerativo: SOLO SI LA CARGA LO ARRASTRA, RESBALAMIENTO NEGATIVO]**

*Considerar Explícitamente:*

- Dinámica Continua motor con arranque directo (DOL) con contactor**  
 (ej. ley torque carga cuadrática creciente, torque impulsor simplif.): *Estado continuo: velocidad*
  - Autómata Discreto (lógica secuencial) de conmutación ON/OFF: Estado discreto ON/OFF**  
**Conexión → Arranque (Aceleración) y Marcha (Motorización o Frenado, según carga)**  
**Desconexión → Desaceleración libre por fricción y carga (Coasting) o Arrastre Libre, según carga)**
- MODELAR Y SIMULAR!**

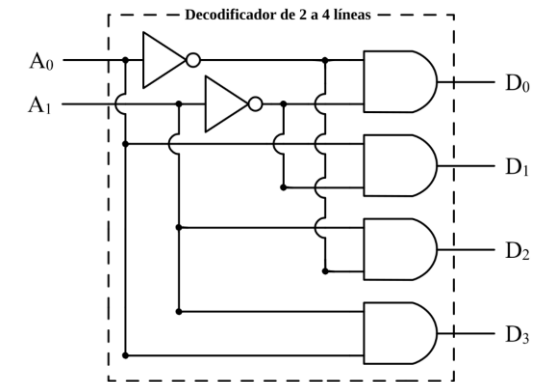
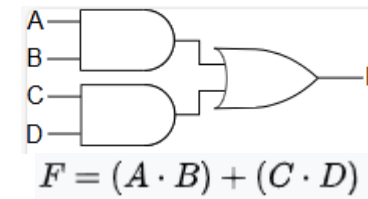


**Modelo HÍBRIDO: SÍ considera Dinámica Continua de Motor**

# Sistemas (Electrónicos) Digitales [Hw/Sw]: Lógica Booleana

**Sistema o Dispositivo DIGITAL:** genera, procesa, transmite y/o almacena señales, datos, o información *codificada* en Valores o Estados Discretos (Discontinuos), representables como combinación de bits (binary digits: {0,1}) → **Algebra de Boole** (operadores lógicos AND, OR, NOT) para Análisis y Síntesis de Sistemas Digitales.

1. Sistemas Digitales **COMBINACIONALES** (inmediatos o Estáticos, SIN memoria): Salidas Digitales solo dependen del valor instantáneo actual de sus Entradas Digitales.



| Tabla de verdad |   | A <sub>1</sub> | A <sub>0</sub> | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |
|-----------------|---|----------------|----------------|----------------|----------------|----------------|----------------|
| 0               | 0 | 0              | 0              | 0              | 0              | 1              |                |
| 0               | 1 | 0              | 0              | 1              | 0              |                |                |
| 1               | 0 | 0              | 1              | 0              | 0              |                |                |
| 1               | 1 | 1              | 0              | 0              | 0              |                |                |

Minitérminos

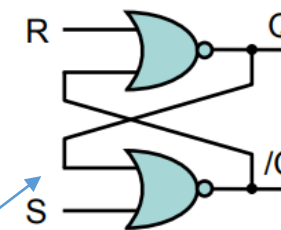
$$D_0 = \bar{A}_1 \cdot \bar{A}_0$$

$$D_1 = \bar{A}_1 \cdot A_0$$

$$D_2 = A_1 \cdot \bar{A}_0$$

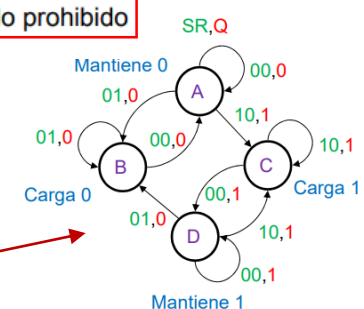
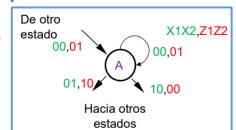
$$D_3 = A_1 \cdot A_0$$

2. Sistemas Digitales **SECUENCIALES = AUTÓMATAS** (demorados o Dinámicos, Con MEMORIA): Salidas Digitales dependen del valor instantáneo actual de sus Entradas Digitales, Y de la Secuencia o Evolución (dinámica) de los valores anteriores, o **Estado** Discreto Interno (Memoria).



| S | R | Q | /Q |                    |
|---|---|---|----|--------------------|
| 0 | 0 | Q | /Q | → Mantener estado  |
| 1 | 0 | 1 | 0  | → Set              |
| 0 | 1 | 0 | 1  | → Reset            |
| 1 | 1 | 0 | 0  | → Estado prohibido |

Diagrama de Estados y Transiciones



Implementación Física: **Hardware** Electrónico, Fijo o **Programable + Software**.  
Modelo: **Máquina Abstracta Secuencial (FSM) = Autómata Finito determinístico.**

# AUTÓMATA de Estados Finitos Determinístico (FSA / FSM):

Finite-State **Automaton** (FSA): **Abstract Sequential Machine** (FSM)

Sistema Digital Secuencial (DEDS, Estados Discretos activados por Eventos)  $X(n+1) = f(X(n), u)$

Sistemas **REACTIVOS** (reaccionan a Eventos)

**Definición Formal** sin explicitar SALIDAS:

**4-tupla**(\*)  $G = (Q, E, f, q_0)$

$Q = \{q_0, q_1, \dots, q_s\}$ : Conjunto Finito de **Estados Discretos (Modos)**.

$E = \{e_1, e_2, \dots, e_E\}$ : **Alfabeto de Entrada**: condición de **Transiciones**  
Conjunto Finito de **Eventos** asincrónicos.

$f = Q \times E \rightarrow Q$ : Función de **Transición** a próx. estado (**PARCIAL**:  
 $f(q, e) = q'$  **NO todas las transiciones están permitidas**).

$q_0 \in Q$ : Estado o Modo **Inicial** o de Arranque.

**Diagrama de Estados Discretos y Transiciones:**

**Grafo dirigido**: vértices=nodos y enlaces c/ dirección

**Ejemplo: Máquina Elemental ("Small Machine")**

$Q = \{\text{Idle}, \text{Work}, \text{Down}\}$

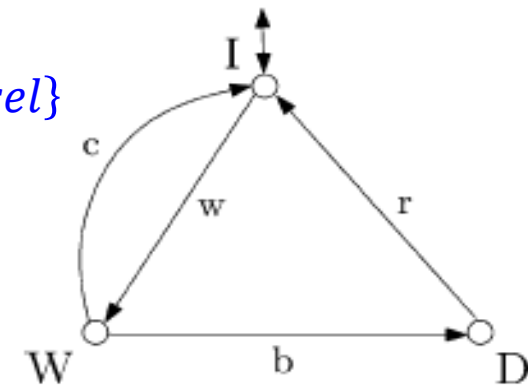
$E = \{\text{work}, \text{break}, \text{reset}, \text{cancel}\}$

$f(I, w) = W$ ;

$f(W, b) = D$ ;  $f(W, c) = I$

$f(D, r) = I$ ;

$q_0 = I$



Esta Definición considera **solo Estados y Evolución (Eventos Activación)** del Autómata  
(**NO** estamos explicitando aún las posibles **SALIDAS = ACCIONES** del Autómata).

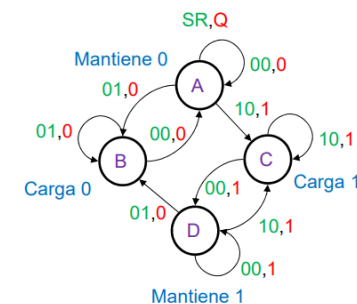
**SALIDAS o ACCIONES**: Efectos o Acciones de **Control Discreto, Supervisión!**

**2 enfoques o Paradigmas** respecto a **cuándo se activan las posibles Salidas**:

**Moore FSA vs Mealy FSA**: Definición  $\rightarrow$  6-tuple (agregar definición de salidas)

(\*) **n-tupla** (matemáticas): *estructura de datos* que tiene lista (secuencia) ordenada y finita de n elementos.

(\*\*) **Grafo** (matemáticas): estructura matemática consistente en vértices (o nodos) + enlaces (o conexiones) entre vértices.



# AUTÓMATA de Estados Finitos Determinístico (FSA / FSM):

## Finite-State *Automaton* (FSA): *Abstract Sequential Machine* (FSM)

Sistema Digital Secuencial (DEDS, Estados Discretos activados por Eventos)

$$X(n+1) = f(X(n), u)$$

Sistemas **REACTIVOS** (reaccionan a Eventos)

**Definición Formal** explicitando **SALIDAS**:

**MOORE FSA**: Salidas actuales dependen solo de Estado actual  
**6-tupla(\*)**  $G = (Q, E, f, q_0, O, g)$

$Q = \{q_0, q_1, \dots, q_s\}$ : Conjunto Finito de **Estados Discretos (Modos)**.

$E = \{e_1, e_2, \dots, e_E\}$ : **Alfabeto de Entrada**: condición de **Transiciones**  
 Conjunto Finito de **Eventos** asincrónicos.

$f = Q \times E \rightarrow Q$ : Función de **Transición** a próx. estado (**PARCIAL**:  
 $f(q, e) = q'$  **NO todas las transiciones están permitidas**).

$q_0 \in Q$ : Estado o Modo **Inicial o de Arranque**.

$O = \{o_1, o_2, \dots, o_O\}$ : **Alfabeto de Salida**: condición de **Transiciones**  
 Conjunto Finito de **Acciones** de Salida.

$g = Q \rightarrow O$ : Función de **Salida** para cada Estado.

**MOORE FSA**: Valor actual de Salidas depende solo de Estado actual.

Las entradas (eventos) influyen el cambio de Estado actual a próximo Estado →

→ Entradas influyen **INDIRECTAMENTE** las salidas (solo a través del Cambio a próximo estado), pero no directamente la salida actual o inmediata.

Ref.: [https://en.wikipedia.org/wiki/Moore\\_machine#cite\\_note-gedanken-1](https://en.wikipedia.org/wiki/Moore_machine#cite_note-gedanken-1)

(\*) **n-tupla** (matemáticas): *estructura de datos* que tiene lista (secuencia) ordenada y finita de n elementos.

(\*\*) **Grafo** (matemáticas): estructura matemática consistente en **vértices** (o **nodos**) + **enlaces** (o **conexiones**) entre vértices.

**Diagrama de Estados Discretos y Transiciones**:

**Grafo dirigido**: vértices=nodos y enlaces c/ dirección  
**Ejemplo: Máquina Elemental ("Small Machine")**

$Q = \{\text{Idle}, \text{Work}, \text{Down}\}$

$E = \{\text{work}, \text{break}, \text{reset}, \text{cancel}\}$

$f(I, w) = W$ ;

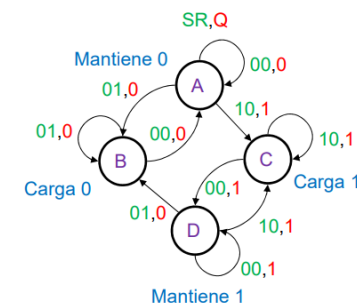
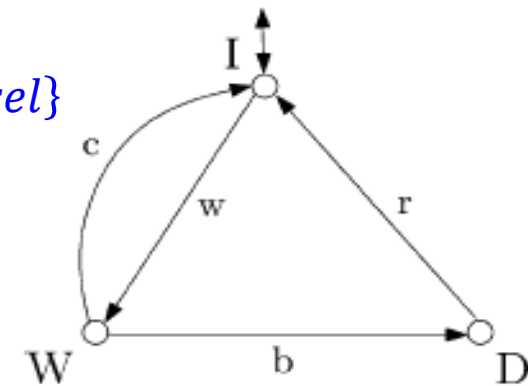
$f(W, b) = D$ ;  $f(W, c) = I$

$f(D, r) = I$ ;

$q_0 = I$

$O = \{o_I, o_W, o_D\}$

$g(I) = o_I$ ;  $g(W) = o_W$ ;  $g(D) = o_D$



# AUTÓMATA de Estados Finitos Determinístico (FSA / FSM):

## Finite-State *Automaton* (FSA): *Abstract Sequential Machine* (FSM)

Sistema Digital Secuencial (DEDS, Estados Discretos activados por Eventos)  $X(n+1) = f(X(n), u)$

Sistemas **REACTIVOS** (reaccionan a Eventos)

**Definición Formal** explicitando **SALIDAS**:

**MEALY FSA:** Salidas actuales dependen de Estado Y Entradas actuales **Grafo dirigido:** vertices y enlaces dirigidos  
**6-tupla(\*)**  $G = (Q, E, f, q_0, O, g)$  Ejemplo: **Máquina Elemental ("Small Machine")**

$Q = \{q_0, q_1, \dots, q_s\}$ : Conjunto Finito de **Estados Discretos (Modos)**.

$E = \{e_1, e_2, \dots, e_E\}$ : **Alfabeto de Entrada:** condición de **Transiciones**  
 Conjunto Finito de **Eventos** asincrónicos.

$f = Q \times E \rightarrow Q$ : Función de **Transición** a próx. estado (**PARCIAL:**  
 $f(q, e) = q'$  **NO todas las transiciones están permitidas).**

$q_0 \in Q$ : Estado o Modo **Inicial o de Arranque**.

$O = \{o_1, o_2, \dots, o_O\}$ : **Alfabeto de Salida:** condición de **Transiciones**  
 Conjunto Finito de **Acciones** de Salida.

$g = Q \times E \rightarrow O$ : Función de **Salida** para cada Estado y/o Evento.

**MEALY FSA:** Valor actual de Salidas depende de Estado actual y/o Eventos actuales.

Las entradas (eventos) influyen el cambio de Estado actual a próximo Estado y también pueden influenciar **DIRECTAMENTE** e **INMEDIATAMENTE** las salidas (NO solo a través del Cambio a próximo estado).

Ref.: [https://en.wikipedia.org/wiki/Mealy\\_machine](https://en.wikipedia.org/wiki/Mealy_machine)

(\*) **n-tupla** (matemáticas): *estructura de datos* que tiene lista (secuencia) ordenada y finita de n elementos.

(\*\*) **Grafo** (matemáticas): estructura matemática consistente en vertices (o nodos) + enlaces (o conexiones) entre vértices.

**Diagrama de Estados Discretos y Transiciones:**

**Grafo dirigido:** vertices y enlaces dirigidos  
 Ejemplo: **Máquina Elemental ("Small Machine")**

$Q = \{\text{Idle}, \text{Work}, \text{Down}\}$

$E = \{\text{work}, \text{break}, \text{reset}, \text{cancel}\}$

$f(I, w) = W$ ;

$f(W, b) = D$ ;  $f(W, c) = I$

$f(D, r) = I$ ;

$q_0 = I$

$O = \{o_I, o_W, o_D\}$

$g(I) = o_I$ ;  $g(W) = o_W$ ;  $g(D) = o_D$

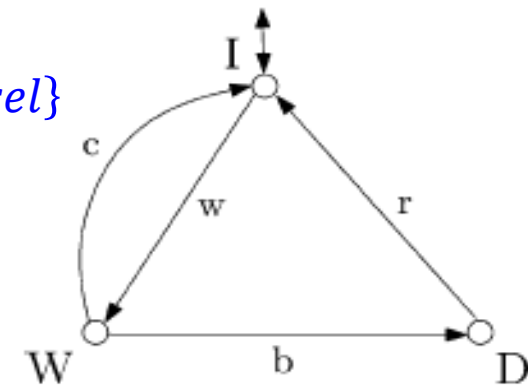
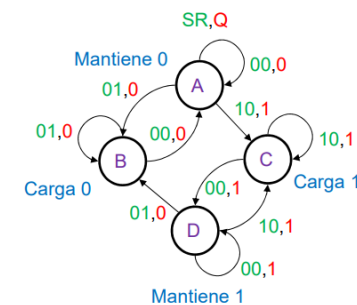
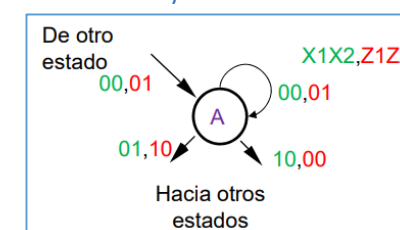


Diagrama de Estados y Transiciones



# AUTÓMATA de Estados Finitos Determinístico (FSA / FSM):

Finite-State *Automaton* (FSA): *Abstract Sequential Machine* (FSM)

Sistema Digital Secuencial (DEDS, Estados Discretos activados por Eventos)  $X(n+1) = f(X(n), u)$

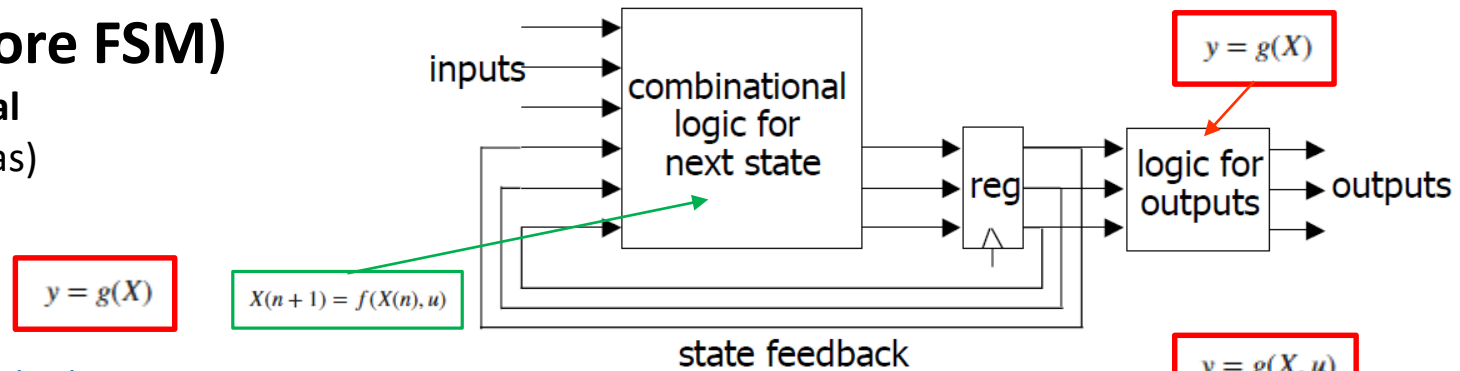
Ref.: <https://www.mathworks.com/help/stateflow/ug/overview-of-mealy-and-moore-machines.htm>

## DEDS – Paradigmas de Autómatas (Sists. Digitales Secuenciales)

### • Autómata de Moore (Moore FSM)

**Salidas** dependen sólo de estado actual  
(no dependen directamente de entradas)

**Salidas** cambian en forma  
demorada con cambio de entradas  
(**síncronas** con cambio en estados)

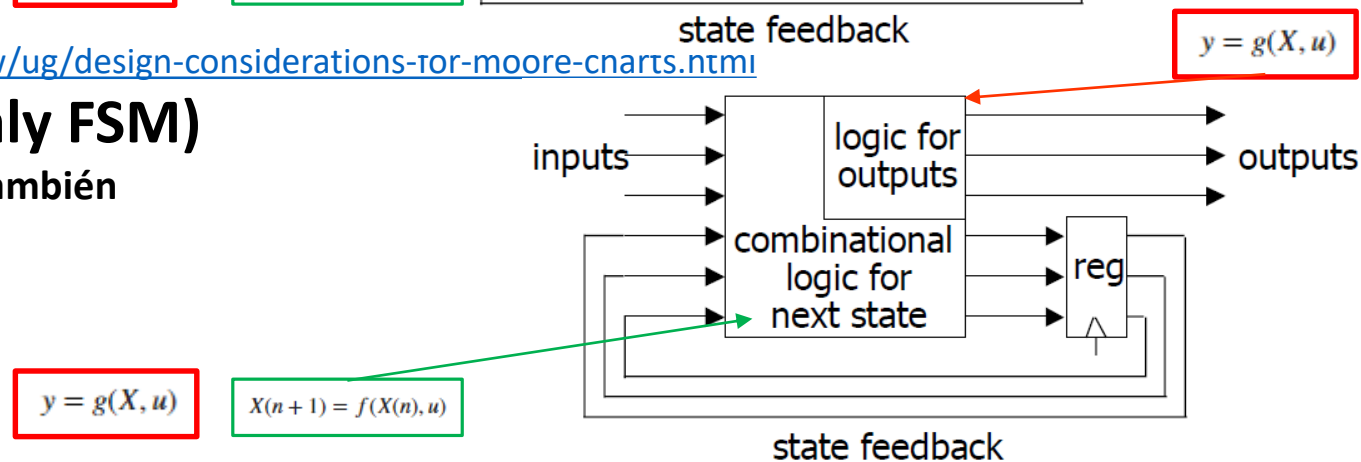


Ver: <https://www.mathworks.com/help/stateflow/ug/design-considerations-for-moore-charts.html>

### • Autómata de Mealy (Mealy FSM)

**Salidas** dependen de estado actual y también  
de entradas (transferencia directa)

**Salidas** pueden cambiar en forma  
inmediata con cambio de entradas  
(**asíncronas** con cambio en estados)



Ver: <https://www.mathworks.com/help/stateflow/ug/design-considerations-for-mealy-charts.html>

# AUTÓMATA de Estados Finitos Determinístico (FSA / FSM):

## SD SIMBÓLICOS: Estados DISCRETOS, activados x EVENTOS (DEDS)

“JUMP Variables”

**FSM** → **STATECHARTS: FSM Jerárquicos y Concurrentes**

ver Documentos:

1. AyCD\_01a\_DEDS\_Autómatas\_FSM\_STATEFLOW.pdf (11 págs.) →

Nota: Resumen extractado principalmente de Manual de Ayuda (online Help) de Simulink / Stateflow

Contenido:

**Discrete event dynamic systems (DEDS)**

**Stateflow**

**Finite State Machine (FSM) Concepts**

- FSM concept, representation, notation, semantics.
- STATE concept, hierarchy, decomposition & concurrency, notation.
- State ACTIONS, options.
- TRANSITION concept, hierarchy, notation.
- Transition EVENTS or CONDITIONS.

2. AyCD\_01a\_Harel D\_STATECHARTS - Visual Formalism for Complex Systems.pdf →

Paper, Fundamentos sobre los que se basa en parte STATEFLOW.

Discrete event dynamic systems (DEDS)

In *control engineering*, a **discrete event dynamic system (DEDS)** is a discrete-state, event-driven system of which the state evolution depends entirely on the occurrence of asynchronous discrete events over time. Although similar to continuous-variable dynamic systems (CVDS), DEDS consists solely of discrete state spaces and event-driven state transition mechanisms.

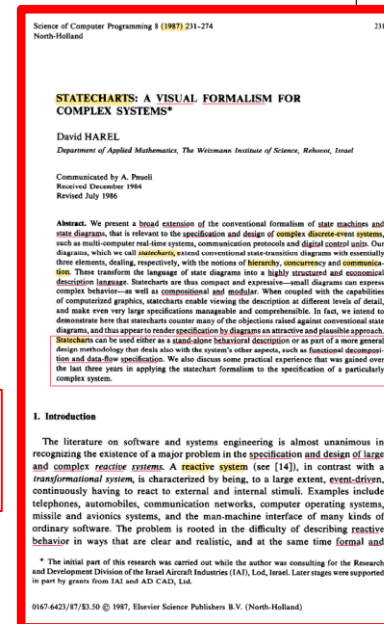
STATEFLOW

- Development environment for modeling, designing and simulating (a animating) complex combinatorial and sequential decision logic (reactive or event-driven or discrete-event systems) based on (finite) state machines and flow charts.
- Allows combining graphical and tabular representations of classic (default: full set of semantics), *Mealy* (output is a function of inputs and state) and *Mooore* (output is a function of state only) finite-state machines: state transition diagrams, flow charts, state transition tables/matrices, and truth tables.
- Deterministic execution semantics with hierarchy, parallelism, temporal operators, and events
- Model how the system reacts to events, time-based conditions, and external input signals.
- State diagram animation, state activity logging, data logging, and integrated debugging: static and run-time checks for transition conflicts, cyclic problems, state inconsistencies, data-range violations, and overflow conditions (for testing design consistency and completeness, and detecting runtime errors before implementation).

Applications: supervisory control, task scheduling, and fault management applications.

Difference between Flow Graphs and State Charts  
A flow graph is a *stateless* flow chart because it cannot maintain its active state between updates. As a result, a flow graph always begins executing from a default transition and ends at a terminating junction (a junction that has no valid outgoing transitions).

By contrast, a state chart stores its current state in memory to preserve local data and activity between updates. As a result, state charts can begin executing where they left off in the previous time step, making them suitable for modeling reactive or supervisory systems that depend on history. In these kinds of systems, the current result depends on a previous result.



### 1. Introduction

The literature on software and systems engineering is almost unanimous in recognizing the existence of a major problem in the specification and design of large and complex reactive systems. A reactive system (see [14]), in contrast with a *transformational system*, is characterized by being, to a large extent, event-driven, continuously having to react to external and internal stimuli. Examples include telephones, automobiles, communication networks, computer operating systems, missile and avionics systems, and the man-machine interface of many kinds of ordinary software. The problem is rooted in the difficulty of describing reactive behavior in ways that are clear and realistic, and at the same time formal and

\* The initial part of this research was carried out while the author was consulting for the Research and Development Division of the Israel Aircraft Industries (IAI), Ltd., Lod, Israel. Later stages were supported in part by grants from IAI and AD CAD, Ltd.

# STATECHARTS: FSM jerárquicos y concurrentes

## Discrete-State, Event-driven Dynamic Systems (DEDS)

$$X(n+1) = f(X(n), u)$$

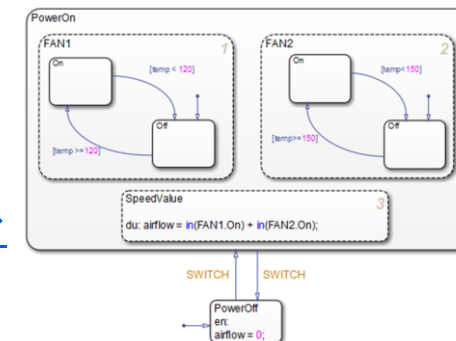
- **ESTADOS** (Etapas) **DISCRETOS** (contables) de duración variable, cuya **evolución secuencial** (mecanismo de **TRANSICIONES** hacia Nuevo/s Estado/s) depende de validación según Estado/s Activo (**Memoria**) y la ocurrencia o activación de **EVENTOS** discretos **asincrónicos** en el tiempo (entradas discretas).
- **ACCIONES** = salidas dependen solo de Estados [Paradigma **MOORE**]  $y = g(X)$   
o de Estados y/o Transiciones [Paradigma **MEALY**]  $y = g(X, u)$

## STATECHART:

Ref. inicial: AyCD\_01a\_ Harel D\_ **STATECHARTS - Visual Formalism for Complex Systems**.pdf →

*Paper, Fundamentos sobre los que se basa en parte STATEFLOW*

- Modelo de **comportamiento** de DEDS: Transiciones entre Estados Discretos.
- Generalización de **FSM** (Finite-State Machine): **jerarquía (encapsulamiento)**, **conurrencia (paralelismo)**, **operadores temporales y eventos**, semántica de ejecución determinística (Similar o alternativa a GRAFCET / SFC).



## Stateflow (Simulink):

- Entorno de Desarrollo para **modelado, diseño y simulación** (con **animación**) de Sistemas lógicos **combinacionales** y **secuenciales** complejos (DEDS, control supervisor, sistemas reactivos a eventos, condiciones temporales y entradas externas).
- Basado en **Flow charts** y **FSM** → **STATECHARTs** (Classic, Moore, Mealy), etc.

# STATECHARTS: FSM jerárquicos y concurrentes

## Notación, Sintaxis o ESTRUCTURA:

define conjunto de **Objetos + Reglas** que controlan las relaciones entre tales Objetos.

Stateflow chart **notation**:

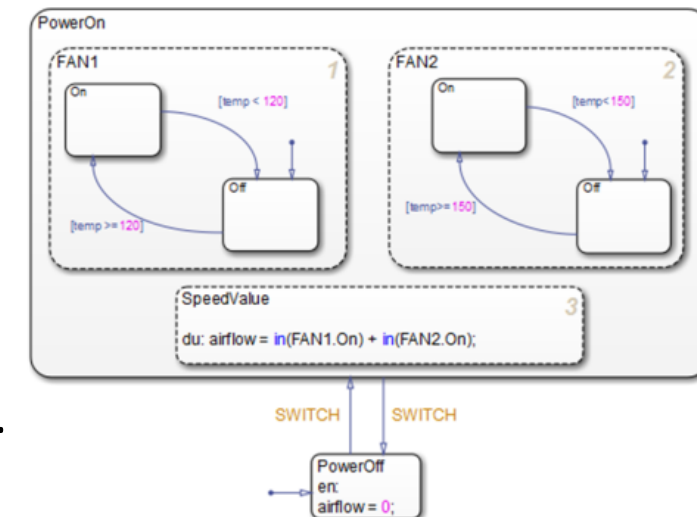
Provides a way to communicate the design information in a **Stateflow chart** (**modelo de comportamiento**).

It consists of these **elements**:

- A set of **graphical objects**
- A set of nongraphical **text-based objects**
- Defined **relationships** between those objects

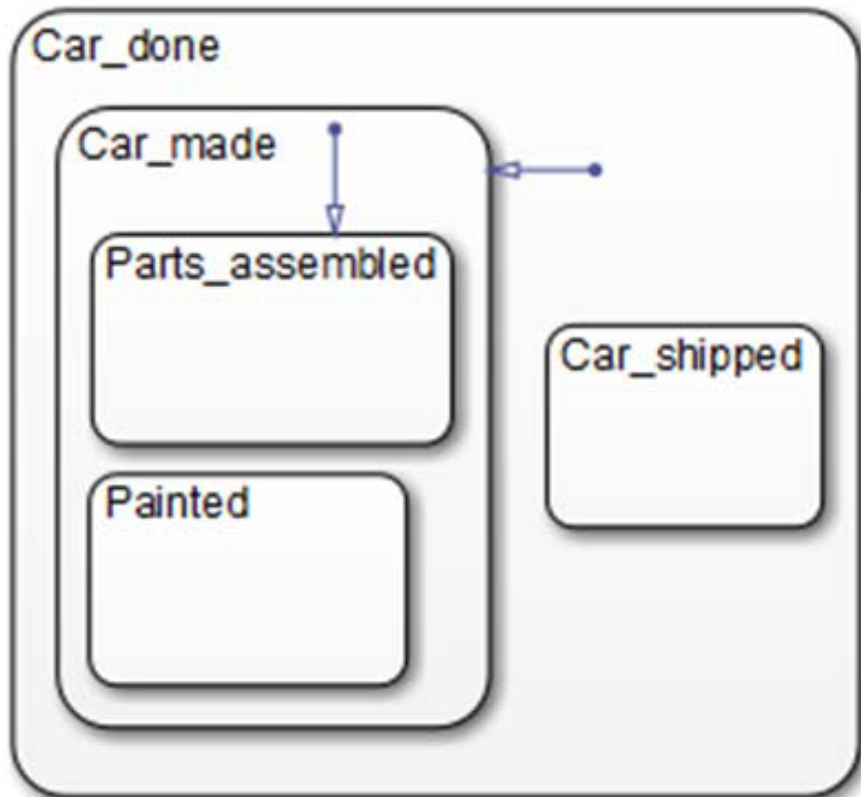
## Semántica o INTERPRETACIÓN:

- describe significado o cómo interpretar la notación del Statechart.
- A typical Stateflow chart contains **actions** associated with transitions and states.
- The semantics describe the **sequence of these actions** during chart execution.



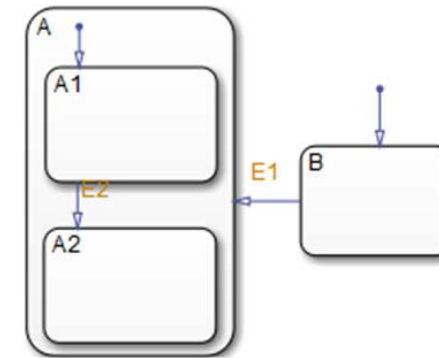
# STATECHARTS: FSM jerárquicos y concurrentes

**JERARQUÍA** de Estados:

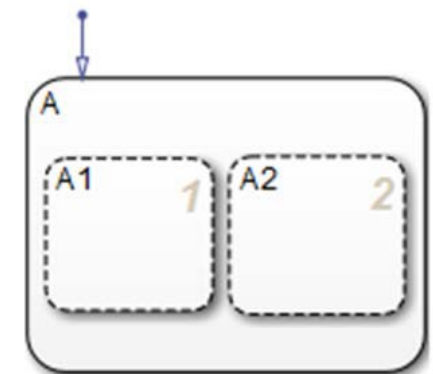
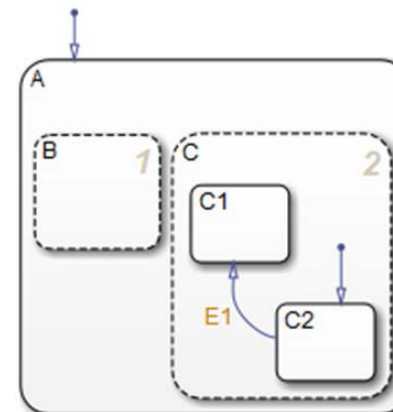


**DESCOMPOSICIÓN** de Estados:

a) **OR: Exclusiva o Excluyente** (NO concurrente)



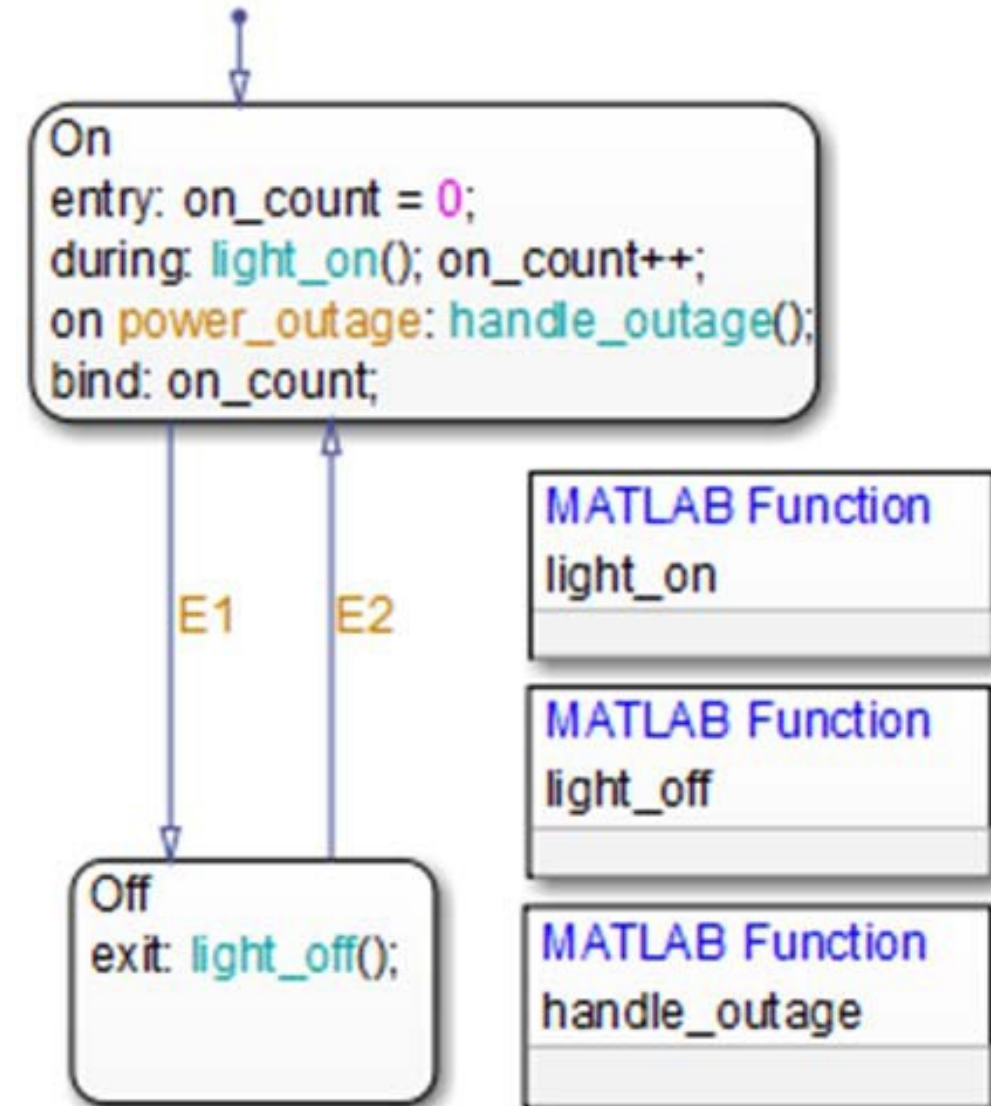
b) **AND: Concurrente o Paralelo** (Simultáneo)



# STATECHARTS: FSM jerárquicos y concurrentes

## Etiquetas de Estados y Acciones:

- name/
- entry: entry actions
- during: during actions
- exit: exit actions
- on event\_name: on event\_name actions
- bind: events, data

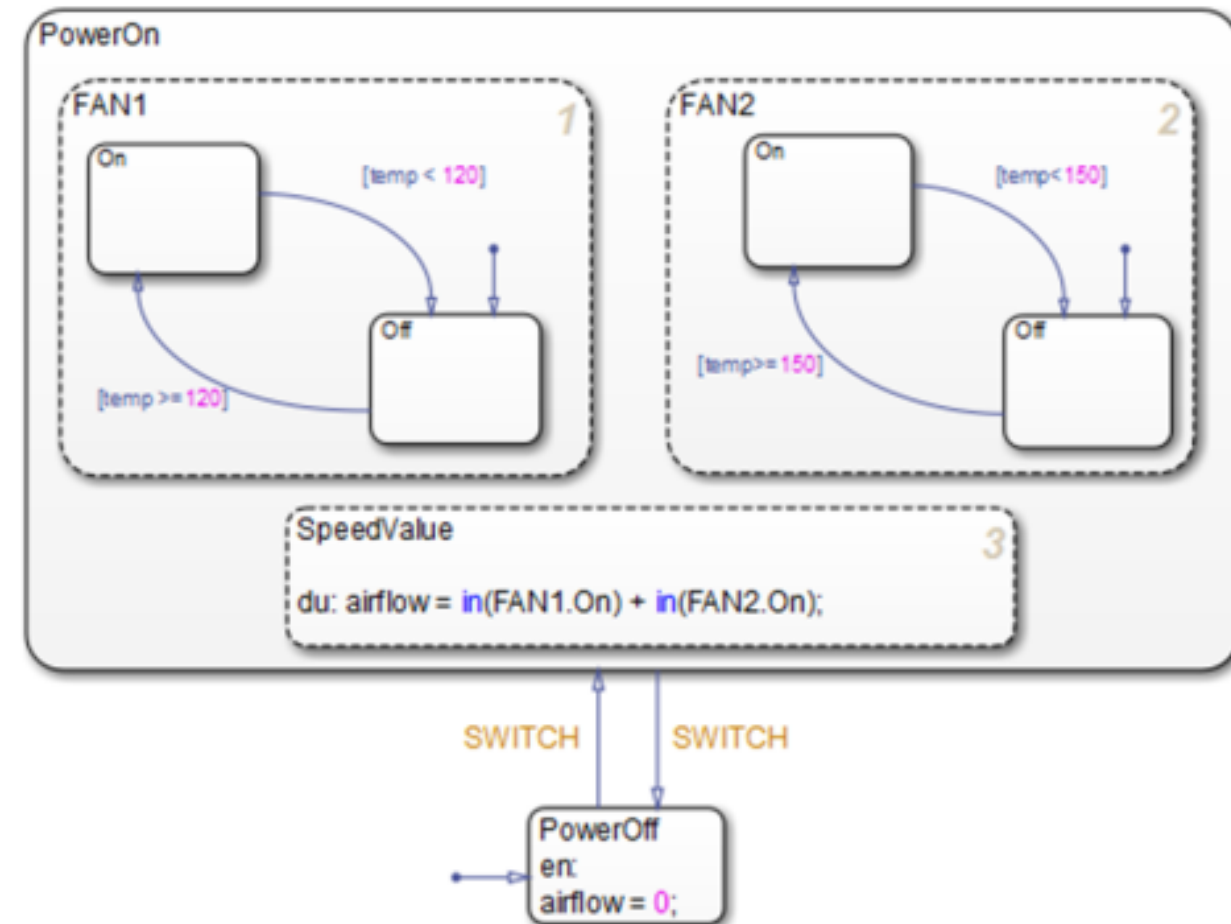


# STATECHARTS: FSM jerárquicos y concurrentes

**Etiquetas de Estados:** ejemplo de nombres internos “repetidos” sin conflicto (datos “Locales” o internos vs “Globales”)

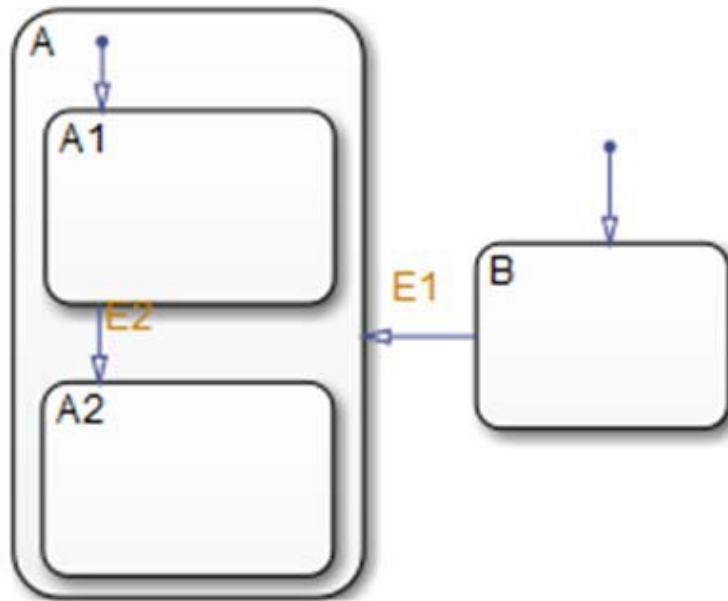
Nombres completos (estructura de datos):

- PowerOn.FAN1.On
- PowerOn.FAN1.Off
- PowerOn.FAN2.On
- PowerOn.FAN2.Off

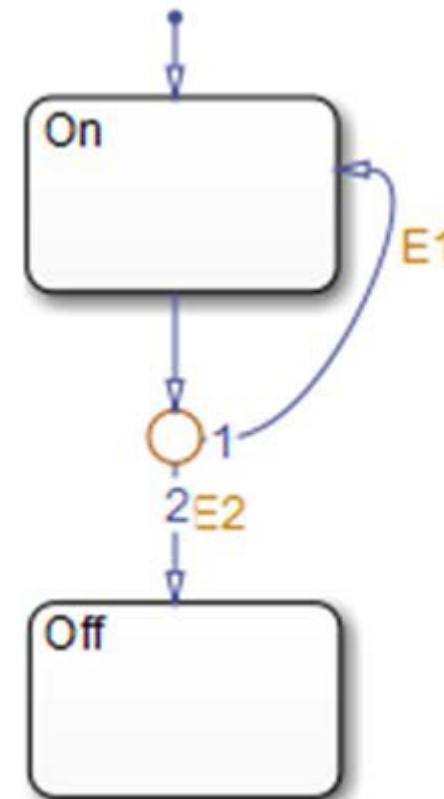


# STATECHARTS: FSM jerárquicos y concurrentes

Transiciones entre Estados:

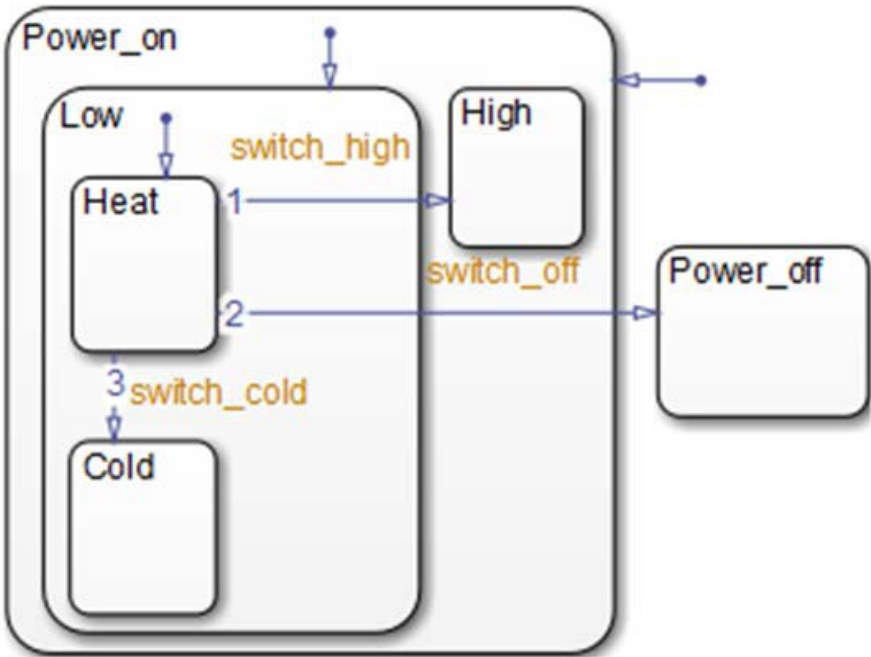


Uniones (Junctions) y  
Segmentos de Transición  
Prioridades de Transición (1, 2)



# STATECHARTS: FSM jerárquicos y concurrentes

JERARQUÍA de Transiciones:



Parentesco de Transiciones:

| Transition Label | Transition Parent | Transition Source  | Transition Destination |
|------------------|-------------------|--------------------|------------------------|
| switch_off       | /                 | /Power_on.Low.Heat | /Power_off             |
| switch_high      | /Power_on         | /Power_on.Low.Heat | /Power_on.High         |
| switch_cold      | /Power_on.Low     | /Power_on.Low.Heat | /Power_on.Low.Cold     |

# STATECHARTS: FSM jerárquicos y concurrentes

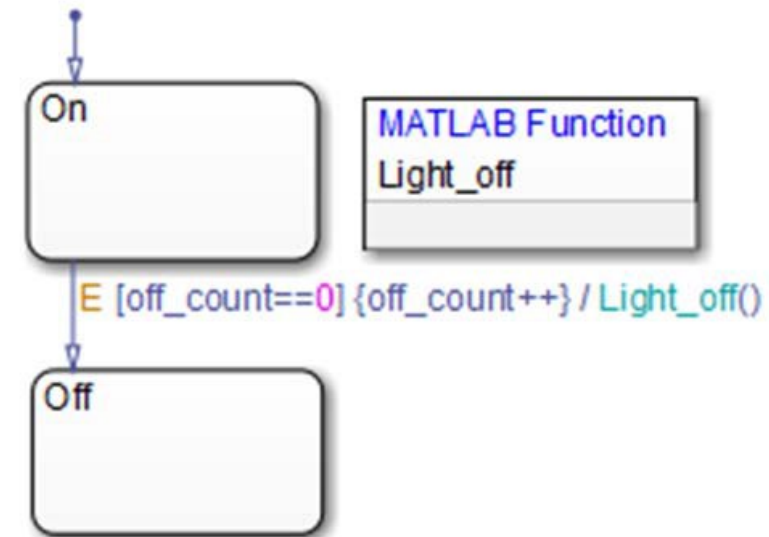
Notación de **ETIQUETAS** de Transición:

Formato general

**event**[**condition**]{**condition\_action**} /  
**transition\_action**

Etiquetas de transición válidas:

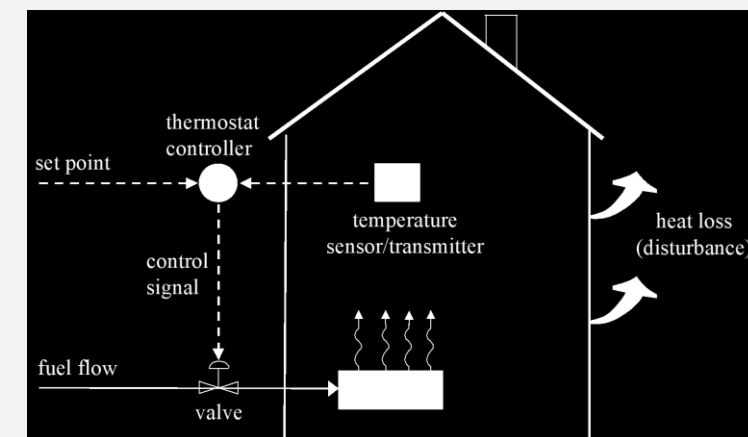
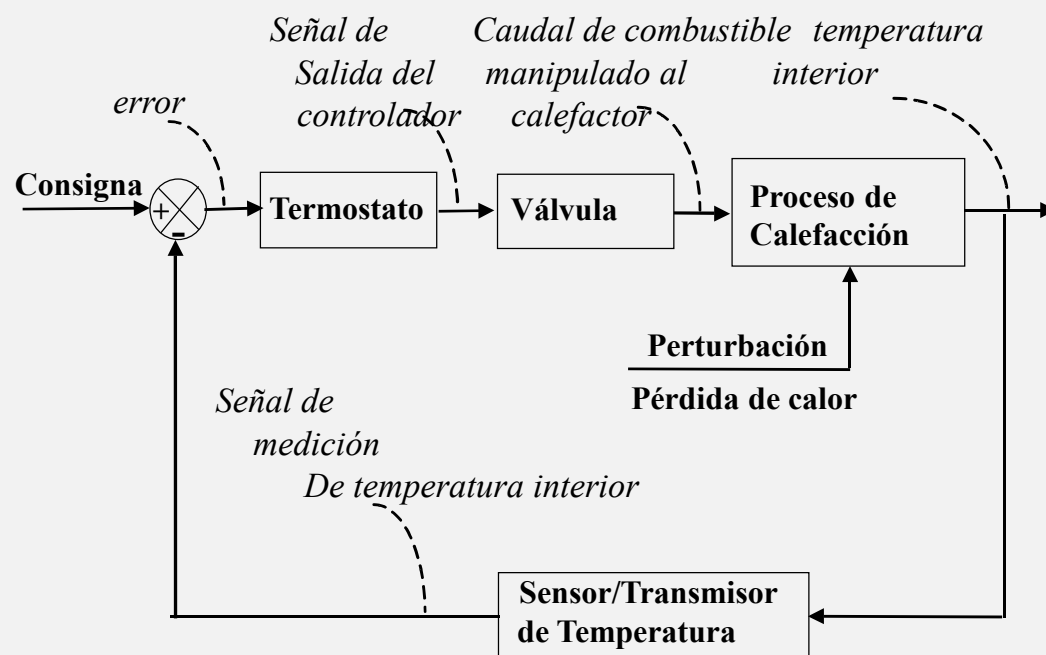
**Combinaciones posibles**



| Transition Label    | Is Valid If...                              |
|---------------------|---------------------------------------------|
| Event only          | That event occurs                           |
| Event and condition | That event occurs and the condition is true |
| Condition only      | Any event occurs and the condition is true  |
| Action only         | Any event occurs                            |
| Not specified       | Any event occurs                            |

# TERMOSTATO para Calefacción

## Esquema general y componentes del Sistema:



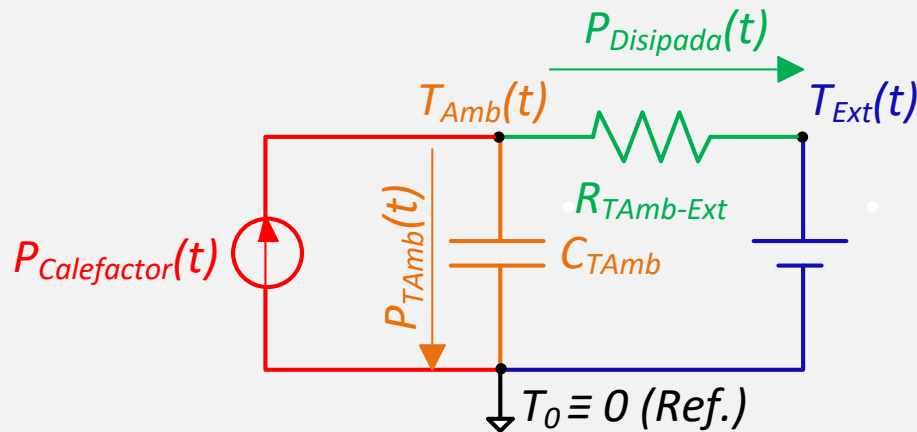
## Objetivo de Control:

Regular la temperatura ambiente interior alrededor de un valor de consigna seleccionable, frente a bajas temperaturas invernales, mediante control ON/OFF con Termostato.

# TERMOSTATO para Calefacción

*Modelo simplificado de Planta: Sistema Térmico (SDCTC)*

Circuito térmico equivalente:



Modelo dinámico (Ec. Estado):

$$P_{Calefactor}(t) [W] = P_{TAmb}(t) + P_{Disipada}(t) [W]$$

$$P_{Calefactor}(t) [W] = C_{TAmb} \cdot \frac{dT_{Amb}(t)}{dt} + \frac{1}{R_{TAmb-Ext}} \cdot (T_{Amb}(t) - T_{Ext}(t)) [W]$$

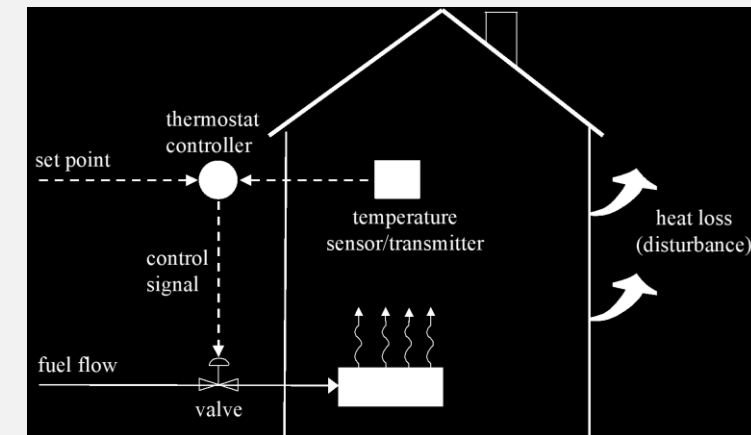


Diagrama de bloques de Estado: Hacer → Implementar en Simulink → **Simular!**

# TERMOSTATO para Calefacción

## Modelo simplificado del Controlador: Termostato (DEDS)

Ley de Control Básica (ON/OFF Automático):

$$\text{si } T_{Amb} < T_{Consigna} - T_{Delta} \Rightarrow Ctrl ON \Rightarrow \\ \Rightarrow P_{calefactor}(t) = P_{ON}[W];$$

$$\text{si } T_{Amb} > T_{Consigna} + T_{Delta} \Rightarrow Ctrl OFF \Rightarrow \\ \Rightarrow P_{calefactor}(t) = 0[W];$$

Implementación: Relé (Histéresis)

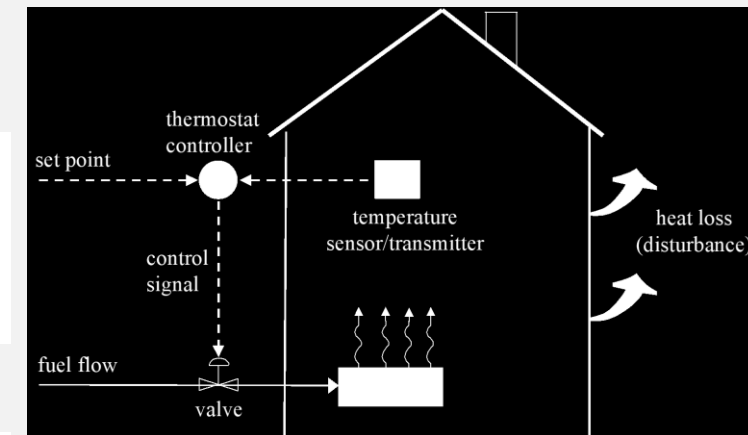
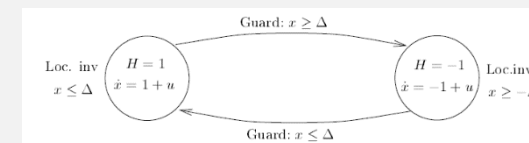
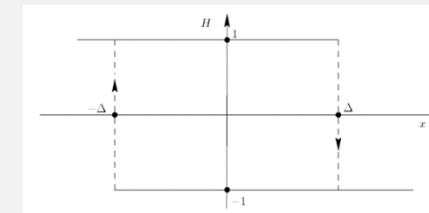
Hacer (agregar a Diagrama de Bloques de Estado);

→ Implementar en Simulink → **Simular!**

→ Agregar **interruptor Manual** de encendido/apagado;

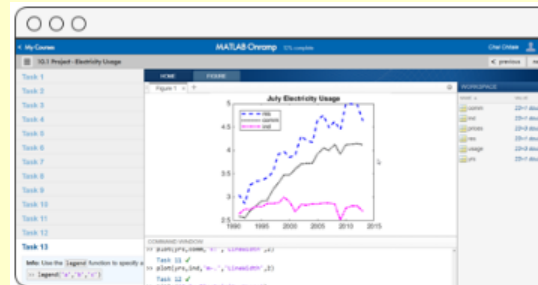
→ Plantear **Diagrama de Transición de Estados** (Autómata o Máquina de Estados Finitos);

→ Implementar en Stateflow → **Simular!** → **Identificar Características**



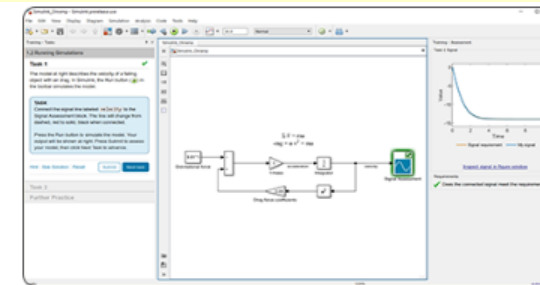
# Introducción a MATLAB / Simulink

**MATLAB and Simulink** *free* online **Tutorials** (requiere registrar cuenta propia libre)  
<https://www.mathworks.com/support/learn-with-matlab-tutorials.html>



MATLAB Onramp

Quickly learn the essentials of MATLAB®.



Simulink Onramp

Learn to create, edit, and troubleshoot Simulink® models.

1. MATLAB Onramp (ver video introd.: 1:18 min. → **tutorial: 2 hs.**, a ritmo propio)  
<https://www.mathworks.com/learn/tutorials/matlab-onramp.html>

2. Simulink Onramp (ver video introd.: 1:41 min. → **tutorial: 3 hs.**, a ritmo propio)  
<https://www.mathworks.com/learn/tutorials/simulink-onramp.html>

3. Stateflow Onramp (ver video introd.: 1:23 min. → **tutorial: 2:25 hs.**, a ritmo propio)  
<https://www.mathworks.com/learn/tutorials/stateflow-onramp.html>

# Introducción a Stateflow

**MATLAB and Simulink** *free* online **Tutorials** (requiere registrar cuenta propia libre)

<https://www.mathworks.com/support/learn-with-matlab-tutorials.html>

**Stateflow Onramp** (ver video introd.: 1:23 min. → **tutorial: 2:25 hs.**, a ritmo propio)

<https://www.mathworks.com/learn/tutorials/stateflow-onramp.html>

**STATEFLOW Tutorials:**

<https://www.mathworks.com/help/stateflow/getting-started.html>

**Videos:**

**Getting Started with Stateflow, E. Marquez (Mathworks) 2021-01-26**

<https://www.youtube.com/watch?v=cXnT5-zY3YI> (13 min.)

**Getting Started with Stateflow, M. Carone (Mathworks) 2017**

**Part 1:** States, Transitions, flow charts.

<https://www.youtube.com/watch?v=thBxzulFuyg> (28 min.)

**Part 2:** Parallel States (Concurrency), Truth Tables, Temporal logic.

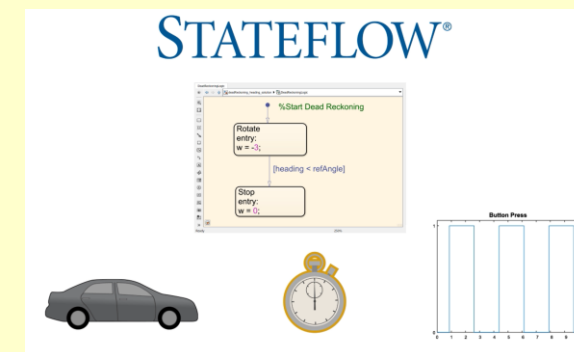
<https://www.youtube.com/watch?v=jvSjBDnrbxE> (27 min.)

**Part 3:** Embedded MATLAB, Simulink functions, Events.

<https://www.youtube.com/watch?v=64iuG25g-Og> (26 min.)

**Introduction to Stateflow** (versión compacta y revisada, 2017)

<https://www.youtube.com/watch?v=xWWDeYYix1I> (55 min.)



# Introducción a Stateflow

**MathWorks Advisory Board (MAB) Guidelines:**

<https://www.mathworks.com/solutions/mab-guidelines.html>

**MAB Modeling Guidelines:** <https://www.mathworks.com/help/simulink/mab-modeling-guidelines.html>

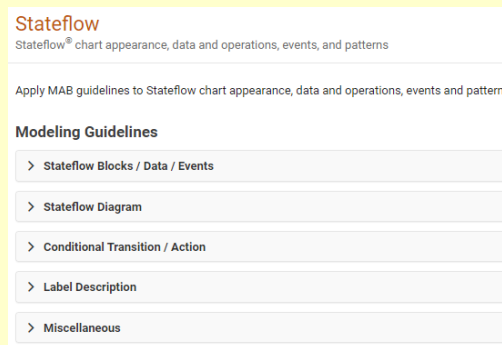
**Best practices** for

- developing,
- maintaining and
- documenting

control algorithm **models**

used in the development of **production software**.

**STATEFLOW:**



<https://www.mathworks.com/help/simulink/stateflow.html>

## MathWorks Advisory Board (MAB) Guidelines

The MAB Guidelines are a set of modeling guidelines developed by an independent industry working group for the usage of MATLAB, Simulink, Stateflow and Embedded Coder. The guidelines are the culmination of years of engineering expertise and best practices for developing, maintaining and documenting control algorithm models used in the development of production software.

The MAB guidelines are an important basis for project success and teamwork - both in-house and when cooperating with partners or subcontractors. Observing the guidelines is one key prerequisite to achieve:

- System integration without problems
- Well-defined interfaces
- Uniform appearance of models, code and documentation
- Reusable models
- Readable models
- Problem-free exchange of models
- A simple, effective process
- Professional documentation
- Understandable presentations
- Fast software changes
- Cooperation with subcontractors
- Handing over of research or predevelopment projects to product development

- Model Checks for MAB Guidelines
- Model Checks for JMAAB Guideline
- ⌚ Edit-Time Checking to Find Compliance Issues Earlier (1:28)
- ⌚ Modeling Standards Compliance (2:56)

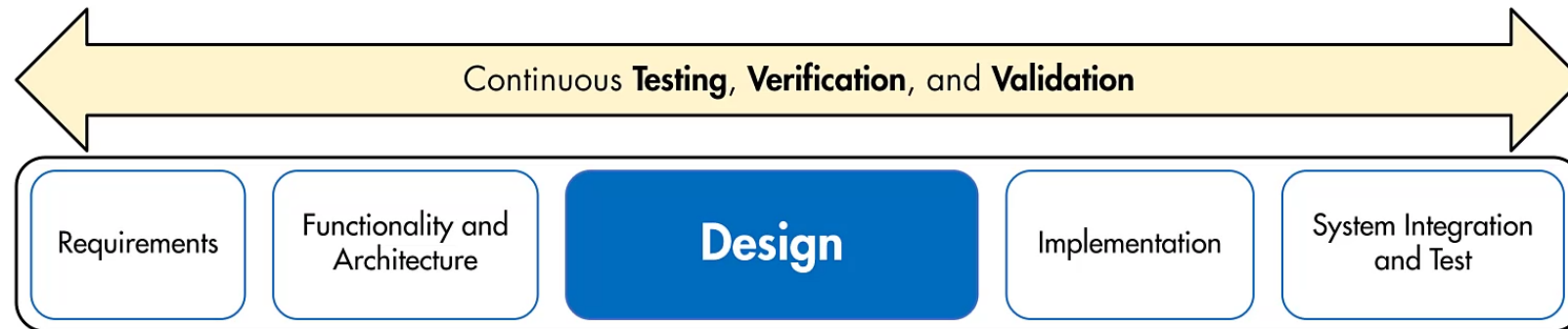


# Introducción a Stateflow

*MathWorks Advisory Board (MAB) Guidelines:*

<https://www.mathworks.com/solutions/mab-guidelines.html>

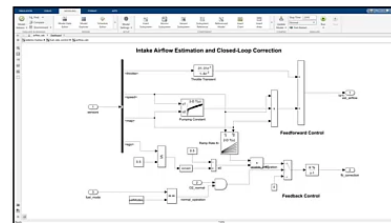
## Modeling Standards Compliance



**ISO 26262**

**DO-178C**

**DO-254**



**Develop**



**Maintain**



**Document**



**IEC 61508**

**ISO 25119**

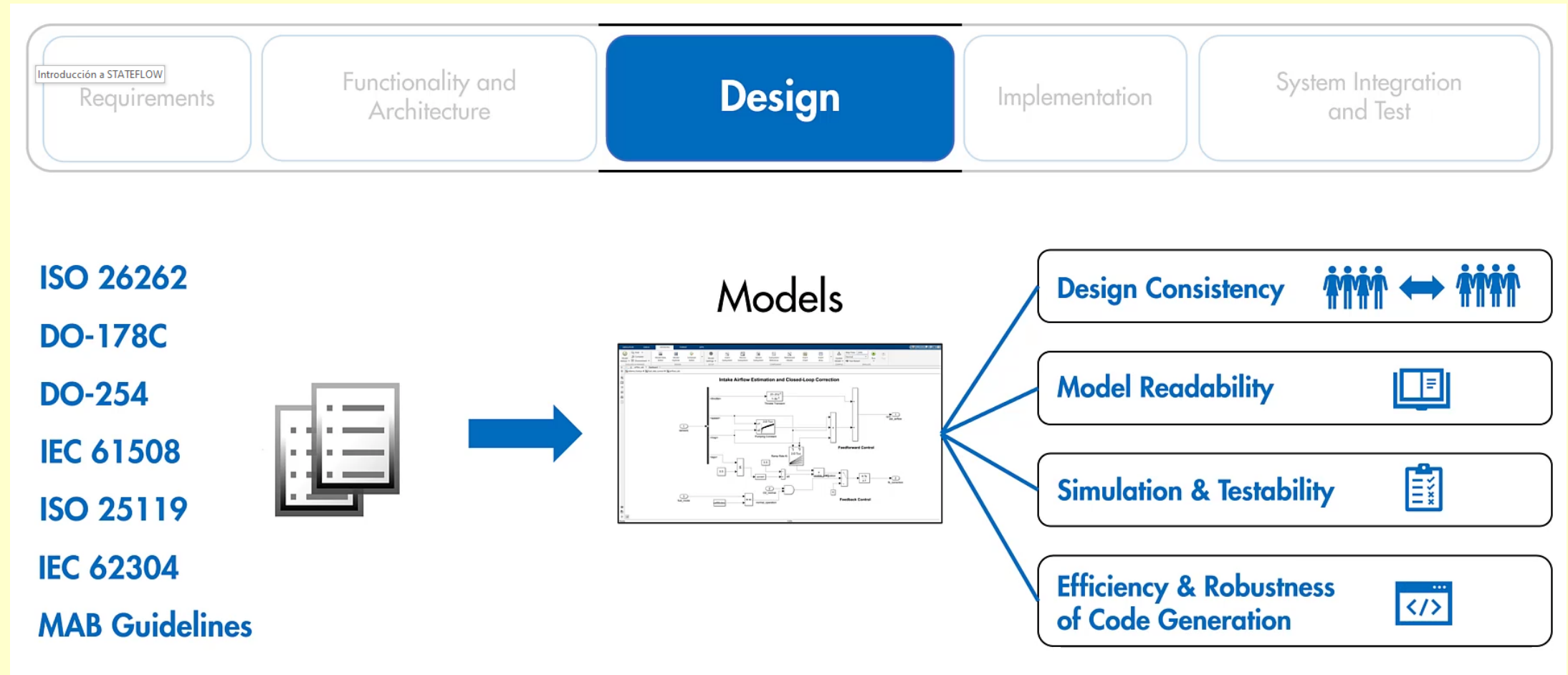
**IEC 62304**

**MAB Guidelines**

# Introducción a Stateflow

*MathWorks Advisory Board (MAB) Guidelines:*

<https://www.mathworks.com/solutions/mab-guidelines.html>





# Unidad 1

## Autómatas, Sistemas Discretos y Sistemas Híbridos

---

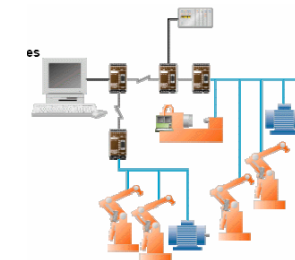


# AUTÓMATAS y CONTROL DISCRETO

## 1. *Autómatas, Sistemas Discretos y Sistemas Híbridos*

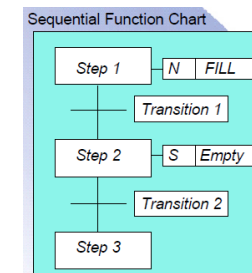
### A. Concepto y Aplicaciones

- Automatización y Control Industrial: Concepto y esquema general. Componentes y estructura.
- Aplicaciones Mecatrónicas → Control automático secuencial + control de tiempo discreto.



### B. Fundamentos de Autómatas Secuenciales de Control

- Sistemas secuenciales: estados discretos activados por eventos. Conceptos.
- Autómatas finitos. Modelado y representación. FSM. STATECHARTS.
- Simulación. STATEFLOW.

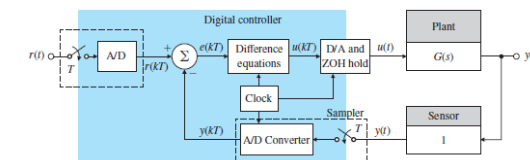


### C. Petri Nets (ISO/IEC 15909). GRAFCET (IEC 60848)

- Metodologías de representación y diseño. Redes de Petri.
- Lenguaje GRAFCET. Estructuras Jerarquizadas.

### D. Sistemas Híbridos. Práctica de Modelado y Simulación

- Modelado y Simulación en STATEFLOW. Representación en GRAFCET.



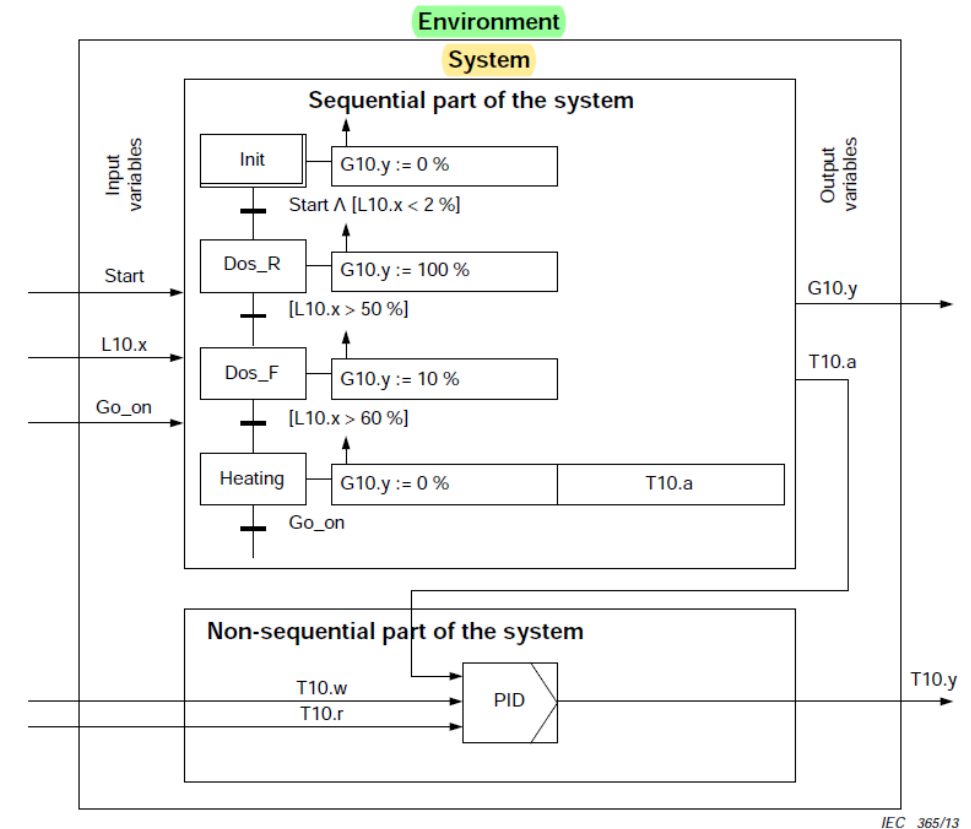
# AUTÓMATA de Control: Control Lógico Secuencial

Subsistema de Control Lógico Secuencial  
(ej. Nivel de Control Supervisor)

- Representación Gráfica: Diagrama de Bloques
- Lógica Secuencial** (Memoria → **Estados** Discretos: **Etapas, Modos; Transiciones** entre Etapas: Eventos (Dinámica=evolución)

ESPECIFICACIÓN de Comportamiento lógico **secuencial**  
(Estados Discretos activados por Eventos):

- Autómata Finito (FSM) → STATE CHART**
- GRAFCET (IEC 60848)**



L10.x tank level

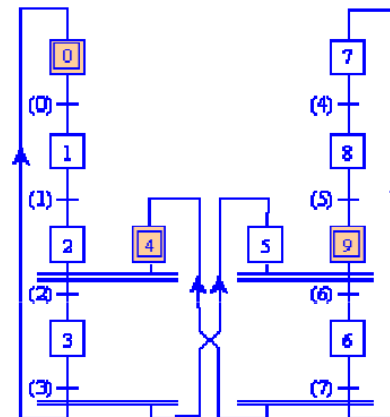
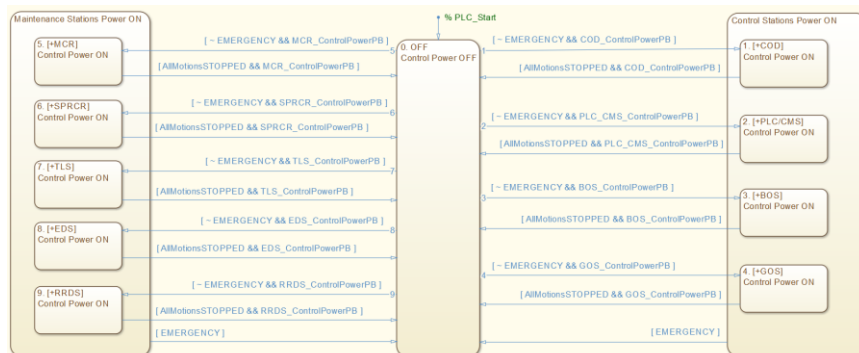
T10.a temperature loop – automatic mode

T10.r temperature loop – measured value

G10.y dosing valve – position

T10.w temperature loop – setpoint

T10.y temperature loop – manipulated value



# IEC: International Electrotechnical Commission



- **The world of IEC:** An animated video that shows the **breadth and scope of IEC work**. The crucial role of International standards and Conformity assessment in making the world **safer and more efficient**.

<https://youtu.be/mrh3gnHBo9Y> (Video 1:27 min - 2020-08)

- **The importance of IEC International Standards:**

A look at the importance of IEC standards throughout "The Electric Century".

<https://www.youtube.com/watch?v=0Byh7VLHGDY> (Video 6:38 min)

## IEC 60050 - ELECTROPEDIA: Electrotechnical Vocabulary

<https://www.electropedia.org/>

Algunas Normas IEC (IEC Standards) de **aplicación en AyCD**:

- **IEC 60848: GRAFCET Specification Language for Sequential Function Charts.**
- **IEC 61131: Programmable Controllers.**

Etc...

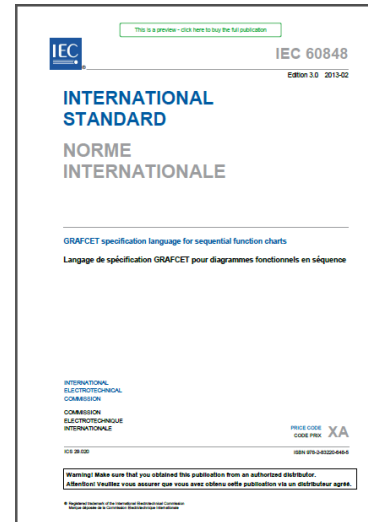
# GRAFCET: diagrama funcional secuencial

GRAFCET: ( **G**raphe Fonctionnel de **C**ommande **É**tape-**T**ransition = **G**rafo **F**uncional de **C**ontrol **E**tapa-**T**ransición ):

Lenguaje gráfico de **ESPECIFICACIÓN** para descripción funcional (= **modelo**) del **comportamiento lógico determinista** de la **parte secuencial** de sistemas de automatización y control.

**IEC 60848 Ed. 3.0 (2013-02): GRAFCET Specification Language for Sequential Function Charts**  
(STABILITY: 2028)

→ IEC 61131-3: **SFC** (Sequential Function Chart): Lenguaje gráfico de **Programación**



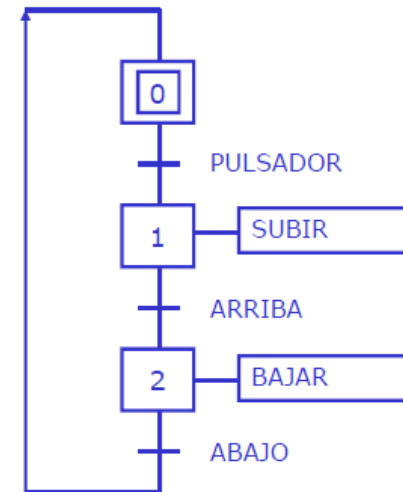
## GRAFCET: Componentes Básicos

### Estructura

- ETAPAS o PASOS (E) →
- TRANSICIONES (T) ←
- Enlaces dirigidos (E→T ; T→E)  
→ **Evolución secuencial** de Etapa activa  
(desde Etapa Inicial)

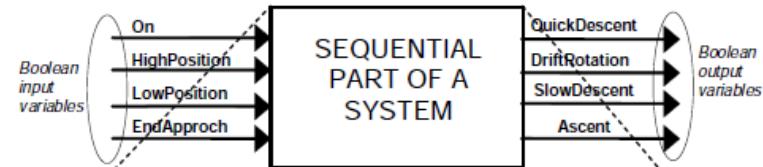
### Interpretación

**Acciones (A)**  
**Condiciones de Transición (C)**



# GRAFCET: diagrama funcional secuencial

IEC 60848 Ed. 3.0 (2013-02):  
GRAFCET Specification Language for Sequential Function Charts



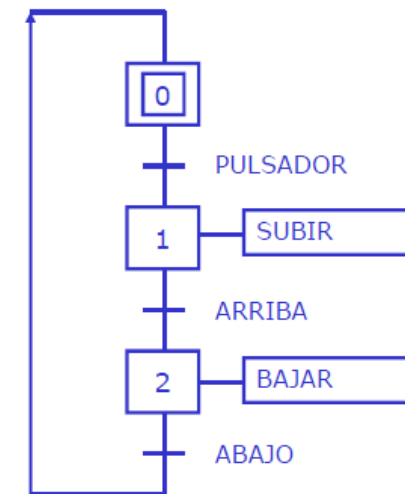
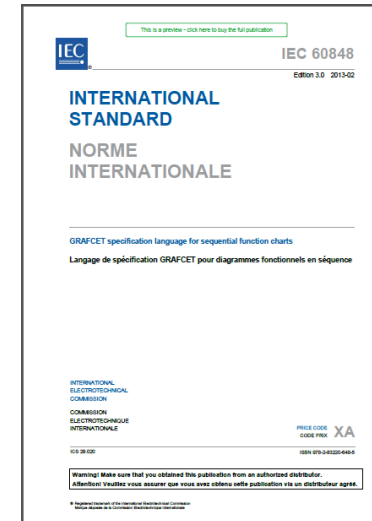
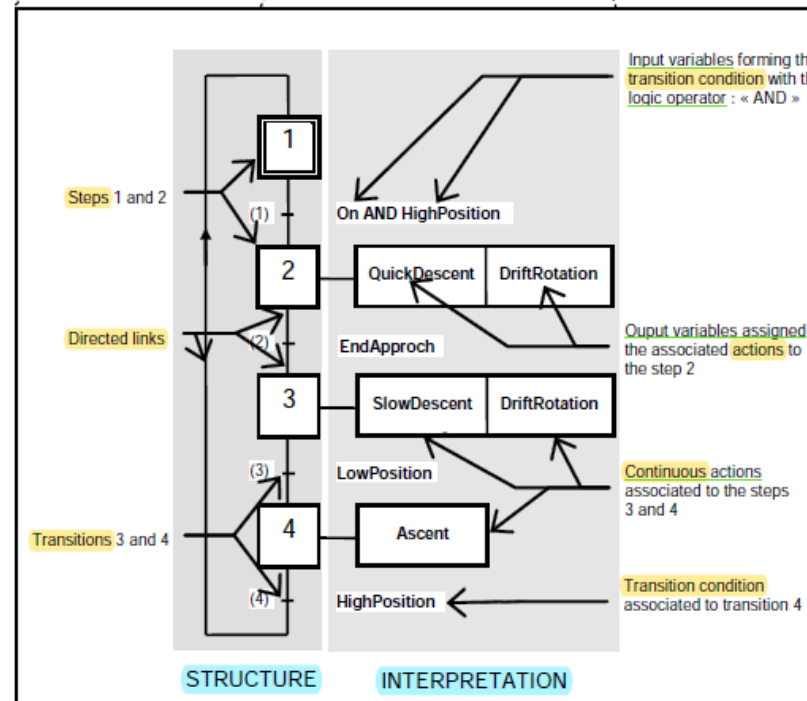
## Componentes básicos

### Estructura

- ETAPAS o Pasos (E)
- TRANSICIONES (T)
- Enlaces dirigidos ( $E \rightarrow T$  ;  $T \rightarrow E$ )
- **Evolución secuencial**  
de Etapa activa (desde Etapa Inicial)

### Interpretación

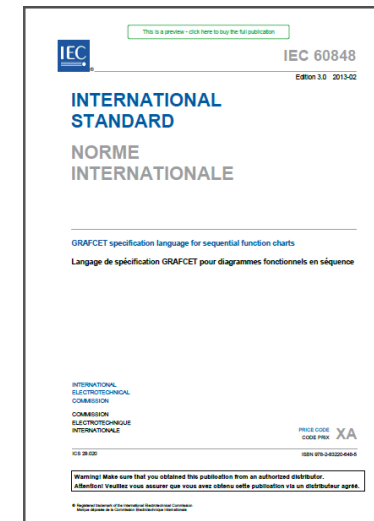
- E → **Acciones (A)**
- T ← **Condiciones de Transición (C)**



# GRAFCET: conceptos y componentes

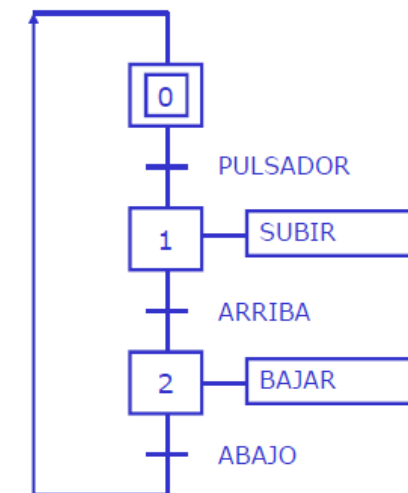
**Estructura:** describe las evoluciones posibles de la situación del sistema.

- **(E) Etapa o Paso (Step)=Estado Discreto** (**Inactivo** ó **Activo**): define la **Situación** (**Estado o Modo de Operación**) actual de parte secuencial del sistema.  
(**Concurrencia**  $\Leftrightarrow$  2 o más etapas activas simultáneamente = **Situación**)
- **(T) Transición** (**invalidada** ó **validada = habilitada**  $\rightarrow$  **ejecutable = despejable = franqueable**): indica la **posibilidad de evolución** entre dos o más etapas.
- **Enlace dirigido:** indica el camino de evolución posible E (1 ó +)  $\rightarrow$  T ó T  $\rightarrow$  E (1 ó +) [**Regla de SINTAXIS**].



**Interpretación:** relaciona variables entrada (C)  $\rightarrow$  Estructura  $\rightarrow$  (A) var. salida.

- **Evento de entrada:** cambio de estado lógico = flanco ( $\uparrow$  o  $\downarrow$ ) de al menos una de las variables de entrada booleanas.
- **Evento interno:** cambio en la situación del sistema: activación/desactivación de etapa, despeje de transición.
- **(C) Condición de Transición** (**Falsa** ó **Verdadera**; asociada a cada Transición): evento, o resultado lógico de expression booleana (variables de entrada o internas).
- **(A) Acción** sobre variable de salida o variable interna (asociada a una Etapa): asignación s/**estado** (acción **continua**) o asignación s/**evento** (acción **memorizada**).



# GRAFCET: Reglas de EVOLUCIÓN

## Regla 1: Situación inicial

Situación (= conjunto de Etapas **activas**) en **instante inicial**, i.e. al energizar el controlador secuencial → **elegir Situación inicial SEGURA**.

## Regla 2: Ejecución = Despeje = Franqueamiento de una Transición

Transición **validada = habilitada**  $\Leftrightarrow$  todas las Etapas inmediatas precedentes **activas**.  
Si Transición **validada = habilitada** y **Condición de Transición Verdadera**  $\Rightarrow$   
 $\Rightarrow$  Transición **ejecutada = despejada** inmediatamente (tiempo nulo).

## Regla 3: Evolución de Etapas activas

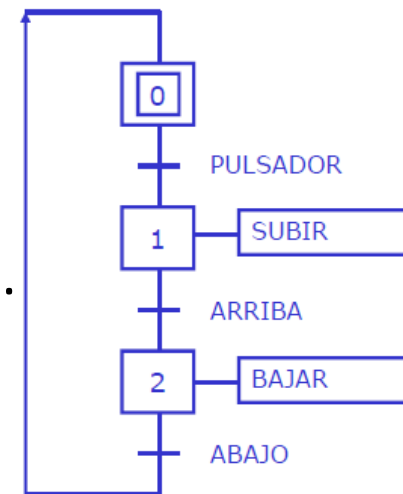
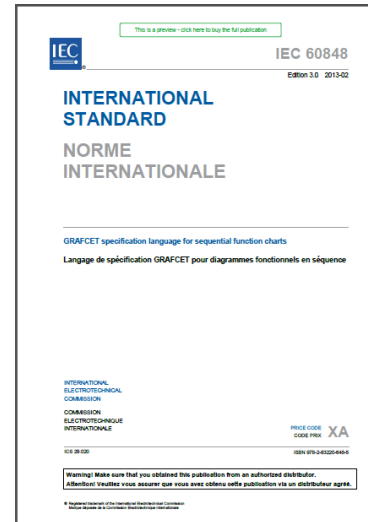
Ejecución = despeje de una Transición  $\Rightarrow$  **simultáneamente:**  
**activación** de todas las Etapas inmediatamente siguientes y  
**desactivación** de todas las Etapas inmediatamente precedentes.

## Regla 4: Evoluciones simultáneas (consistencia en t)

Varias Transiciones ejecutables simultáneamente son ejecutadas simultáneamente (**cambio instantáneo completo de situación** original a nueva situación, sin intermedios).

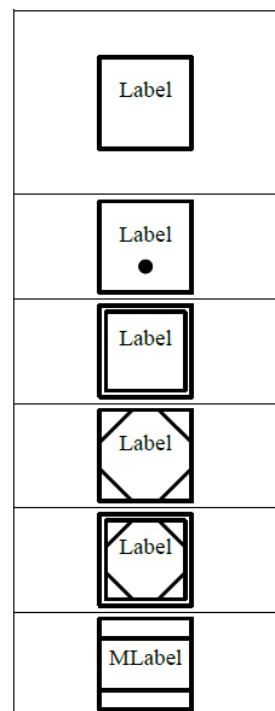
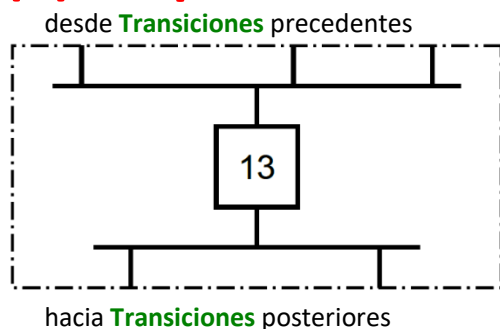
## Regla 5: Activación y desactivación simultánea de una Etapa

Si una Etapa activa es simultáneamente desactivada y activada  $\rightarrow$  **permanece activa**.



# GRAFCET: símbolos, representación gráfica

## (E) Etapas o Pasos



### Etapa:

**XLabel** : variable de estado booleano  
(0: inactivo; 1: activo)

### Etapa con estado **activo**

**XLabel** = 1)

### Etapa **inicial**

(participa de la **Situación inicial**)

### Etapa **encapsulante**

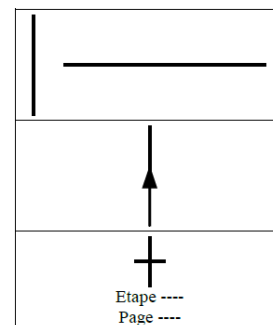
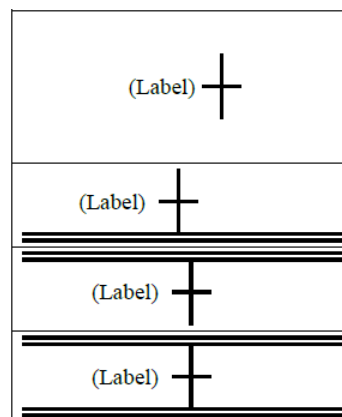
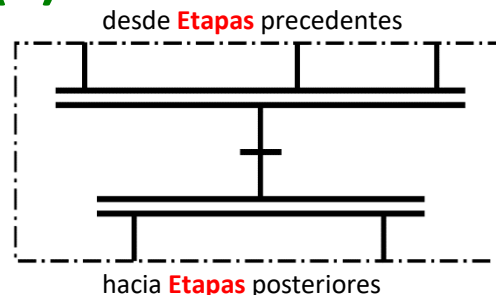
(contiene otras etapas «encapsuladas»)

### Etapa encapsulante **inicial**

### **Macro-Etapa**

(representa parte detallada de GRAFCET:  
«**expansión**» de la Macro-Etapa)

## (T) Transiciones



### **Transición:** línea perpendicular

**(Label)** : marcador opcional (izquierda)

### **Concurrencia:**

Activación de **secuencias paralelas**

**Sincronización** de **secuencias paralelas**

**Sincronización y activación** de **secuencias paralelas**

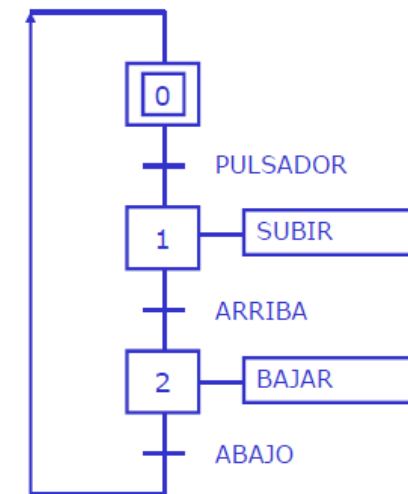
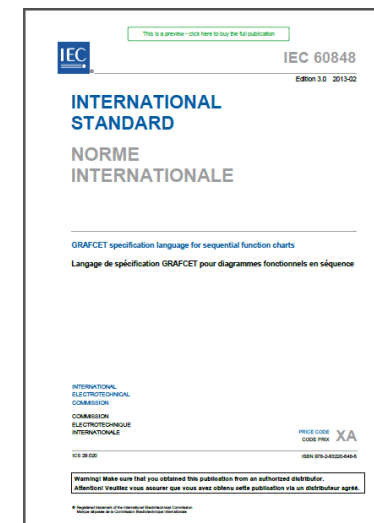
### **Enlace dirigido u orientado:**

Camino de evolución entre etapas (E→T;  
T→E)

### Enlace vertical **invertido**

### Marcador de Enlace

(si enlace debe interrumpirse)



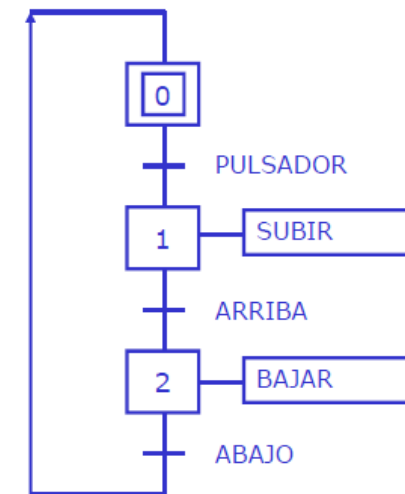
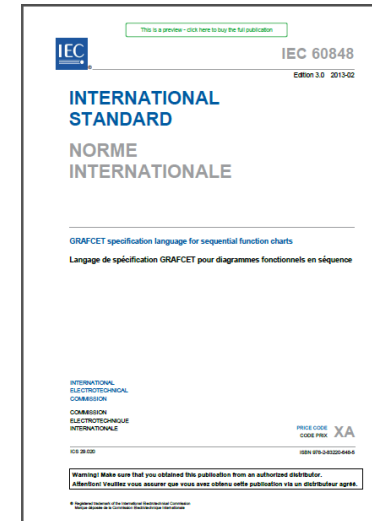
# GRAFCET: símbolos, representación gráfica

## (C) Condiciones (→ Transiciones)

|  |                                                                                                                                                             |
|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <b>Condición de Transición:</b><br>Proposición lógica (V / F): expresión o variable booleana → <b>condición de despeje de transición</b> → <b>evolución</b> |
|  | <b>Condición siempre V</b>                                                                                                                                  |
|  | <b>Flanco ascendente</b><br>(0→1) de variable o expresión lógica                                                                                            |
|  | <b>Flanco descendente</b><br>(1→0) de variable o expresión lógica                                                                                           |
|  | <b>Flancos no simultáneos</b>                                                                                                                               |
|  | <b>Condición temporizada</b><br>Condición: V después de t1 desde que var ↑V → F después de t2 desde que var ↓F                                              |
|  | <b>Simplificación usual</b><br>Condición: F cuando var ↓F                                                                                                   |
|  | <b>Valor booleano de un predicado</b><br>Ej. $[3.25 < Pos < 5.84]$                                                                                          |

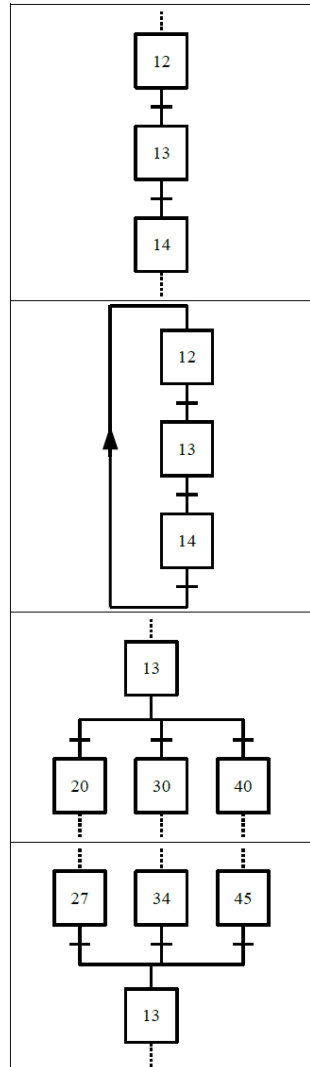
## (A) Acciones (← Etapas)

|  |                                                                                                                                                   |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <b>Acción continua: (default)</b><br>Siempre asociada a una etapa<br>Acción (en gral.) : asignación de una salida (mientras la etapa esté activa) |
|  | <b>Varias acciones</b><br>asociadas a una misma etapa                                                                                             |
|  | <b>Condición de asignación (estable)</b><br>Proposición lógica → condiciona la ejecución de acción.                                               |
|  | <b>Acción dependiente de Evento</b><br>Si Etapa activa y ocurre evento → ejecución de acción.                                                     |
|  | <b>Acción temporizada</b><br>Si Etapa activa y temp. V → ejecución de acción.                                                                     |
|  | <b>Acción retardada</b><br>Si Etapa activa y t1 después de activación → ejecución.                                                                |
|  | <b>Acción limitada en tiempo</b><br>Si Etapa activa y t1 después de activación → ejecución.                                                       |
|  | <b>Afectación o memorización de valor a var.</b><br>Ej. $C := C + 1$ (ante evento)                                                                |



# GRAFCET: estructuras de secuencias

## Estructuras básicas

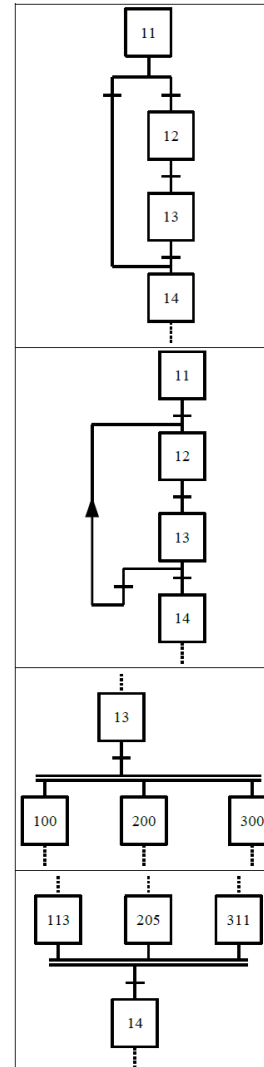


**Secuencia:** Sucesión de etapas  
Cada etapa: **una sola transición posterior** (excepto la última); y **una sola transición precedente** (excepto la primera).

**Ciclo** de una sola Secuencia

**Selección** de Secuencias: Elección de caminos de evolución **NO CONCURRENTES** (selección **inclusiva** o **exclusiva**, dependiendo de definición de Condiciones y posible simultaneidad → **priorizar**)

**Reagrupamiento** de Secuencias: Secuencia única luego de selección de secuencias **NO CONCURRENTES** (en general exclusivas)



**Salto** de etapas

Caso particular de Selección de secuencias (en general, **exclusivas** para funcionamiento correcto)

**Repetición** de secuencia parcial

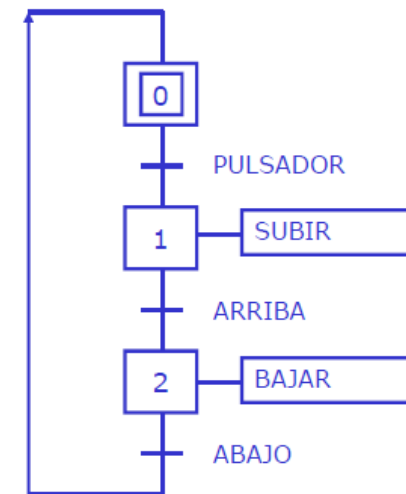
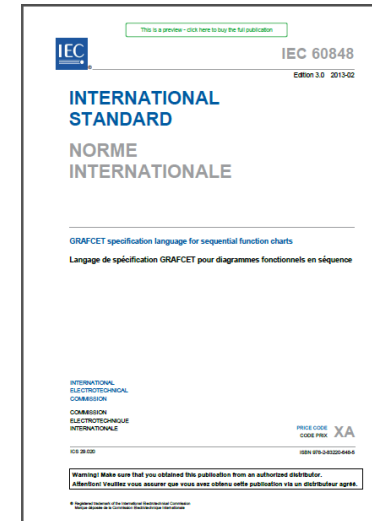
Caso particular de Selección de secuencias (en general, **exclusivas** para funcionamiento correcto)

Activación de **secuencias paralelas**

Ejecución simultánea (**conurrencia**) de varias secuencias a partir de una sola transición.

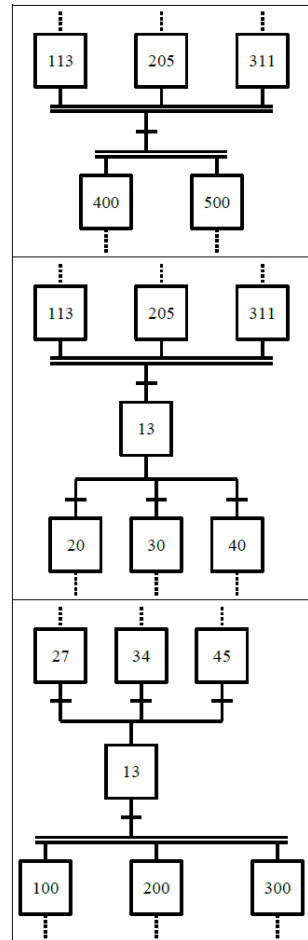
**Sincronización** de secuencias **CONCURRENTES**

Transición validada  $\leftrightarrow$  todas las etapas de entrada están activas.



# GRAFCET: estructuras de secuencias

## Estructuras básicas

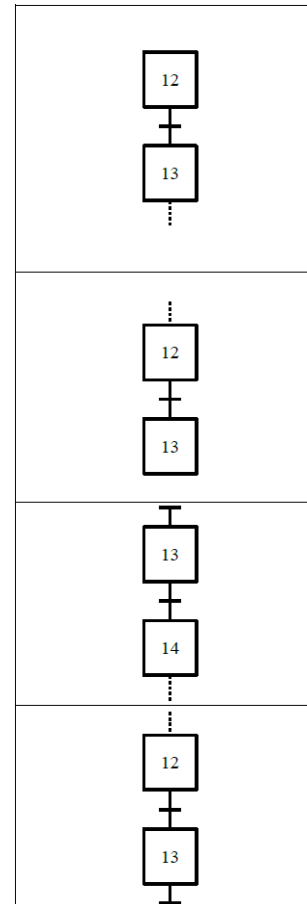


**Sincronización** seguida de **activación de secuencias paralelas**:  
Una sola transición, validada cuando todos las etapas anteriores activas.

**Sincronización** seguida de **selección de secuencias**:  
Etapla intermedia E13 **obligatoria**.

**Reagrupamiento** seguido de **activación de secuencias paralelas**:  
Etapla intermedia E13 **obligatoria**.

## Estructuras particulares



### Etapla fuente

No la precede ninguna transición

- Etapa inicial de GRAFCET (activada al energizar)
- Etapa inicial dentro de **Macroetapa**.

### Etapla final

No la sucede ninguna transición.

Se desactiva mediante

- Orden forzada
- Etapa final dentro de **Macroetapa**.

### Transición fuente

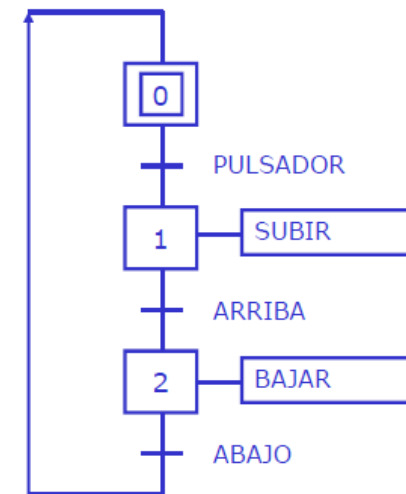
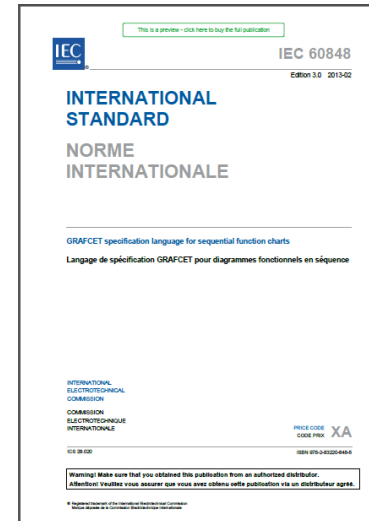
No la precede ninguna etapa.

Siempre está validada.

### Transición final

No la sucede ninguna etapa.

Despeje causa únicamente la desactivación de la etapa precedente.

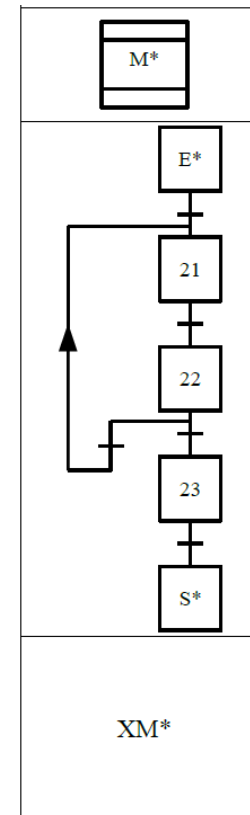


# GRAFCET: estructuras jerarquizadas

Automatismos **complejos** → especificación estructurada por niveles:

1. **Partición** del diagrama GRAFCET (Macro-etapas)
2. **Jerarquización** de tareas (Forzamiento y Encapsulamiento)

## Estructuración por **MACRO-ETAPA**



### Macro-Etapa M\*

Representa parte detallada: **expansión** de macro-etapa

### Expansión de la Macro-Etapa M\*

Partición de GRAFCET con:

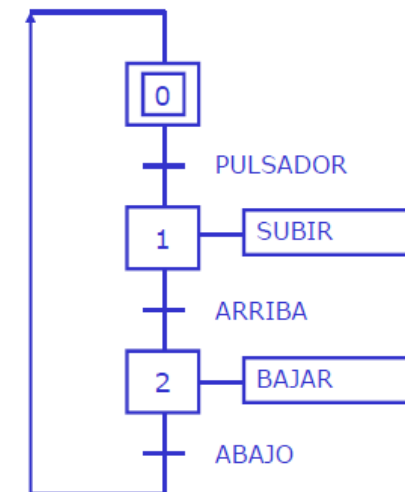
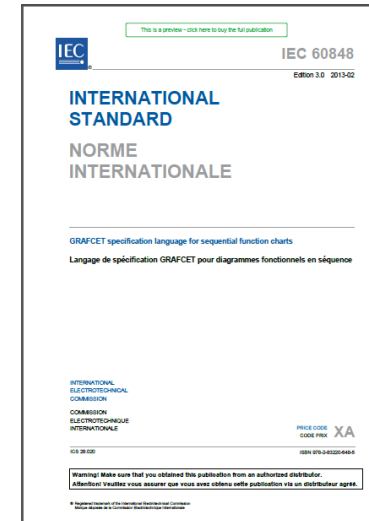
E\*: etapa de **entrada** (se activa simultáneamente con la activación de M\*)

S\*: etapa de **salida** (la transición posterior a M\* es validada cuando S\* activa → despeje de dicha transición produce desactivación simultánea de M\* y S\*)

### Variable de Macro-Etapa

XM# =1: M\* activa, y al menos una etapa de su expansión está activa

XM# =0: M\* inactiva y todas las etapas de su expansión inactivas



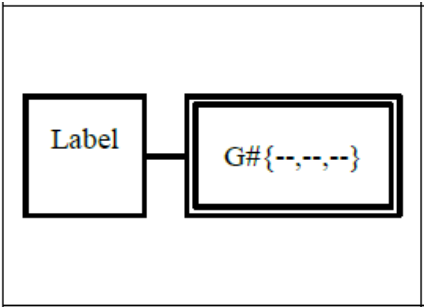
# GRAFCET: estructuras jerarquizadas

Automatismos **complejos** → especificación estructurada por niveles:

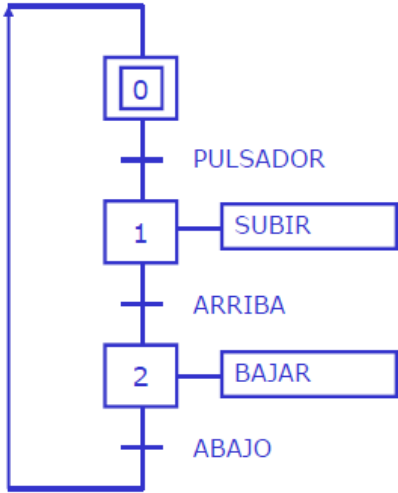
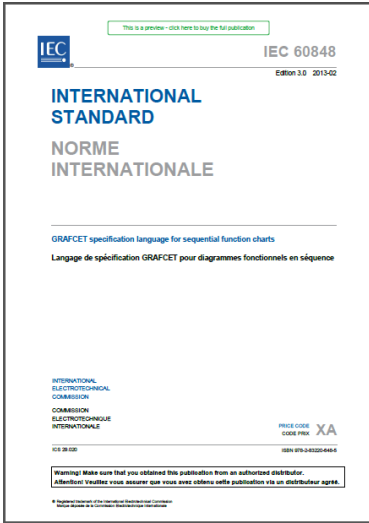
- 1. **Partición** del diagrama GRAFCET (Macro-etapas)
- 2. **Jerarquización** de tareas (Forzamiento y Encapsulamiento)

|                       |                                                                                                                                                |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| G#                    | <b>GRAFCET parcial</b><br>#: etiqueta o nombre                                                                                                 |
| XG#                   | <b>Variable</b> del GRAFCET parcial<br>XG# =1: al menos una etapa está activa<br>XG# =0: todas las etapas inactivas → GRAFCET parcial inactivo |
| G#{étape, étape, ...} | <b>Situación</b> del GRAFCET parcial<br>Conjunto de sus <b>etapas activas</b> actuales                                                         |
| G#{*}                 | <b>Situación actual</b> del GRAFCET parcial<br>En la que se encuentra en un instante dado                                                      |
| G#{ }                 | <b>Situación vacía</b> del GRAFCET parcial<br>Ninguna de sus etapas está activa                                                                |
| G#{INIT}              | <b>Situación inicial</b> del GRAFCET parcial<br>En instante de energización                                                                    |

## Estructuración por **FORZAMIENTO** de GRAFCET parcial



**Acción de forzamiento**  
Orden interna que permite **imponer una situación** particular a un GRAFCET parcial jerárquicamente inferior (al estar **activa la etapa correspondiente del GRAFCET superior**).  
El GRAFCET parcial forzado no puede evolucionar mientras dure la orden de forzado (está congelado)



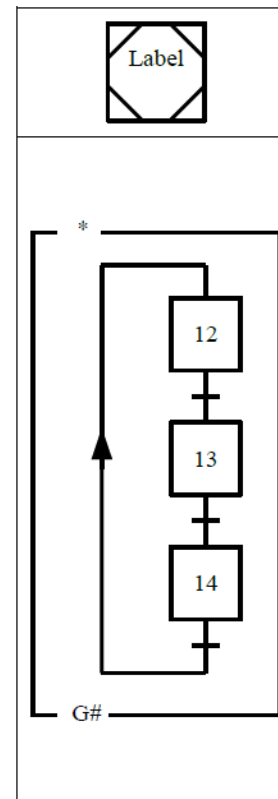
# GRAFCET: estructuras jerarquizadas

Automatismos **complejos** → especificación estructurada por niveles:

1. **Partición** del diagrama GRAFCET (Macro-etapas)
2. **Jerarquización** de tareas (Forzamiento y Encapsulamiento)

|                       |                                                                                                                                                |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| G#                    | <b>GRAFCET parcial</b><br>#: etiqueta o nombre                                                                                                 |
| XG#                   | <b>Variable</b> del GRAFCET parcial<br>XG# =1: al menos una etapa está activa<br>XG# =0: todas las etapas inactivas → GRAFCET parcial inactivo |
| G#{étape, étape, ...} | <b>Situación</b> del GRAFCET parcial<br>Conjunto de sus <b>etapas activas</b> actuales                                                         |
| G#{*}                 | <b>Situación actual</b> del GRAFCET parcial<br>En la que se encuentra en un instante dado                                                      |
| G#{ }                 | <b>Situación vacía</b> del GRAFCET parcial<br>Ninguna de sus etapas está activa                                                                |
| G#{INIT}              | <b>Situación inicial</b> del GRAFCET parcial<br>En instante de energización                                                                    |

## Estructuración por **ENCAPSULAMIENTO** de GRAFCET parcial



### Etapa Encapsulante

Contiene otras etapas «encapsuladas»

### Encapsulamiento X\*/G#

etapa **encapsulante** X\*

GRAFCET parcial **Encapsulado** G#

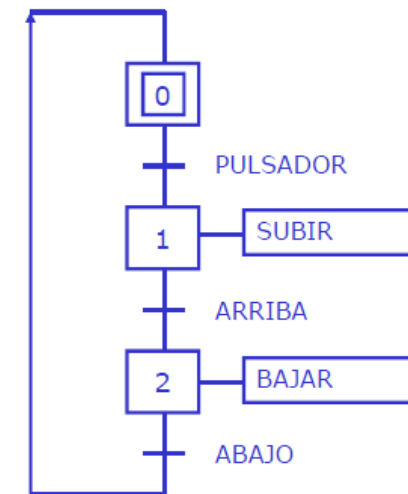
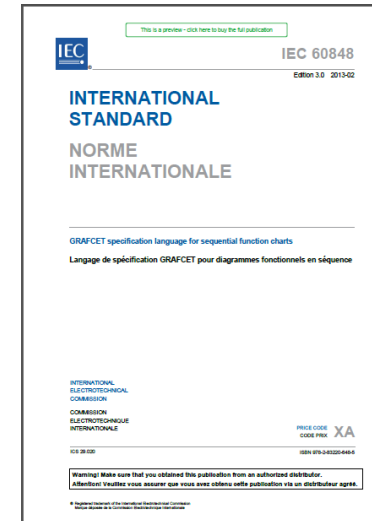
(G# podría contener una sola etapa: encapsulamiento elemental)

### Activación interna:

Etapa encapsulada inicial (\*) a la izquierda) se activa inmediatamente a la activación de X\*.

### Desactivación interna:

Desactivación de X\* implica desactivación de G#

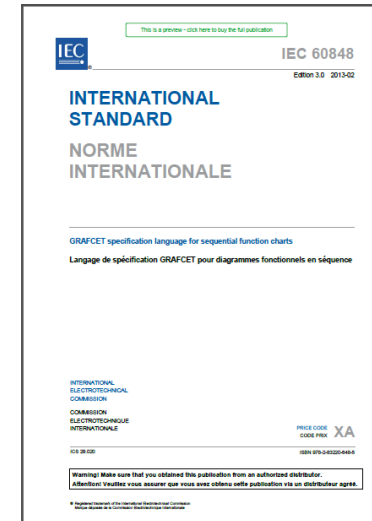


# GRAFCET: Norma IEC 60848

- **IEC 60848: GRAFCET Specification Language for Sequential Function Charts (→SFC)**

ver Documento de la Norma Ed. 3.0:

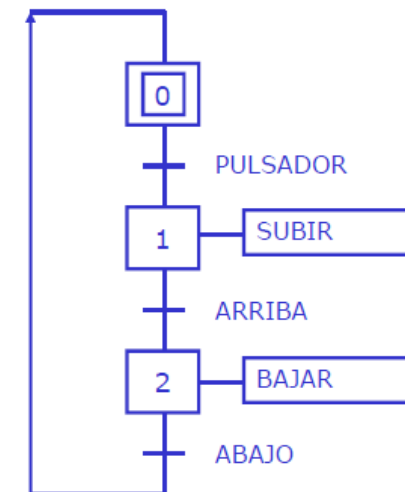
AyCD\_01b\_ **IEC 60848** \_Ed**3.0** \_2013-02\_  
**GRAFCET Specification Language for Sequential Function Charts.pdf**  
(114 págs.: 54 en inglés – 55 en francés)



IEC 60848 Ed. 3.0 :2013:

- Defines the **GRAFCET specification language** for the functional description of the behavior of the sequential part of a control system.
- Specifies the symbols and rules for the graphical representation (structure) of this language, as well as for its interpretation.
- Has been prepared for automated production systems of industrial applications.

This **edition 3.0** cancels and replaces the second edition published in 2002 and constitutes a global technical revision with the extended definition of the concept of variables introducing: **internal variable, input variable and output variable**.



# GRAFCET: Norma IEC 60848

- **IEC 60848:** GRAFCET Specification Language for Sequential Function Charts (→SFC)

## Herramienta de Software (a explorar)

### Grafcet-Studio

(basado en Grafcet DIN EN 60848)

MHJ Software for Automation Technology

<https://www.mhj-tools.com/>

<https://www.mhj-tools.com/?page=grafcet-studio>

(Versión de prueba 30 días disponible a pedido)

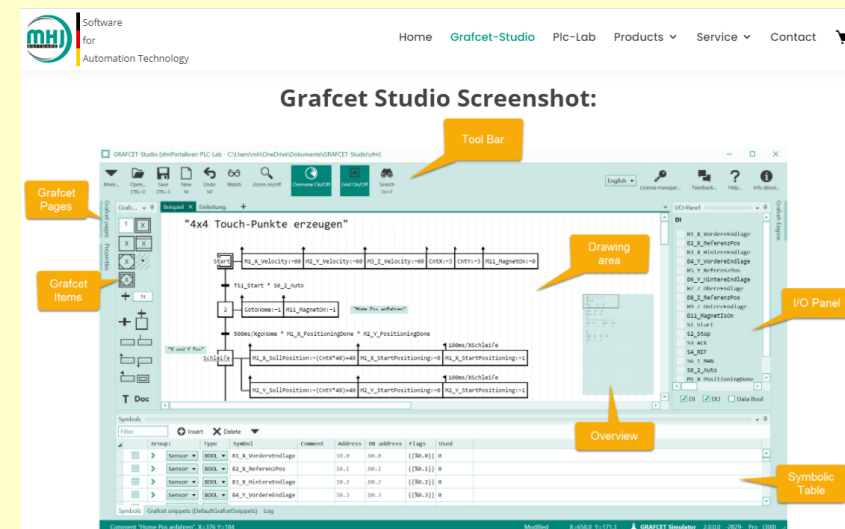
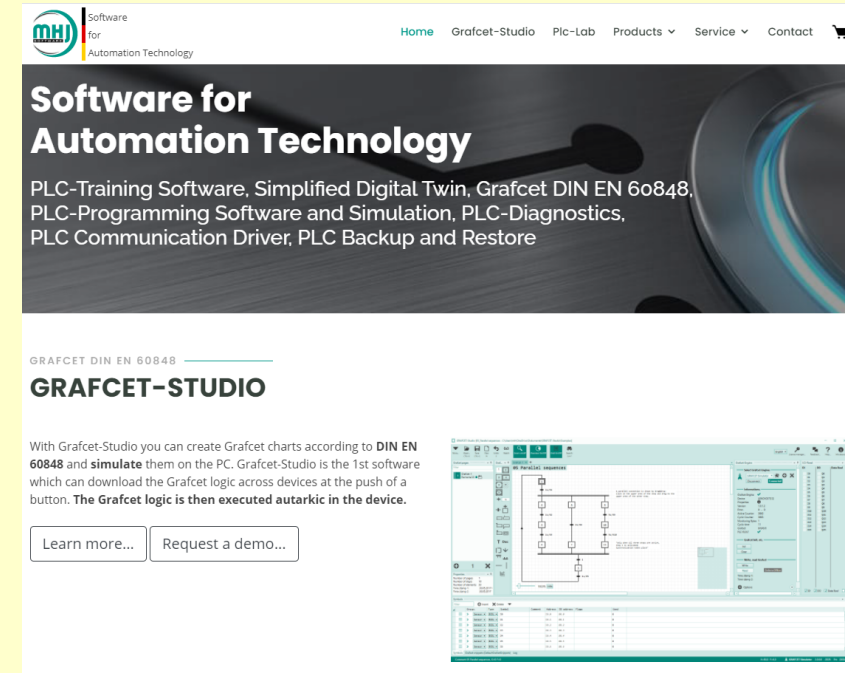
<https://www.tia-expert.com/grafcet-studio/version-demo/>

MANUALES, etc.: Wiki MHJ software

<https://www.mhj-wiki.de/de/mhj/handbuecher/>

<https://www.mhj-wiki.de/de/grafcet-studio/>

→ Explorar y comparar con CODESYS - SFC  
(lenguaje de PROGRAMACIÓN IEC 61131-3)



Grafcet-Studio is a modern software tool for the Grafcet standard DIN EN 60848, making it easy and fast to draw a Grafcet chart with **copy & paste** and **auto-completion**.

# Autómata simple en Sistemas Híbrido: *Ejercicio*

## 4. Termostato para Calefacción

### Modelo simplificado del Controlador: Termostato (SDED)

Ley de Control Básica (ON/OFF Automático):

si  $T_{Amb} < T_{Consigna} - T_{Delta} \Rightarrow Ctrl\ ON \Rightarrow$   
 $\Rightarrow P_{calefactor}(t) = P_{ON}[W];$

si  $T_{Amb} > T_{Consigna} + T_{Delta} \Rightarrow Ctrl\ OFF \Rightarrow$   
 $\Rightarrow P_{calefactor}(t) = 0[W];$

Implementación: Relé (Histéresis)

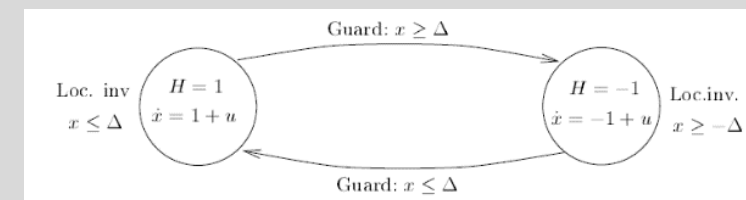
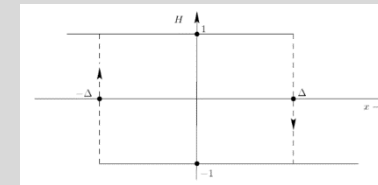
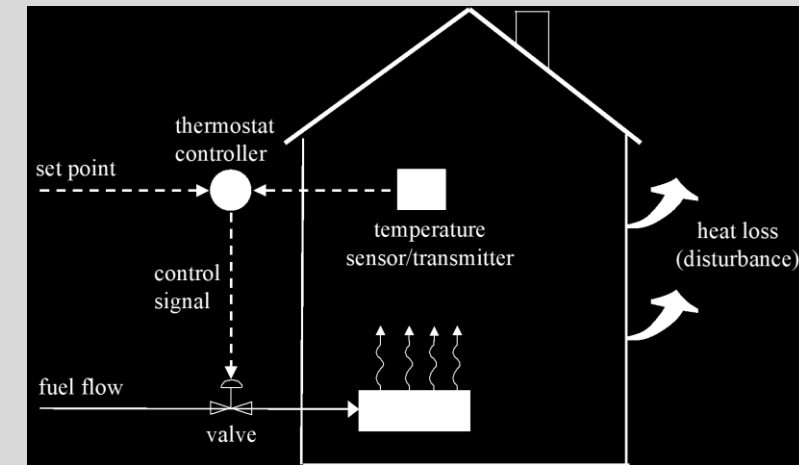
Hacer (agregar a Diagrama de Bloques de Estado);

→ Implementar en Simulink → **Simular!**

→ Agregar **interruptor Manual** de encendido/apagado;

→ Plantear **Diagrama de Transición de Estados**  
**(Autómata o Máquina de Estados Finitos);**

→ Implementar en Stateflow → **Simular!** → **Identificar Características**

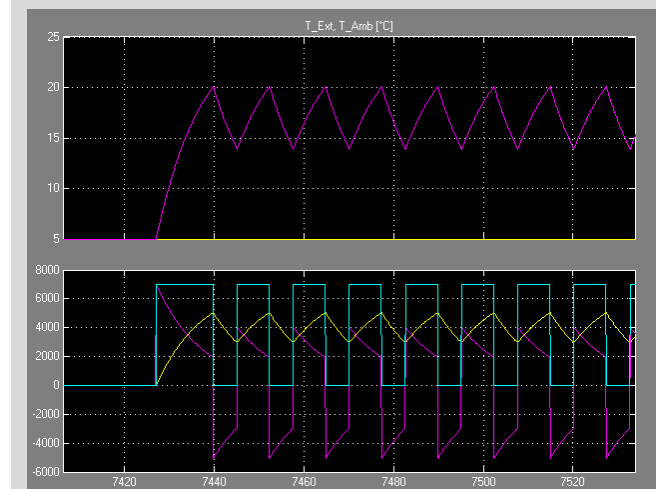
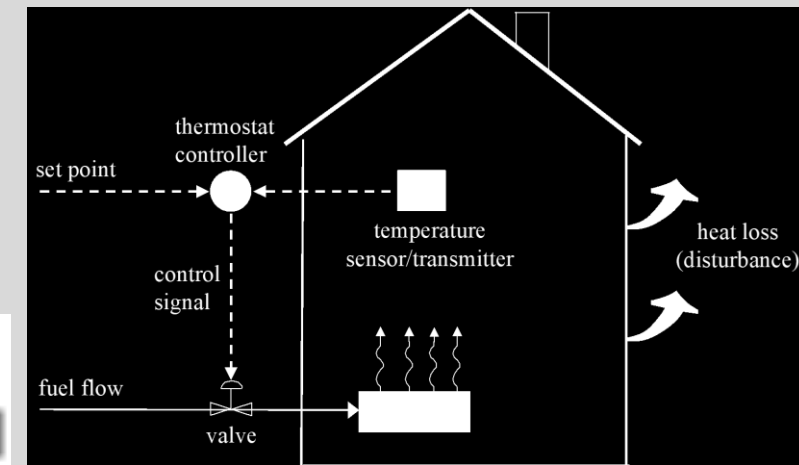
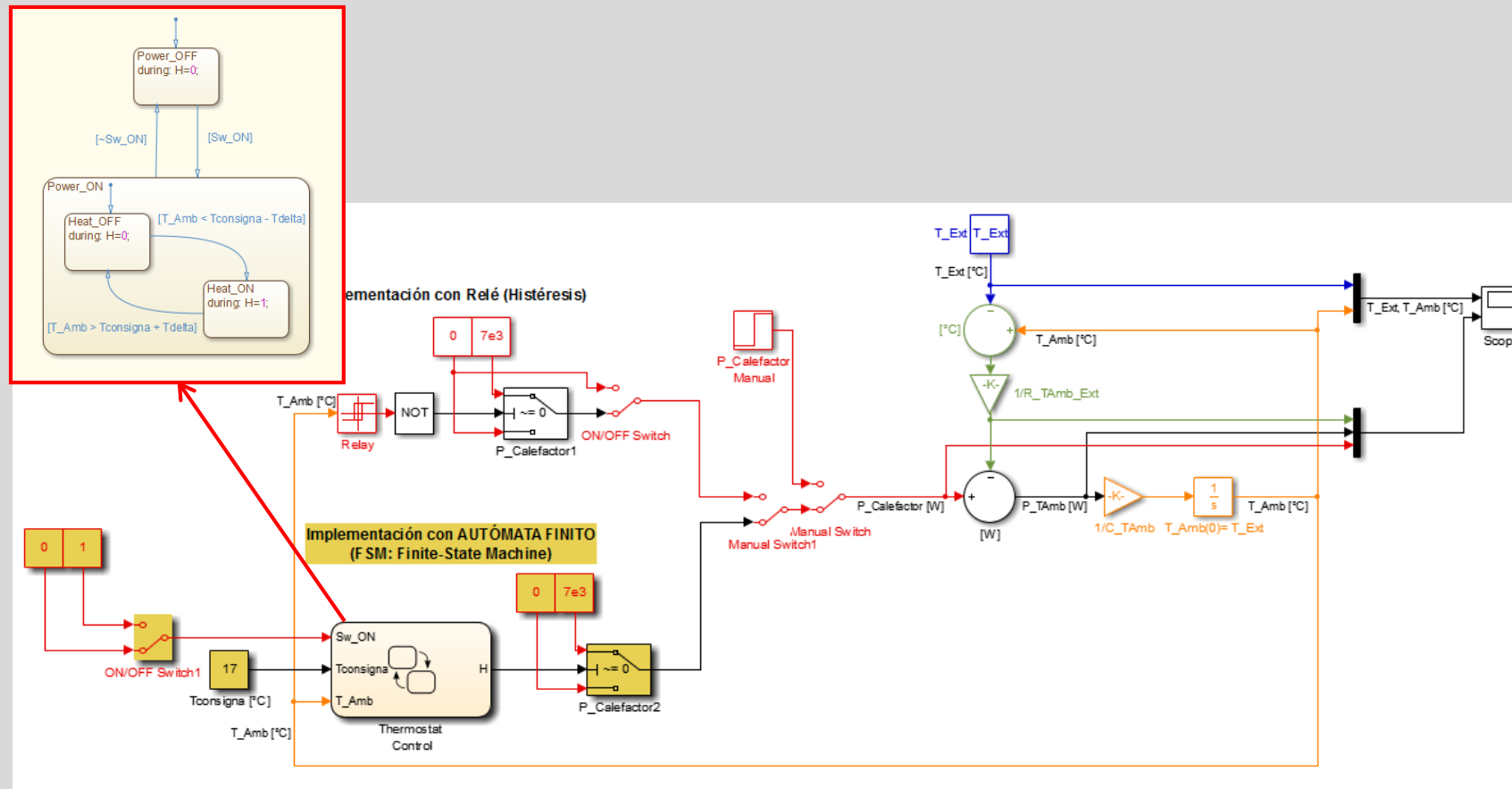


# Autómata simple en Sistemas Híbridos: *Ejercicio*

## 4. Termostato para Calefacción

Modelo simplificado de **Planta+Controlador: (SDH)**

→ Implementación FSM en Stateflow → Simulación





# HDS: Autómatas Híbridos simples: Ejercicio

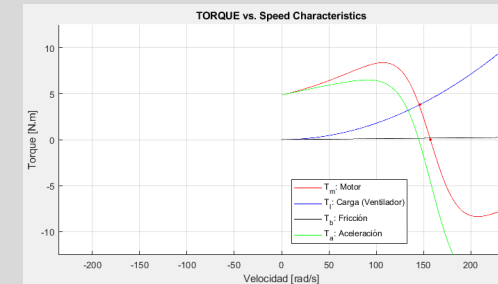
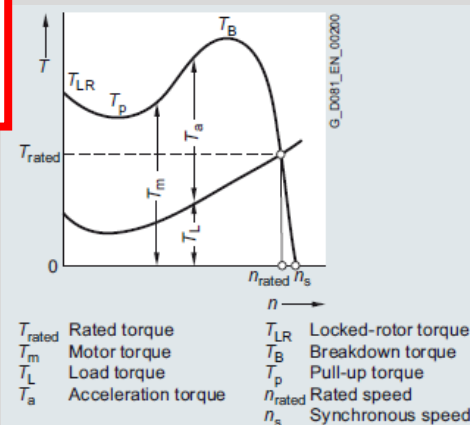
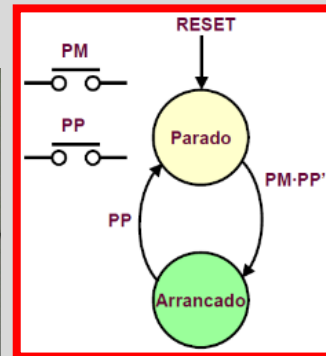
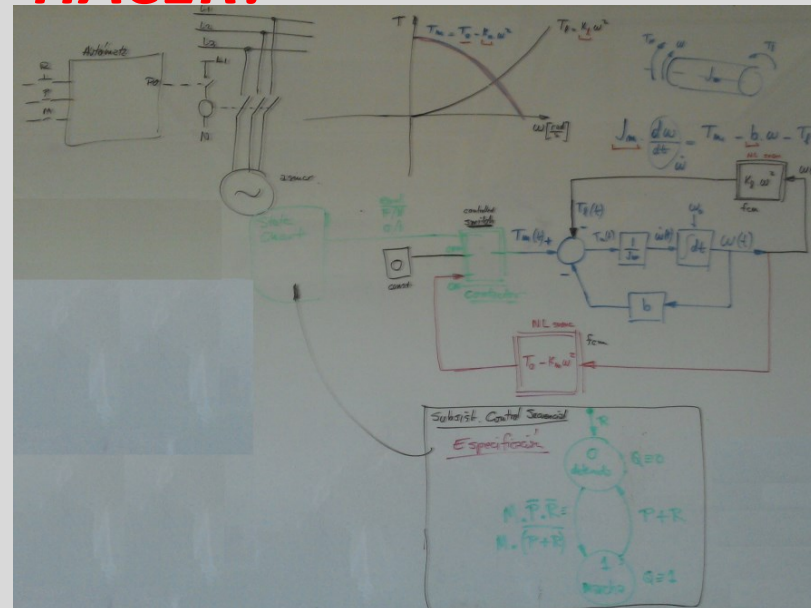
## 1. Control discreto (ON/OFF) de Marcha / Parada de Motor Eléctrico

Conexión Directa a Línea (DOL) 1 Cuadrante motorización FWD (Avance)

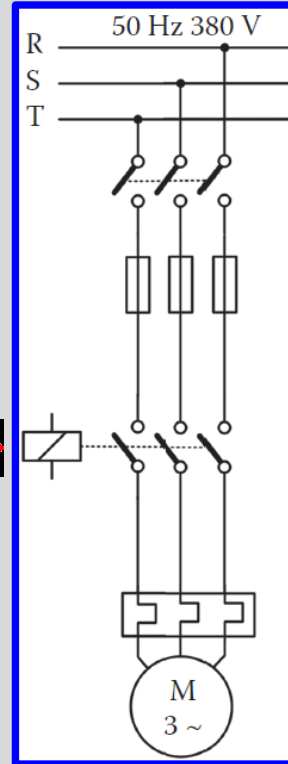
integrando:

- **dinámica motor con arranque directo (DOL) con contactor (continuo)** (ej. ley torque carga cuadrática creciente, torque impulsor simplif.); **Estado: velocidad**
  - **autómata (lógica secuencial) de control ON/OFF (conmutación)**
- Evaluar proceso de aceleración – marcha – desaceleración

**HACER!**



**Modelo HÍBRIDO: Sí considera Dinámica Continua de Motor**



# HDS: Autómatas Híbridos simples: *Ejercicio*

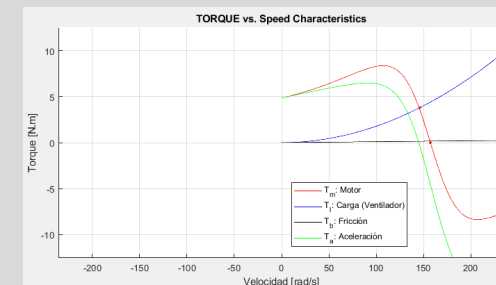
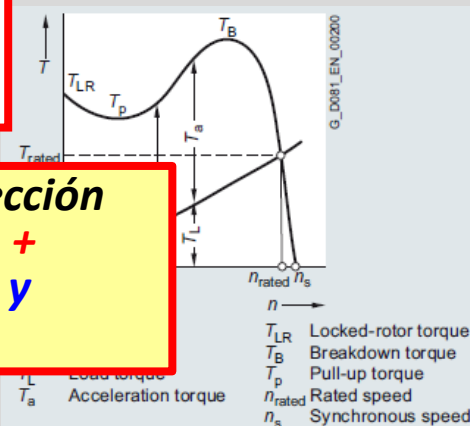
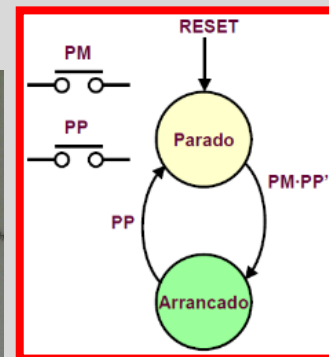
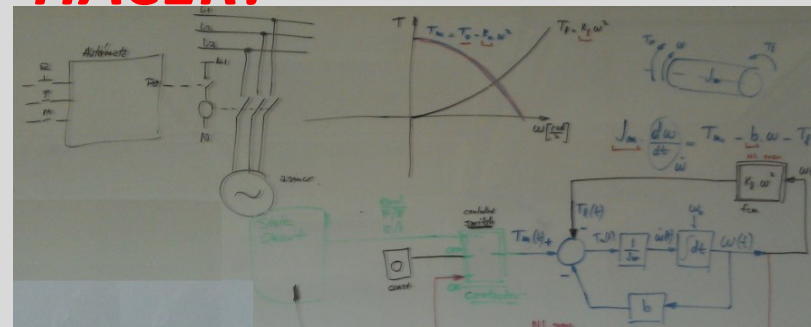
## 1. Control discreto (ON/OFF) de *Marcha / Parada* de Motor Eléctrico

Conexión Directa a Línea (DOL) 1 Cuadrante motorización FWD (Avance)

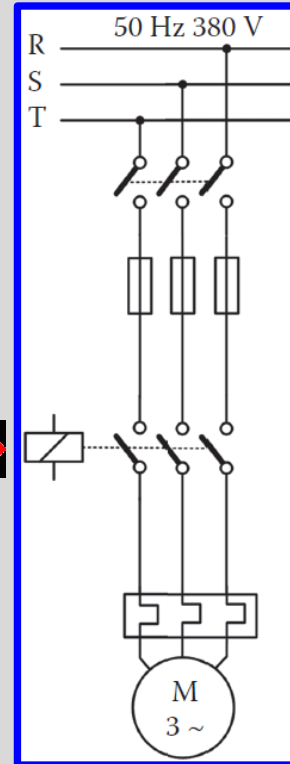
*integrando:*

- **dinámica motor con arranque directo (DOL) con contactor (continuo)**  
(ej. ley torque carga cuadrática creciente, torque impulsor simplif.); *Estado: velocidad*
  - **autómata (lógica secuencial) de control ON/OFF (conmutación)**
- **Evaluar proceso de aceleración – marcha – desaceleración**

**HACER!**



**Modelo HÍBRIDO: Sí considera Dinámica Continua de Motor**



→ **Incluir MODELO TÉRMICO (Imagen Térmica) → Protección Parada por sobretemp. e inhibición hasta enfriamiento + alarma (Control Híbrido) (interacción estados discretos y continuos)**

# DEDS: Autómatas simples: *Ejercicio*

## 2. Control discreto (ON/OFF) de *Marcha / Parada de Motor Eléctrico*

Conexión Directa a Línea (DOL) con **INVERSIÓN**: 2 Cuadrantes motorización FWD / REV (Avance / Retroceso)

### Entradas/Salidas (I/O) del controlador:

- 4 Entradas discretas: Relé TÉRMICO / Pulsadores FWD / REV / PARADA
- 2 Salidas discretas: Contactores FWD / REV de alimentación Motor

### Especificación de Operación:

- Relé TÉRMICO activado → desconecta el motor [prioridad 1].
- Pulsador PARADA → desconecta el motor [prioridad 2].
- Pulsadores FWD / REV → conectan el motor **FWD ó REV** **mutuamente excluyentes!** [prioridad 3] (si está desconectado y no está activado simultáneamente el Relé Térmico o el Pulsador de Parada)

### → Hacer:

1. Implementar en papel, en formas convencionales conocidas:

- **lógica cableada** → lógica de contactos (ladder)
- bloques de funciones lógicas
- Pseudocódigo informático de alto nivel (similar a Basic, Pascal, etc.)

2. Hacer: **Tabla de Verdad** y **Diagrama de Tiempos**

3. Plantear **Diagrama de Estados Discretos y Transiciones**

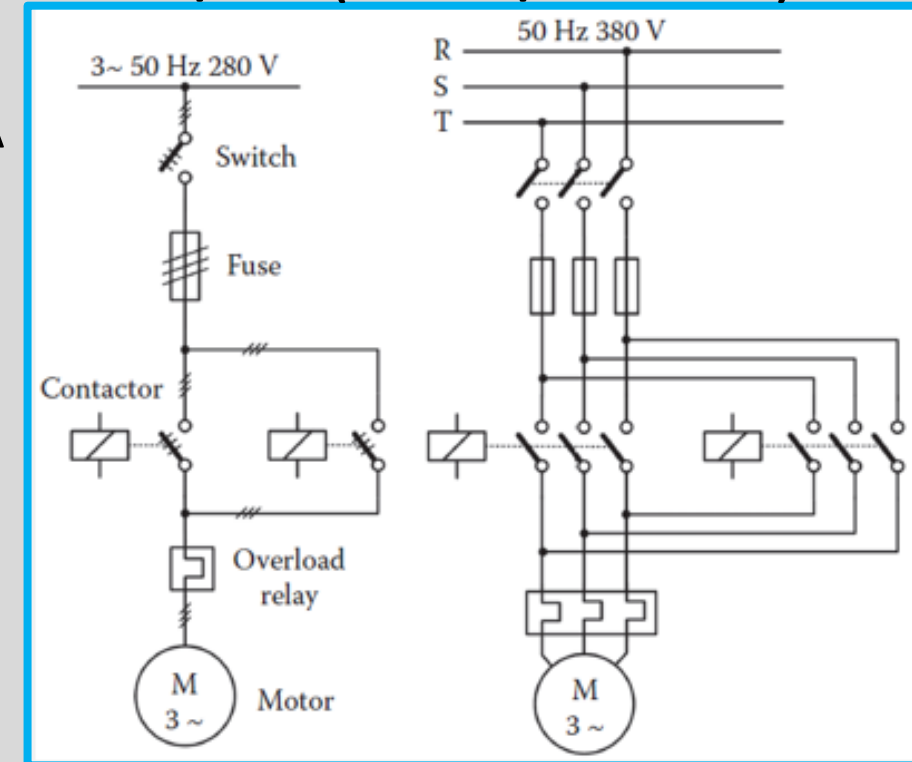


Diagrama Unifilar → Trifilar

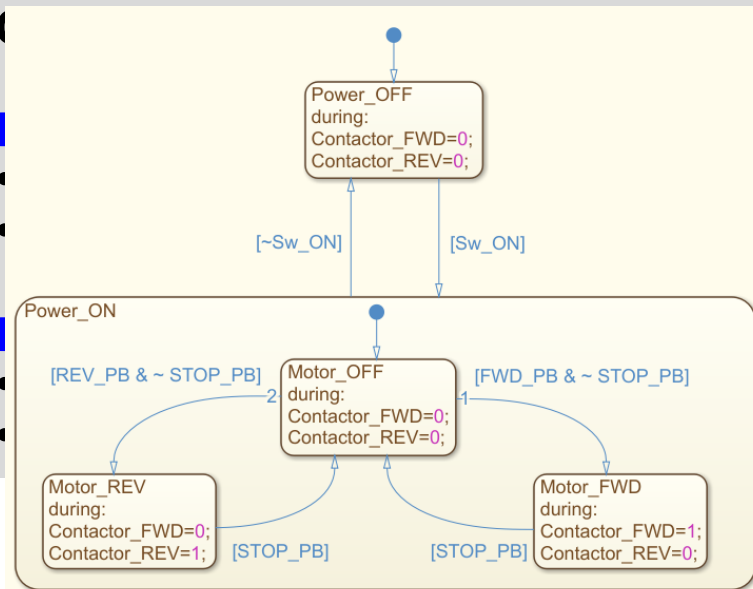
**Modelo DISCRETO:** NO considera Dinámica Continua de Motor



# DEDS: Autómatas simples: *Ejercicio*

## 2. Control discreto (ON/OFF) de *Marcha / Parada de Motor Eléctrico*

**INVERSIÓN:** 2 Cuadrantes motorización FWD / REV (Avance / Retroceso)



ador:

CO / Pulsadores FWD / REV / PARADA  
FWD / REV de alimentación Motor

Conecta el motor [prioridad 1].  
el motor [prioridad 2].

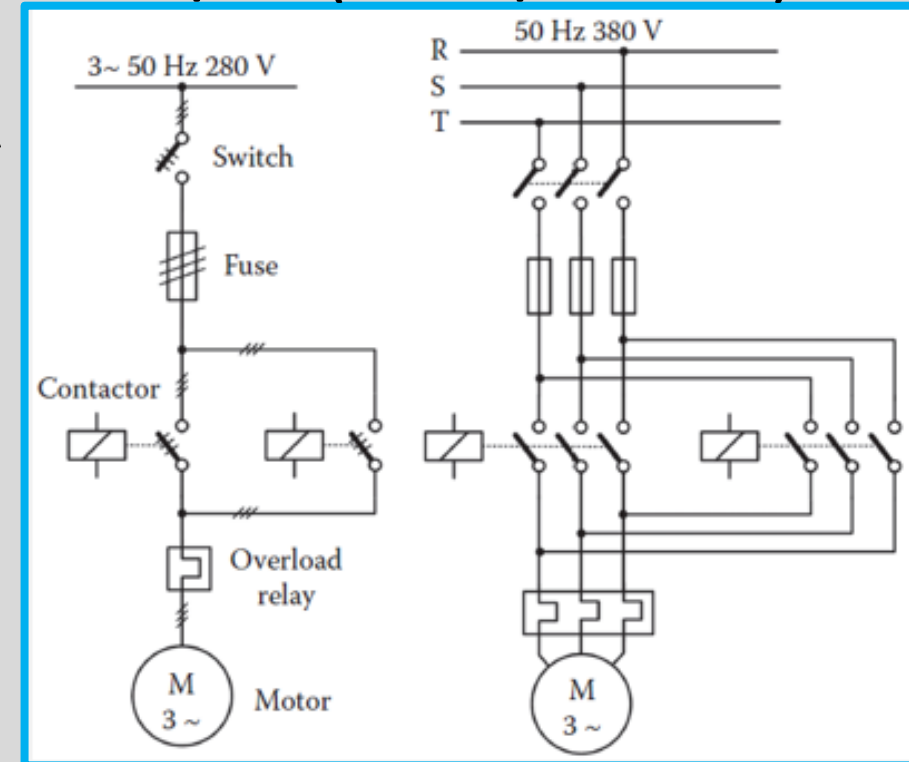
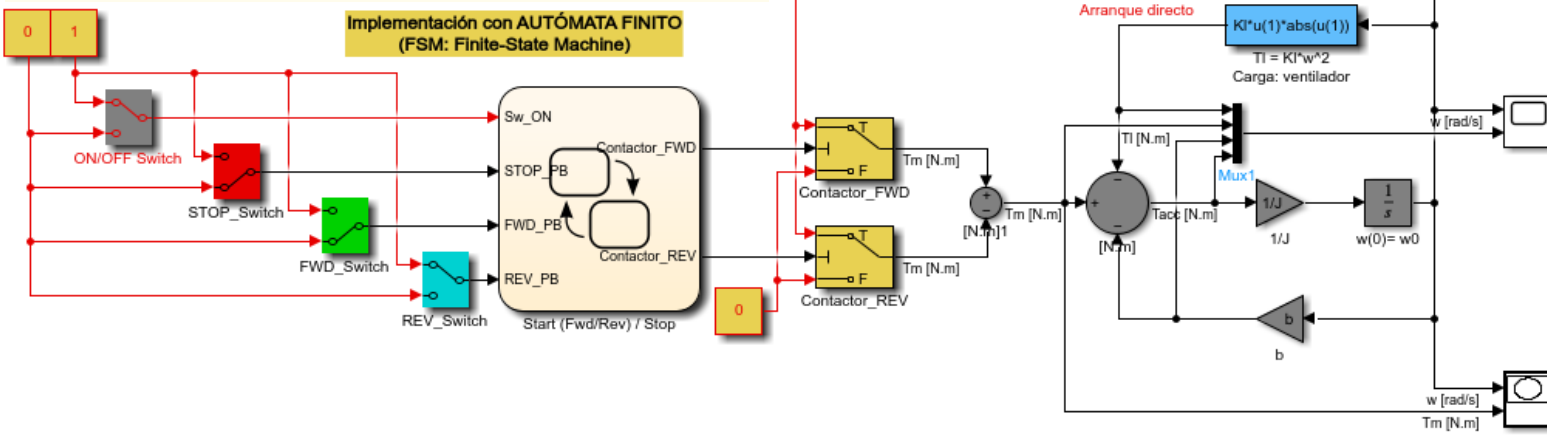


Diagrama Unifilar → Trifilar  
etc.)

**Modelo HÍBRIDO:** Sí considera  
Dinámica Continua de Motor





# Unidad 1

## Autómatas, Sistemas Discretos y Sistemas Híbridos

---

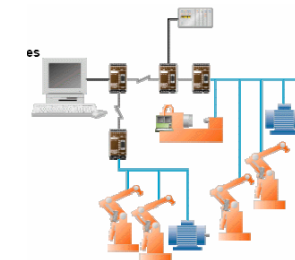


# AUTÓMATAS y CONTROL DISCRETO

## 1. *Autómatas, Sistemas Discretos y Sistemas Híbridos*

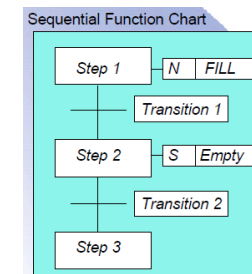
### A. Concepto y Aplicaciones

- Automatización y Control Industrial: Concepto y esquema general. Componentes y estructura.
- Aplicaciones Mecatrónicas → Control automático secuencial + control de tiempo discreto.



### B. Fundamentos de Autómatas Secuenciales de Control

- Sistemas secuenciales: estados discretos activados por eventos. Conceptos.
- Autómatas finitos. Modelado y representación. FSM. STATECHARTS.
- Simulación. STATEFLOW.

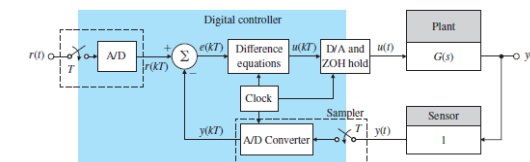


### C. Petri Nets (ISO/IEC 15909). GRAFCET (IEC 60848)

- Metodologías de representación y diseño. Redes de Petri.
- Lenguaje GRAFCET. Estructuras Jerarquizadas.

### D. Sistemas Híbridos. Práctica de Modelado y Simulación

- Modelado y Simulación en STATEFLOW. Representación en GRAFCET.



# Ejemplo: Control de una Prensa de Moldeo

Ref.: Example in **Annex A, IEC 60848 Ed. 3.0 2013 GRAFCET**

[AyCD\\_01b\\_IEC 60848\\_Ed3.0\\_2013-02\\_GRAFCET specification language for sequential function charts.pdf](#)

## Molding Press – Especificación de Requisitos:

A molding press for compressed powders works as shown on the Grafcet chart of Figure A.1.

When the press is in stand-by at **step 1**, the stamp and the die are in high position and a "ready" signal is lighted to indicate to the operator to put in a new charge.

The actions are successively performed as shown on the Grafcet chart.

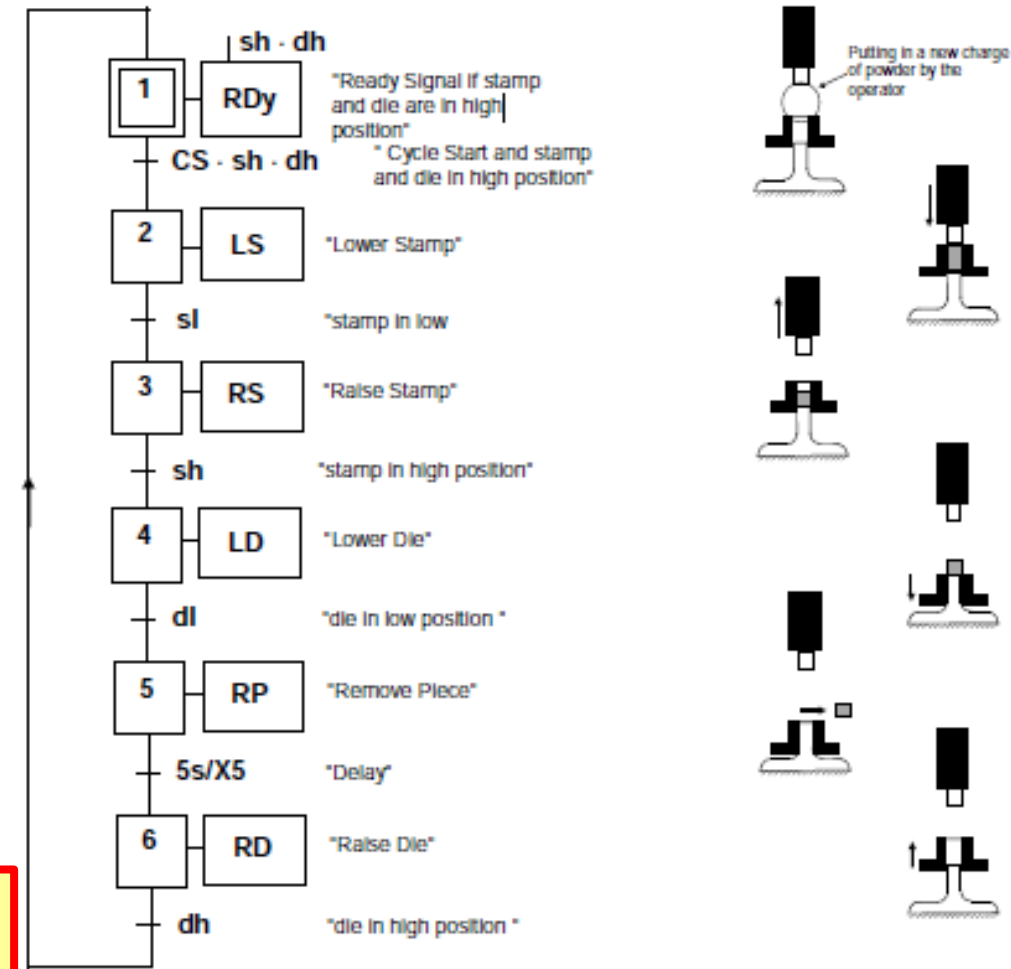
Codes:

Inputs

Outputs

|    |                        |     |              |
|----|------------------------|-----|--------------|
| CS | Cycle Start            | RDy | Ready signal |
| sh | stamp in high position | LS  | Lower Stamp  |
| sl | stamp in low position  | RS  | Raise Stamp  |
| dh | die in high position   | LD  | Lower Die    |
| dl | die in low position    | RP  | Remove Piece |
|    |                        | RD  | Raise Die    |

Figure A.1 – Representation of the working press using a grafcet



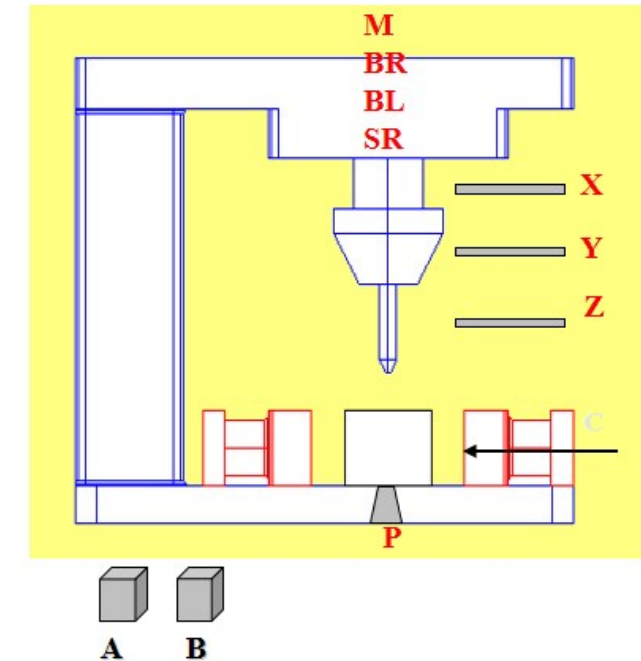
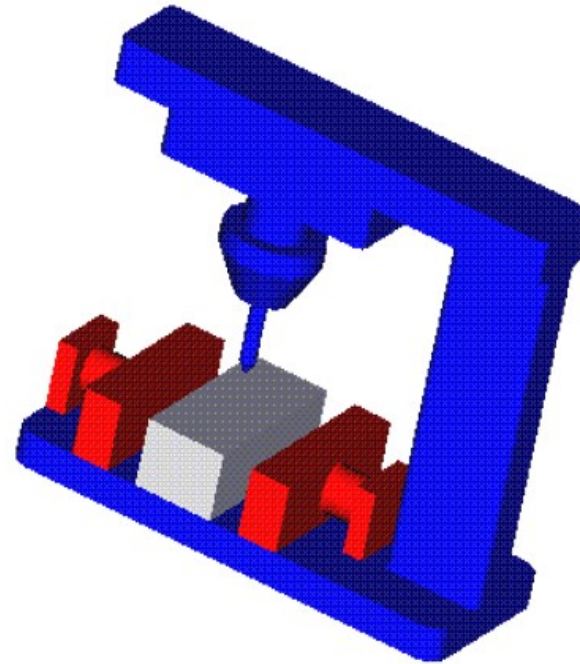
**Modelar Control Discreto en Simulink / STATEFLOW (Statechart) → Simular agregando 2 Actuadores Continuos de Traslación Vertical (Stamp + Die).**

# Ejercicio: Máquina Herramienta Taladradora

## Especificaciones de Requisitos de Funcionamiento:

### ► Funcionamiento de la taladradora:

- Existe un pulsador “B”, de inicialización del sistema, con objeto de que el motor adquiera una velocidad de giro de régimen permanente, que se obtiene accionando M. El taladro posee varias velocidades en el sentido longitudinal del eje, léase bajada lenta del utensilio del taladro BL, bajada rápida BR y subida rápida SR.
- La pieza en la que se va a realizar el taladro se detecta mediante un detector inductivo P, y se sujeta mediante dos sujeciones accionadas por C. La tarea de realizar un taladro sigue la siguiente secuencia: primero se detecta la pieza mediante el detector inductivo, posteriormente se pulsa el botón “A” de inicio de operación con lo que actúan las sujeciones de la pieza y al mismo tiempo se inicia el descenso rápido de la broca “BR”.
- Antes de empezar a realizar el taladro propiamente dicho a la pieza, el detector “Y” provoca el paso de descenso rápido de la broca a descenso lento “BL”, el cual se interrumpe cuando se detecta el final de carrera “Z”. Inmediatamente se produce la subida rápida de la broca hasta alcanzar la posición de reposo “X”.



Diseñar un autómatas híbrido de control coordinado de movimientos Giro y Axial + toma/liberación de pieza a mecanizar (a partir de especificación)

Especificar Comportamiento de Control Discreto en GRAFCET → Modelar Control Discreto en Simulink / STATEFLOW (Statechart) → Simular agregando 2 Actuadores Continuos: Traslación Vertical y Giro.

# Ejemplo: Pesaje-Mezclado Automático

Ref.: Example in **Annex B, IEC 60848 Ed. 3.0 2013 GRAFCET**

[AyCD\\_01b\\_IEC 60848\\_Ed3.0\\_2013-02\\_GRAFCET specification language for sequential function charts.pdf](#)

## **Automatic Weighing-Mixer - Especificación de Requisitos:**

Products A and B, previously weighed on a weighing unit C, and soluble bricks, brought one by one on a belt, are fed into a mixer N. The automatic system described in Figure B.1 allows a mixture of these three components to be obtained.

### Cycle

Actuating the push-button "cycle start" (CS) causes the **simultaneous** weighing of products and the transport of bricks as follows:

- **weighing** product A up to the mark "a" of the weighing unit, and then dosing product B up to the mark "b" followed by emptying weighing unit C into the mixer;
- **transport** of two bricks.

The cycle ends with the mixer rotation (MR) and its final tipping (TM) after time "t1". The rotation of the mixer continues during emptying.

## **SOLUCIÓN (Diferentes Opciones posibles con mismo resultado externo):**

**a) PLANA (con Acciones Continuas o Acciones Memorizadas) vs**

**b) ESTRUCTURADA (con Macroetapas, Forzamiento o Etapas Encapsulantes).**

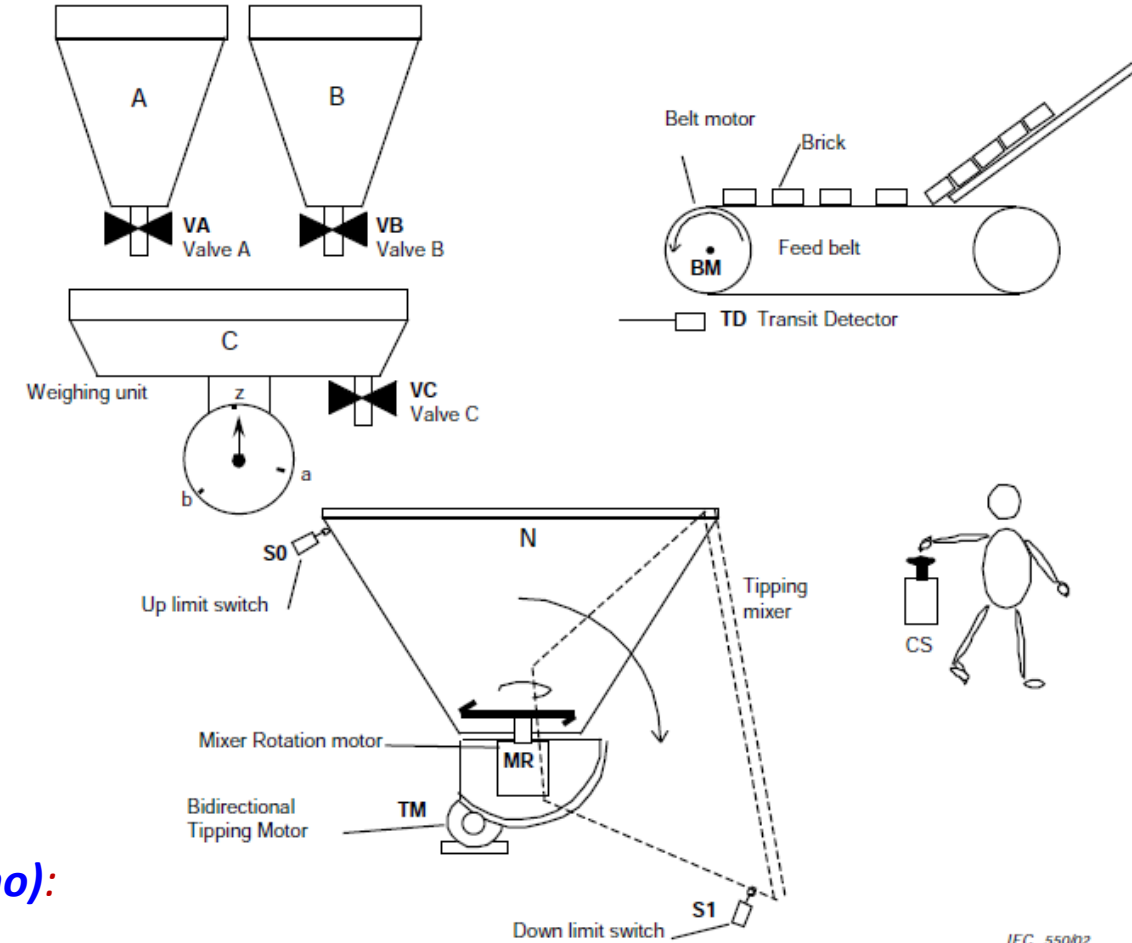


Figure B.1 - Overview diagram of weighing-mixing system

IEC 550/02

# Ejemplo: Pesaje-Mezclado Automático

Ref.: Example in Annex B, IEC 60848 Ed. 3.0 2013 GRAFCET

AyCD\_01b\_IEC 60848\_Ed3.0\_2013-02\_GRAFCET specification language for sequential function charts.pdf

## a) SOLUCIÓN PLANA:

### 1. Acciones Continuas

### 2. Acciones Memorizadas

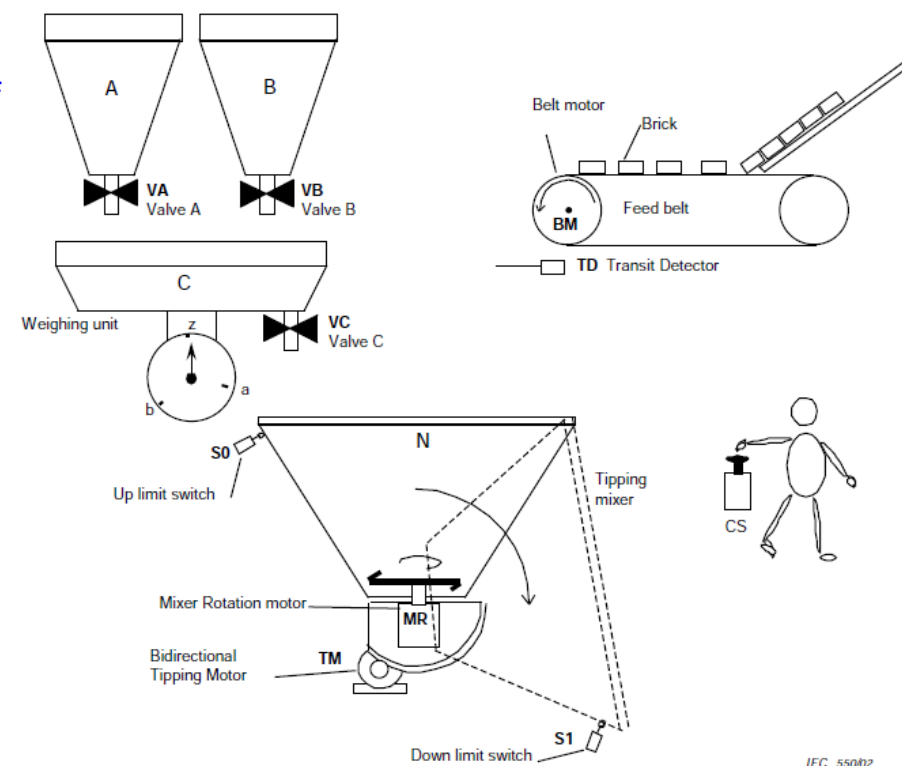


Figure B.1 – Overview diagram of weighing-mixing system

Codes:

| Inputs |                            | Outputs |                      |
|--------|----------------------------|---------|----------------------|
| CS     | Cycle Start                | BM      | Belt Motor           |
| TD     | Transit Detector           | MR      | Mixer Rotation motor |
| a      | Fluid weight A reached     | TM+     | Tipping Motor (down) |
| b      | Fluid weight A + B reached | TM-     | Tipping Motor (up)   |
| z      | Empty weighing unit        | VA      | Opening Valve A      |
| S0     | Mixer up                   | VB      | Opening Valve B      |
| S1     | Mixer down                 | VC      | Opening Valve C      |

Figure B.2 – Grafcet of a weighing-mixing involving only continuous actions

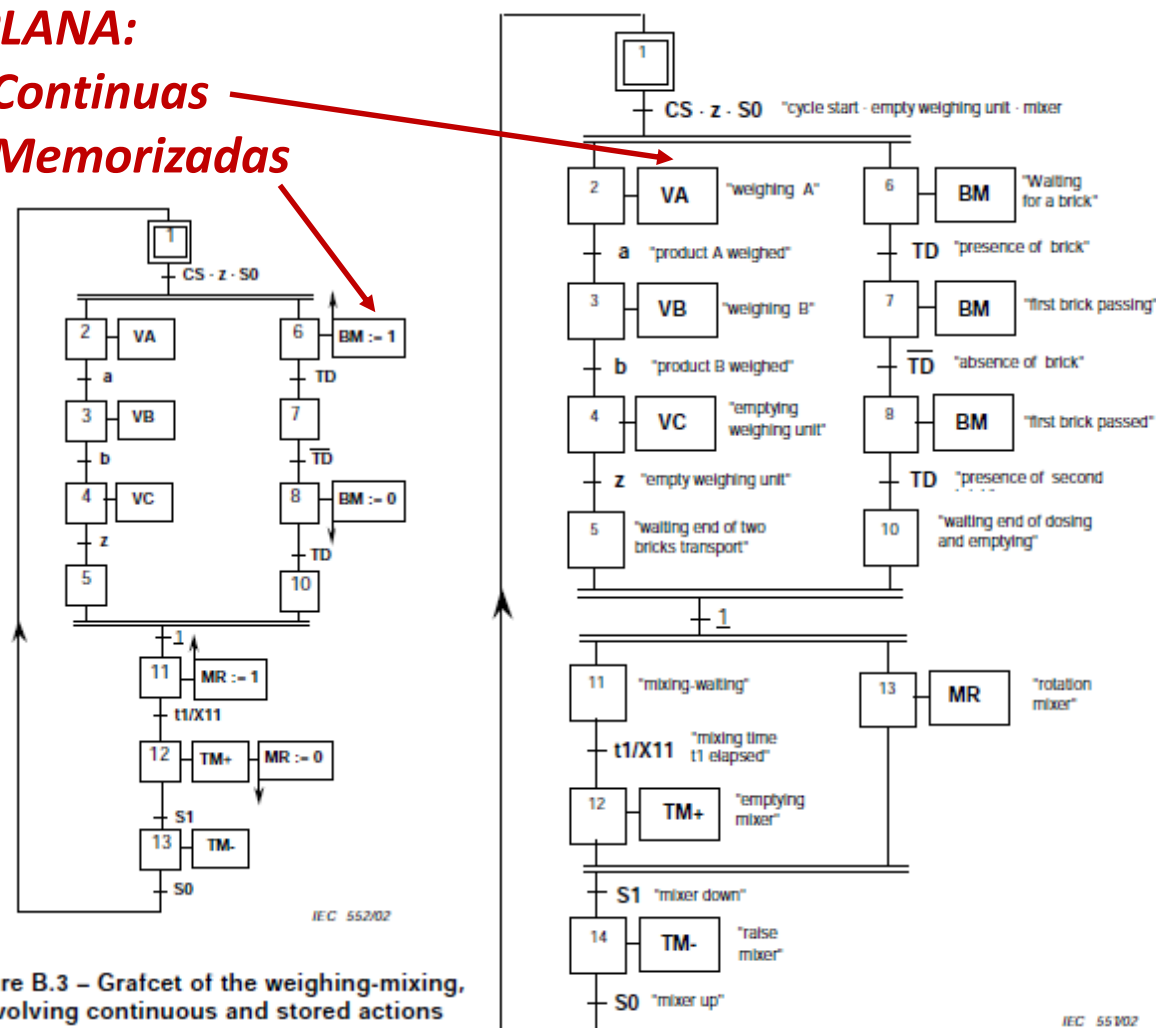


Figure B.3 – Grafcet of the weighing-mixing, involving continuous and stored actions

IEC 552/02

IEC 551/02

IEC 550/02

→ Modelar Control Discreto en Simulink / STATEFLOW (Statechart) → Simular agregando Actuadores Continuos.

# Ejemplo: Pesaje-Mezclado Automático

Ref.: Example in Annex B, IEC 60848 Ed. 3.0 2013 GRAFCET  
AyCD\_01b\_IEC 60848\_Ed3.0\_2013-02\_GRAFCET specification language for sequential function charts.pdf

b) SOLUCIÓN ESTRUCTURADA

1. Macroetapas

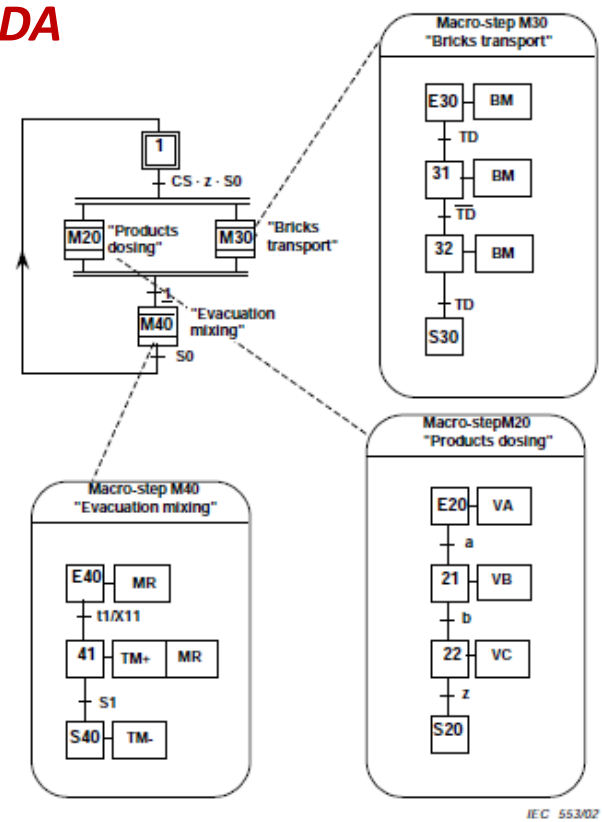


Figure B.4 – Grafcet of the weighing-mixing, divided into a global description using macro-steps and a description detailed by the macro-step expansions

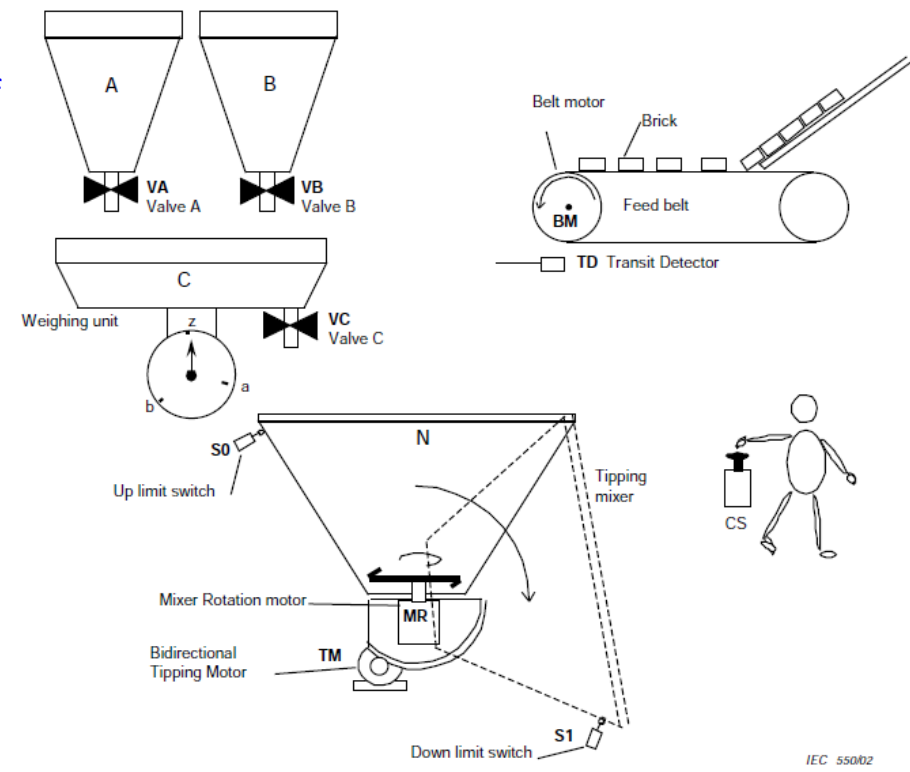


Figure B.1 – Overview diagram of weighing-mixing system

Codes:

| Inputs |                            | Outputs |                      |
|--------|----------------------------|---------|----------------------|
| CS     | Cycle Start                | BM      | Belt Motor           |
| TD     | Transit Detector           | MR      | Mixer Rotation motor |
| a      | Fluid weight A reached     | TM+     | Tipping Motor (down) |
| b      | Fluid weight A + B reached | TM-     | Tipping Motor (up)   |
| z      | Empty weighing unit        | VA      | Opening Valve A      |
| S0     | Mixer up                   | VB      | Opening Valve B      |
| S1     | Mixer down                 | VC      | Opening Valve C      |

Figure B.2 – Grafcet of a weighing-mixing involving only continuous actions

→ Modelar Control Discreto en Simulink / STATEFLOW (Statechart) → Simular agregando Actuadores Continuos.

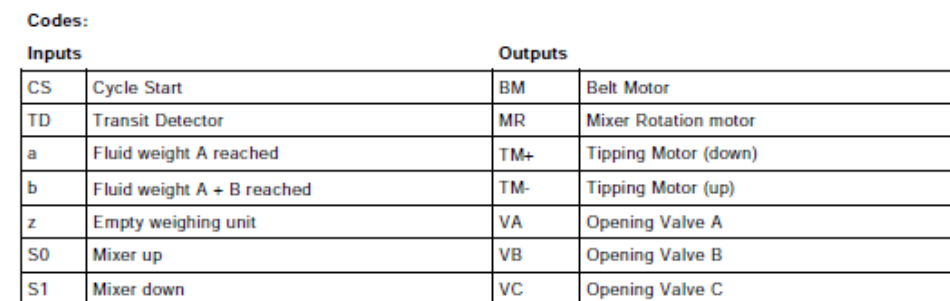
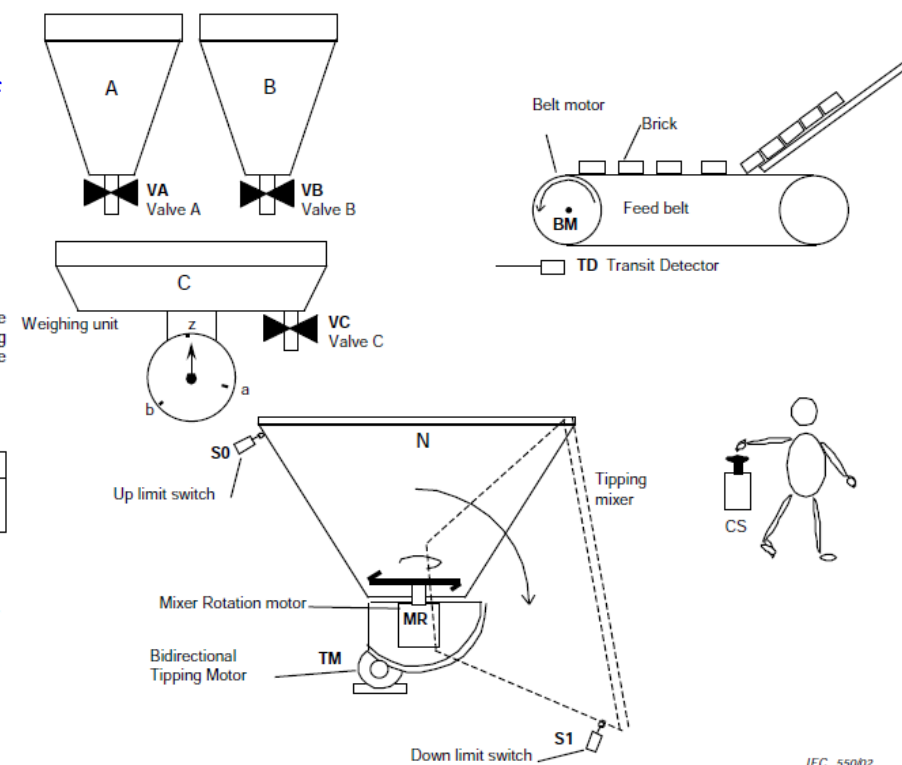
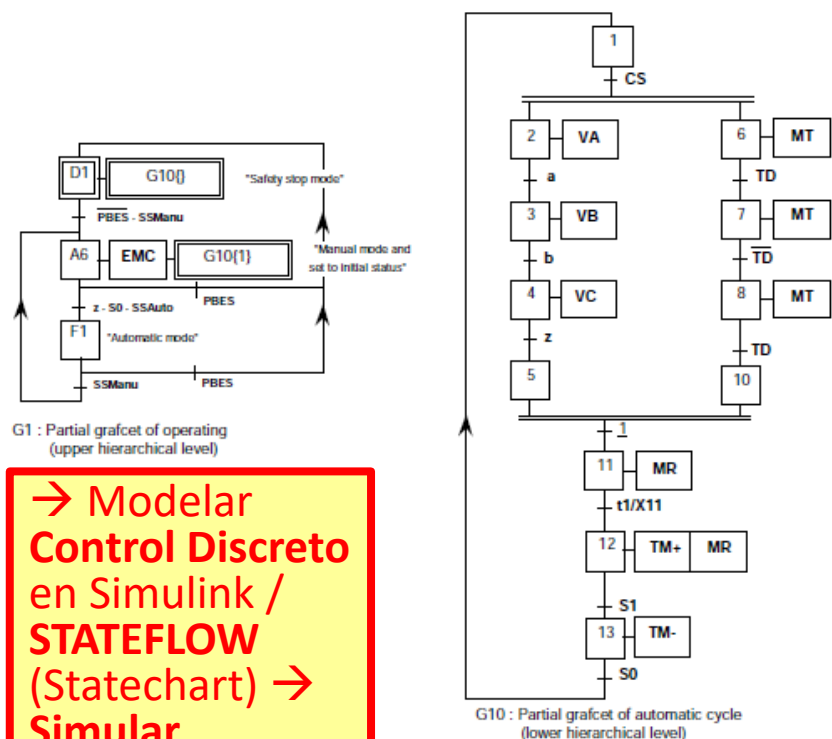
# Ejemplo: Pesaje-Mezclado Automático

Ref.: Example in **Annex B, IEC 60848 Ed. 3.0 2013 GRAFCET**

[AyCD\\_01b\\_IEC 60848\\_Ed3.0\\_2013-02\\_GRAFCET specification language for sequential function charts.pdf](#)

## b) SOLUCIÓN ESTRUCTURADA

### 2. Modos de Operación y Forzamiento



→ Modelar Control Discreto en Simulink / STATEFLOW (Statechart) → Simular actuadores continuos.

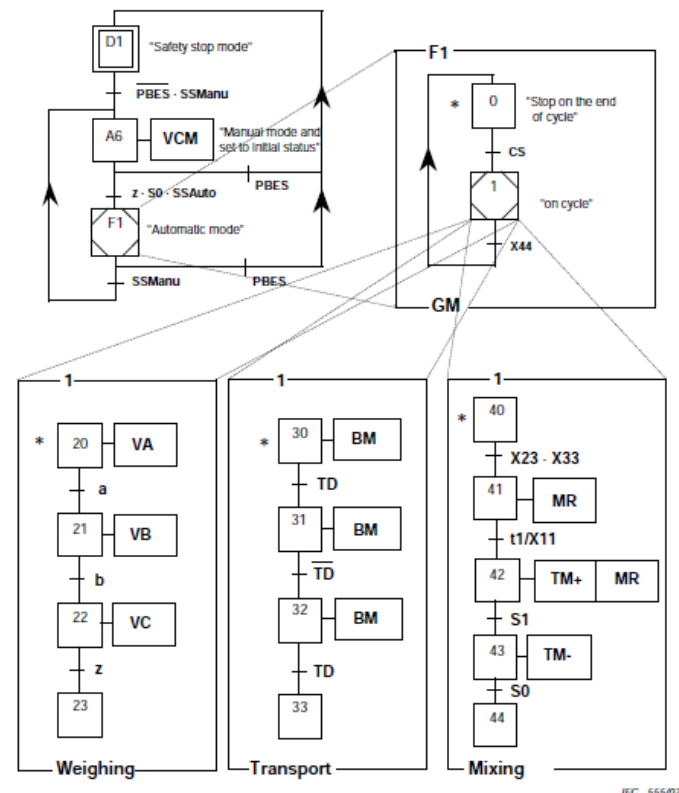
# Ejemplo: Pesaje-Mezclado Automático

Ref.: Example in Annex B, IEC 60848 Ed. 3.0 2013 GRAFCET

[AyCD\\_01b\\_IEC 60848\\_Ed3.0\\_2013-02\\_GRAFCET specification language for sequential function charts.pdf](#)

## b) SOLUCIÓN ESTRUCTURADA

### 3. Modos de Operación y Etapa Encapsulante



The enclosure hierarchy involves three levels.

When step D1 is active, no other step is active.

When step A6 is active, no other step is active and the manual controls (EMC) are enabled.

When the enclosing step F1 is activated, step 0 of its enclosure GM is also activated.

When the enclosing step 1 is activated, step 20 of its "weighing" enclosure, step 30 of its "transport" enclosure and step 40 of its "mixing" enclosure are also activated.

Figure B.6 – Structuring with operating modes using enclosing step

→ Modelar Control Discreto en Simulink / STATEFLOW (Statechart) → Simular agregando Actuadores Continuos.

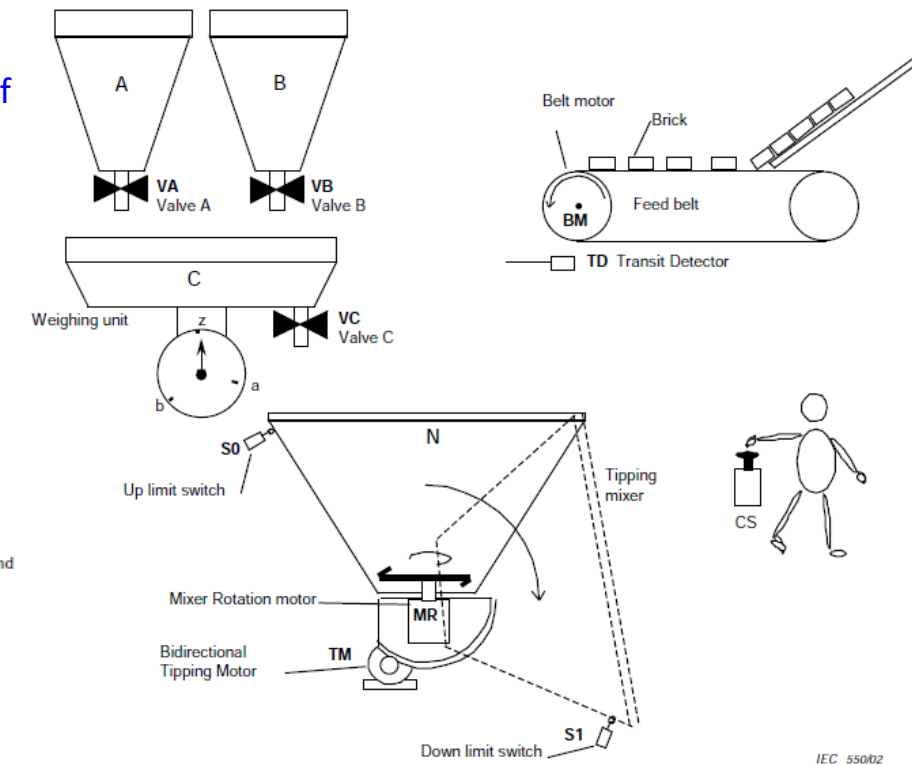


Figure B.1 – Overview diagram of weighing-mixing system

Codes:

| Inputs |                            | Outputs |                      |
|--------|----------------------------|---------|----------------------|
| CS     | Cycle Start                | BM      | Belt Motor           |
| TD     | Transit Detector           | MR      | Mixer Rotation motor |
| a      | Fluid weight A reached     | TM+     | Tipping Motor (down) |
| b      | Fluid weight A + B reached | TM-     | Tipping Motor (up)   |
| z      | Empty weighing unit        | VA      | Opening Valve A      |
| S0     | Mixer up                   | VB      | Opening Valve B      |
| S1     | Mixer down                 | VC      | Opening Valve C      |

Figure B.2 – Grafcet of a weighing-mixing involving only continuous actions