

# Fase di Programmazione dei Nodi

## Introduzione

La **fase di programmazione** è il momento in cui il sistema distribuito viene configurato e reso coerente dal punto di vista logico. In questa fase non avviene automazione operativa, ma **definizione delle regole** che governeranno il comportamento futuro della rete.

È una fase **temporanea, esplicita e separata** dal funzionamento normale del sistema.

---

## Motivazione architetturale

La separazione tra fase di programmazione e fase di runtime è una scelta architetturale fondamentale che consente di:

- mantenere i Nodi semplici e deterministici
- evitare logica dinamica o ambigua durante il funzionamento
- consentire la riconfigurazione del sistema senza modifiche firmware
- rendere il comportamento del sistema ispezionabile e verificabile

Questa separazione introduce due regimi operativi distinti:

- **Modalità Programmazione:** rete orchestrata, deterministica, sequenziale
  - **Modalità Runtime:** rete distribuita, event-driven, autonoma
- 

## Strati separati dalla fase di programmazione

La fase di programmazione separa chiaramente i seguenti strati:

### 1. Strato fisico

Il bus e il mezzo di comunicazione restano invariati.

### 2. Strato descrittivo

I Nodi dichiarano cosa sono e cosa sanno fare.

### 3. Strato logico

Vengono definite le relazioni evento → azione.

### 4. Strato operativo

Viene sospeso temporaneamente per evitare interferenze.

Questa separazione impedisce che la logica di configurazione contaminino il comportamento operativo.

---

## Attori coinvolti

Durante la fase di programmazione sono presenti due ruoli distinti:

- **Configuratore (Master di Programmazione)**  
Entità temporanea che guida la configurazione.
- **Nodi**  
Entità passive che rispondono alle richieste e ricevono configurazione.

Il Master non è parte integrante del sistema runtime.

---

## Transizione in modalità Programmazione

La fase di programmazione inizia con un comando globale che richiede a tutti i Nodi di:

- sospendere la pubblicazione di eventi runtime
- interrompere l'esecuzione delle regole
- entrare in stato di ascolto configurazione

Da questo momento, la rete non è più event-driven ma segue una comunicazione orchestrata.

---

## Step 1 – Discovery dei Nodi

Il primo passo consiste nell'identificazione di tutti i Nodi presenti sulla rete.

Ogni Nodo:

- segnala la propria presenza
- espone il proprio UID
- dichiara il proprio stato operativo

Il Master costruisce così una vista completa e coerente della rete.

---

## Step 2 – Scoperta delle capacità

Una volta individuati i Nodi, il Master li interroga singolarmente per ottenere:

- elenco degli Event Generators disponibili
- tipi di eventi generabili
- elenco dei Device / Actuators
- azioni supportate da ciascun Device

Questa fase costruisce il **catalogo funzionale** del sistema.

---

## Step 3 – Assegnazione delle etichette semantiche

Le etichette semantiche rappresentano la conoscenza dell'utente sul sistema fisico.

In questa fase:

- l'utente associa nomi logici a Event Generators e Device
- la semantica resta esterna ai Nodi
- i Nodi continuano a operare su identificatori tecnici

Le etichette sono metadati di configurazione e non influenzano il comportamento operativo del Nodo; possono tuttavia essere memorizzate localmente sul Nodo come informazioni ausiliarie per debug, diagnostica o interfacce di programmazione.

---

## Step 4 – Definizione della logica evento → azione

Utilizzando le etichette assegnate, l'utente definisce le regole di comportamento del sistema.

Ogni regola consiste in:

- un evento sorgente
- una o più azioni target

Le regole sono espresse in forma dichiarativa e non contengono riferimenti alla topologia fisica.

---

## Step 5 – Distribuzione delle regole

Il Master traduce la logica definita dall'utente in configurazioni locali.

Per ciascun Nodo:

- vengono indicati gli eventi da osservare
- vengono caricate le reazioni locali da eseguire

Ogni Nodo riceve solo le regole rilevanti per il proprio ruolo.

---

## Step 6 – Verifica e commit

Prima del ritorno al runtime, il sistema può eseguire:

- verifica di coerenza delle regole
- validazione delle configurazioni ricevute
- conferma di corretta memorizzazione

Questa fase riduce il rischio di stati parziali o inconsistenti.

---

## Ripristino della modalità Runtime

Concludere le operazioni di configurazione:

- il Master invia il comando di uscita dalla modalità Programmazione
- i Nodi riprendono la pubblicazione degli eventi
- le regole entrano in vigore

Il sistema torna a operare in modalità event-driven distribuita.

---

## Proprietà garantite

La fase di programmazione garantisce che:

- il runtime sia privo di ambiguità
  - il comportamento sia deterministico
  - la logica sia modificabile senza aggiornamenti firmware
  - la rete resti autonoma dopo la configurazione
- 

## Sintesi

La fase di programmazione è il meccanismo che consente di trasformare una rete di Nodi neutri in un sistema domotico coerente.

Separando configurazione ed esecuzione, il sistema mantiene semplicità locale, flessibilità globale e osservabilità nel tempo.