

**Trabajo Práctico**

# **Control de llenado de tanque de agua**

INFORMATICA II

Sbruzzi, Hugo  
2-9-2023

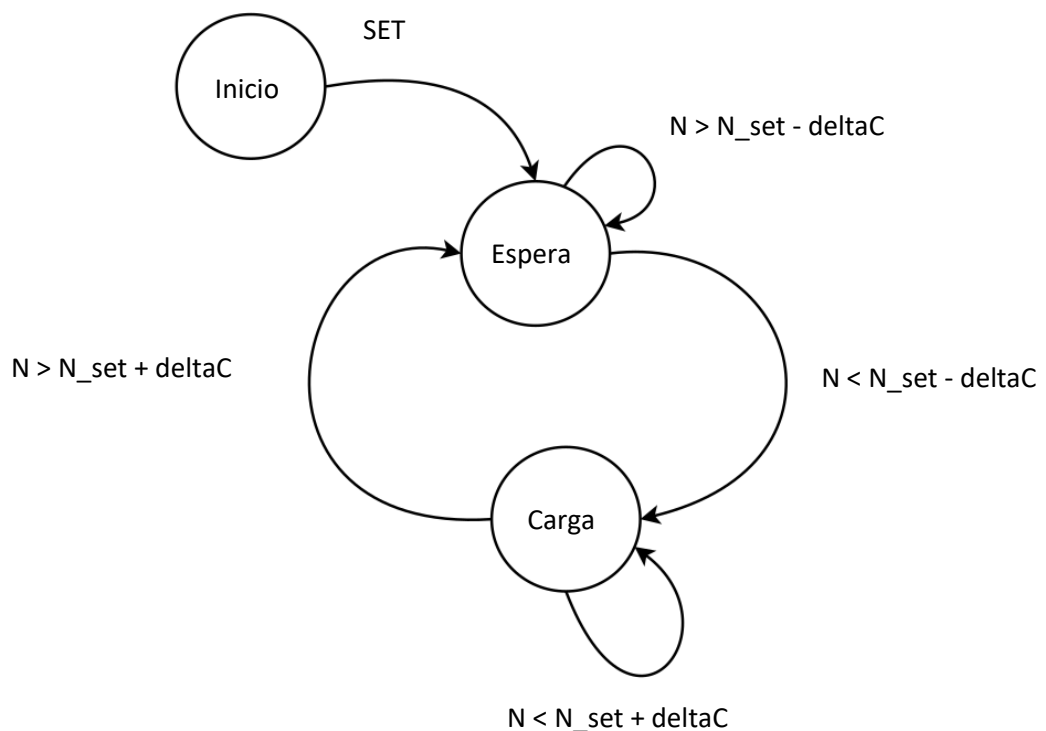
## Control de llenado de tanque de agua

### Memoria descriptiva

El sistema representa un control de llenado de tanque de agua del tipo ON-OFF con nivel de carga y delta de variación (Agua para consumo) es configurable. Se mantiene el nivel configurado utilizando el encendido y apagado de una bomba centrífuga comandada mediante un relé. El nivel de llenado es leído por un único sensor que puede ser de resistencia variable o por ultrasonido. A los fines didácticos el sensor y el relé de control de la bomba centrífuga serán emulados.

Primero el sistema llena el tanque mientras que el nivel sea menor a la nivel configurada *más* el delta y vuelve a cargar cuando el nivel sea inferior al configurado *menos* "el delta" de nivel. Esto garantiza una nivel de carga para consumo regulable.

### Máquina de estado - Control de llenado de tanque de agua



- **SET** : Bandera de fin de inicialización
- **N**: Nivel Leído en sensor GP2D12
- **c\_set** : Nivel configurado (De flotación. En Centímetros)
- **deltaN** : Delta de Nivel (En Centímetros)

## Código

- Archivo de cabecera **mylib.h**

```
#ifndef MY_LIB
#define MY_LIB

#include <stdio.h>

typedef enum {
    espera = 0,
    cargar = 1
} estados_c;

typedef struct {
    int n;           // Nivel actual (cambiado a tipo int para mayor nivel)
    int n_set;       // Nivel seteado
    int deltaN;      // Delta de nivel
} nivel_c;

nivel_c f_inicio(void); // Lee el archivo de configuración y carga las
variables.
estados_c f_espera(nivel_c);
estados_c f_cargar(nivel_c);

#endif
```

- Archivo **main.c**

con Switch case

```
#include "mylib.h"

int main() {
    nivel_c config;
    estados_c estado = espera; // primer estado

    config = inicio();
    while(1){
        switch (estado) {
            case espera: estado = f_espera(config);
                        break;
            case cargar: estado = f_cargar(config);
                        break;
        }
    }
    return 0;
}
```

- Con Punteros a funciones

```
#include "mylib.h"

int main() {
    nivel_c config;
    estados_c estado = espera; // primer estado
    estados_c (*fsm[])(nivel_c) = {f_espera, f_cargar}
    config = inicio();
    while(1) estado = (*fsm[estado])(config);

    return 0;
}
```

- Archivo **config.conf**

```
# Nivel de carga
c_set 50 cm
# Delta de carga
deltaC 20 cm
```

```
// mylib.c
#include "mylib.h"
nivel_c f_inicio(void) {
    nivel_c config;
    config.n = 0;
    config.n_set = 50;
    config.deltaN = 20;
    return config;
}
estados_c f_espera(nivel_c config) {
    if (config.n < config.n_set) {
        // Si se cumple la condición, retornamos el estado de carga.
        return cargar;
    } else {
        // Si no se cumple la condición, permanecemos en el estado de espera.
        return espera;
    }
}
estados_c f_cargar(nivel_c config) {
    config.n = config.n + config.deltaN

    // Después de realizar la carga, verificamos si el nivel alcanzó el nivel
    seteado.
    if (config.n >= config.n_set) {
        // Si se cumple la condición, volvemos al estado de espera.
        return espera;
    } else {
        // Si no se cumple la condición, permanecemos en el estado de carga.
        return cargar;
    }
}
```