



Código PID

Veamos un ejemplo :)

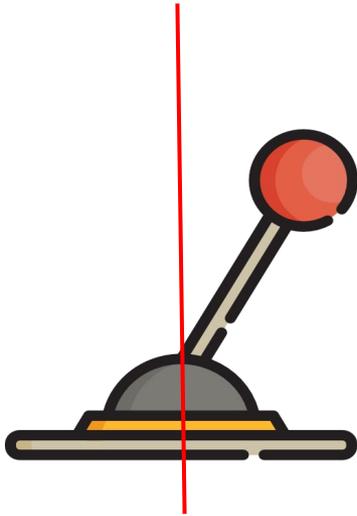
Curso: Mecatrónica - ME4250

Profesora: Carolina Silva

Auxiliar: Makarena Torres

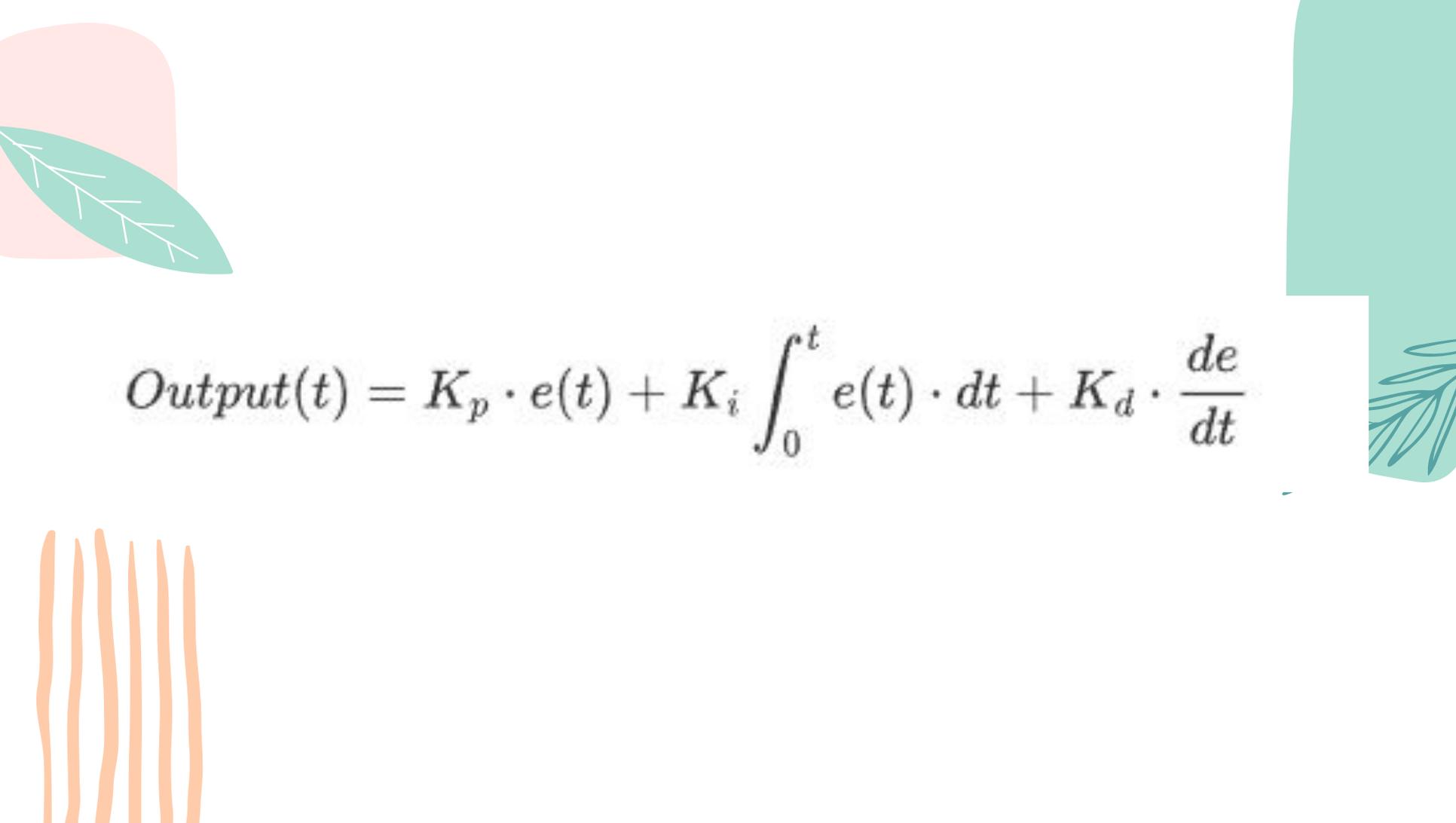
01

Teoría



SetPoint

- Proporcional -> presente
- Integrativo -> pasado
- Derivativo -> predice

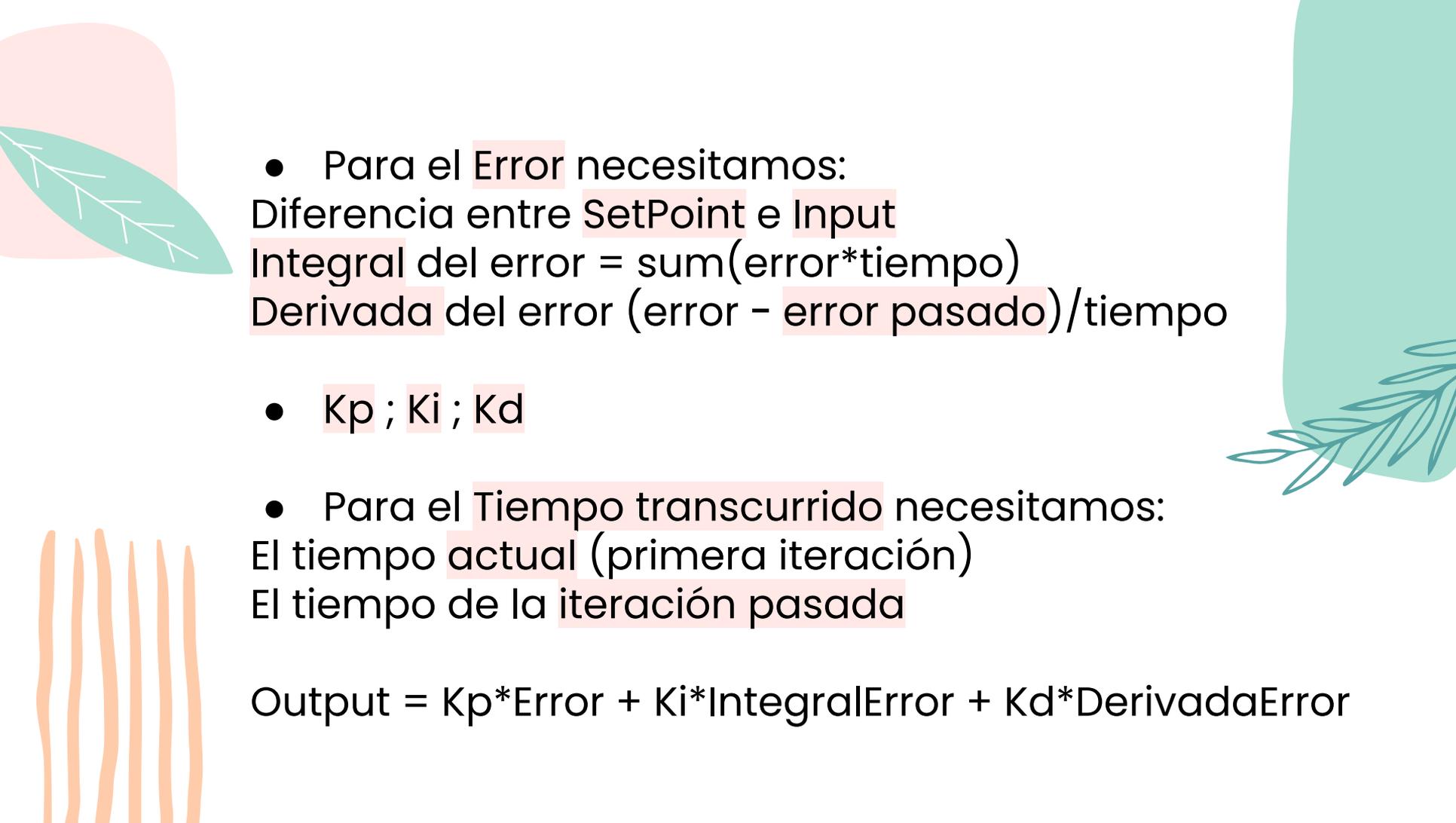

$$\text{Output}(t) = K_p \cdot e(t) + K_i \int_0^t e(t) \cdot dt + K_d \cdot \frac{de}{dt}$$

02 Variables



- Error
- K_p ; K_i ; K_d
- Tiempo transcurrido

$$Output(t) = K_p \cdot e(t) + K_i \int_0^t e(t) \cdot dt + K_d \cdot \frac{de}{dt}$$



- Para el Error necesitamos:
Diferencia entre SetPoint e Input
Integral del error = $\text{sum}(\text{error} * \text{tiempo})$
Derivada del error $(\text{error} - \text{error pasado}) / \text{tiempo}$

- K_p ; K_i ; K_d

- Para el Tiempo transcurrido necesitamos:
El tiempo actual (primera iteración)
El tiempo de la iteración pasada

$\text{Output} = K_p * \text{Error} + K_i * \text{IntegralError} + K_d * \text{DerivadaError}$

03

Ejemplo código

```
// Asignaciones pins
const int PIN_INPUT = A0;
const int PIN_OUTPUT = 3;

// Constantes del controlador
double Kp=2, Ki=5, Kd=1;

// variables externas del controlador
double Input, Output, Setpoint;

// variables internas del controlador
unsigned long currentTime, previousTime;
double elapsedTime;
double error, lastError, cumError, rateError;

void setup()
{
  Input = analogRead(PIN_INPUT);
  Setpoint = 100;
}
```

04 Operaciones

```
double computePID(double inp){  
    currentTime = millis(); // obtener el tiempo actual  
    elapsedTime = (double)(currentTime - previousTime); // calcular el tiempo transcurrido  
  
    error = Setpoint - Input; // determinar el error entre la co  
    cumError += error * elapsedTime; // calcular la integral del error  
    rateError = (error - lastError) / elapsedTime; // calcular la derivada del error  
  
    double output = kp*error + ki*cumError + kd*rateError; // calcular la salida del PID  
  
    lastError = error; // almacenar error anterior  
    previousTime = currentTime; // almacenar el tiempo anterior  
  
    return output;  
}
```



```
void loop() {
```

```
    pidController.Compute();
```

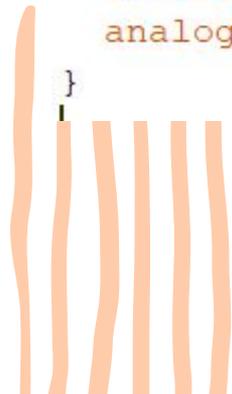
```
    Input = analogRead(PIN_INPUT);    // leer una entrada del controlador
```

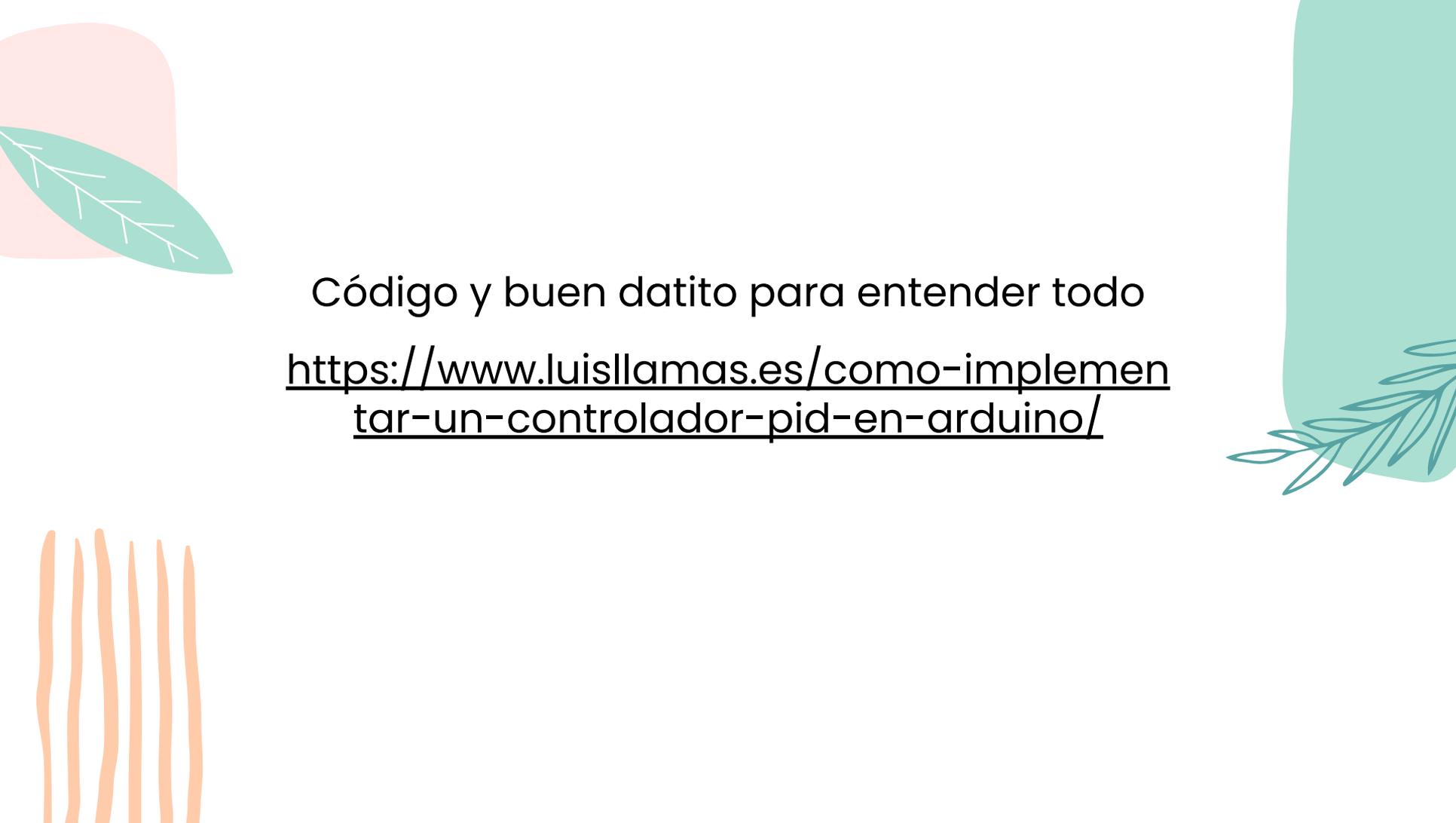
```
    Output = computePID(Input);    // calcular el controlador
```

```
    delay(100);
```

```
    analogWrite(PIN_OUTPUT, Output);    // escribir la salida del controlador
```

```
}
```





Código y buen datito para entender todo
<https://www.luisllamas.es/como-implementar-un-controlador-pid-en-arduino/>

05 Librería

```
#include <PIDController.hpp>

const int PIN_INPUT = 0;
const int PIN_OUTPUT = 3;

PID::PIDParameters<double> parameters(4.0, 0.2, 1);
PID::PIDController<double> pidController(parameters);

void setup()
{
  pidController.Input = analogRead(PIN_INPUT);
  pidController.Setpoint = 100;

  pidController.TurnOn();
}

void loop()
{
  pidController.Input = analogRead(PIN_INPUT);
  pidController.Update();

  analogWrite(PIN_OUTPUT, pidController.Output);
}
```



Gracias <3