

# Inspira Crea Transforma

# **ESCUELA DE CIENCIAS APLICADAS E INGENIERÍA INGENIERÍA AGRONÓMICA**

## **ST0299 PENSAMIENTO COMPUTACIONAL II Configurando un entorno para desarrollo de sistemas ciber-físicos**

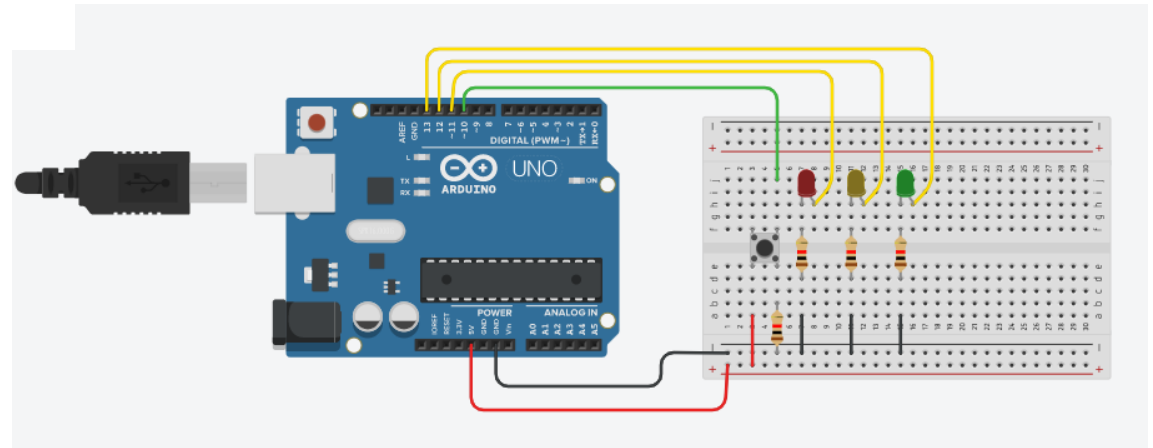
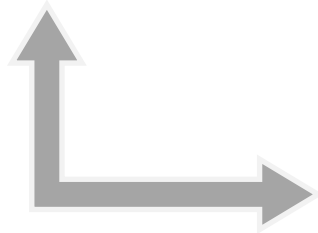
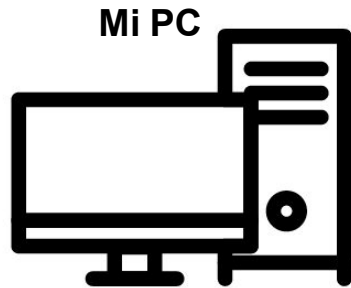
**Yomin Jaramillo M**

Docente | Escuela de Ingeniería | Ingeniería Agronómica

Correo: [yejaramilm@eafit.edu.co](mailto:yejaramilm@eafit.edu.co)

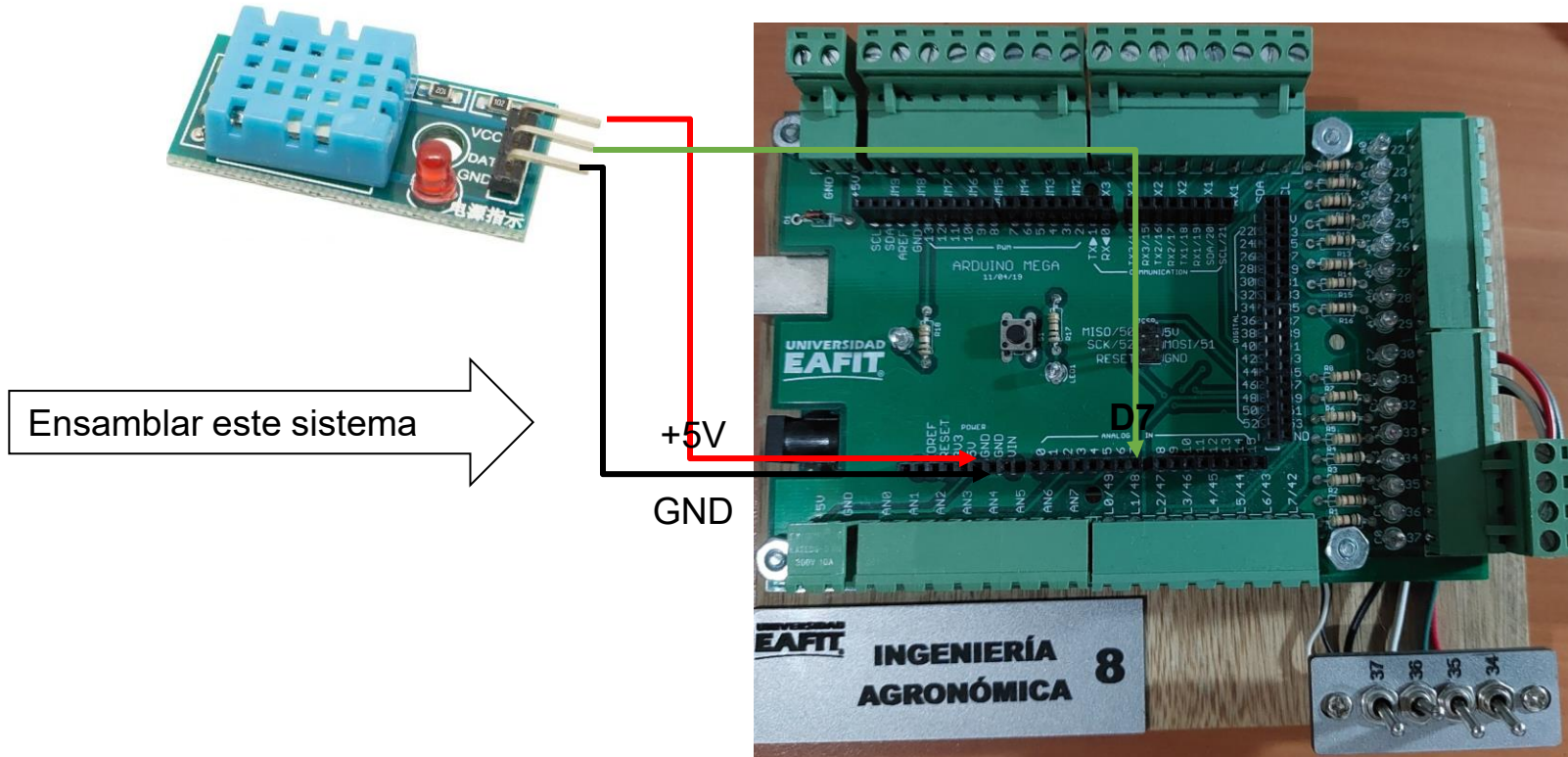
# Preparar el Entorno

# Que necesito para implementar este sistema?



Mi Tarjeta

# Como sería un circuito básico?

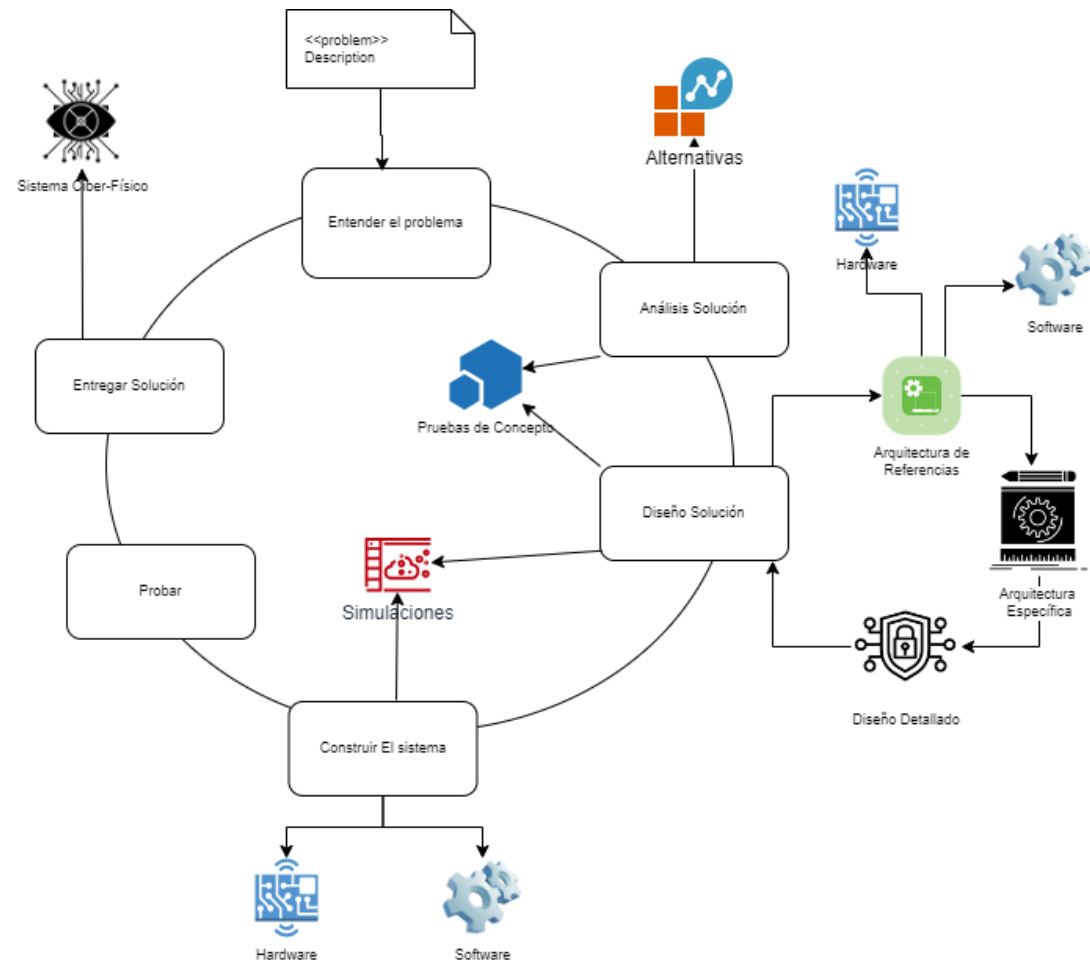


# Ciclo desarrollo de CPS

La solución se va desarrollando de forma evolutiva.

Cada evolución se realiza en una iteración, la iteración anterior es la entrada de la siguiente.

Cada paso que se de en la evolución se debe asegurar que está libre de fallos y se ha completado.



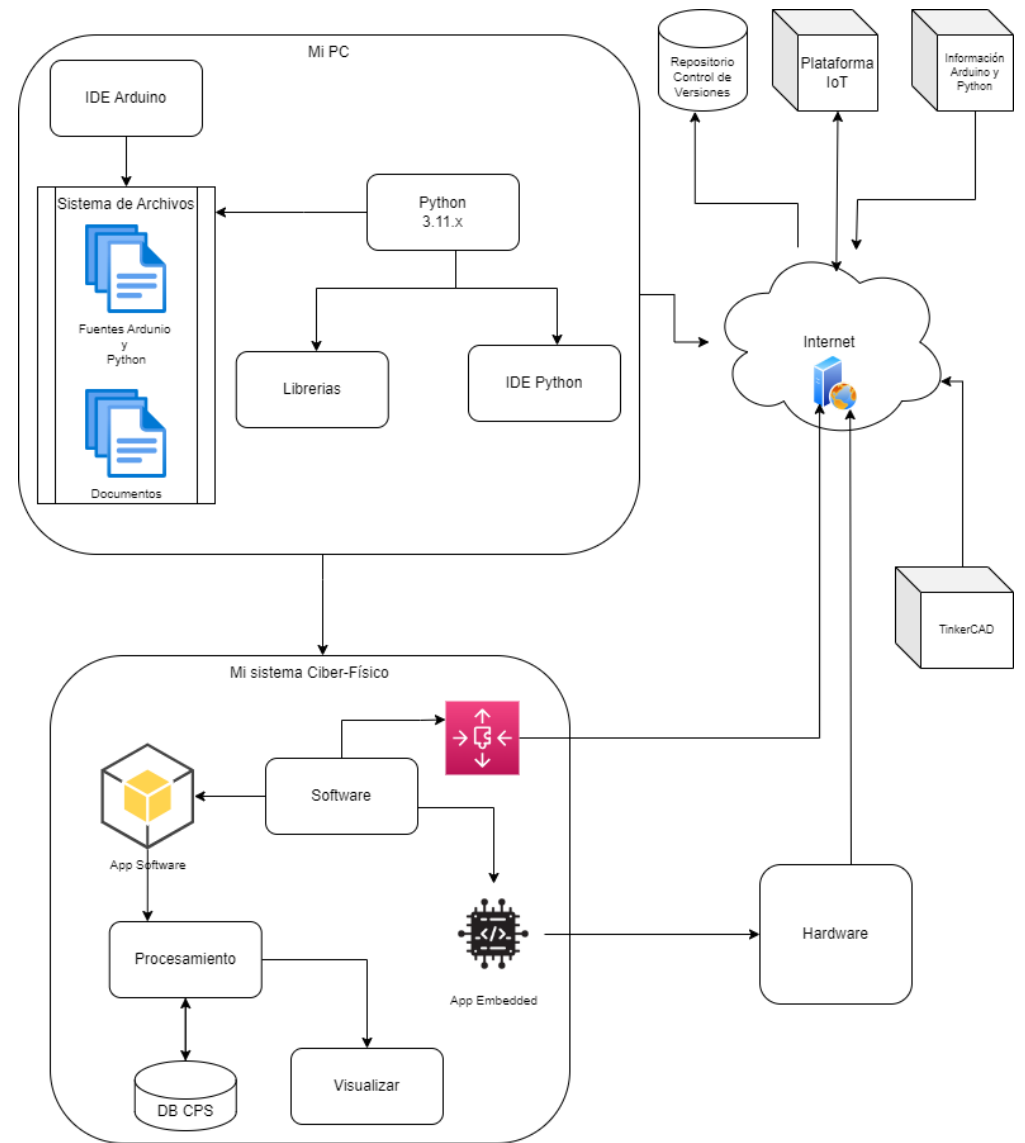
# Entorno de trabajo

Organizar el entorno para trabajar, nos asegura orden y evita reprocesar.

Toda la información del proyecto se debe centralizar y mantener organizados lo que se conoce como productos de trabajo

Los productos de trabajo son los documentos, los diseños, el código fuente de los programas.

Si no configura su entorno, perderá trabajo y le genera reprocesos innecesarios.



# Entorno Arduino



# Instalación de IDE Arduino

Para programar la tarjeta necesitas el entorno de desarrollo Arduino, el cual lo puedes descargar del siguiente:

- <https://www.arduino.cc/en/software>

Ejecuta el instalador del programa y sigue los pasos de instalación:

- Acepta los términos y condiciones de la licencia
- Selecciona todas las opciones para que instale todos los complementos y drivers necesarios.
- Selecciona la ruta de instalación y presiona “install”.
- Monte uno de los programas que realizó en el simulador y realice la compilación, si tiene errores corrija hasta que quede compilado con éxito.,

# Programar tarjeta

Identifique los componentes de la tarjeta y en donde están instalados los diferentes sensores.

Conectar la tarjeta (board) a uno de los puertos del USB.

Configure por la opción de herramientas el puerto y seleccione la tarjeta Arduino indicada, en esta caso es una tarjeta Arduino Mega

Cargue el programa que prende y apaga el LED, al cargar debe de poder ver el LED titilando.

Puedes usar el siguiente enlace de apoyo

<https://arduino.cl/como-instalar-arduino-en-windows/#:~:text=%2D%20Instala%20el%20entorno%20de%20desarrollo,instalaci%C3%B3n%20y%20presiona%20%E2%80%9Cinstall%E2%80%9D.>

# Configurar Python

# Configurar el proyecto

Vamos a asegurarnos de que ha hecho lo siguiente antes de seguir adelante con la programación de Python:

Asegurar que se tiene Python y los paquetes de Python pySerial instalado en su ordenador

Instalar un editor de texto para escribir códigos Python

Asegurar que los componentes de hardware estén configurados, como se defina en el diseño del sistema

Se conecta el Arduino al ordenador mediante un cable USB

# Instalar Python

En primer lugar, deberá descargar una copia de Python desde [http: https://www.python.org/getit](https://www.python.org/getit)

Debe tener cuidado con la versión de Python que está descargando.

Compruebe si el sistema operativo es de 32 bits o 64 bits.

Instalar una versión 3.x de Python, recomendando la versión 3.11.x

Se recomienda que se instale la versión de 32-bit por compatibilidad de librerías.

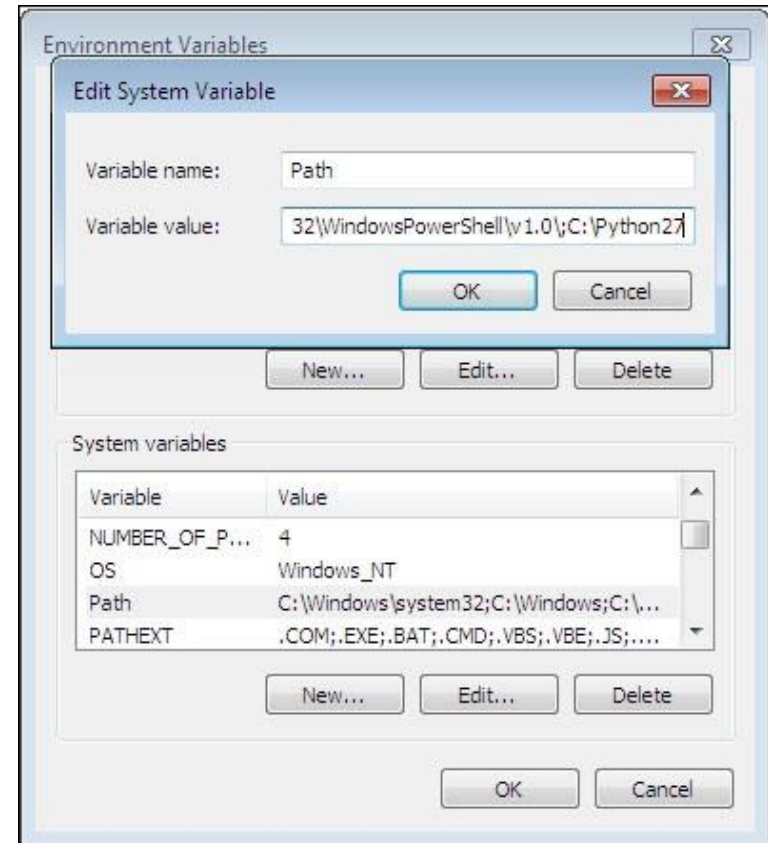
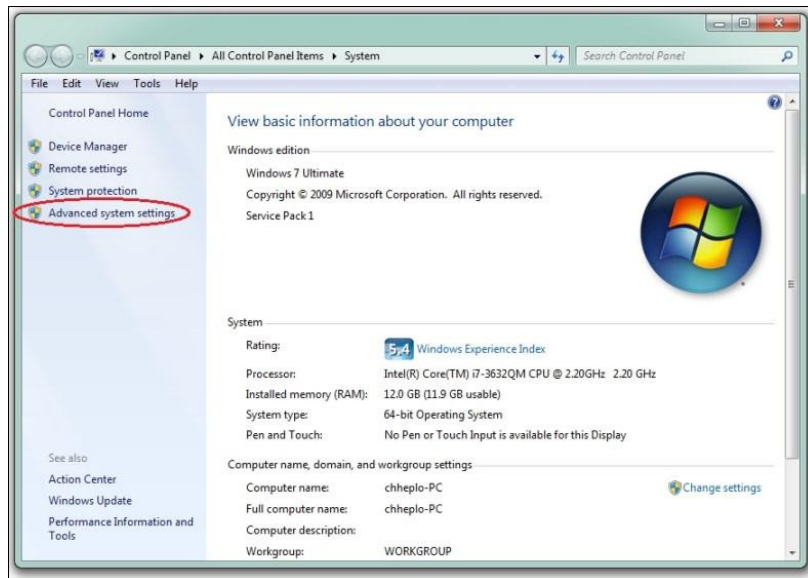
**En la instalación, chequee la opción que se configure la ruta del Python para poder ejecutar de cualquier directorio los comando**

Si no lo hace, debe adicionar en las variables de entorno el PATH de Python, donde quedó instalado.

Compruebe que quedó instalado usando el editor de línea de comando de Python.

# Configuración

Si no marcó el check que la instalación le realice la configuración de la variable de entorno para poder acceder Python de cualquier lugar, debe realizar lo siguiente:



# Prueba de su instalación

Ingresa a una consola de Windows (cmd) o PowerShell

Ejecute los siguientes comandos para validar que tiene Python instalado correctamente:

- *python*: debe entrar a una consola de Python donde puede realizar comando del lenguaje.
- *python -h*: debe mostrarle una lista de argumentos que puede utilizar con Python
- *Python -V*: le muestra la versión que fue instalada

# Protocolo PySerial



# Utilidad de *pySerial*

La biblioteca *pySerial* permite la comunicación con Arduino encapsulando el acceso para el puerto serie.

Este módulo proporciona acceso a la configuración del puerto serie a través de propiedades *Python* y le permite configurar el puerto serie directamente a través del intérprete.

*pySerial* será el puente para cualquier comunicación futura entre el Python y Arduino.

Instale *pySerial* así desde la línea de comando de IDLE de Python o desde una consola de sistema operativo:

**> *pip install pyserial***

# Valide que la librería quedó instalada

Ingresa al IDE por defecto de Python

Puede también utilizar otro IDE como Visual Studio Code o cualquier otro que usted conozca y sea de su dominio

Realice las siguientes instrucciones:

- `>import serial` // No debe dar error
- `>puerto = 'COM7'` // No debe dar error
- `>arduino = serial.Serial(port, 9600)` // Debe dar error, porque?

# Actividades de fortalecimiento

Para fortalecer lo que se viene realizando con el simulador realice las siguientes dos actividades específicas:

## Actividad 1

- Basados en lo que ha logrado hacer en TinkerCAD, trate de hacer un semáforo inteligente de tal manera que le de paso a las personas presionando un botón y que quede en espera en rojo unos 60 segundos.
- Realice una lista de preguntas y dudas que le generan el ejercicio.

## Actividad 2

- Con lo que ha realizado en TinkerCad (Semáforos), lleve este a la tarjeta, ensamble el semáforo con LEDS
- Lleve a la tarjeta física y ensamble con el sensor de temperatura y humedad, realice el semáforo basado en los niveles de temperatura, si es mayor a 27 grados prende el rojo, entre 20 y 27 el naranja, menos de 20 el verde
- Liste todas las inquietudes y dificultades que tenga

# Actividad de entrenamiento en Python

## En su entorno configurado en Python:

Realice un programa que tenga controles como decisiones y ciclos.

Realice diferentes operaciones matemáticas en las que se hagan transformaciones de datos análogos a digitales.

Evolucione su programa para que realice diferentes operaciones con ingreso de información desde la consola, por ejemplo un valor límite y este se valide, se saque un mensaje de alerta.

Realice varios ejemplos en el IDE de Arduino que le compilen, use los que realizó en TinkerCAD.

Estudio el programa que se adjunta en el material de apoyo, es un código básico que adquiere datos del sensor HT11, escriba todas las dudas que le genera.

# FIN DE TEMA

**Configurando un entorno para desarrollo de  
sistemas ciber-físicos**