

UNCUYO  
UNIVERSIDAD  
NACIONAL DE CUYO



FACULTAD  
DE INGENIERÍA

# Proyecto Integrador: Simulación de control de movimiento de Puma 560 mediante Realidad Aumentada

## Realidad Virtual

Agustín Lezcano

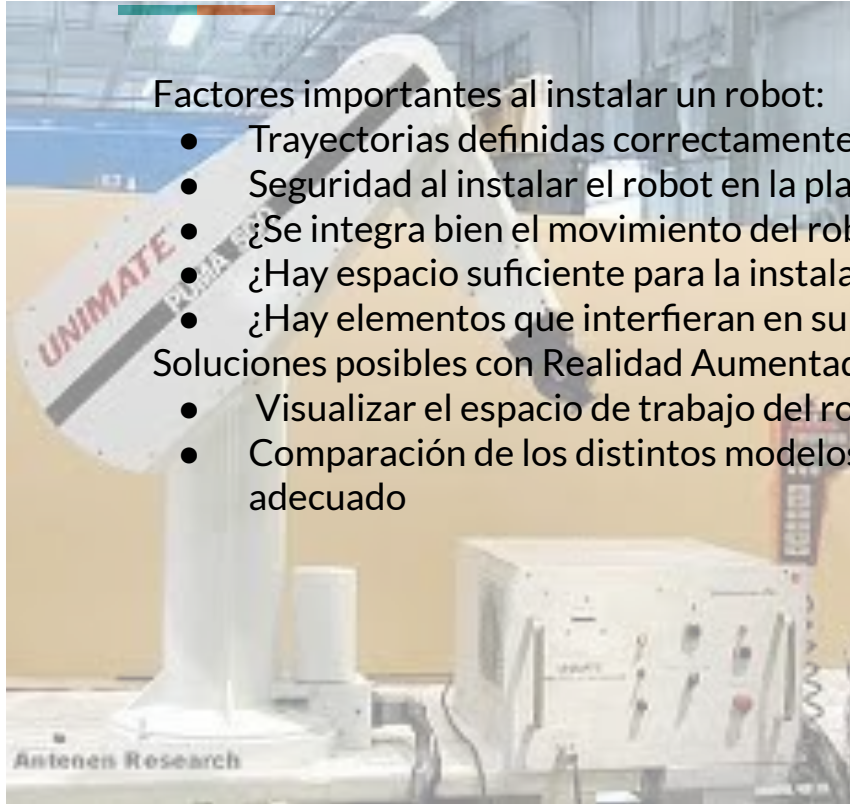
# Introducción

Factores importantes al instalar un robot:

- Trayectorias definidas correctamente, libres de singularidades
- Seguridad al instalar el robot en la planta
- ¿Se integra bien el movimiento del robot al funcionamiento de la planta?
- ¿Hay espacio suficiente para la instalación del robot?
- ¿Hay elementos que interfieran en su espacio de trabajo?

Soluciones posibles con Realidad Aumentada:

- Visualizar el espacio de trabajo del robot y los movimientos que realiza
- Comparación de los distintos modelos de robots para poder seleccionar el adecuado



## Software Utilizado



vuforia™



**Unity**

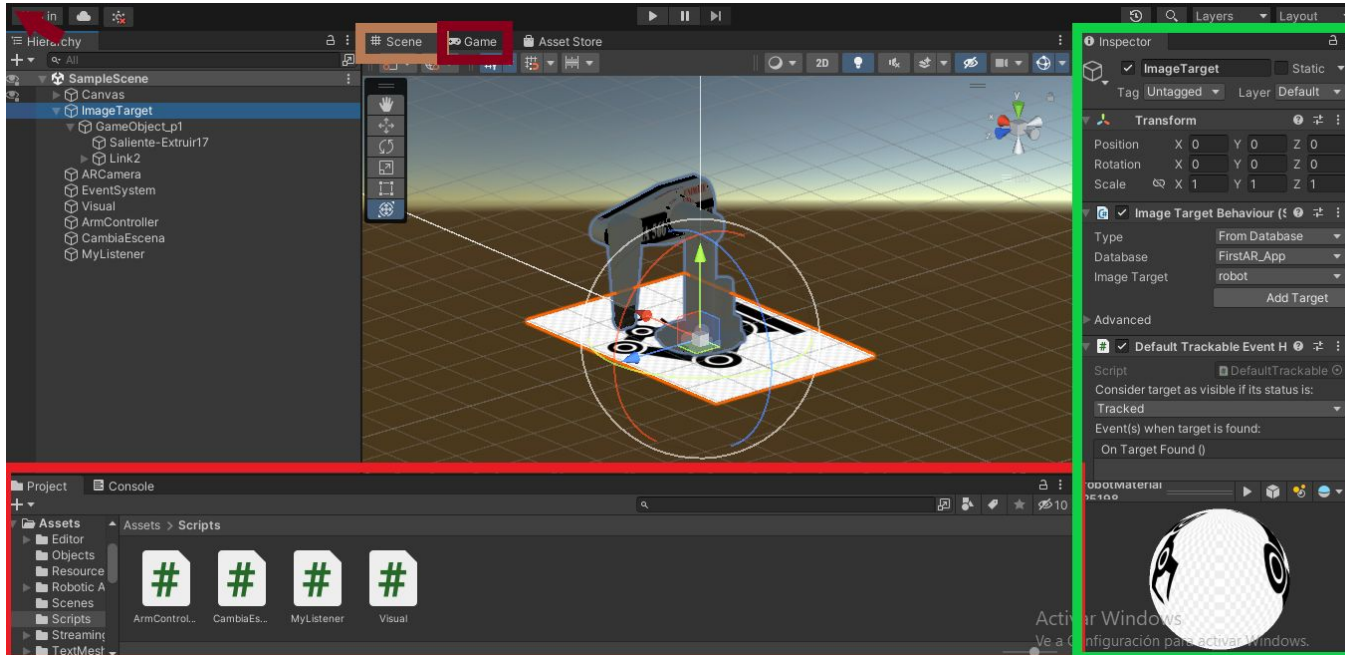


Visual Studio Code



python™

# Entorno de Unity



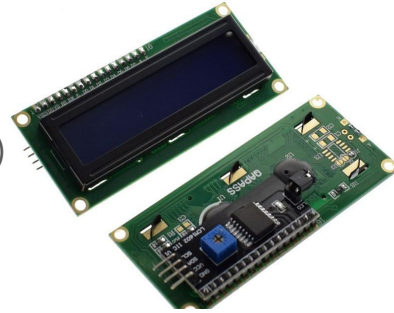
Ventanas:

1. Proyecto
2. Inspector
3. Escena
4. Game View
5. Menú

# Hardware Utilizado

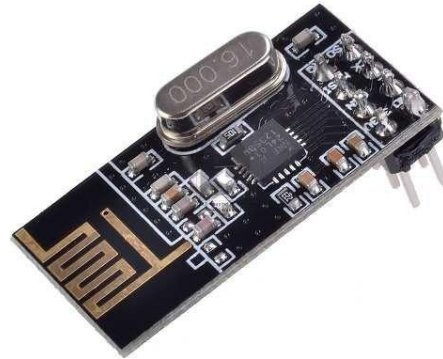
## Control Remoto:

- Módulo de carga (batería)
- Step-up (NRF 24L01)
- Módulo NRF 24L01
- Regulador de tensión

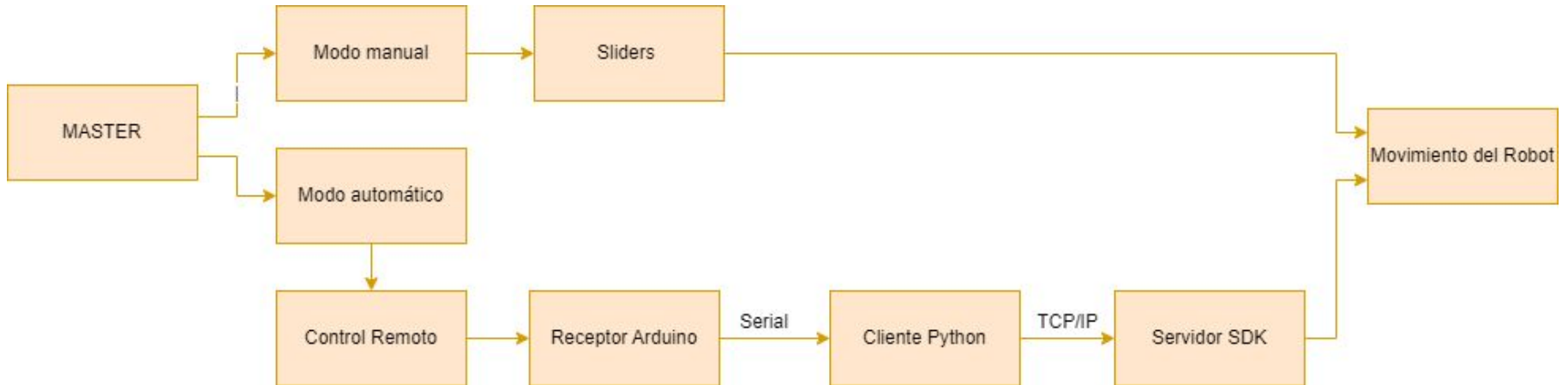


## Receptor:

- Módulo NRFL01
- Arduino Uno
- LCD 1602



## Interfaz: operaciones de movimiento





# Modo Manual

Clase ArmController:

Métodos:

- CheckInput();
- ProcessMovement();

Atributos:

- Límites articulares
- Posiciones Articulares





## Interpolación suave

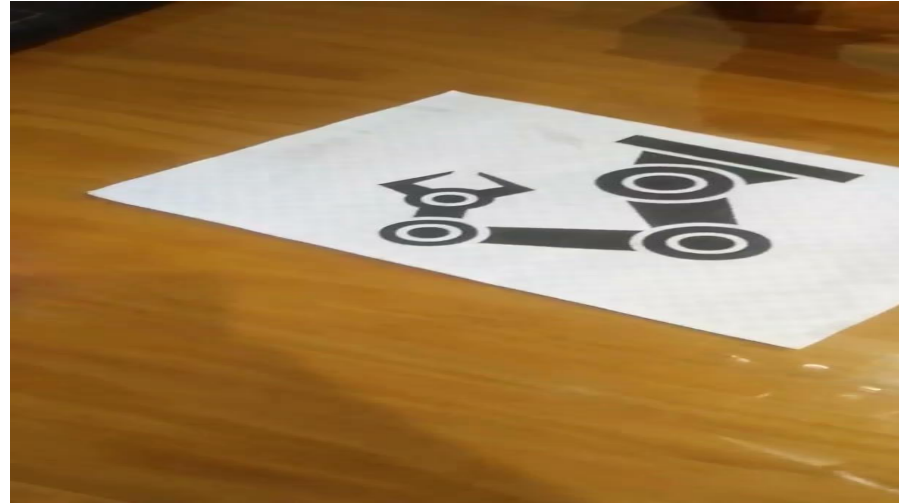
```
//Interpolación esférica suave
for(int i=0;i<6;i++){
    smoothRotation[i] = Quaternion.Slerp(startRotation[i], endRotation[i], 0.1f);
}
// Asigna la rotación suave al objeto
for(int i=0;i<6;i++){
    Link[i].rotation = smoothRotation[i];
}
```



# Modo Remoto

Funcionamiento:

- Inicio de comunicación TCP/IP
- Encendido de CR, activación de RF
- Posicionamiento (potenciómetro)
- Enviar consignas (trama)

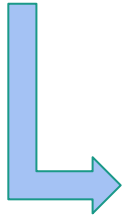


## Modo Remoto: Comunicación TCP/IP

```
# SOCK_STREAM: TCP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_address = (IP, PORT)
print('Conectándose a {} puerto {}'.format(*server_address))
```



```
COM11 - Arduino Uno (COM11)
Número de Puerto: 11
OK
Ingrese la IP del celular: 127.0.0.1
Conectándose a 127.0.0.1 puerto 25001
```



```
TcpClient client = listener.AcceptTcpClient();
NetworkStream ns = client.GetStream();
StreamReader reader = new StreamReader(ns);
msg = reader.ReadToEnd();
stringArray = ParseData(msg);
```



## Modo Remoto: Co-rutina

```
IEnumerator RotateObject(float duration)
{
    float elapsedTime = 0f;

    while (elapsedTime < duration)
    {
        for(int i=0;i<6;i++){
            smoothRotation[i] = Quaternion.Slerp(startRotation[i], endRotation[i], elapsedTime / duration);
        }
        // Asigna la rotación suave al objeto
        for(int i=0;i<6;i++){
            Link[i].rotation = smoothRotation[i];
        }
        elapsedTime += Time.deltaTime;
        yield return null;
    }

    //Hace al valor actual ser el inicial para el siguiente frame
    for(int i=0;i<6;i++){
        startRotation[i]=Link[i].rotation;
    }
}
```



## Continuo de la Virtualidad



Mixed  
reality (MR)

The diagram illustrates the Virtuality Continuum as a horizontal blue double-headed arrow. Above the arrow, a bracket spans from the 'Augmented reality (AR)' position to the 'Augmented virtuality (AV)' position, with the text 'Mixed reality (MR)' centered above it. Below the arrow, four labels are positioned from left to right: 'Real environment', 'Augmented reality (AR)', 'Augmented virtuality (AV)', and 'Virtual environment'.

Real  
environment

Augmented  
reality (AR)

Augmented  
virtuality (AV)

Virtual  
environment

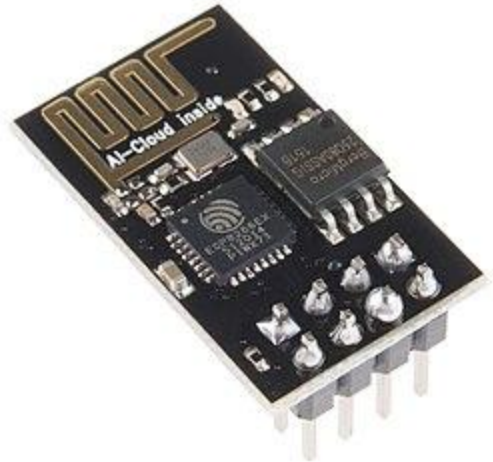
# Resultados



- Se creó una aplicación que simula el movimiento de las articulaciones de un robot Puma 560 de seis grados de libertad.
- Se integró la aplicación con un control tipo joystick para una mayor inmersión y emulación de controladores industriales.
- El movimiento del robot fue fluido y se ajustó correctamente a las posiciones de los sliders en el modo local y en la pantalla del control remoto en el modo remoto.
- Los tiempos de seguimiento de las consignas y la comunicación entre el cliente (computadora y control) y el servidor (aplicación en el celular) fueron aceptables.

## Implementaciones a futuro

- Escala real del robot
- Definición de trayectorias con RTB (Peter Corke)
- Agregar distintos modelos del robot
- Trayectorias predefinidas
- Agregar el LCD al control remoto
- Cambiar la interfaz de conexión (WiFi, Bluetooth)
- Diseño de Control más similar al industrial





# Conclusión

- Inclusión y afianzamiento de conocimientos adquiridos en la cátedra.
- Integración de conceptos de otras cátedras
- Repaso, experimentación y selección de diversas herramientas de desarrollo
- Adquisición de conceptos de programación