

**Grado en Informática Industrial y Robótica**

**Entornos Operativos para  
Robótica e Informática Industrial (EOII)**

Departamento de Informática de Sistemas y Computadoras (DISCA)

*Universitat Politècnica de València*

Trabajo de asignatura

Francisco Blanes

Patricia Balbastre



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

- ▶ Evaluar los conocimientos adquiridos tanto en las sesiones de teoría como en los seminarios y practicas
    - Comunicaciones TCP,UDP,IP
    - Ignition
    - OPC-UA
- } 50% nota final asignatura
- ▶ Los conocimientos no se limitan solo a cuestiones de programación si no también de instalación y configuración de las aplicaciones, así como la comprensión de el funcionamiento de sistemas distribuidos en redes IP



Cliente  
Visualización  
Datos en  
Designer

**Ignition**!  
by inductive automation

**SO Windows**

Thread 1 Cliente OPC-UA

Thread 2 Servidor OPC-UA

Python APP

Thread 3 Puente UDP





- ▶ El origen de los datos será la aplicación **Prosys Simulation Server**
- ▶ Deberá estar accesible en “localhost”
- ▶ Habrá que configurarse los siguientes datos simulados:
  - ▶ **Counter:** de 0 100
  - ▶ **Random:** entre -100 y 100
  - ▶ **Senoidal:** periodo 1 Hz y valores entre -10 y 10
  - ▶ Todos ellos generan un dato nuevo cada 100 milisegundos

## Python bridge

Thread 1 Cliente OPC-UA

Thread 2 Servidor OPC-UA

Thread 3 Puente UDP

El puente será una aplicación Python con tres threads

- Thread 1: tiene el rol de cliente **OPC-UA**
- Thread 2: tiene el rol de servidor **OPC-UA**
- Thread 3: es un puente a **UDP**

### ► El comportamiento de los threads es el siguiente

- Thread 1: como cliente lee del servidor Prosys los datos counter, random y senoidal
- Thread 2: tiene el rol de servidor **OPC-UA** del dato senoidal (solo ese dato)
- Thread 3: es un puente que envía los datos counter y random encapsulados en un datagrama formato **UDP** en formato **JSON**
- Todos ellos deberán tener la periodicidad necesaria para no perder ningún dato
- Entre el thread 1 (que es quien lee los datos del servidor de Prosys) y el resto de threads la comunicación será mediante objetos compartidos protegidos con semáforo



- ▶ **En Ignition deberán configurarse:**
  - ▶ La conexión cliente **OPC-UA** a la aplicación **Python**
  - ▶ El device **UDP** para recibir datos por **UDP** en el puerto **9000**
  - ▶ Todo aquello que se considere necesario
  
- ▶ **Mediante el designer desarrollar una aplicación de interfaz que permita visualizar:**
  - ▶ El dato de la senoidal en un easy chart en modo real time
  - ▶ El dato counter con un tanque de nivel
  - ▶ El dato random un easy chart en modo historico

- ▶ **Fechas de entrega: en horario de practicas**
  - ▶ PL3-A1 Lunes 24 Noviembre
  - ▶ PL3-A2 Jueves 27 Noviembre
- ▶ **La entrega constará de dos fases:**
  - ▶ De forma presencial, el grupo presentará el trabajo, lo que se ha implementado y lo que no.
  - ▶ Se explicará y se harán preguntas para evaluar el conocimiento de las soluciones aplicadas.
  - ▶ Una entrega de documentación por medio de una tarea con:
    - ▶ Código fuente
    - ▶ Un documento PDF con una explicación de lo que se ha solucionado y cómo se ha solucionado.
- ▶ **Evaluación (sobre 10):**
  - ▶ Aplicación Python 5 puntos: 1p por cada thread implementado, 1p si se comunican con objetos compartidos protegidos, 1p si no hay pérdida de datos
  - ▶ Configuración Ignition 1 punto: 0.5p conexión cliente OPC-UA, 0.5p device UDP
  - ▶ Aplicación designer 2 puntos: 0.5p senoidal, 0.5p counter y 1p random
  - ▶ Documentación: 2 puntos