# Simulador Simu\_docker\_rpi

Este es un sistema de un entorno de trabajo sobre un simulador de Raspberry Pi que funciona junto con contendores Docker. En este sentido, el entorno está conformado por diversas partes, que se instalan automáticamente a través del instalador:

* **Imágenes Docker:**  
  Este conformado por varias imágenes Docker. La primera de ellas es un servidor Pigpio, que emula los GPIO de la Raspberry dentro contenedores Docker, que se puede controlar a través de un cliente que se ejecuta en el host de la PC. La segunda imagen que se instala en la PC es la de un Broker Mqtt Mosquitto. Este sirve para realizar pruebas remotas en una red local. El sistema está diseñado para permitir agregar más imágenes Docker en un futuro, tal como un servidor Http, entre otros.
* **Cliente Pigpio:**  
  El cliente Pigpio es el que programa que el usuario para utilizar los GPIO de la Raspberry, que están emulados dentro del Servidor Pipio dentro de Docker. El cliente se debe ejecutar desde el S.O host de la PC del usuario. Es importante mencionar que cuando se ejecuta el cliente, se verá en forma gráfica los sensores y actuadores que están conectados a los pines GPIO de la Raspberry Pi simulada.

## Estructura de Directorios

El directorio que genera el instalador, está organizado de la siguiente manera:

* **DirectorioDocker:**  
  Conformado por el archivo Docker-Compose.yml, que es utilizado para generar las imágenes del Servidor Pigpio y del Borker Mqtt Mosquitto en la PC. Además, contiene archivos de configuración del sistema.
* **Directorio Example:**  
  Este directorio contiene ejemplos de proyecto donde se usan los sensores y actuadores que se pueden simular en el sistema. Este a su vez, se encuentra organizado en dos subdirectorios:
  + **Only Raspberry:**  
    Directorio que contiene ejemplos sencillos usando únicamente la Raspberry emulada.
  + **Raspberry-Android-Mqtt**-  
    Directorio que contiene un ejemplo completo, en donde se representaría la alarma de una casa. Este está conformado por dos partes. Por un lado, la emulación de los sensores y actuadores conectados a la Raspberry Pi, mientras que por otro, la aplicación desarrollada en Android Studio, que permite controlar la Raspberry Pi en forma remota a través de Mqtt. Es importante mencionar, que el control desde la app Android se puede hacer a través de un Broker en la Nube, o usando el Broker Mosquitto instalado en la imagen Docker de la PC.