# Importar flujo en Node-Red

Cuando se tiene el programa *Node-Red* instalado, lo más habitual es comenzar a programar en base a ejemplos creados por otros usuarios, por lo que el proceso de importar un flujo externo es una de las cosas iniciales a aprender.

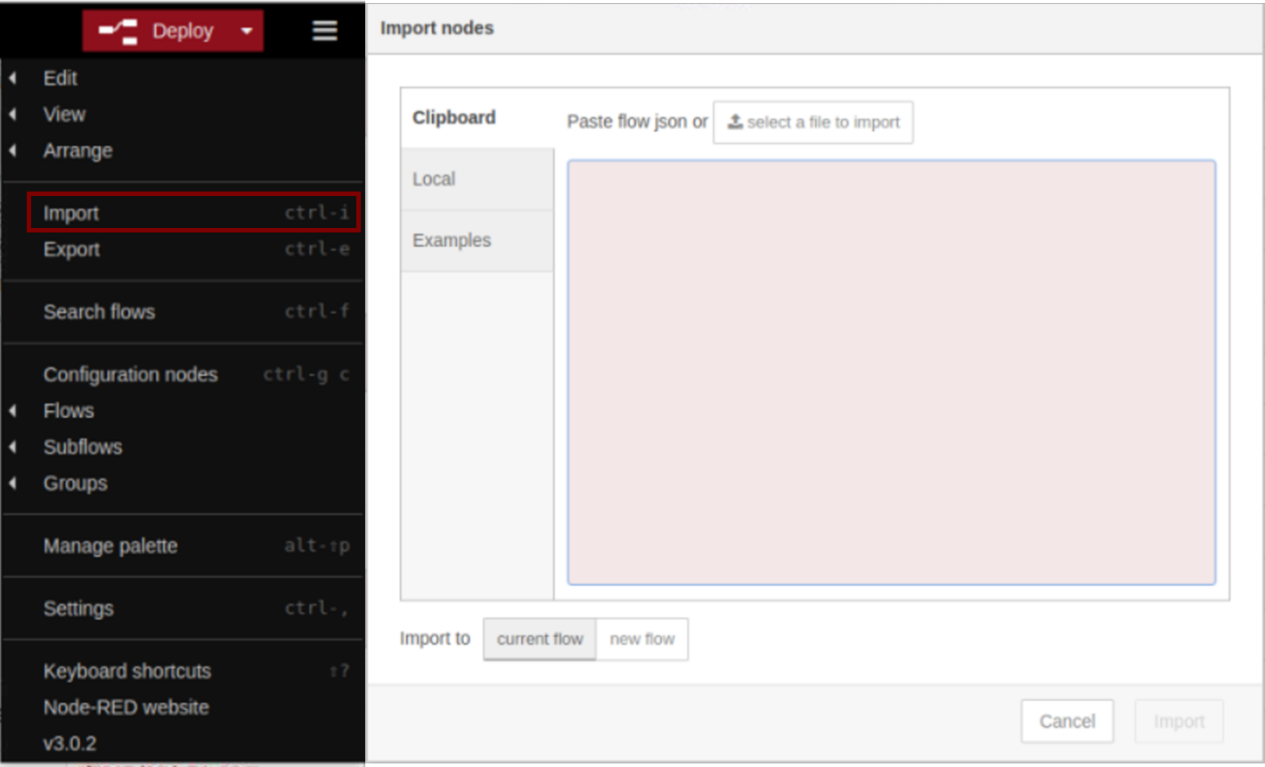


Figura ..- Importar un flujo

Como se muestra en la Figura 1.1, haciendo clic en las tres barras horizontales de la esquina superior izquierda, se abre un menú, en el cual aparece la palabra *Import*. Al hacer clic en ella, se abre la otra ventana mostrada en la figura. En el cuadro de diálogo que aparece en esta ventana deberá de ser insertado el código referente al flujo que se quiere importar.

# Código flujo Proyecto ESP32

A continuación, se dejará escrito el código necesario para importar el flujo creado para el proyecto de un dispositivo que adquiera tres señales fisiológicas en la placa *ESP32*.

|  |
| --- |
| [  {  "id": "8b1077b1c4a722bb",  "type": "tab",  "label": "ESP32",  "disabled": false,  "info": "",  "env": []  },  {  "id": "7ed7242358f1d250",  "type": "mqtt in",  "z": "8b1077b1c4a722bb",  "name": "POX",  "topic": "esp32\_ima/POX",  "qos": "2",  "datatype": "auto-detect",  "broker": "a34b9b453d1a1755",  "nl": false,  "rap": true,  "rh": 0,  "inputs": 0,  "x": 170,  "y": 260,  "wires": [  [  "f7817d93dc74e034",  "7ea32195d16521ee"  ]  ]  },  {  "id": "b062bafc47228925",  "type": "mqtt in",  "z": "8b1077b1c4a722bb",  "name": "GSR",  "topic": "esp32\_ima/GSR",  "qos": "2",  "datatype": "auto-detect",  "broker": "a34b9b453d1a1755",  "nl": false,  "rap": true,  "rh": 0,  "inputs": 0,  "x": 170,  "y": 360,  "wires": [  [  "a7089608435f8ff9",  "58c06e59c43834f0"  ]  ]  },  {  "id": "f7817d93dc74e034",  "type": "file",  "z": "8b1077b1c4a722bb",  "name": "GuardarPOX",  "filename": "/home/pi/Desktop/Data/esp32/POX.csv",  "filenameType": "str",  "appendNewline": true,  "createDir": true,  "overwriteFile": "false",  "encoding": "none",  "x": 390,  "y": 260,  "wires": [  []  ]  },  {  "id": "a7089608435f8ff9",  "type": "file",  "z": "8b1077b1c4a722bb",  "name": "GuardarGSR",  "filename": "/home/pi/Desktop/Data/esp32/GSR.csv",  "filenameType": "str",  "appendNewline": true,  "createDir": true,  "overwriteFile": "false",  "encoding": "none",  "x": 390,  "y": 360,  "wires": [  []  ]  },  {  "id": "7ea32195d16521ee",  "type": "debug",  "z": "8b1077b1c4a722bb",  "name": "debug 1",  "active": false,  "tosidebar": true,  "console": false,  "tostatus": false,  "complete": "payload",  "targetType": "msg",  "statusVal": "",  "statusType": "auto",  "x": 360,  "y": 220,  "wires": []  },  {  "id": "58c06e59c43834f0",  "type": "debug",  "z": "8b1077b1c4a722bb",  "name": "debug 2",  "active": false,  "tosidebar": true,  "console": false,  "tostatus": false,  "complete": "payload",  "targetType": "msg",  "statusVal": "",  "statusType": "auto",  "x": 360,  "y": 320,  "wires": []  },  {  "id": "677167b594c080dd",  "type": "mqtt in",  "z": "8b1077b1c4a722bb",  "name": "ECG",  "topic": "esp32\_ima/ECG",  "qos": "2",  "datatype": "auto-detect",  "broker": "a34b9b453d1a1755",  "nl": false,  "rap": true,  "rh": 0,  "inputs": 0,  "x": 170,  "y": 460,  "wires": [  [  "ec48841d9d16a73d",  "770eac6237d03a1a"  ]  ]  },  {  "id": "ec48841d9d16a73d",  "type": "file",  "z": "8b1077b1c4a722bb",  "name": "GuardarECG",  "filename": "/home/pi/Desktop/Data/esp32/ECG.csv",  "filenameType": "str",  "appendNewline": true,  "createDir": true,  "overwriteFile": "false",  "encoding": "none",  "x": 390,  "y": 460,  "wires": [  []  ]  },  {  "id": "770eac6237d03a1a",  "type": "debug",  "z": "8b1077b1c4a722bb",  "name": "debug 3",  "active": true,  "tosidebar": true,  "console": false,  "tostatus": false,  "complete": "payload",  "targetType": "msg",  "statusVal": "",  "statusType": "auto",  "x": 360,  "y": 420,  "wires": []  },  {  "id": "a34b9b453d1a1755",  "type": "mqtt-broker",  "name": "raspi\_lab",  "broker": "localhost",  "port": "1883",  "clientid": "",  "autoConnect": true,  "usetls": false,  "protocolVersion": "4",  "keepalive": "60",  "cleansession": true,  "birthTopic": "",  "birthQos": "0",  "birthPayload": "",  "birthMsg": {},  "closeTopic": "",  "closeQos": "0",  "closePayload": "",  "closeMsg": {},  "willTopic": "",  "willQos": "0",  "willPayload": "",  "willMsg": {},  "userProps": "",  "sessionExpiry": ""  }  ] |