



**TECNOLÓGICO
NACIONAL DE MÉXICO**



Tecnológico Nacional de México

Campus La Laguna

Ingeniería en Sistemas Computacionales

Sistemas Programables

Unidad 6

Practica 6.1 Conexión Wi-Fi

Equipo 1

Nombre del alumno – Número de control – Email:

Adame Sandoval José Misael – 18131209 – jmisaeladame@gmail.com

Ruíz Martínez Kevin Alejandro – 18131280 – kev.ruiz4696@gmail.com

Sotomayor Saldívar Daniel – 18131284 – daniel15ss02@gmail.com

Docente:

M. S. C. Miguel Ángel De La Vara Ramírez

Torreón, Coahuila

5 de diciembre de 2021

Índice

Objetivo de la práctica	1
Materiales usados	1
Método	1
Conexión	1
Aplicación.....	2
Resultados.....	2

Objetivo de la práctica

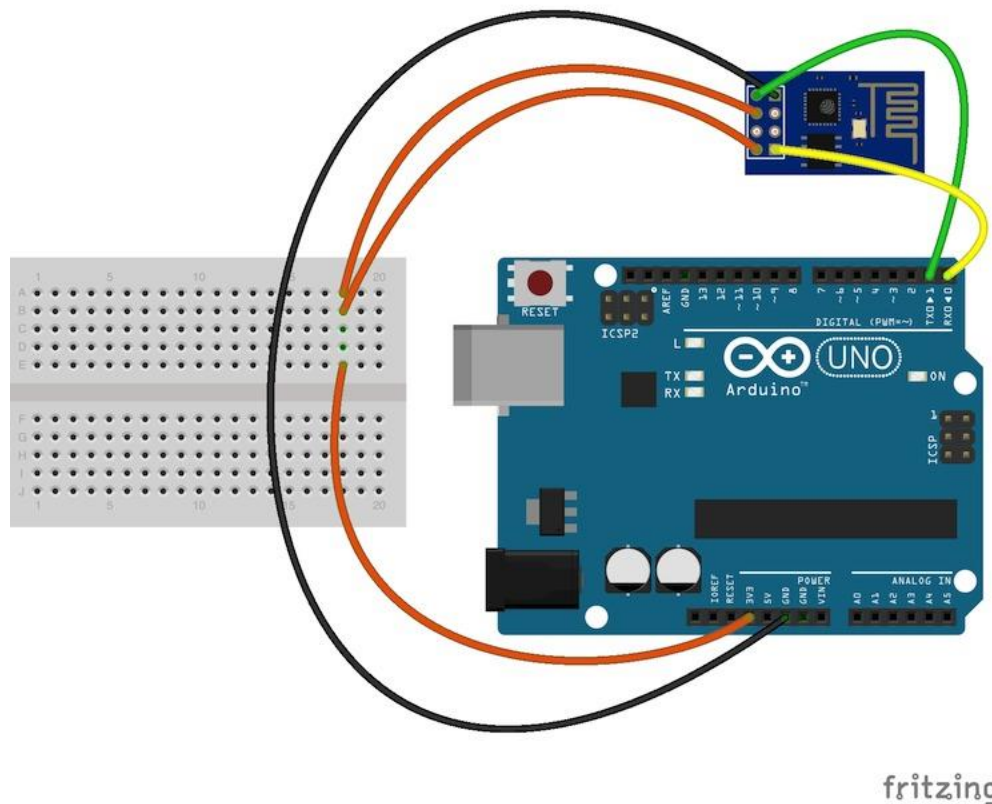
Utilizando el módulo WIFI ESP8266 nos conectaremos por medio de wifi al Arduino.

Materiales usados

- 1 Arduino UNO
- 1 módulo WIFI ESP8266
- Cables (jumpers)

Método

Conexión



La conexión es simple, el módulo WIFI ESP8266 va conectado de la siguiente manera al Arduino UNO:

- Vcc del módulo va conectado al proboard y este al 3.3V del Arduino UNO
- GND del módulo va conectado al GND del Arduino UNO

- TXD del módulo va conectado al pin 1 del Arduino UNO
- RXD del módulo va conectado al pin 0 del Arduino UNO

Aplicación

En esta primera practica que consiste en conectarnos al módulo WIFI utilizaremos los comandos AT, donde "AT" significa "**attention**". Los cuales son utilizados para controlar el módulo de WIFI.

Un comando AT tiene el formato de "AT<x><n>", o "AT&<x><n>", donde "<x>" es el comando específico y "<n>" son los argumentos para ese comando.

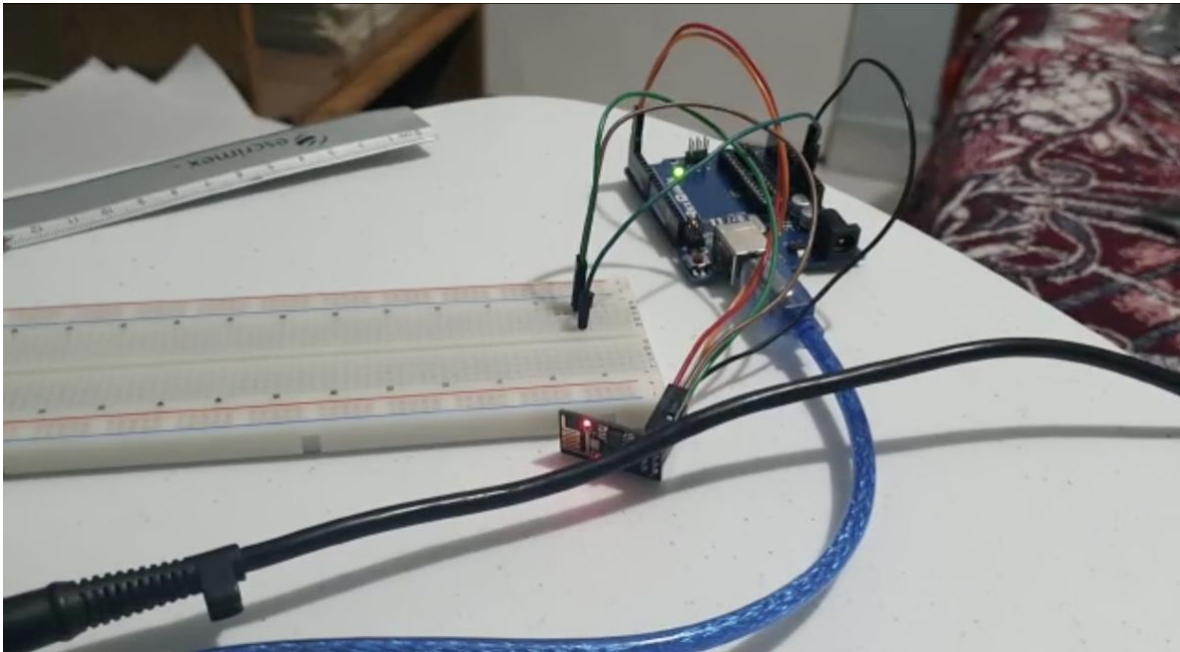
Tipos de comandos AT y sus respuestas:

Probar	AT+<x>=?	Este comando retorna la lista de parámetros y rangos de valores establecidos con el comando correspondiente para escribir o por procesos internos.
Leer	AT+<x>?	Este comando retorna el valor establecido actual del parámetro o parámetros.
Escribir	AT+<x>=<...>	Este comando establece los parámetros que se pueden definir por el usuario.
Ejecutar	AT+<x>	Este comando lee parámetros que no se pueden modificar, afectados únicamente por el dispositivo.

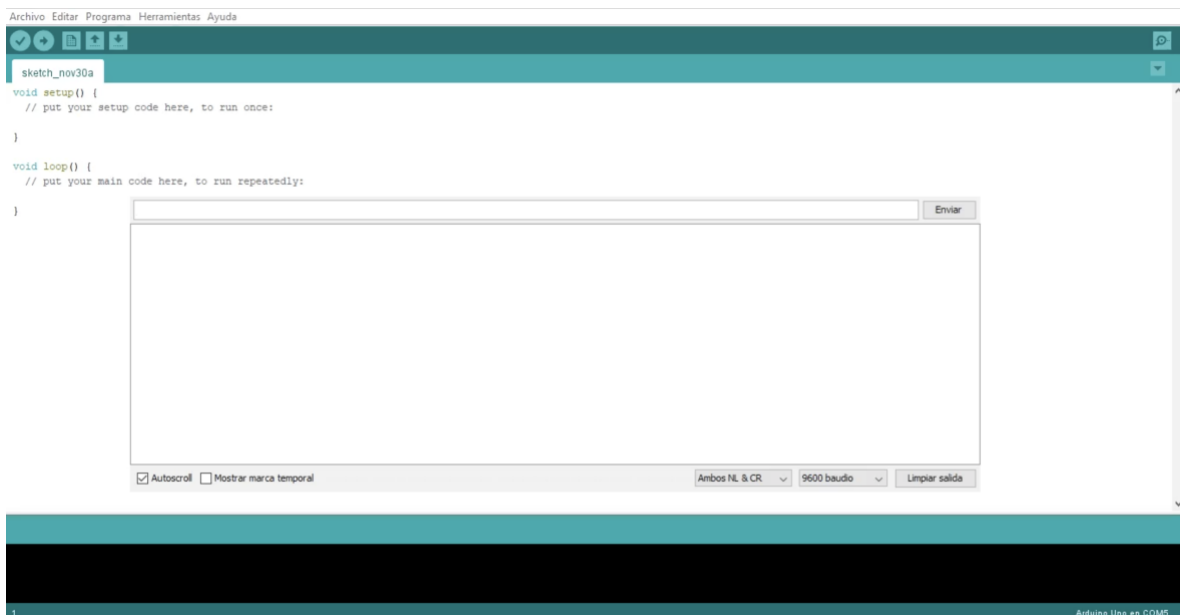
Resultados

Hay dos videos en los que se pueden ver los resultados completamente, son los archivos "SP0106PR01A.mp4" y "SP0106PR01B.mp4", los objetivos se cumplieron.

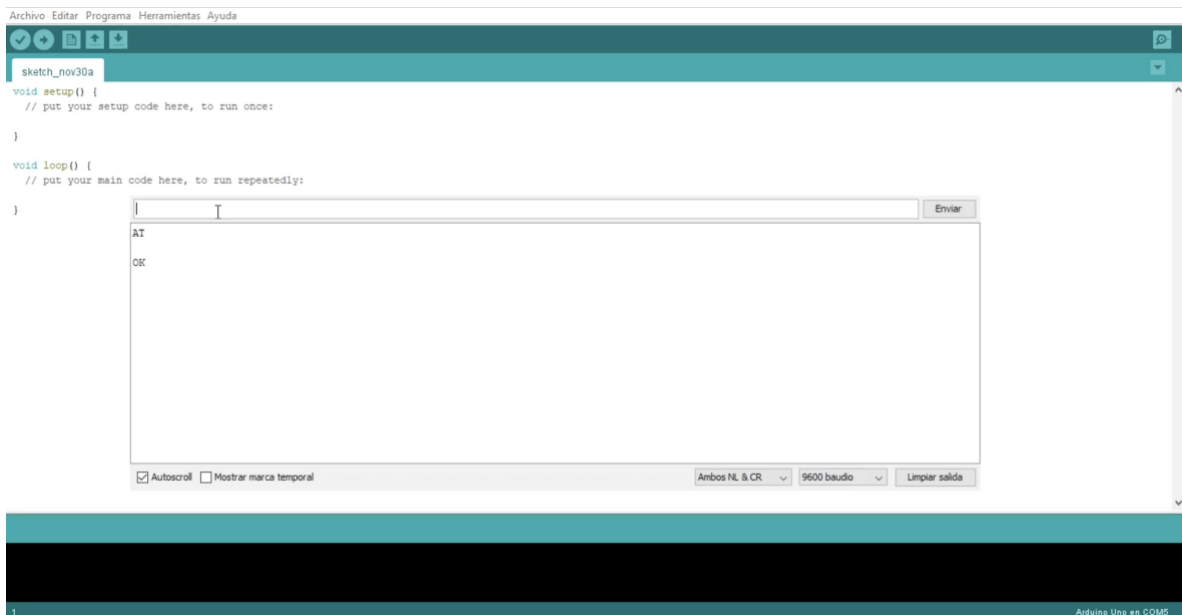
Conexión de Arduino UNO a Laptop.



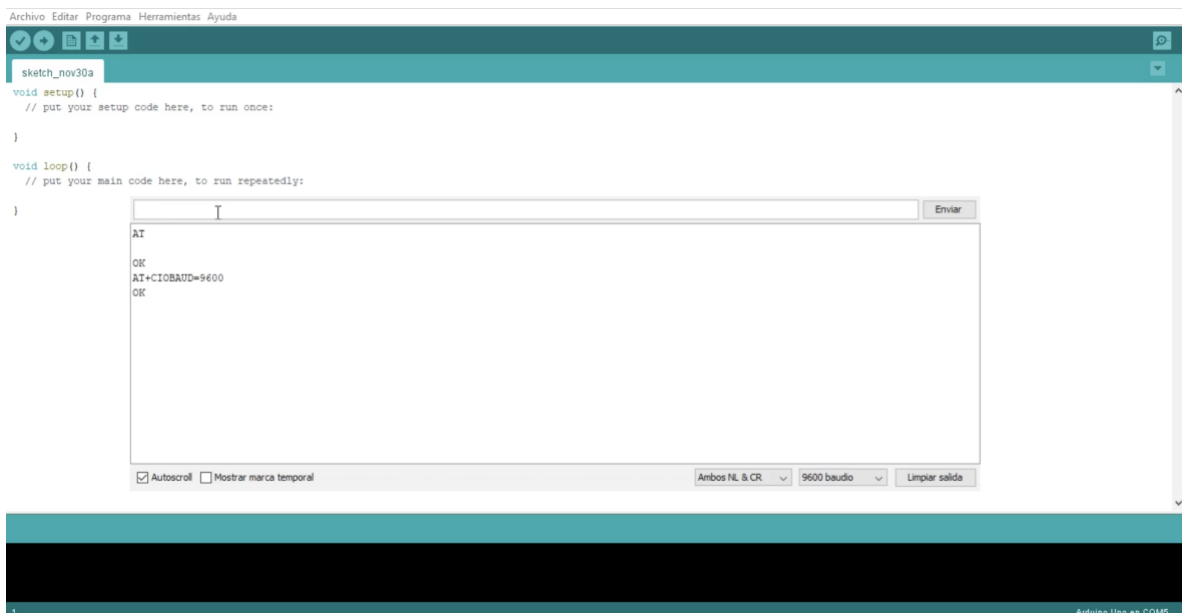
Se coloca el monitor serie con la funcionalidad “Ambos NL & CR” y con velocidad de 9600 baudio.



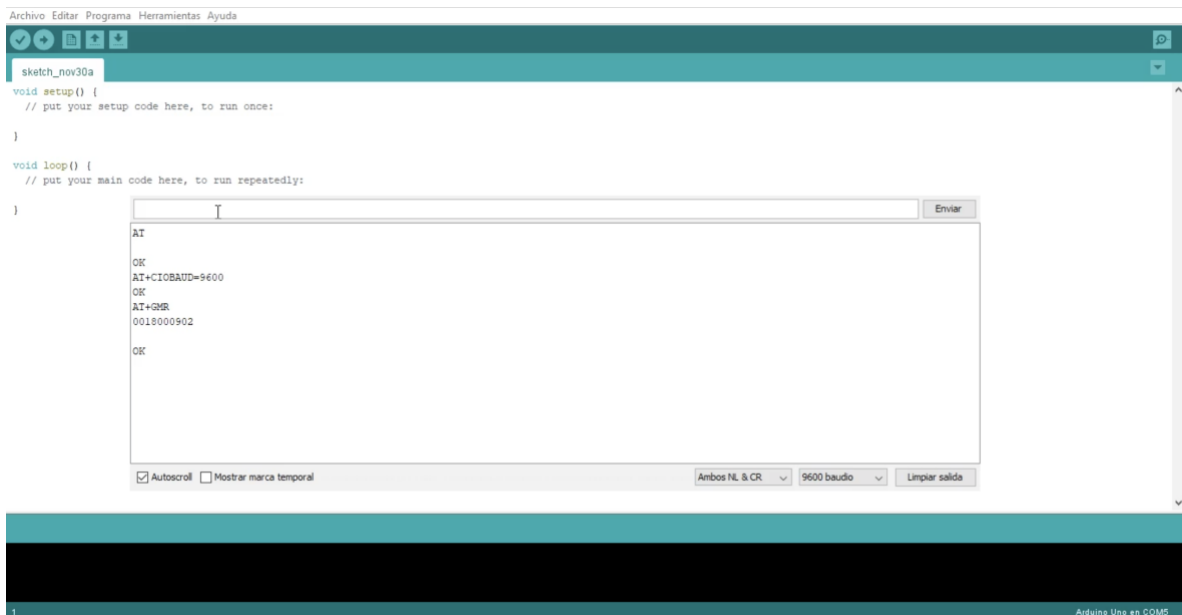
Colocamos el comando “AT” para comprobar que el módulo este recibiendo nuestros comandos.



Siguientes comando: “AT+CIOBAUD=9600” el cual nos permitira neter el módulo con la velocidad en 9600.



El comando “AT+GMR” nos proporciona la versión de módulo que tenemos.



The screenshot shows the Arduino IDE interface with the serial monitor open. The code in the editor is as follows:

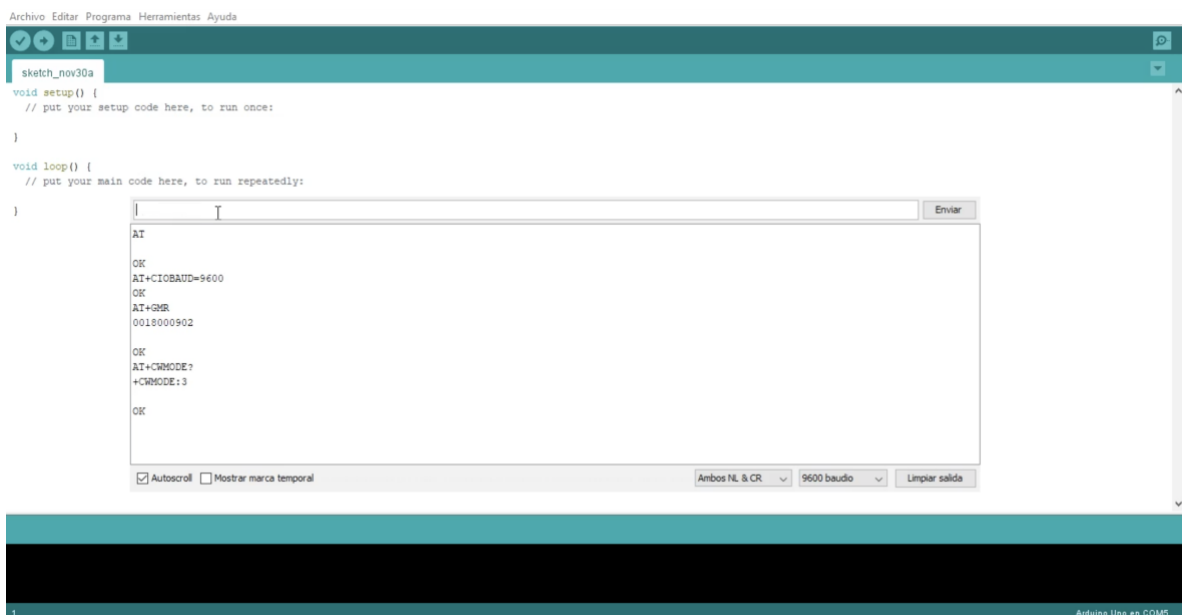
```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

The serial monitor displays the following output:

```
AT  
OK  
AT+CIOBAUD=9600  
OK  
AT+QMR  
0018000902  
OK
```

At the bottom of the serial monitor, the settings are: ☒ Autoscrol, ☐ Mostrar marca temporal, Ambos NL & CR, 9600 baudo, and Limpiar salida.

El comando “AT+CWMODE?” nos dira en que modo AP esta actualmente el módulo.

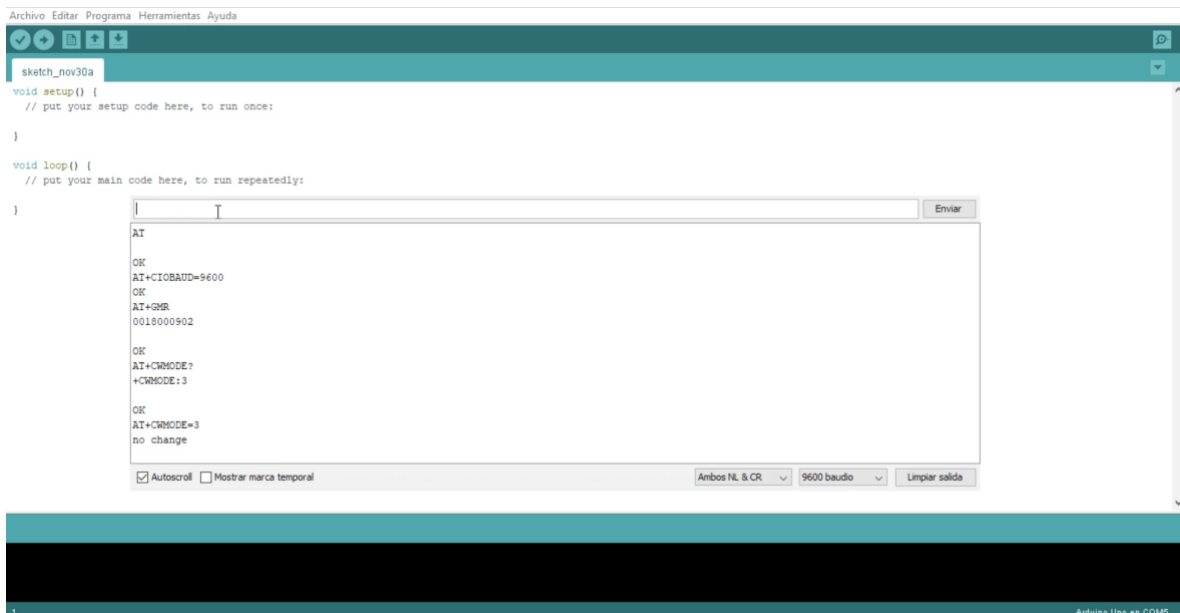


The screenshot shows the Arduino IDE interface with the serial monitor open. The code in the editor is the same as in the previous screenshot. The serial monitor displays the following output:

```
AT  
OK  
AT+CIOBAUD=9600  
OK  
AT+QMR  
0018000902  
OK  
AT+CWMODE?  
+CWMODE:3  
OK
```

At the bottom of the serial monitor, the settings are: ☒ Autoscrol, ☐ Mostrar marca temporal, Ambos NL & CR, 9600 baudo, and Limpiar salida.

Esta vez ponemos el comando “AT+CWMODE=3” el cuál nos arrojo un sin cambios por que ya se encontraba en este, siendo el modo AP + estación (modo dual).



```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

```
AT  
OK  
AT+IOBAUD=9600  
OK  
AT+GMR  
0018000902  
OK  
AT+CWMODE?  
+CWMODE:3  
OK  
AT+CWMODE=3  
no change
```

Con el comando “AT+CWJAP” el cual nos proporciona la lista los Access Points disponibles para conectarse.

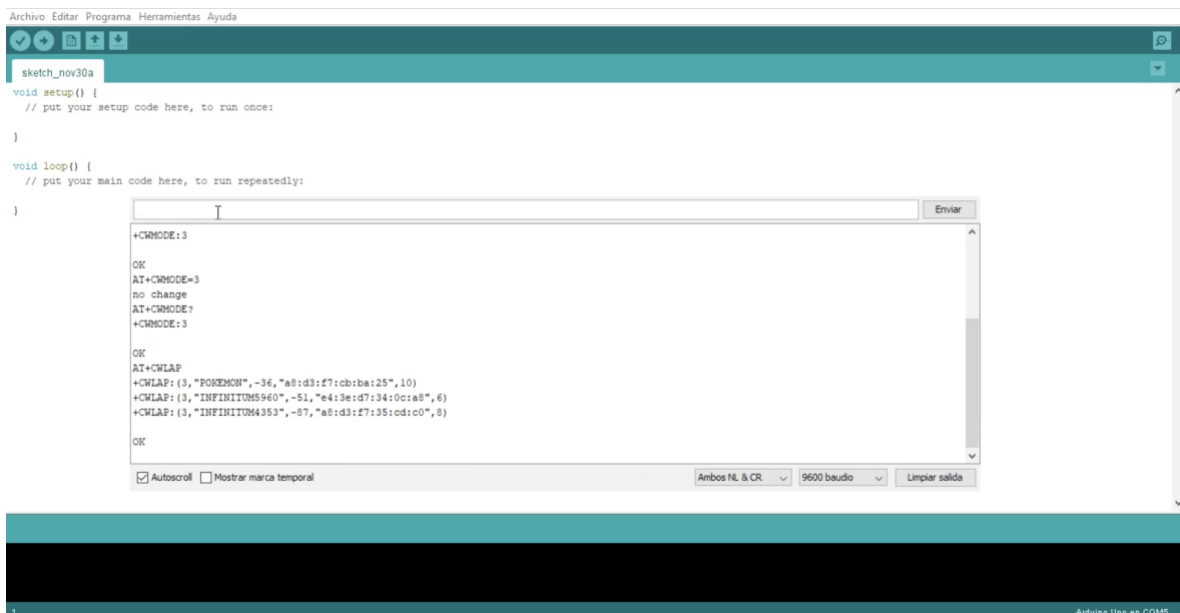
0 = Abierto

1 = WEP

2 = WPA PSK

3 = WPA2 PSK

4 = WPA WPA2 PSK

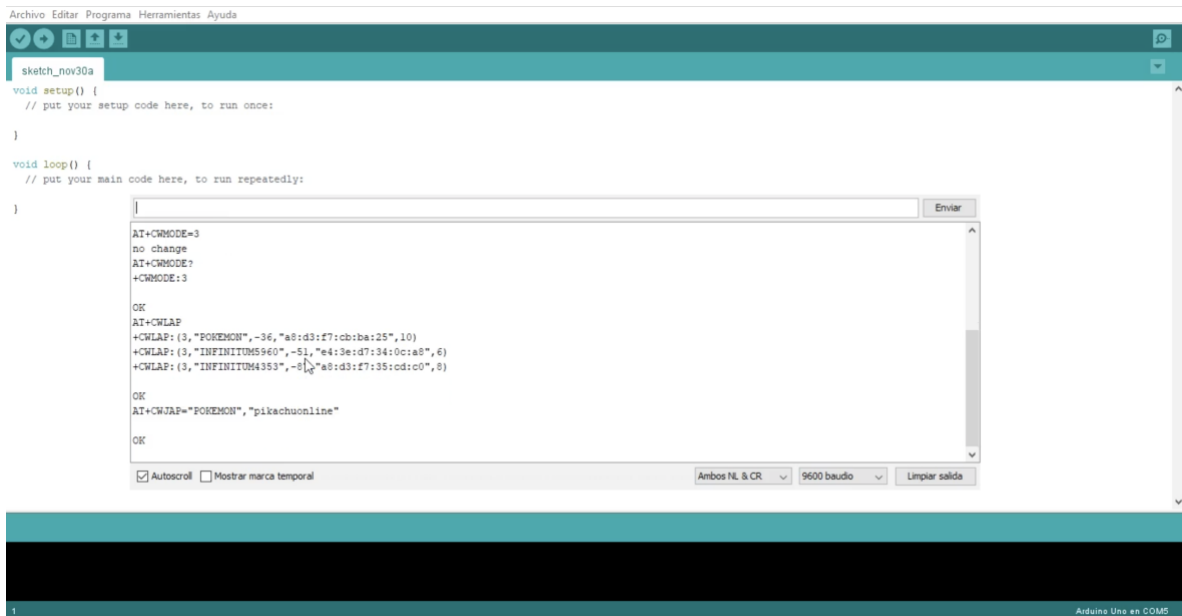


```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

```
+CWMODE:3  
OK  
AT+CWMODE=3  
no change  
AT+CWMODE?  
+CWMODE:3  
OK  
AT+CWJAP  
+CWJAP: (3,"POKEMON",-36,"a8:d3:f7:cb:ba:25",10)  
+CWJAP: (3,"INFINITUM960",-51,"e4:3e:d7:34:0c:a8",6)  
+CWJAP: (3,"INFINITUM4353",-87,"a8:d3:f7:35:cd:c0",8)  
OK
```

Como último código ya para poder realizar la conexión es el “AT+CWJAP=”POKEMON”,”pikachuonline”. Como primer parametro mandamos el nombre

de la red a la que nos queremos conectar y en el segundo pasamos la contraseña de la red, si la conexión es exitosa nos mandara un “Ok”.



The screenshot shows the Arduino IDE interface. The top menu bar includes 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. The toolbar contains icons for opening, saving, and running. The sketch is named 'sketch_nov30a'. The code in the editor is as follows:

```
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

The serial monitor window is open, showing the following output:

```
AT+CRMODE=3  
no change  
AT+CRMODE?  
+CRMODE:3  
  
OK  
AT+CWJAP  
+CWJAP: (3, "POKEMON", -36, "a8:d3:f7:cb:ba:25", 10)  
+CWJAP: (3, "INFINITUM5960", -51, "e4:3e:d7:34:0c:a8", 6)  
+CWJAP: (3, "INFINITUM4353", -8, "a8:d3:f7:35:cd:c0", 8)  
  
OK  
AT+CWJAP="POKEMON", "pikachuonline"  
  
OK
```

At the bottom of the IDE, the status bar indicates 'Arduino Uno en COM5'.