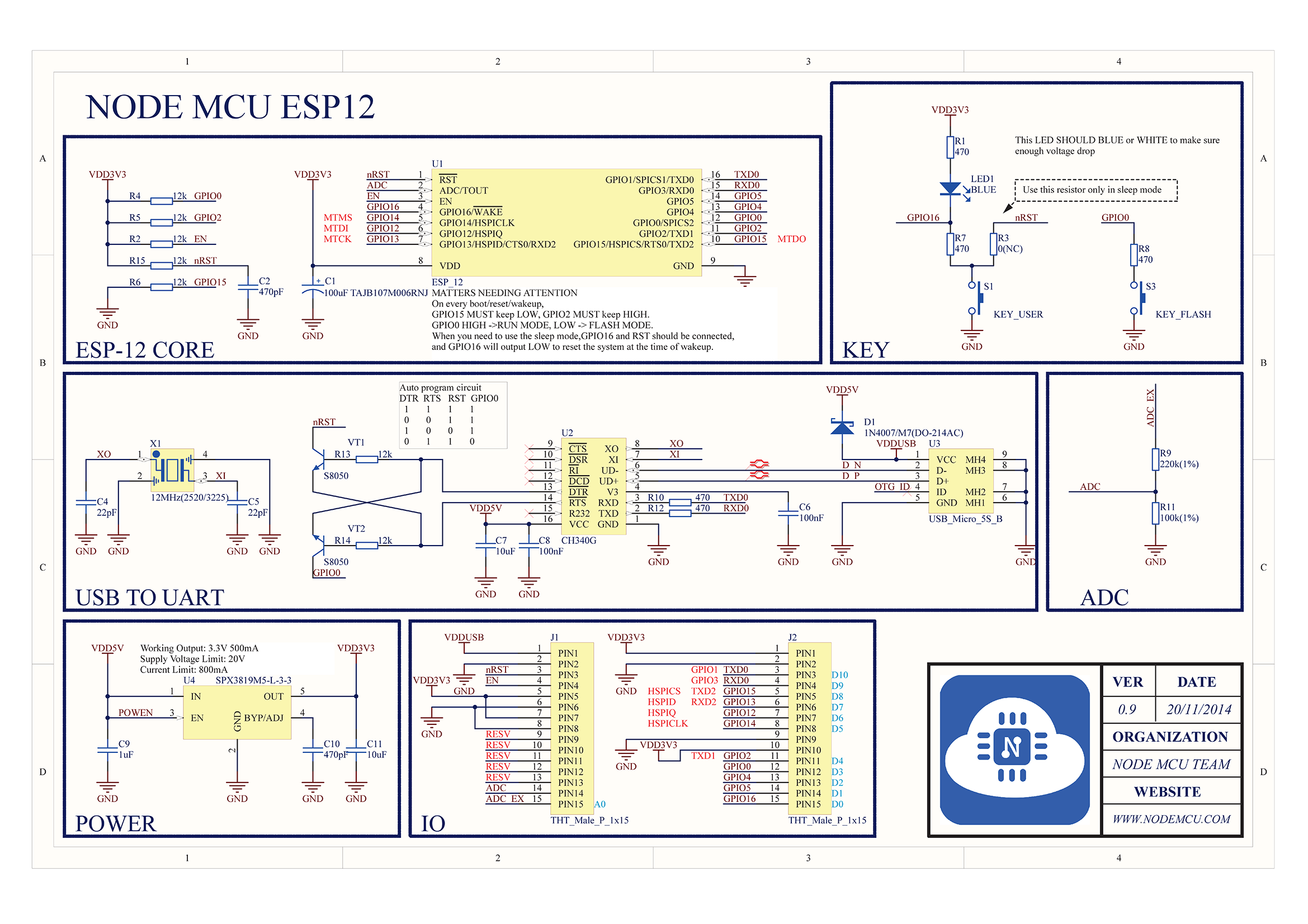
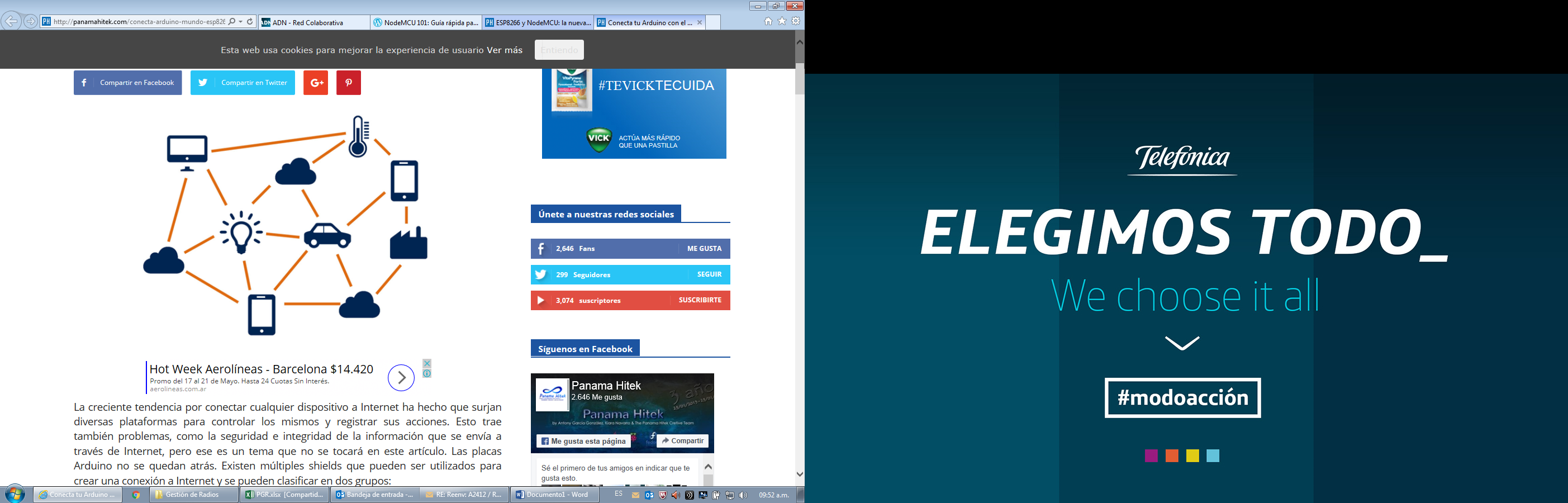
****

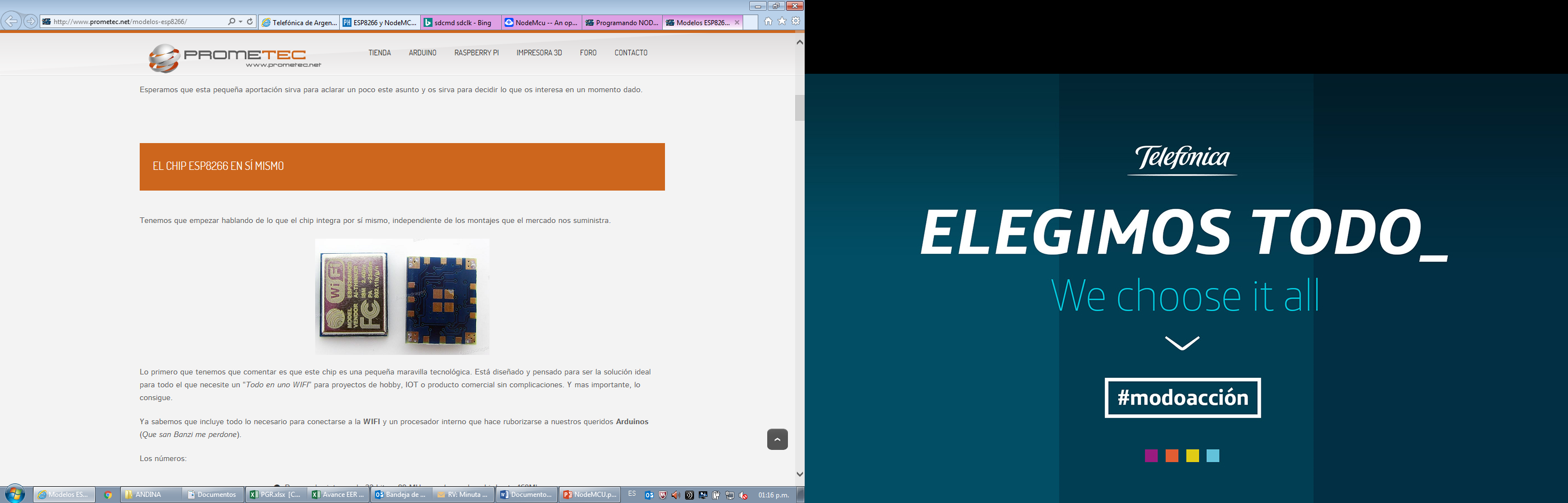
**ESP8266 y NodeMCU: la nueva generación de sistemas embebidos**

La tendencia actual a la programación de microcontroladores está desplazándose rápidamente hacia el área del Internet of Things (IoT) o “Internet de las cosas”. Esto quiere decir que los dispositivos a emplear deben de ser capaces de conectarse para poder utilizar las últimas librerías y ambientes de trabajo.



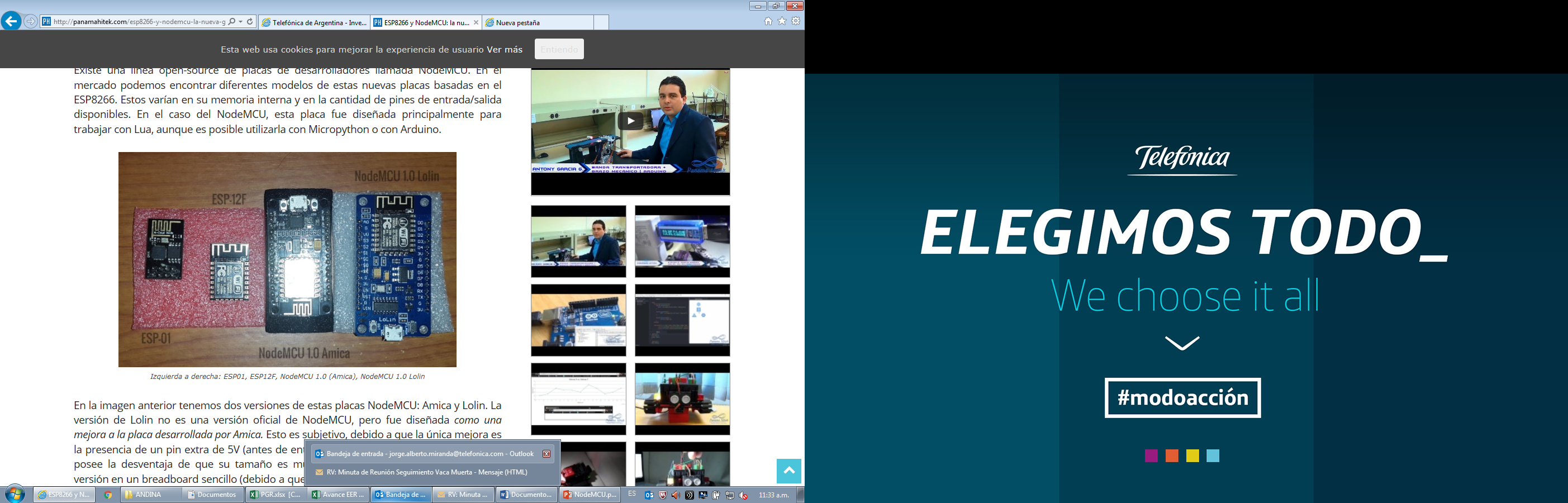
Los dispositivos utilizados comúnmente de la línea de Arduino no tienen conectividad a Internet. Sus últimos modelos han sido introducidos con este enfoque en mente sin embargo la línea que ha dado mejores resultados es la de los dispositivos basados en el chip ESP8266.

El ESP8266 es un pequeño microcontrolador desarrollado por la firma Espressif, de bajo costo y con Wifi integrado. Su bajo costo y pequeño tamaño lo hace ideal para dotar de conectividad Wifi a cualquier sensor o actuador para IoT. Además se puede programar utilizando Arduino IDE y muchas de las librerías que ya están disponibles para Arduino.



A partir de este chip han surgido una gran cantidad de nuevos modelos y marcas de placas de desarrollo similares al Arduino pero con capacidades nativas integradas, tales como conectividad Bluetooth y/o WiFi. Estas placas difieren en su memoria interna y en la cantidad de pines de entrada/salida disponibles. Una de estas placas es la línea llamada **NodeMCU**.

**NodeMCU** es una iniciativa open Source para el desarrollo de un modelo sencillo de integrar la IOT en donde haga falta. Para ello desarrollan modelos de hardware y software que a su vez facilite el desarrollo de programas y aplicaciones basados en WIFI.



En la imagen anterior tenemos dos versiones de estas placas NodeMCU: Amica y Lolin. La versión de Lolin no es una versión oficial de NodeMCU pero fue diseñada como una mejora a la placa desarrollada por Amica. La única mejora es la presencia de un pin extra de 5V (antes de entrar al regulador de voltaje). Sin embargo posee la desventaja de que su tamaño es mucho mayor y no es posible utilizar esta versión en un breadboard sencillo (debido a que no tiene columnas sobrantes a los lados para hacer conexiones).

Una ventaja con respecto a otros módulos que integran el **ESP2866** es que todos sus pines están disponibles en el exterior, en un montaje que se puede colocar en una protoboard y además incluye un conector mini USB para programar el chip interno y comunicarse con la PC si es necesario.

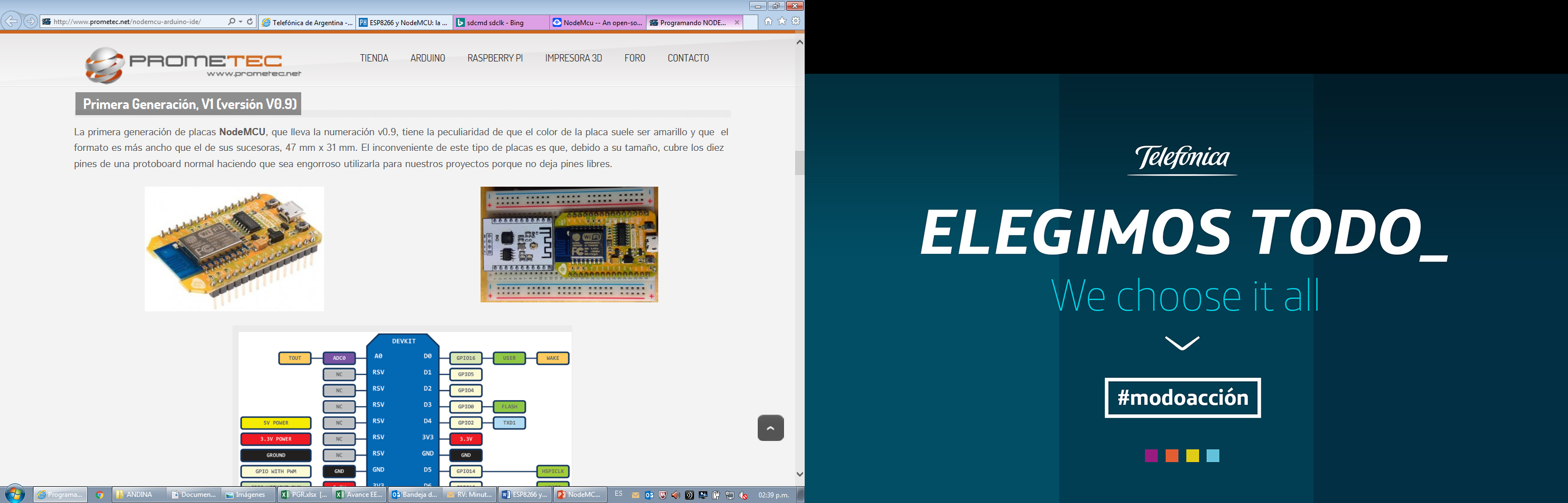
**Variantes de NodeMCU**

A continuación se presentan los diferentes modelos de las placas NodeMCU dado que, al ser un proyecto open-source, existen una serie de variantes ya que cualquiera puede producir y vender placas NodeMCU.

Los fabricantes son principalmente tres firmas, **Amica**, **Doit**/**Smartduino** y **Lolin**/**WeMos** pero se detallaran las versiones de **NodeMCU Amica**.

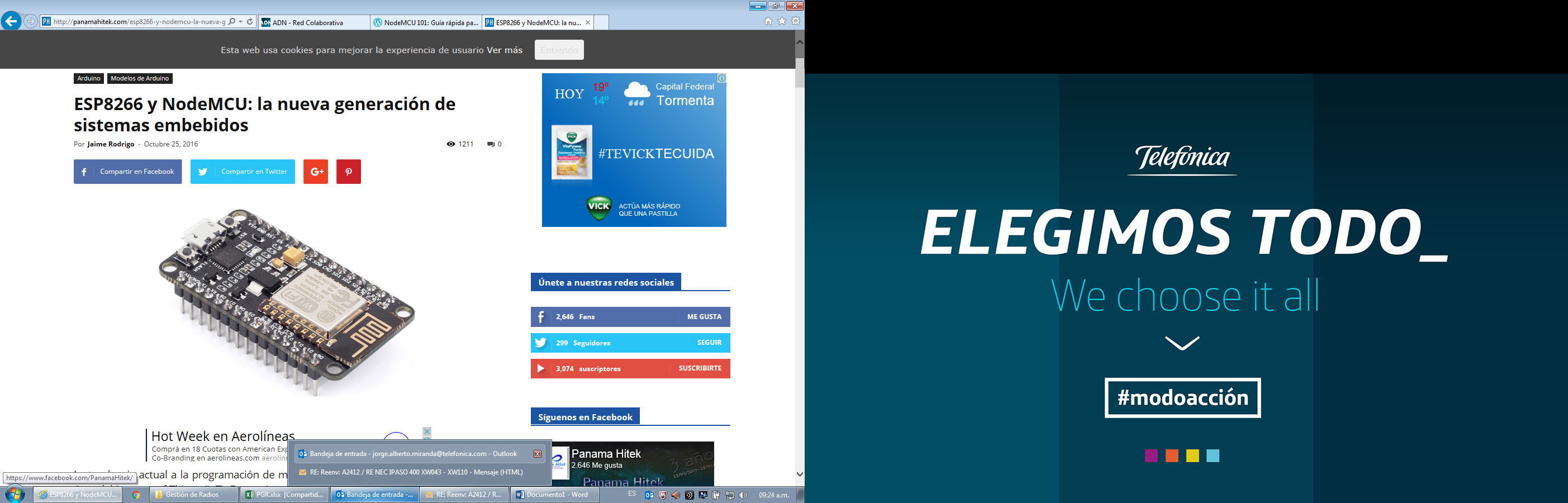
* **Primera Generación, V1 (versión V0.9)**

La primera generación de placas NodeMCU, que lleva la numeración v0.9, tiene la peculiaridad de que el color de la placa suele ser amarillo y que  el formato es más ancho que el de sus sucesoras, 47mm x 31mm. El inconveniente de este tipo de placas es que, debido a su tamaño, cubre los diez pines de una protoboard normal haciendo que sea engorroso utilizarla para nuestros proyectos porque no deja pines libres.



* **Segunda Generación, V2 (versión V1.0)**

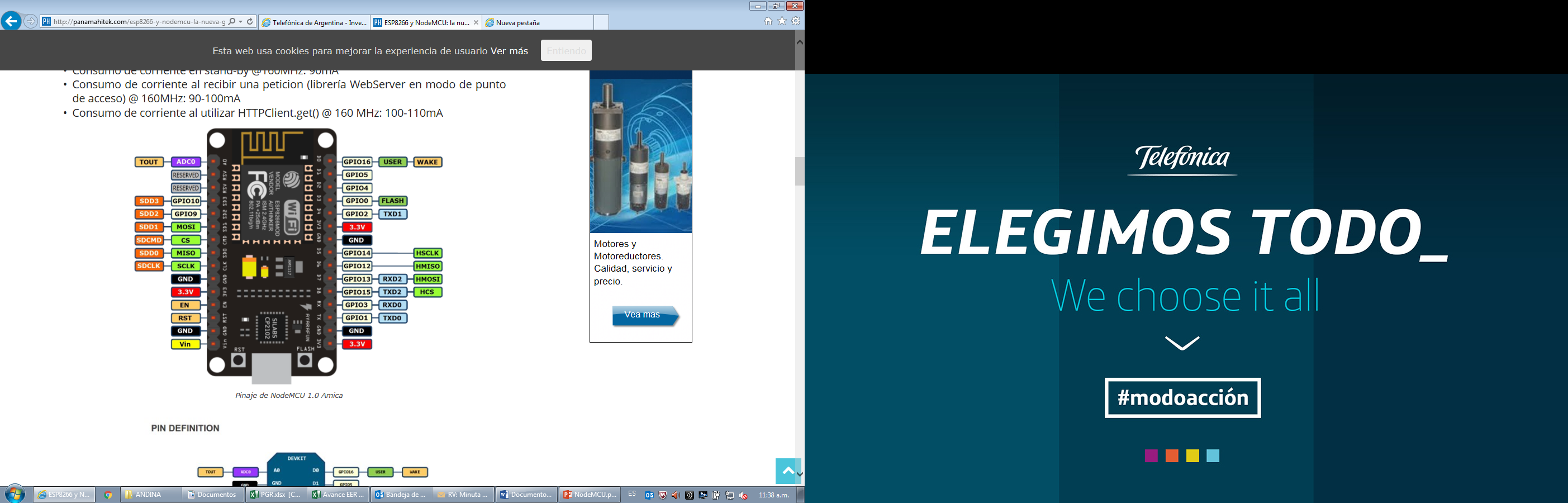
En la segunda generación de placas se soluciona el problema del tamaño y es posible usar las nuevas NodeMCU v1.0 en protoboards sin incomodidades, ya que deja una fila de pines disponibles a cada lado. Además llevan los nuevos chips mejorados ESP12E



**Especificaciones técnicas de NodeMCU 1.0 (Amica)**

* Voltaje de entrada (USB): 5V
* Voltaje de salida en los pines: 3.3V
* Voltaje de referencia en el ADC: 3.3V
* Corriente nominal por pin: 12mA
* Frecuencia de procesador: 80MHz (160MHz máx.)
* 4MB Flash
* Consumo de corriente en stand-by @80MHz: 80mA
* Consumo de corriente al recibir una petición (librería WebServer en modo de punto de acceso) @ 80MHz: 90mA
* Consumo de corriente al utilizar HTTPClient.get() @ 80 MHz: 100-110mA
* Consumo de corriente en stand-by @160MHz: 90mA
* Consumo de corriente al recibir una petición (librería WebServer en modo de punto de acceso) @ 160MHz: 90-100mA
* Consumo de corriente al utilizar HTTPClient.get() @ 160 MHz: 100-110mA

**Pin Out**



TOUT: 10 bit 0v-1.1v analog input

ADC: Analog to Digital Converter

GPIO: General Purpose Input/Ouput

MOSI: Master Out Slave In (Bus SPI)

CS: Slave Select (Bus SPI)

MISO: Master In Slave Out (Bus SPI)

SCLK: Serial Clock (Bus SPI)

GND: Ground

3.3V: Alimentacion

Vin:

EN:

RST:

RXD2 (GPIO13): Received Data 2 (RS232/RS485 Channel)

TXD2 (GPIO15): Transmitted Data 2 (RS232/RS4585 Channel)

RXD0 (GPIO3): Received Data 0 (RS232/RS485 Channel)

TXD0 (GPIO1): Transmitted Data 0 (RS232/RS485 Channel)

USER (GPIO16):

WAKE (GPIO16): Deep Sleep Mode

FLASH (GPIO0):

HSCLK (GPIO14):

HMISO (GPIO12):

HMOSI (GPIO13):

HCS (GPIO15):

SDCMD: Command (SD Standard)

SDCLK: Clock (SD Standard)

SDD0: Data 0 (SD Standard)

SDD1: Data 1 (SD Standard)

SDD2: Data 2 (SD Standard)

SDD3: Data 3 (SD Standard)

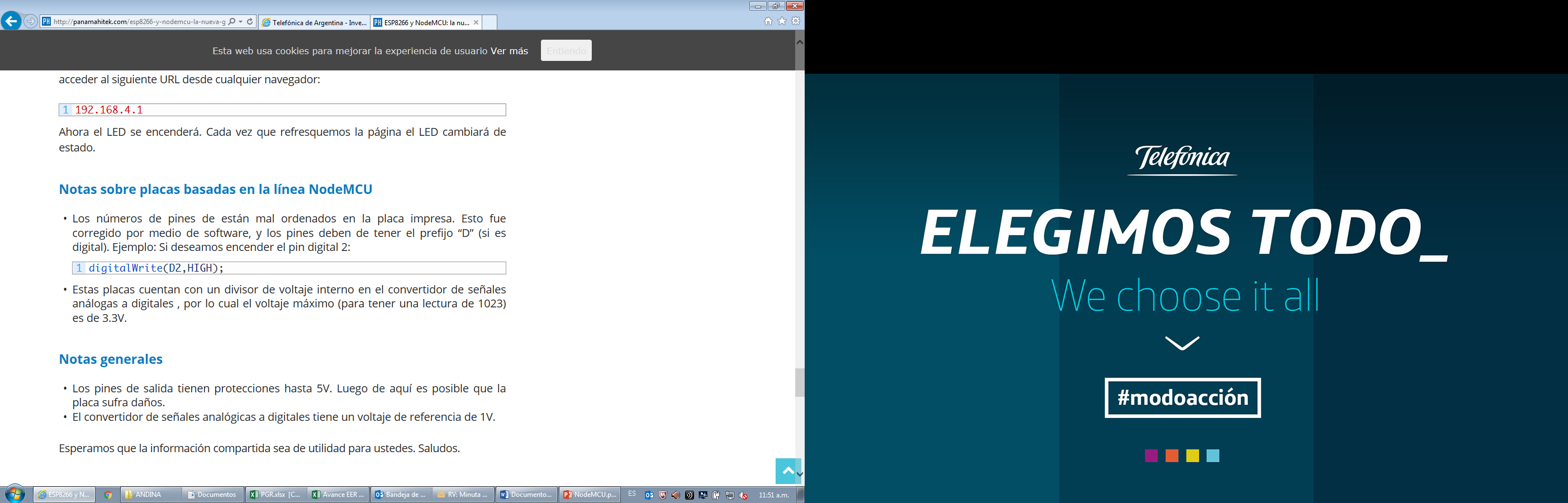
**Lenguajes de programación**

El NodeMCU fue diseñado principalmente para trabajar con el lenguaje Lua (lenguaje estructurado y ligero basado en C y Perl), aunque es posible programarlo con Micropython o con Arduino IDE.

El lenguaje Arduino es una plataforma con la que es posible compartir código entre diferentes placas y posee una gran cantidad de librerías existentes lo cual hace más versátil su utilización. Las otras plataformas (Micro Python y Lúa) tienen sus ventajas y desventajas. Una de las grandes deficiencias que se pueden mencionar es que no son estables (especialmente Lúa) y que son lenguajes interpretados (esto quiere decir que procesan el código un poco más lento que las contrapartes en Arduino).

**Notas sobre placas basadas en la línea NodeMCU**

* Dato a confirmar: Los números de pines de están mal ordenados en la placa impresa. Esto fue corregido por medio de software, y los pines deben de tener el prefijo “D” (si es digital). Ejemplo: Si deseamos encender el pin digital 2:



* Estas placas cuentan con un divisor de voltaje interno en el convertidor de señales analógicas a digitales, por lo cual el voltaje máximo (para tener una lectura de 1023) es de 3.3V.

**Notas generales**

* Los pines de salida tienen protecciones hasta 5V. Luego de aquí es posible que la placa sufra daños.
* El convertidor de señales analógicas a digitales tiene un voltaje de referencia de 1V.