**Memoria de proyecto**

1. Se actualiza el firmware de la placa de desarrollo NodeMCU y se instalan los drivers necesarios en la PC para establecer su comunicación. se instalan las librerías en el IDE de Arduino para poder programar módulos ES8266. (Fuente: [Everything ESP8266](https://www.esp8266.com/))
2. Código v0.1: Se inicia un servidor wifi desde el NodeMCU. Mediante monitor serie el micro informa su estado, se da un tiempo de espera para la conexión. Si logra conectarse a la red devuelve su dirección IP. Caso contrario devuelve error. Se utiliza un código html básico para encender y apagar un led mediante dirección IP.
3. Código v0.2: Se agrega al código la lectura del sensor de temperatura y humedad DHT11. Se sigue trabajando en el archivo html del servidor, para lograr visualizar los valores leídos del sensor en el mismo.
4. Código v0.3: Se agrega la función de activar/desactivar cerradura, utilizando un servomotor. Debe agregarse alimentación externa para evitar problemas de funcionamiento del motor.
5. Código v0.4: Se agrega el sensor de gas. Al activarse el título del servidor cambia, indicando la alerta. El mismo continúa igual hasta que se de click en un nuevo botón que aparece entonces en el servidor: Desactivar alarma. Se trabaja con interrupciones, después de varios intentos se logra implementar. Las funciones de la librería Servo.h parecen verse afectadas por el uso de interrupciones, por lo que se recurre a utilizar PWM mediante analogRead. En ángulo se ve ligeramente afectado. Se introducen ligeros cambios en el código en general para mayor eficiencia.
6. Código v0.5. Se agrega el sensor PIR. Se hace una función de control de servo propia, intentando solucionar el problema generado tras la implementación de las interrupciones. Sigue sin funcionar, al igual que las funciones de la librería Servo.h. Finalmente se recurre a implementar parte del código de la función creada dentro de la acción tras recibir la orden del servidor, quedando el servo en funcionamiento. **Se adiciona además un transistor para como etapa intermedia, para excitar el control del servo con 5v en lugar de los 3,3v del micro.** Se agregan actuadores automáticos para las alarmas: extractor para la de gas y sirena para la de movimiento. Se agrega un nuevo botón al servidor: Activar alarma. Se muestra el estado de la alarma debajo de la temperatura y la humedad.
7. Código v0.6. Se agrega el sensor LDR. Se agrega al servidor la opción de automatizar las luces exteriores, y que enciendan o apaguen acorde a lo que el sensor detecte, o que sean controladas de modo manual. A esta altura se empieza a notar la limitación del microcontrolador, ya que no quedan prácticamente pines puramente GPIO para usar, sino que son pines compartidos con diversas funciones del micro, como comunicaciones serie o pines que usa en el arranque. Se hacen intercambios de pines, la salida del buzzer pasa a ser la entrada del sensor PIR y viceversa. Se utiliza un pin reservado, con la condición de que esté en modo pull-up, por lo que se utiliza con lógica negativa, es decir activando la carga al estar en 0 y desactivándola al estar en 1, drenando corriente el puerto del micro. Por estas limitaciones el proyecto llega hasta este punto en cuanto a adiciones de periféricos, quedando sólo dos controles de luces: una habitación (hasta el momento “Activar o desactivar LED”) y el control de las luces exteriores.
8. Código v0.7. Se implementa el “Modo Vacaciones”. Para esto se agrega un contador de segundos al código, que siempre se está ejecutando en segundo plano. Al activar el modo, el contador se pone 0 y cada cierto tiempo predefinido, las luces interiores se encienden y apagan automáticamente. El resto de las funcionalidades pueden seguir utilizándose normalmente. Se agregan colores al texto del servidor. Se cambia el título de “ESP 8266 Server” a “Control del hogar”.
9. Código v0.8: Se reemplazan los leds indicadores por módulos de relé, teniendo que cambiar la lógica ya que funcionan en bajo. Se agregan líneas de código para fijar la dirección IP, ya que al usar tantos pines extra la comunicación serie se ve afectada, y muchas veces el monitor no muestra la IP a la que se conectó el módulo. Se agrega un destello en el led del NodeMCU para indicar el éxito al conectarse a la red.