**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

(*Universidad del Perú*, Decana de América)



**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**Curso: Internet de las Cosas**

**Tema: Aplicación de semáforos inteligentes haciendo uso de simulación en CISCO packet tracer 7.3.1**

**Docente:** **Herrera Quispe, Jose Alfredo**

**Grupo: hardworKINGS**

**Integrantes:**

* **Palomino Loa, Junior Jose 18200172**
* **Sánchez Saldaña, Néstor Joaquín 18200191**
* **Sánchez Muñoz, José Anderson 18200236**

LIMA – PERÚ

**2021**

Índice

[**Homework** 3](#_Toc73443729)

[**Diseño (Idea previa)** 4](#_Toc73443730)

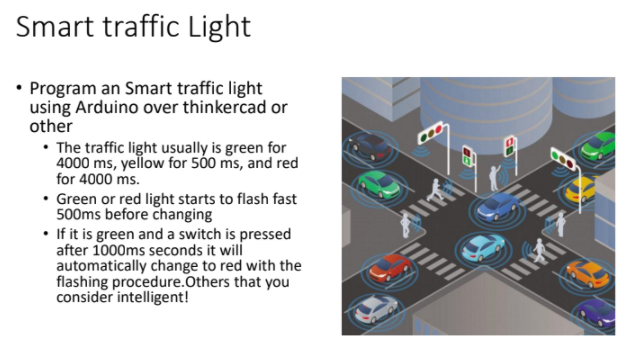
[**Materiales usados** 5](#_Toc73443731)

[**Cambios realizados** 5](#_Toc73443732)

[**Análisis del código** 9](#_Toc73443733)

[**Bibliografía** 15](#_Toc73443734)

# **Homework**



# 

# **Diseño (Idea previa)**

Diagrama

Descripción generada automáticamente

1- El switch envía una señal al **ARDUINO MAESTRO** (si se enciende o se apaga) y esta será repetida al **ARDUINO ESCLAVO**, ejecutando la misma orden.

2- El **botón maestro** envía una señal al **ARDUINO MAESTRO** para cambiar de luz (roja a verde), permitiendo así el paso del peatón. Esta señal se replica al **ARDUINO ESCLAVO**, haciendo que este cambie también (verde a roja), evitando así que se pueda producir un accidente.

3- El **botón esclavo** envía una señal al **ARDUINO ESCLAVO** para cambiar de luz (roja a verde), permitiendo así el paso del peatón. Esta señal se replica al **ARDUINO MAESTRO**, haciendo que este cambie también (verde a roja), evitando así que se pueda producir un accidente.

4- Tanto el **ARDUINO MAESTRO** como el **ARDUINO ESCLAVO** dirigen el comportamiento de un **semáforo peatona**l y uno **automovilístico**, permitiendo que estos enciendan sus luces correspondientes en el tiempo indicado.

# **Materiales usados**

|  |  |
| --- | --- |
| Qué significan los colores de los cables eléctricos? » MN Del Golfo  **Cables (x16)** | LAS ENTRADAS DIGITALES DE ARDUINO | Tienda y Tutoriales Arduino  **Botones (x2)** |
| 2-Mode Rocker Button Switch Module for Arduino - Free shipping - DealExtreme  **Switch (x1)** | Uno R3  **Arduino (x2)** |

# **Cambios realizados**

**LED’S**

Por defecto los LED’S en en packet tracer tienen la forma de uno convencional, como se conseguiría en las tiendas.

Sin embargo, packet tracer nos ofrece la posibilidad de cambiar el formato de este, por lo que lo aprovecharemos para darles una forma más parecida a lo que deseamos (luces de semáforo), para ello usaremos plantillas (estas se encuentran en el drive).

![Imagen que contiene Logotipo

Descripción generada automáticamente](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAEgAAABBCAIAAAD9gHXYAAAA0ElEQVR42u3b2Q3DMAwEUQFJGXRa3dZTQQALIcXDMw3Y78+wVuvzeq/I9LvQ564ImPYrDZNH5WByzeGFrr9hiikTpvgSYDrVUZjOdgimjLZhtglTXoEwZRcCU42A3YOpUm4w1QtYK9Ut20Nhqh2wh8DUIWDAusLUJ2DAgAEDBgwYsKowPoKBAQuH8TMHWB0Yv7gbwjhGaggbe6LJ4XpP2Nidx+RlzuQtVZv1m03dK9rUhalN3QSbx4q7nMoLVnF3b1NvStjUuy1XMCwtYMBq9AWe0t/quf3WzwAAAABJRU5ErkJggg==) ![Forma, Círculo

Descripción generada automáticamente](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAEgAAAA/CAIAAADMqBlpAAAAuUlEQVR42u3ayRGAMAxD0dRCPTRLh1QAh+BFFl+TBt4tI2uttBzX+fTWuLxgRiI3PAOEISotXixJwpZEauYVqBpsZapSW7GqyNaiSrc1qhJt7aosmydMRBVsk1JF2jxhgqoYmydMVvXVBmwUTFy1bwMGTAE2QrVjAwYMGDBgwIAB41fFJxgY7QBlDrCfdaZU3JyRuP1xreW4bruIYJnDluoX6zfnvaLzwtR5E5xnY3cvL1zjYoUh6bkB1AbugjXv1R8AAAAASUVORK5CYII=)

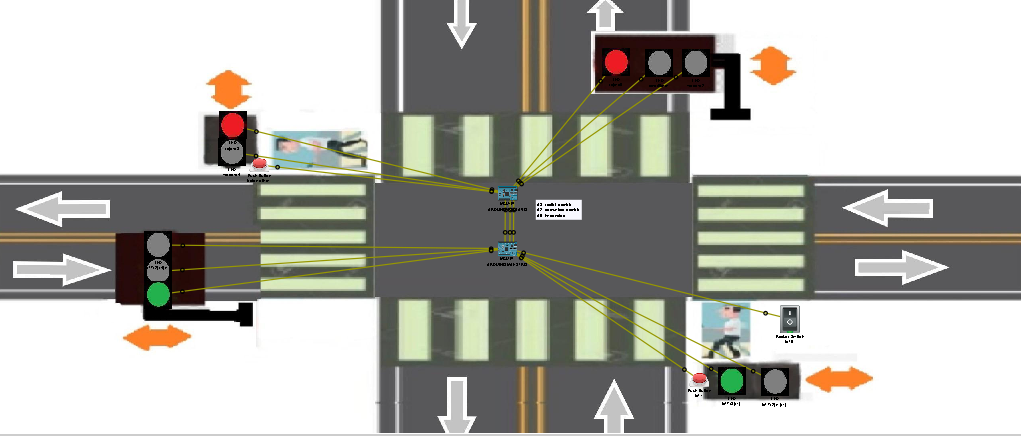
**Fondo**

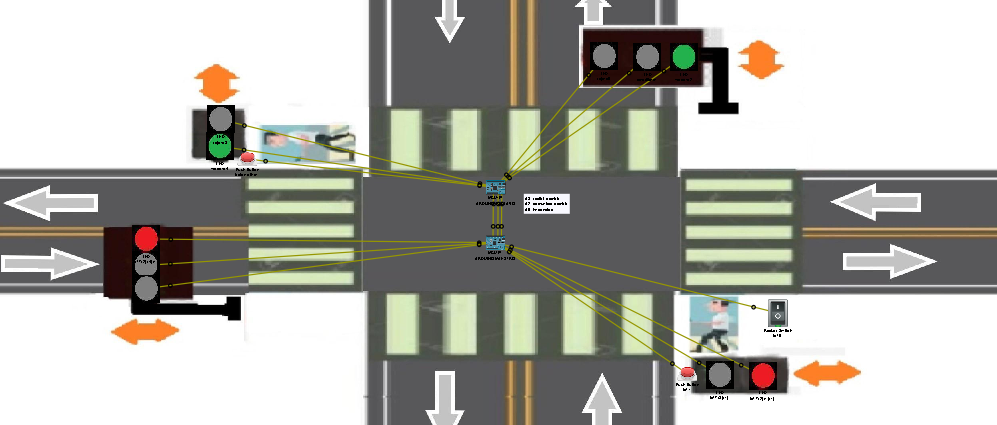
Packet tracer nos permite agregar un fondo que nos ayudará a una simulación más apreciable (este se encontrará en el drive).

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente

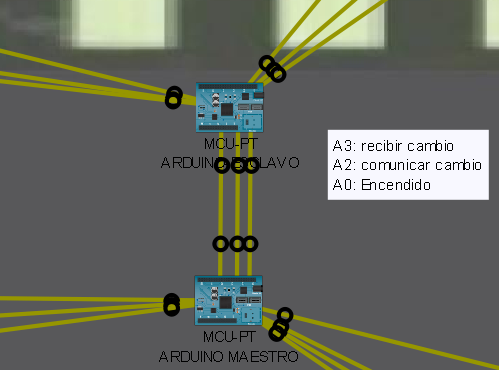
**Vista del proyecto en packet tracer 7.3.1**



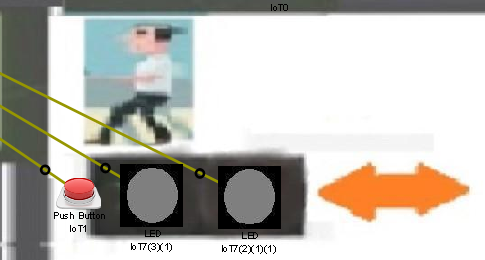


(\*) La simulación se encuentra en el video “Demo - Semaforos inteligentes (hardworKINGS).mp4” (se encontrará en el drive)

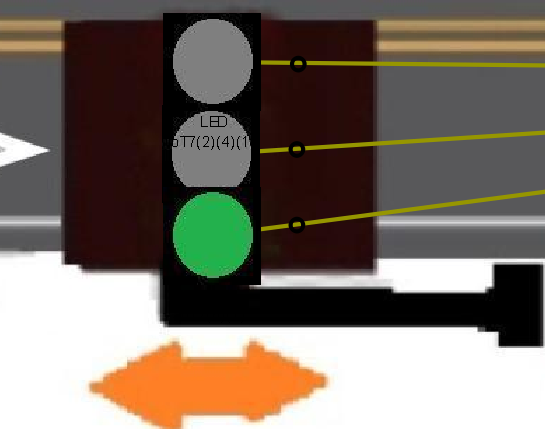
**Arduinos conectados**



**Vista del semáforo peatonal con su respectivo botón inteligente**



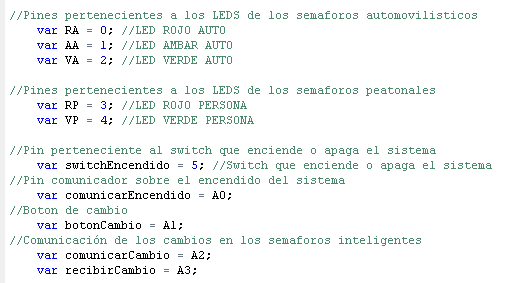
**Vista del semáforo automovilístico**



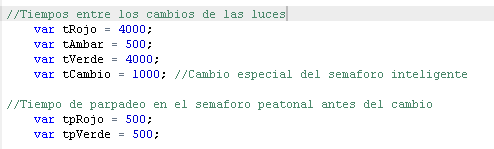
# **Análisis del código**

**Nota: Se tomará como referencia el código del ARDUINO MAESTRO para la explicación.**

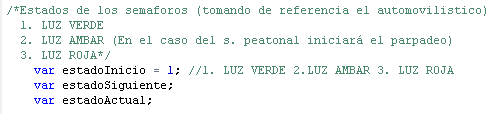
1- Declaramos los identificadores que correspondan a los pines para su fácil identificación.



2- Declaramos los identificadores que correspondan a los tiempos de los que hará uso cada luz sea el caso.



3- Declaramos los identificadores que correspondan a los estados de los semáforos (luces ejecutándose)

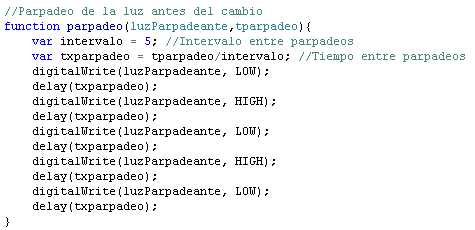


(\*) Nota: En este caso el ARDUINO MAESTRO inicia en 1 (luz verde) mientras que el ARDUINO ESCLAVO iniciará en 3 (luz roja).

4- Declaramos la función correspondiente a la acción de parpadeo en las luces en ejecución.

Parámetros:

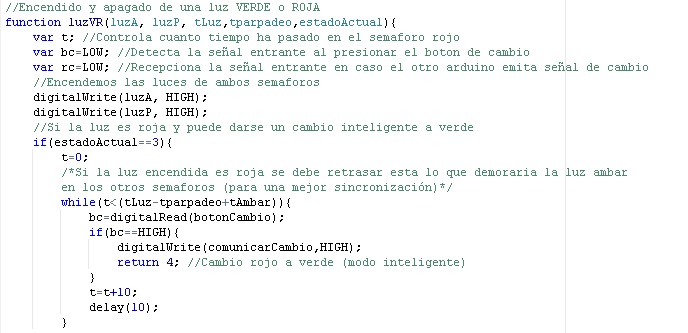
* luzParpadeante: luz que va a realizar el parpadeo
* tparpadeo: es el tiempo en milisegundos que durará el parpadeo

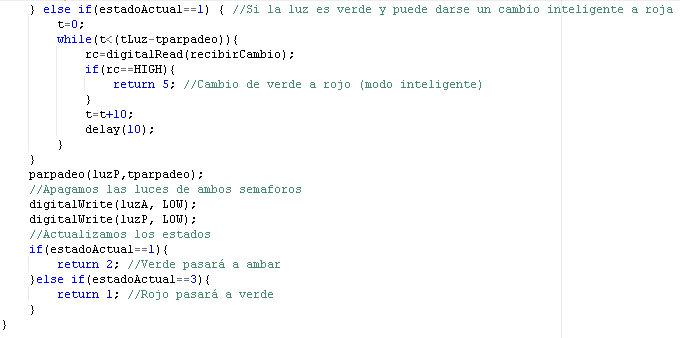


5-Declaramos la función correspondiente a la acción de encender y apagar las luces.

Parámetros:

* luzA: luz correspondiente al semáforo automovilístico
* luzP: luz correspondiente al semáforo peatonal
* tLuz: tiempo en que la luz estará encendida
* tparpadeo: tiempo en que la luz estará parpadeando
* estadoActual: estado que informa que luces se están ejecutando

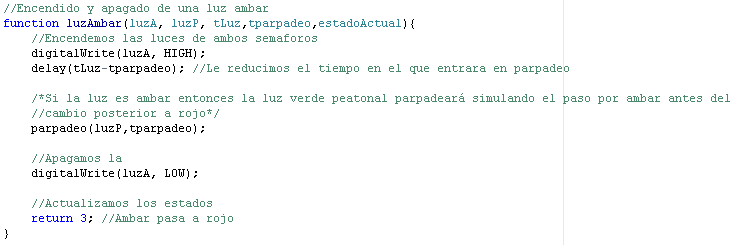




(\*) Nota: El curso normal de esta función cambia en caso se detecte que se presionó el botón que activa el modo inteligente (sea que ocurriera en el ARDUINO MAESTRO o en el ESCLAVO)

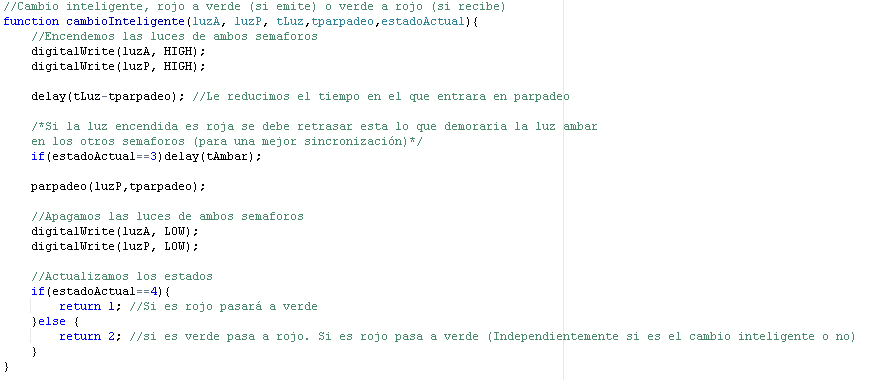
6- Esta función permite el prendido y el apagado de la luz ámbar de los semáforos automovilísticos, a su vez de proceder con la función de “parpadeo” de los semáforos peatonales. Toma como parámetros de referencia:

* luzA: luz correspondiente al semáforo automovilístico
* luzP: luz correspondiente al semáforo peatonal
* tLuz: tiempo en que la luz estará encendida
* tparpadeo: tiempo en que la luz estará parpadeando
* estadoActual: estado que informa que luces se están ejecutando



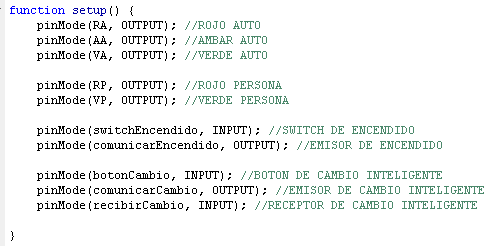
7- Esta función ejecuta el cambio de luz cuando se presiona los botones situados en los semáforos peatonales. El único caso donde se presenta este es cuando el semáforo peatonal se encuentra en rojo y se busca el cambio de este a verde para poder cruzar. Consecuentemente los semáforos automovilisticos se adaptan al cambio, pasando de su **estado de verde a rojo**, permitiendo un paso seguro para el transeúnte. Toma como parámetros de referencia:

* luzA: luz correspondiente al semáforo automovilístico
* luzP: luz correspondiente al semáforo peatonal
* tLuz: tiempo en que la luz estará encendida
* tparpadeo: tiempo en que la luz estará parpadeando
* estadoActual: estado que informa que luces se están ejecutando



(\*) Nota: tomar en cuenta que el tiempo que tarda en cambiar un semáforo automovilístico que se encuentra en rojo a verde debe de considerar el tiempo de la luz ámbar para evitar una desincronización futura con los semáforos peatonales. Además, se debe reflejar el parpadeo en las luces de estos.

8- Se define en el setup los modos de uso de cada pin, sean estos del tipo INPUT u OUTPUT.



9- En esta función se va a tener en cuenta el código que se va a ejecutar continuamente. Para el caso del arduino Maestro el if-else va a depender del estado de switchEncendido ya que dependiendo de esto, el sistema se enciende o se apaga.Y en el caso del arduino Esclavo el if-else va a depender de la variable recibir Encendido y no tendrá la opción de comunicar al arduino Maestro que es lo que debe hacer.

Dentro del if , hay un switch que dependiendo del estadoActual representado principalmente por los casos 1 , 2 y 3 que son los colores de las luces.

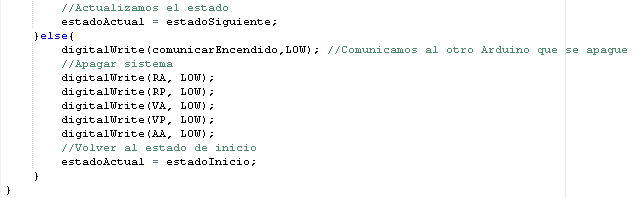
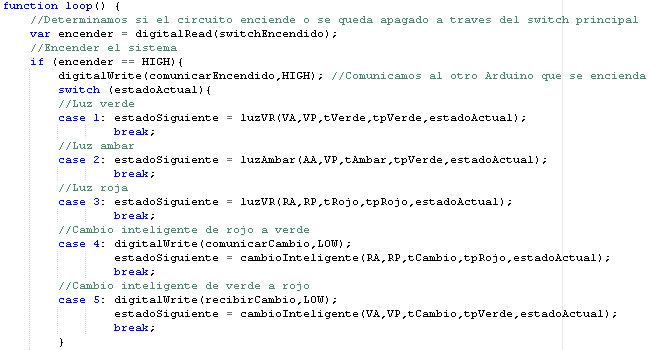
Hay que tener en cuenta que los casos de luz verde y roja hay una función que les permite ingresar los parámetros del tipo de luz del semáforo de autos y personas, el tiempo de cada uno de estos semáforos y el estado actual para que calculen el estado siguiente al que serían asignados.

Los casos 4 y 5 son casos especiales ya que dependen de los botones ubicados en los semáforos de peatones.

El caso 4 es el caso del botón del arduino Maestro, cuando el semáforo del peatón esté en rojo, es necesario realizar un cambio inteligente, para que el semáforo pase a verde.

El caso 5 es el caso del botón del arduino Esclavo, cuando el semáforo del otro extremo está en rojo, es necesario realizar un cambio inteligente, para que el semáforo pase a verde.

Dentro del else, significa que el switch estará apagado por lo que se apagan los semáforos de ambos tipos, y el estadoActual toma el valor inicial de estadoInicio.



(\*) Nota: Las líneas de código relacionadas a la variable **“comunicarEncendido”** sólo están presentes en el ARDUINO MAESTRO, siendo que el ARDUINO ESCLAVO tendrá el **“recibirEncendido”** que le comunicará cuando este se encienda o apague para replicar su comportamiento.

# **Bibliografía**

* Universidad Católica de Colombia. (2019). Arquitectura IoT para la Prestación del Servicio de Semaforización Inteligente en Bogotá. <https://repository.ucatolica.edu.co/bitstream/10983/23709/1/Semaforizacion%20Inteligente%20_625387.pdf>