



## Secuencias de trabajo para Taller de Ciclo Básico

### Secuencia de Trabajo Nº 3 (Para estudiantes de primer ciclo de ETP)

#### “Programación definiendo valores constantes...”

Esta secuencia de trabajo estará organizada con las siguientes actividades:

1. **Explicativa:** Realización del circuito y la programación de leds utilizando “#define”.
2. **Problemática circuital y de programación.** (Método de Pólya)
3. **Problemática de programación.** (Método de Pólya)
4. **Situación Problema Tecnológica.** “Problemática Abierta” (Metodología Proyectual). \*
5. **Situación Problema Tecnológica.** “Problemática Abierta” (Metodología Proyectual). \*

\*En estas dos últimas actividades se pretende que las y los estudiantes de taller del primer ciclo (Ciclo Básico), resuelvan una problemática tecnológica articulando lo distintos saberes desarrollados durante su formación, a través de la metodología proyectual y que este proceso de como resultado la fabricación de un prototipo maquetoizado y automatizado.

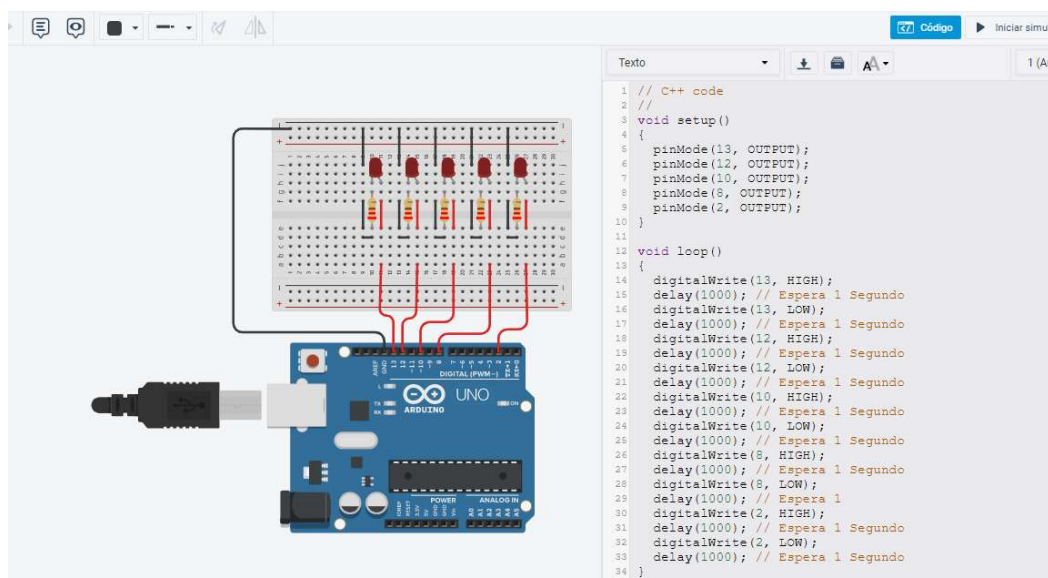
### 1 Actividad Explicativa

#### Objetivos

- Utilizar la variable #define
- Encender alternativamente 5 led
- Controlar el tiempo encendido de los leds

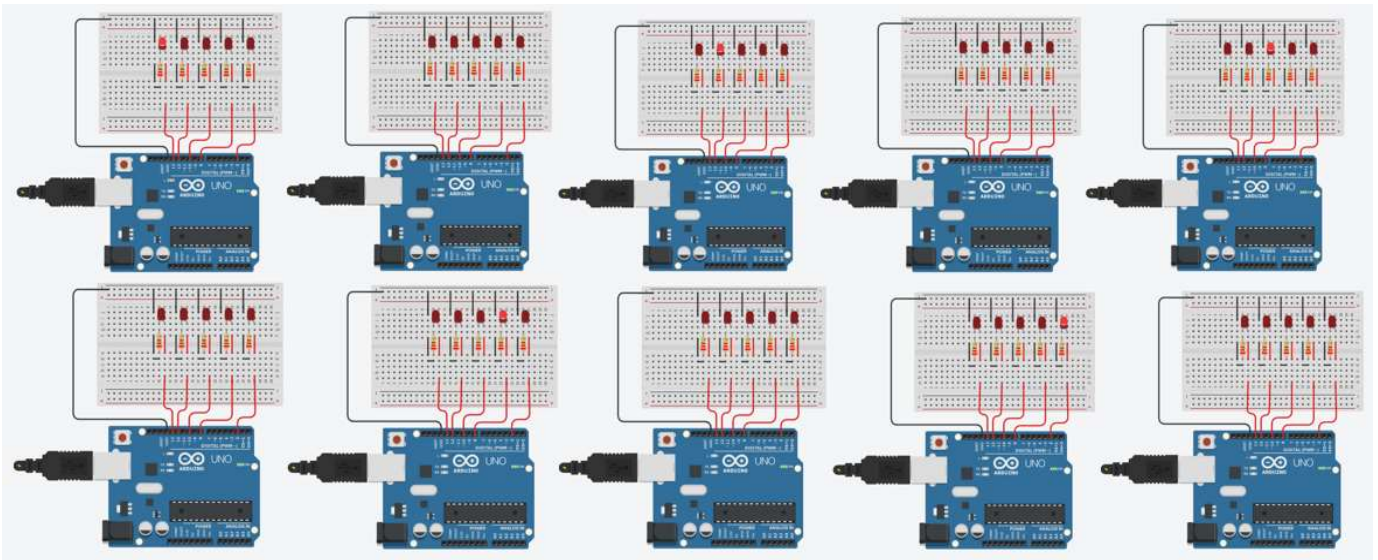
#### Situación a resolver:

Encender 5 led rojos de forma alternada, utilizando una salida para cada led. El lapso de tiempo que permanecerá encendido cada uno será de un segundo y la secuencia se repetirá de forma permanente.



## Analizamos la programación:

```
1 // C++ code
2 //
3 void setup()//Escribo los parametros del programa
4 {
5   pinMode(13, OUTPUT);//El pin 13 va a ser una salida
6   pinMode(12, OUTPUT);//El pin 12 va a ser una salida
7   pinMode(10, OUTPUT);//El pin 10 va a ser una salida
8   pinMode(8, OUTPUT);//El pin 8 va a ser una salida
9   pinMode(2, OUTPUT);//El pin 2 va a ser una salida
10 }
11
12 void loop()
13 {
14   digitalWrite(13, HIGH);// Escritura digital alta
15   delay(1000); // Espera 1 Segundo
16   digitalWrite(13, LOW);// Escritura digital baja
17   delay(1000); // Espera 1 Segundo
18   digitalWrite(12, HIGH);// Escritura digital alta
19   delay(1000); // Espera 1 Segundo
20   digitalWrite(12, LOW);// Escritura digital baja
21   delay(1000); // Espera 1 Segundo
22   digitalWrite(10, HIGH);// Escritura digital alta
23   delay(1000); // Espera 1 Segundo
24   digitalWrite(10, LOW);// Escritura digital baja
25   delay(1000); // Espera 1 Segundo
26   digitalWrite(8, HIGH);// Escritura digital alta
27   delay(1000); // Espera 1 Segundo
28   digitalWrite(8, LOW);// Escritura digital baja
29   delay(1000); // Espera 1 Segundo
30   digitalWrite(2, HIGH);// Escritura digital alta
31   delay(1000); // Espera 1 Segundo
32   digitalWrite(2, LOW);// Escritura digital baja
33   delay(1000); // Espera 1 Segundo
34 }
```



### Donde:

“void setup ()” será donde se escriban los parámetros del programa:

“pinMode (13, OUTPUT);” El pin 13 es una salida

“pinMode (12, OUTPUT);” El pin 12 es una salida

“pinMode (10, OUTPUT);” El pin 10 es una salida

“pinMode (8, OUTPUT);” El pin 8 es una salida

“pinMode (2, OUTPUT);” El pin 2 es una salida

**void loop()** Es donde escribo el programa o bucle del programa

**“digitalWrite(13, HIGH);”** Significa que la escritura digital tendrá estado alto

**“delay(1000);”** Significa que en ese estado permanecerá 1 segundo

**“digitalWrite(13, LOW);”** Significa que la escritura digital tendrá estado bajo

**“delay(1000);”** Significa que en ese estado permanecerá 1 segundo

**“digitalWrite(12, HIGH);”** Significa que la escritura digital tendrá estado alto

**“delay(1000);”** Significa que en ese estado permanecerá 1 segundo

**“digitalWrite(12, LOW);”** Significa que la escritura digital tendrá estado bajo

**“delay(1000);”** Significa que en ese estado permanecerá 1 segundo

**“digitalWrite(10, HIGH);”** Significa que la escritura digital tendrá estado alto

**“delay(1000);”** Significa que en ese estado permanecerá 1 segundo

**“digitalWrite(10, LOW);”** Significa que la escritura digital tendrá estado bajo

**“delay(1000);”** Significa que en ese estado permanecerá 1 segundo

**“digitalWrite(8, HIGH);”** Significa que la escritura digital tendrá estado alto

**“delay(1000);”** Significa que en ese estado permanecerá 1 segundo

**“digitalWrite(8, LOW);”** Significa que la escritura digital tendrá estado bajo

**“delay(1000);”** Significa que en ese estado permanecerá 1 segundo

**“digitalWrite(2, HIGH);”** Significa que la escritura digital tendrá estado alto

**“delay(1000);”** Significa que en ese estado permanecerá 1 segundo

**“digitalWrite(2, LOW);”** Significa que la escritura digital tendrá estado bajo

**“delay(1000);”** Significa que en ese estado permanecerá 1 segundo

Como observamos la programación del sistema funciona correctamente. Pero para su mejor comprensión o posible modificación utilizaremos la constante **“#define”**

```
1 #define Rojo1 13
2 #define Rojo2 12
3 #define Rojo3 10
4 #define Rojo4 8
5 #define Rojo5 2
6
7 void setup()//Escribo los parametros del programa
8 {
9   pinMode(13, OUTPUT);//El pin 13 va a ser una salida
10  pinMode(12, OUTPUT);//El pin 12 va a ser una salida
11  pinMode(10, OUTPUT);//El pin 10 va a ser una salida
12  pinMode(8, OUTPUT);//El pin 8 va a ser una salida
13  pinMode(2, OUTPUT);//El pin 2 va a ser una salida
14 }
15 void loop()
16 {
17   digitalWrite(Rojo1, HIGH);// Escritura digital alta
18   delay(1000); // Espera 1 Segundo
19   digitalWrite(Rojo1, LOW);// Escritura digital baja
20   delay(1000); // Espera 1 Segundo
21   digitalWrite(Rojo2, HIGH);// Escritura digital alta
22   delay(1000); // Espera 1 Segundo
23   digitalWrite(Rojo2, LOW);// Escritura digital baja
24   delay(1000); // Espera 1 Segundo
25   digitalWrite(Rojo3, HIGH);// Escritura digital alta
26   delay(1000); // Espera 1 Segundo
27   digitalWrite(Rojo3, LOW);// Escritura digital baja
28   delay(1000); // Espera 1 Segundo
29   digitalWrite(Rojo4, HIGH);// Escritura digital alta
30   delay(1000); // Espera 1 Segundo
31   digitalWrite(Rojo4, LOW);// Escritura digital baja
32   delay(1000); // Espera 1 Segundo
33   digitalWrite(Rojo5, HIGH);// Escritura digital alta
34   delay(1000); // Espera 1 Segundo
35   digitalWrite(Rojo5, LOW);// Escritura digital baja
36   delay(1000); // Espera 1 Segundo
37 }
```

El **#define** es un componente útil de C que permite que el programador dé un nombre a un valor constante antes de que se compile el programa. Las constantes definidas en Arduino no usan memoria del programa en el microprocesador. El compilador substituirá referencias a estas constantes por el valor definido durante la compilación.

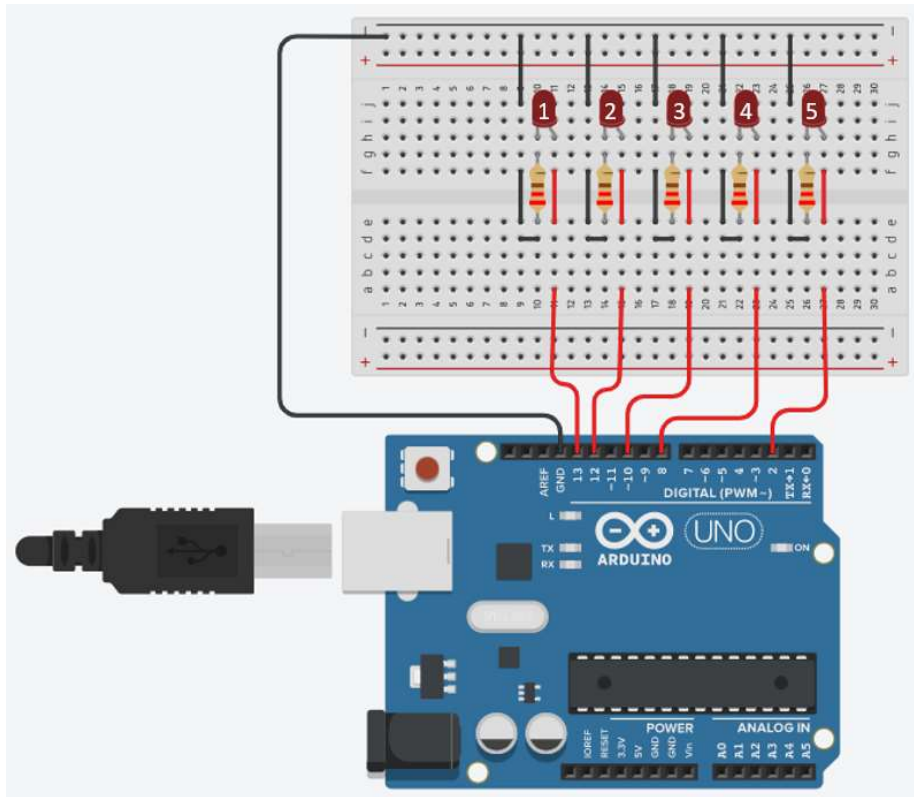
**“#define Rojo1 13”** Significa que el pin 13 corresponderá al led Rojo1

**“#define Rojo2 12”** Significa que el pin 12 corresponderá al led Rojo2

**“#define Rojo3 10”** Significa que el pin 10 corresponderá al led Rojo3

**“#define Rojo4 8”** Significa que el pin 8 corresponderá al led Rojo4

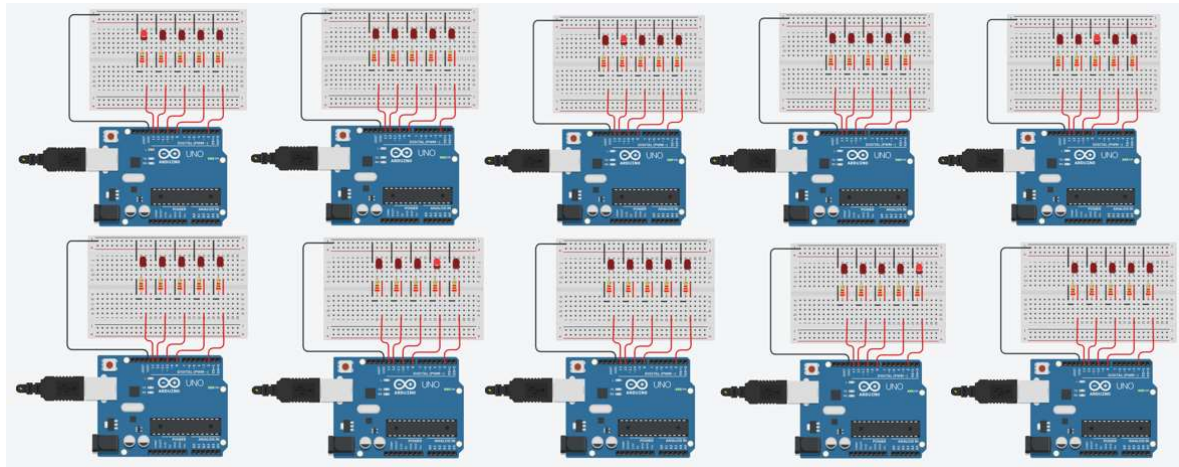
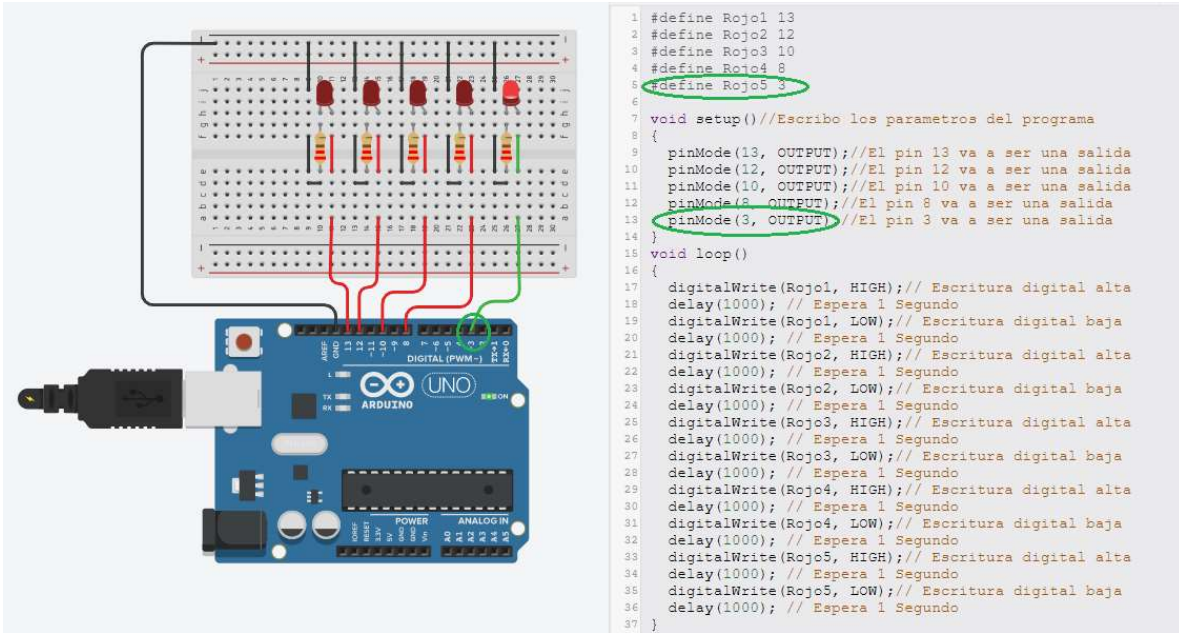
**“#define Rojo5 2”** Significa que el pin 13 corresponderá al led Rojo5



A partir de esas definiciones realizamos la programación colocando en el void loop las constantes definidas anteriormente.

```
1 #define Rojo1 13
2 #define Rojo2 12
3 #define Rojo3 10
4 #define Rojo4 8
5 #define Rojo5 2
6
7 void setup()//Escribo los parametros del programa
8 {
9   pinMode(13, OUTPUT);//El pin 13 va a ser una salida
10  pinMode(12, OUTPUT);//El pin 12 va a ser una salida
11  pinMode(10, OUTPUT);//El pin 10 va a ser una salida
12  pinMode(8, OUTPUT);//El pin 8 va a ser una salida
13  pinMode(2, OUTPUT);//El pin 2 va a ser una salida
14 }
15 void loop()
16 {
17   digitalWrite(Rojo1, HIGH);// Escritura digital alta
18   delay(1000); // Espera 1 Segundo
19   digitalWrite(Rojo1, LOW);// Escritura digital baja
20   delay(1000); // Espera 1 Segundo
21   digitalWrite(Rojo2, HIGH);// Escritura digital alta
22   delay(1000); // Espera 1 Segundo
23   digitalWrite(Rojo2, LOW);// Escritura digital baja
24   delay(1000); // Espera 1 Segundo
25   digitalWrite(Rojo3, HIGH);// Escritura digital alta
26   delay(1000); // Espera 1 Segundo
27   digitalWrite(Rojo3, LOW);// Escritura digital baja
28   delay(1000); // Espera 1 Segundo
29   digitalWrite(Rojo4, HIGH);// Escritura digital alta
30   delay(1000); // Espera 1 Segundo
31   digitalWrite(Rojo4, LOW);// Escritura digital baja
32   delay(1000); // Espera 1 Segundo
33   digitalWrite(Rojo5, HIGH);// Escritura digital alta
34   delay(1000); // Espera 1 Segundo
35   digitalWrite(Rojo5, LOW);// Escritura digital baja
36   delay(1000); // Espera 1 Segundo
37 }
```

Una de las ventajas es que la constante # no ocupa memoria y que ante cualquier cambio en el hardware no tendremos que modificar el programa. Por ejemplo si modificamos el “pin 2” utilizando en su lugar el “pin 3”, solo modificaríamos “**#define Rojo5 2**” y en su lugar pondremos “**#define Rojo5 3**”, para luego modificar el parámetro “**pinMode (3; OUTPUT);**” de este modo no hará falta modificar el programa.



**\*NOTA:**

No hay “punto y coma (;)” después de la instrucción “#define”. Si se incluye punto y coma, el compilador genera errores.

No se debe incluir el “signo =” después de la instrucción “#define”, también generara un error.

```

1 #define Rojo1 13;
2 #define Rojo2 12;
3
4 void setup()//Escribo los parametros del programa
5 {
6   pinMode(13, OUTPUT);//El pin 13 va a ser una salida
7   pinMode(12, OUTPUT);//El pin 12 va a ser una salida
8 }
9 void loop()
10 {
11   digitalWrite(Rojo1, HIGH);// Escritura digital alta
12   delay(1000); // Espera 1 Segundo
13   digitalWrite(Rojo1, LOW);// Escritura digital baja
14   delay(1000); // Espera 1 Segundo
15   digitalWrite(Rojo2, HIGH);// Escritura digital alta
16   delay(1000); // Espera 1 Segundo
17   digitalWrite(Rojo2, LOW);// Escritura digital baja
18   delay(1000); // Espera 1 Segundo
19 }
    
```

Sorry, it seems like your code has some errors.

```

In function 'void loop()':
1:17: error: expected ')' before ';' token
12:16: note: in expansion of macro 'Rojo1'
12:21: error: expected primary-expression before ';' token
1:17: error: expected ')' before ';' token
14:16: note: in expansion of macro 'Rojo1'
14:21: error: expected primary-expression before ';' token
2:17: error: expected ')' before ';' token
16:16: note: in expansion of macro 'Rojo2'
16:21: error: expected primary-expression before ';' token
2:17: error: expected ')' before ';' token
18:16: note: in expansion of macro 'Rojo2'
18:21: error: expected primary-expression before ';' token

exit status 1
    
```

```

1 #define = Rojo1 13
2 #define = Rojo2 12
3
4 void setup()//Escribo los parametros del programa
5 {
6   pinMode(13, OUTPUT);//El pin 13 va a ser una salida
7   pinMode(12, OUTPUT);//El pin 12 va a ser una salida
8 }
9 void loop()
10 {
11   digitalWrite(Rojo1, HIGH);// Escritura digital alta
12   delay(1000); // Espera 1 Segundo
13   digitalWrite(Rojo1, LOW);// Escritura digital baja
14   delay(1000); // Espera 1 Segundo
15   digitalWrite(Rojo2, HIGH);// Escritura digital alta
16   delay(1000); // Espera 1 Segundo
17   digitalWrite(Rojo2, LOW);// Escritura digital baja
18   delay(1000); // Espera 1 Segundo
19 }
    
```

Sorry, it seems like your code has some errors.

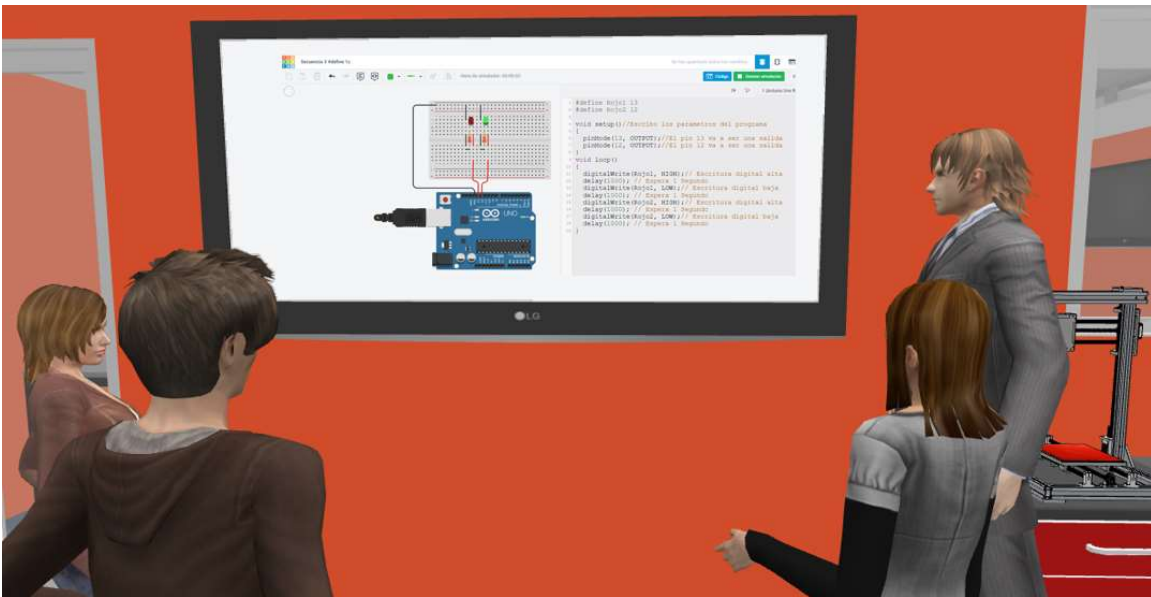
```

1:9: error: macro names must be identifiers
2:9: error: macro names must be identifiers

exit status 1
    
```

## Situación Problema

A continuación se plantearán una serie de situaciones problema para poder trabajar con los estudiantes. A cada una de ellas corresponderá una solución alternativa. Para su solución te invitamos a utilizar el método de Pólya en cual consiste en las siguientes etapas:



**Etapas del método Pólya de Resolución de Problemas**

**1) Comprender el problema.** Reconocer que se pregunta, identificar lo que hay que resolver y las condiciones asociadas.

**2) Elaborar un plan.** Se trata de establecer la vinculación entre los datos presentes y el problema a resolver, determinar los recursos que se utilizarán, verificar la similitud con otros problemas previamente resueltos y también la posibilidad de utilizar teorías o modelos útiles, todo esto en función de buscar una manera de resolver el problema.

**3) Ejecutar el plan.** Desarrollar el resultado de la respuesta, a partir de ejecutar el plan, avanzando y verificando cada paso.

**4) Revisar y verificar.** Controlar que hace y que dice el resultado, con vistas a considerar la posibilidad de transferir la solución a otros problemas.

## Actividades

### 2. Problemática circuital y de programación. (Método de Pólya).

Utilizar la constante define para la programación de un circuito de juego de luces de 3 leds (Rojo, Azul y verde), los cuales deberán encender del siguiente modo:

Bucle	Estado	1º Secuencia:	2º Secuencia:	3º Secuencia
	Encendidos	Rojo y Azul	Verde y Rojo	Verde y Azul
	Apagado	Verde	Azul	Rojo

Este bucle deberá repetirse de forma permanente y el lapso de tiempo que durara cada secuencia será de 3 segundos.

Te invitamos a proponer una solución a esta situación utilizando la metodología de Pólya para resolver problemas y realizar el circuito y la programación que dé respuesta a la problemática:

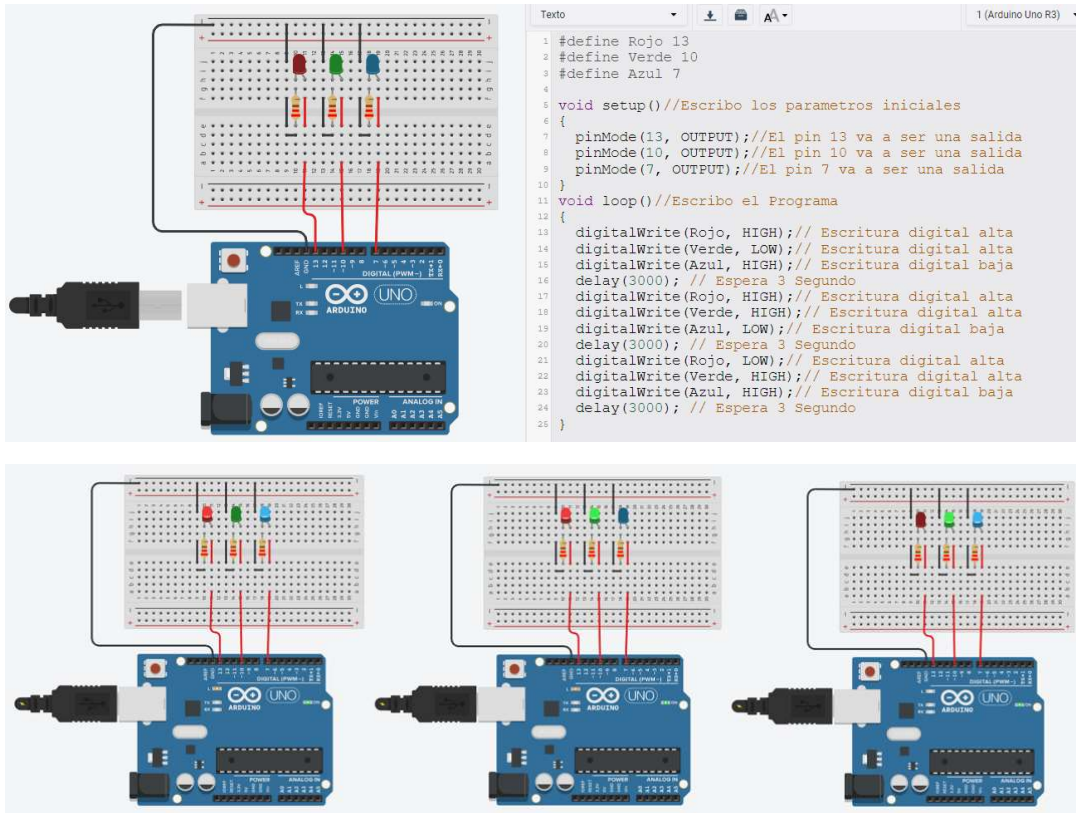


Etapas del Método Pólya		Acciones
1	<b>Comprender el Problema</b>  <i>Lo expresan con sus propias palabras</i>	<b>Analizo y comprendo el problema</b> Analizan y comprenden el problema utilizando procesos de descomposición, lógica, reconocimiento de patrones y abstracción. Debemos realizar la programación de un juego de luces leds rojo, verde y azul. Los cuales encenderán de pares en tres secuencias (Rojo y Azul); (Verde y Rojo); (Verde y Azul). Este bucle será de forma permanente.
2	<b>Elaborar un Plan</b>  <i>Diseñan un plan para resolver el problema</i>	<b>Diseño un plan para resolver el problema</b> Diseñan un circuito y la programación para resolver la problemática a través de un algoritmo que lo resolverá. Deben elegir los componentes diseñar el circuito, seleccionar los pines, realizar la conexión, y realizar la posible programación utilizando "#define".
3	<b>Ejecutar el Plan</b>  <i>Ejecutan el plan diseñado</i>	<b>Realizo el circuito y la programación del plan previamente diseñado</b> Utilizando TINKERCAD realizan el circuito, seleccionan el pin de salida 13 para el led rojo, el pin de salida 10 para el led verde y el pin de salida 7 para el led azul y la programación diseñada previamente.
4	<b>Revisar y Verificar la Solución</b>	<b>Reviso y verifico que la solución diseñada sea la adecuada</b> Ejecuto la simulación del sistema y verifico que funcione correctamente en relación a la problemática a resolver.



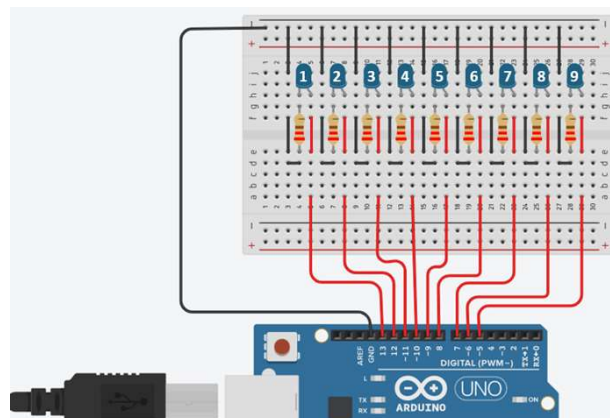
<p>Verifican el funcionamiento del sistema</p>	<p>Analizan que utilidad le pueden dar a este saber en otro contexto o situación física.</p>
--	--

**Alternativa posible:**



**3. Problemática de programación. (Método de Pólya).**

El siguiente circuito consta de 9 leds y debemos programar patrones de luces para que enciendan y apaguen de distinta forma durante lapsos de tiempo. Para el primer patrón necesitamos que enciendan todos los leds durante 4 segundos, después deberán permanecer encendidos solo los leds pares durante 4 segundos, pasado ese periodo de tiempo deben apagarse y encender solo los leds impares durante 4 segundos, luego tendrán que encenderse solo los leds múltiplos de 3 y apagarse el resto durante 4 segundos, para por ultimo permanecer todos los leds apagados durante 4 segundos y volver a repetir el bucle de forma permanente.

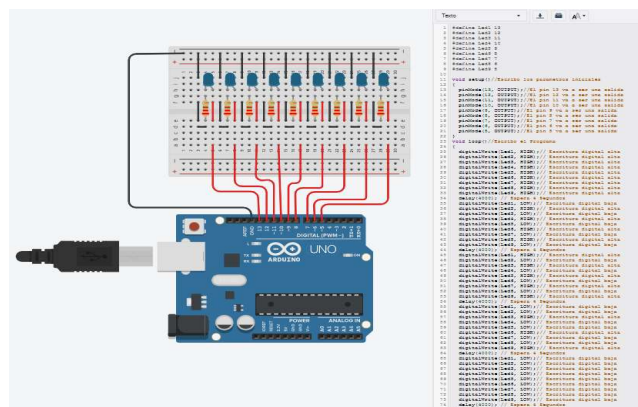


Te invitamos a proponer una solución a esta situación utilizando la metodología de Pólya para resolver problemas y realizar la programación que dé respuesta a la problemática:



Etapas del Método Pólya		Acciones
<b>1</b>	<b>Comprender el Problema</b>  <i>Lo expresan con sus propias palabras</i>	<b>Analizo y comprendo el problema</b>  Analizan y comprenden el problema utilizando procesos de descomposición, lógica, reconocimiento de patrones y abstracción. Deben realizar la programación de un juego de 9 luces leds a través de patrones definidos. El primero me indica que todos los led deben encender durante 4 segundos, el segundo que solo deben permanecer encendidos los leds pares, el tercero los leds impares, el cuarto los led múltiplos de 3 y por ultimo deben permanecer apagados. Cada patrón durara 4 segundos y el bucle de programación será de forma permanente.
<b>2</b>	<b>Elaborar un Plan</b>  <i>Diseñan un plan para resolver el problema</i>	<b>Diseño un plan para resolver el problema</b>  Diseñan la programación para resolver la problemática a través de un algoritmo. Deben observar la conexión del circuito e identificar a que pines de salida corresponderán cada led, y realizar la posible programación utilizando "#define".
<b>3</b>	<b>Ejecutar el Plan</b>  <i>Ejecutan el plan diseñado</i>	<b>Realizo el circuito y la programación del plan previamente diseñado</b>  Utilizando TINKERCAD realizo el circuito, y la programación diseñada previamente.
<b>4</b>	<b>Revisar y Verificar la Solución</b>  <i>Verifican el funcionamiento del sistema</i>	<b>Reviso y verifico que la solución diseñada sea la adecuada</b>  Ejecuto la simulación del sistema y verifico que funcione correctamente en relación a la problemática a resolver.  Analizan que utilidad le pueden dar a este saber en otro contexto o situación física.

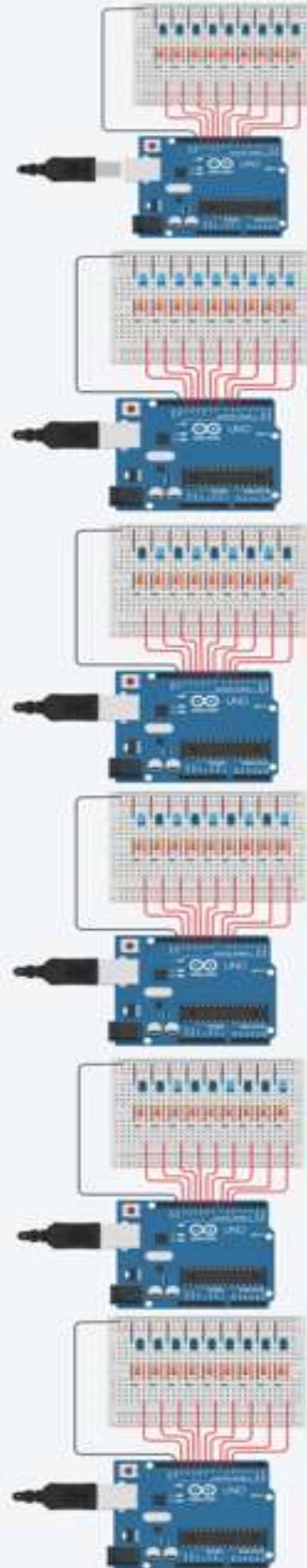
**Alternativa posible:**



```

1 #define Led1 13
2 #define Led2 12
3 #define Led3 11
4 #define Led4 10
5 #define Led5 9
6 #define Led6 8
7 #define Led7 7
8 #define Led8 6
9 #define Led9 5
10
11 void setup()//Escribo los parametros iniciales
12 {
13   pinMode(13, OUTPUT);//El pin 13 va a ser una salida
14   pinMode(12, OUTPUT);//El pin 12 va a ser una salida
15   pinMode(11, OUTPUT);//El pin 11 va a ser una salida
16   pinMode(10, OUTPUT);//El pin 10 va a ser una salida
17   pinMode(9, OUTPUT);//El pin 9 va a ser una salida
18   pinMode(8, OUTPUT);//El pin 8 va a ser una salida
19   pinMode(7, OUTPUT);//El pin 7 va a ser una salida
20   pinMode(6, OUTPUT);//El pin 6 va a ser una salida
21   pinMode(5, OUTPUT);//El pin 5 va a ser una salida
22 }
23 void loop()//Escribo el Programa
24 {
25   digitalWrite(Led1, HIGH);// Escritura digital alta
26   digitalWrite(Led2, HIGH);// Escritura digital alta
27   digitalWrite(Led3, HIGH);// Escritura digital alta
28   digitalWrite(Led4, HIGH);// Escritura digital alta
29   digitalWrite(Led5, HIGH);// Escritura digital alta
30   digitalWrite(Led6, HIGH);// Escritura digital alta
31   digitalWrite(Led7, HIGH);// Escritura digital alta
32   digitalWrite(Led8, HIGH);// Escritura digital alta
33   digitalWrite(Led9, HIGH);// Escritura digital alta
34   delay(4000); // Espera 4 Segundos
35   digitalWrite(Led1, LOW);// Escritura digital baja
36   digitalWrite(Led2, HIGH);// Escritura digital alta
37   digitalWrite(Led3, LOW);// Escritura digital baja
38   digitalWrite(Led4, HIGH);// Escritura digital alta
39   digitalWrite(Led5, LOW);// Escritura digital baja
40   digitalWrite(Led6, HIGH);// Escritura digital alta
41   digitalWrite(Led7, LOW);// Escritura digital baja
42   digitalWrite(Led8, HIGH);// Escritura digital alta
43   digitalWrite(Led9, LOW);// Escritura digital baja
44   delay(4000); // Espera 4 Segundos
45   digitalWrite(Led1, HIGH);// Escritura digital alta
46   digitalWrite(Led2, LOW);// Escritura digital baja
47   digitalWrite(Led3, HIGH);// Escritura digital alta
48   digitalWrite(Led4, LOW);// Escritura digital baja
49   digitalWrite(Led5, HIGH);// Escritura digital alta
50   digitalWrite(Led6, LOW);// Escritura digital baja
51   digitalWrite(Led7, HIGH);// Escritura digital alta
52   digitalWrite(Led8, LOW);// Escritura digital baja
53   digitalWrite(Led9, HIGH);// Escritura digital alta
54   delay(4000); // Espera 4 Segundos
55   digitalWrite(Led1, LOW);// Escritura digital baja
56   digitalWrite(Led2, LOW);// Escritura digital baja
57   digitalWrite(Led3, HIGH);// Escritura digital alta
58   digitalWrite(Led4, LOW);// Escritura digital baja
59   digitalWrite(Led5, LOW);// Escritura digital baja
60   digitalWrite(Led6, HIGH);// Escritura digital alta
61   digitalWrite(Led7, LOW);// Escritura digital baja
62   digitalWrite(Led8, LOW);// Escritura digital baja
63   digitalWrite(Led9, HIGH);// Escritura digital alta
64   delay(4000); // Espera 4 Segundos
65   digitalWrite(Led1, LOW);// Escritura digital baja
66   digitalWrite(Led2, LOW);// Escritura digital baja
67   digitalWrite(Led3, LOW);// Escritura digital baja
68   digitalWrite(Led4, LOW);// Escritura digital baja
69   digitalWrite(Led5, LOW);// Escritura digital baja
70   digitalWrite(Led6, LOW);// Escritura digital baja
71   digitalWrite(Led7, LOW);// Escritura digital baja
72   digitalWrite(Led8, LOW);// Escritura digital baja
73   digitalWrite(Led9, LOW);// Escritura digital baja
74   delay(4000); // Espera 4 Segundos
75 }

```



#### 4. Situación Problema Tecnológica. “Problemática Abierta” (Metodología Proyectual). \*

##### “Sistema de semaforización doble”

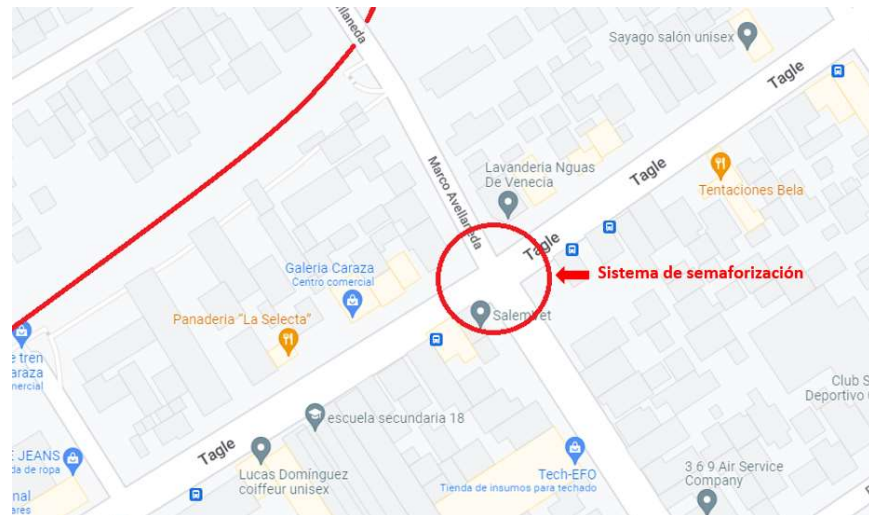
En el barrio de Caraza, partido de Lanús, el municipio creó un cruce vehicular en la intersección de la calle Marcos Avellaneda y la vía del Ferrocarril General Belgrano. Este ramal del ferrocarril cesó de operar en 2017, debido a un anegamiento de vías que provocó el descarrilamiento de una formación a la altura de la estación Puente Alsina.



Esta acción municipal benefició a los vecinos del barrio, ya que existía un solo paso vehicular para la comunicación de las dos zonas del barrio divididas por las vías del ferrocarril, donde en los horarios de más tránsito se provocaban grandes congestiones.

Sin embargo este beneficio trajo una complicación en la intersección de la calle Marcos Avellaneda y la avenida Tagle (Avenida comercial), ya que el tráfico se intensificó debido a que por esta

avenida circulan cuatro líneas de colectivos, más los vehículos particulares, provocando accidentes y grandes embotellamientos en horarios más transitados.



Por tal motivo los vecinos del barrio solicitan al municipio un sistema de semaforización en esta intercepción de calles a fin de solucionar la situación del tránsito.



Te invitamos a que diseñes y fabriques un prototipo maquetizado y automatizado de semaforización doble para colaborar con los vecinos de Villa Caraza para resolver este problema de tránsito utilizando la metodología proyectual para resolver problemas tecnológicos:

La tecnología adopta un esquema de trabajo basado en el método proyectual, el cual posee diferentes etapas o fases para su concreción.

- **Percepción del problema**
- **Búsqueda de alternativas de solución**
- **Selección de la solución adecuada**
- **Diseño de la solución**
- **Organización del trabajo**
- **Construcción de modelos**
- **Evaluación y perfeccionamiento**

Por fines educativos y teniendo en cuenta las características de las y los estudiantes del ciclo básico, esta metodología proyectual, tomara otra forma. La fase o etapa denominada “Percepción del problema”, tomara una rol más orientativo definido por los contenidos, las expectativas de logro y las capacidades que las y los jóvenes deben desarrollar. Esta fase se redefinirá en dos fases simultáneas “Diseño de la situación Problemática” y “Planteo de la problemática”.

### **Método proyectual para estudiantes de Ciclo Básico**

- **Percepción del problema**
  - **Búsqueda de alternativas de solución**
  - **Selección de la solución adecuada**
  - **Diseño de la solución**
  - **Organización del trabajo**
  - **Construcción de modelos**
  - **Evaluación y perfeccionamiento**
- [
- **Diseño de la situación problema**
  - **Planteo de la problemática**

### **Posible Procedimiento**

- **Diseño de la problemática**

La acción comienza pensando una situación problema para las y los estudiantes. En el caso de este ejemplo, les dará respuesta a los vecinos del barrio de Villa Caraza para la semaforización de los cruces de las calles Marco Avellaneda y Tagle.



Las y los docentes involucrados se reúnen para pensar una situación problema para plantearle a sus estudiantes. Para su diseño deben tener en cuenta los ejes estructurantes, los contenidos y las capacidades que sus estudiantes deben desarrollar. Esta es una instancia en donde se articulan los espacios o módulos que conforman el taller. Para su planteo deben pensar un enunciado que contextualice una situación real que las y los jóvenes consideren útil para la comunidad.

- **Planteo de la Problemática**

Una vez diseñada la problemática, las o los docentes deben plantearlas a sus estudiantes de la forma más abierta posible. La descripción del problema será realizada de modo tal que pueda ser considerada desde diferentes perspectivas. Esto se refiere a que los y las estudiantes tengan una percepción global en relación al problema como etapa previa a la búsqueda de una solución.



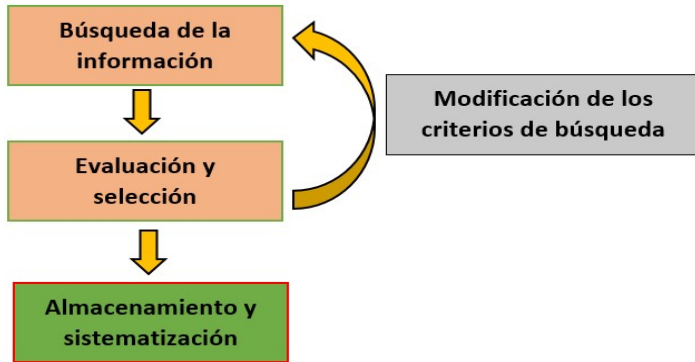
Este planteo de la situación a resolver, puede estar acompañado por documentos, imágenes, artículos periodísticos, videos y todo tipo de información que sirva para contextualizar la situación problema que deberán resolver.

- **Búsqueda de alternativas de solución**

Si bien los docentes deben hacer un andamiaje durante todo el proceso. En esta etapa será de suma importancia sus estrategias de mediación para lograr que los jóvenes puedan consensuar en una propuesta de solución lo más colectiva posible y que este dentro de su alcance.

Esta etapa del método proyectual implica la necesidad, que los estudiantes adquieran saberes relacionados con la búsqueda y análisis de información. Los procesos de búsqueda de información son complejos y cíclicos, e implican una serie de actividades tales como:

- a) Búsqueda, evaluación y selección de la información.
- b) Almacenamiento de resultados parciales.
- c) Comparación y análisis de la información obtenida.
- d) Modificación de los criterios de búsqueda: ampliar, especificar o redefinir los criterios.



- **Selección de la solución adecuada**

En esta instancia los estudiantes tendrán que seleccionar la solución que mejor cumpla con la problemática planteada por el o los docentes.

En esta fase de la metodología proyectual debe predominar la ubicuidad. Estamos hablando de la búsqueda de la solución que pueden apropiarse. La solución debe ser posible de realizar con los recursos que tienen o que pueden llegar a tener.



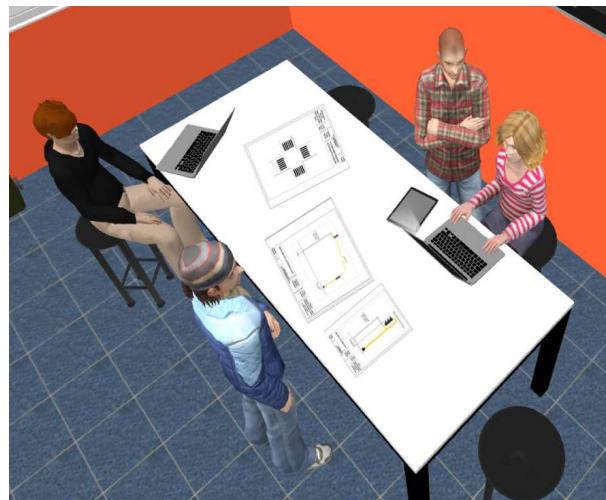
En la selección de la solución, los docentes deberán realizar formulaciones a través de diálogos, realizando preguntas al grupo en relación a las características del prototipo que van a fabricar para dar respuesta al problema planteado.

- **La etapa de diseño**

El diseño es la parte creativa del proyecto tecnológico, porque es aquí donde, a partir de la información que las y los jóvenes han consultado y del conocimiento de la situación en la realidad, realizarán su propia respuesta al problema.



Involucra aprendizajes personales y colectivos y requiere de métodos tales como: croquis, bocetos y planos; diseñar el circuito y la programación, cálculos y estimaciones (de costos, de magnitudes físicas, etc.); la elaboración de planes de acción, etc.



- **Organización del trabajo**

Una vez que las y los estudiantes han optado por una alternativa de solución y realizado el diseño, y antes de comenzar a trabajar en la fabricación, es conveniente que dediquen un tiempo a pensar en cómo organizar y sistematizar las tareas que les permitirán llevar adelante la solución seleccionada.

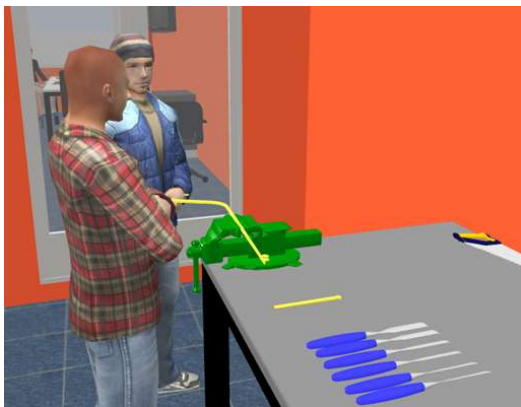


Esto quiere decir, planificar el modo que se llevara a cabo la fabricación del prototipo, detallando todas las tareas a realizar, su secuencia, el tiempo estimado, previendo los recursos necesarios para la ejecución del proyecto y la distribución y asignación de estas tareas entre los integrantes del equipo de trabajo.

- **Construcción de modelos**

Durante esta etapa las y los estudiantes deberán seleccionar y utilizar los materiales, herramientas, aplicar los sistemas involucrados, utilizar máquinas, instrumentos y sus distintos procedimientos, planificando su aplicación eficiente, aplicar técnicas manuales y técnicas digitales de fabricación, explicar a terceros cómo se desarrolla el trabajo. Seleccionar caminos alternativos cuando aparezcan dificultades y solicitar ayuda cuando la necesiten.

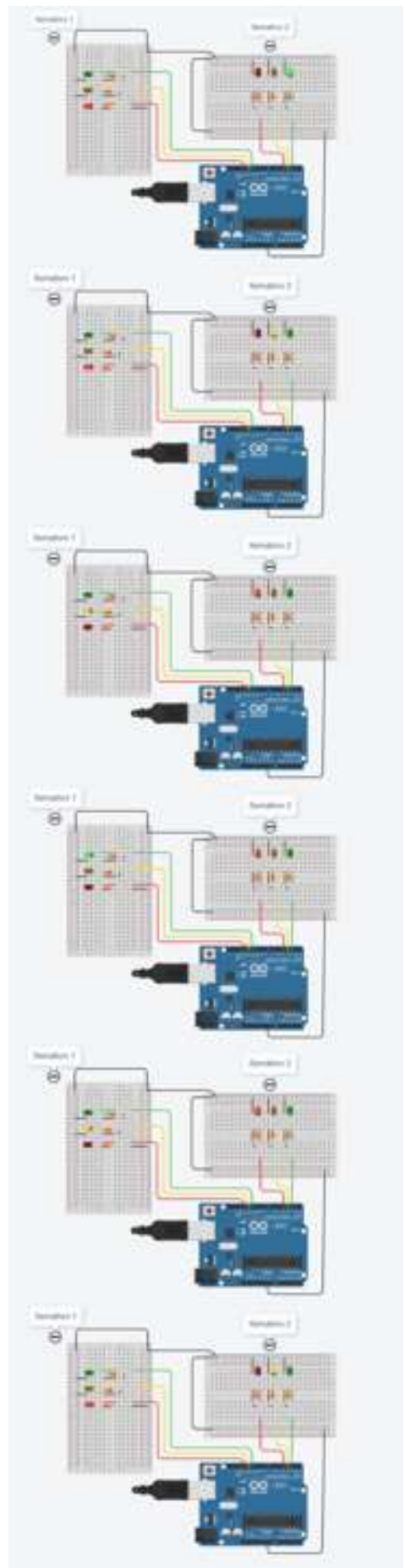
En esta instancia las y los jóvenes aplicaran saberes de medición, trazado, corte, plegado, agujereado, diseño y fabricación digital, técnicas de unión y ensamblado, aplicaran saberes de electrónica y aplicaran la programación del sistema previamente diseñada y todos los procedimientos para la fabricación del prototipo.



```

1  #define rojo1 13
2  #define amarillo1 12
3  #define verde1 11
4  #define rojo2 4
5  #define amarillo2 3
6  #define verde2 2
7
8  void setup()
9  {
10     pinMode(13, OUTPUT);
11     pinMode(12, OUTPUT);
12     pinMode(11, OUTPUT);
13     pinMode(4, OUTPUT);
14     pinMode(3, OUTPUT);
15     pinMode(2, OUTPUT);
16 }
17 void loop()
18 {
19     digitalWrite(rojo1,HIGH);//Semaforo 1 prendido
20     digitalWrite(amarillo1,LOW);//Semaforo 1 apagado
21     digitalWrite(verde1,LOW);//Semaforo 1 apagado
22     digitalWrite(rojo2,LOW);//Semaforo 2 apagado
23     digitalWrite(amarillo2,LOW);//Semaforo 2 apagado
24     digitalWrite(verde2,HIGH);//Semaforo 2 prendido
25     delay(4000); //Espera 4 segundos
26
27     digitalWrite(rojo1,HIGH);//Semaforo 1 prendido
28     digitalWrite(amarillo1,LOW);//Semaforo 1 apagado
29     digitalWrite(verde1,LOW);//Semaforo 1 apagado
30     digitalWrite(rojo2,LOW);//Semaforo 2 apagado
31     digitalWrite(amarillo2,HIGH);//Semaforo 2 prendido
32     digitalWrite(verde2,LOW);//Semaforo 2 apagado
33     delay(4000);//Espera 4 segundos
34
35     digitalWrite(rojo1,LOW);//Semaforo 1 apagado
36     digitalWrite(amarillo1,HIGH);//Semaforo 1 prendido
37     digitalWrite(verde1,LOW);//Semaforo 1 apagado
38     digitalWrite(rojo2,HIGH);//Semaforo 2 prendido
39     digitalWrite(amarillo2,LOW);//Semaforo 2 apagado
40     digitalWrite(verde2,LOW);//Semaforo 2 apagado
41     delay(4000);//Espera 4 segundos
42
43     digitalWrite(rojo1,LOW);//Semaforo 1 apagado
44     digitalWrite(amarillo1,LOW);//Semaforo 1 apagado
45     digitalWrite(verde1,HIGH);//Semaforo 1 prendido
46     digitalWrite(rojo2,HIGH);//Semaforo 2 prendido
47     digitalWrite(amarillo2,LOW);//Semaforo 2 apagado
48     digitalWrite(verde2,LOW);//Semaforo 2 apagado
49     delay(4000);//Espera 4 segundos
50
51     digitalWrite(rojo1,LOW);//Semaforo 1 apagado
52     digitalWrite(amarillo1,HIGH);//Semaforo 1 prendido
53     digitalWrite(verde1,LOW);//Semaforo 1 apagado
54     digitalWrite(rojo2,HIGH);//Semaforo 2 prendido
55     digitalWrite(amarillo2,LOW);//Semaforo 2 prendido
56     digitalWrite(verde2,LOW);//Semaforo 2 apagado
57     delay(4000);//Espera 4 segundos
58
59     digitalWrite(rojo1,HIGH);//Semaforo 1 prendido
60     digitalWrite(amarillo1,LOW);//Semaforo 1 apagado
61     digitalWrite(verde1,LOW);//Semaforo 1 apagado
62     digitalWrite(rojo2,LOW);//Semaforo 2 apagado
63     digitalWrite(amarillo2,HIGH);//Semaforo 2 prendido
64     digitalWrite(verde2,LOW);//Semaforo 2 apagado
65     delay(4000);//Espera 4 segundos
66 }

```



- **La etapa de evaluación y perfeccionamiento**

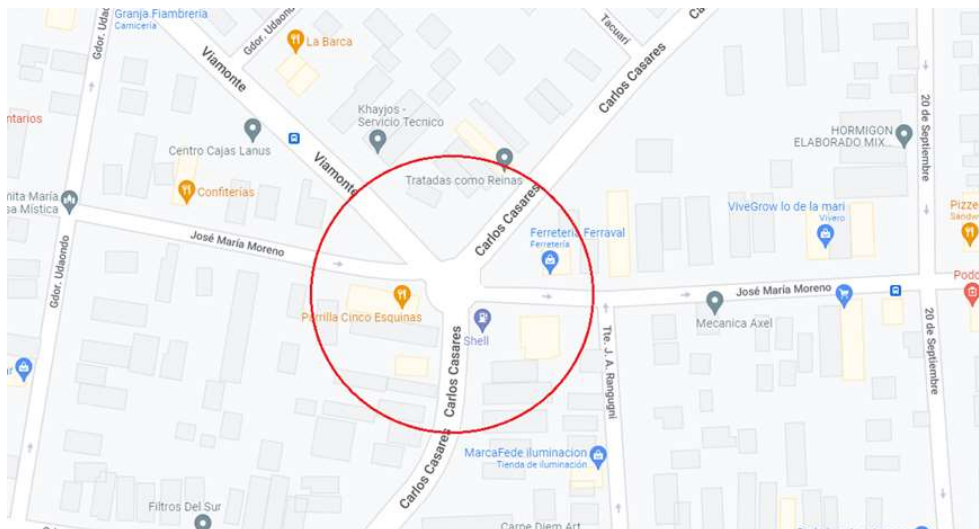
En esta etapa, se revisara todo el proceso que llevo a la fabricación del prototipo, se compararan el resultado obtenido con los objetivos iniciales. Probaran su funcionamiento e incluso podrán sugerir cambios y mejoras en cada una de las fases anteriores. Analizaran las consecuencias deseadas y las no deseadas.

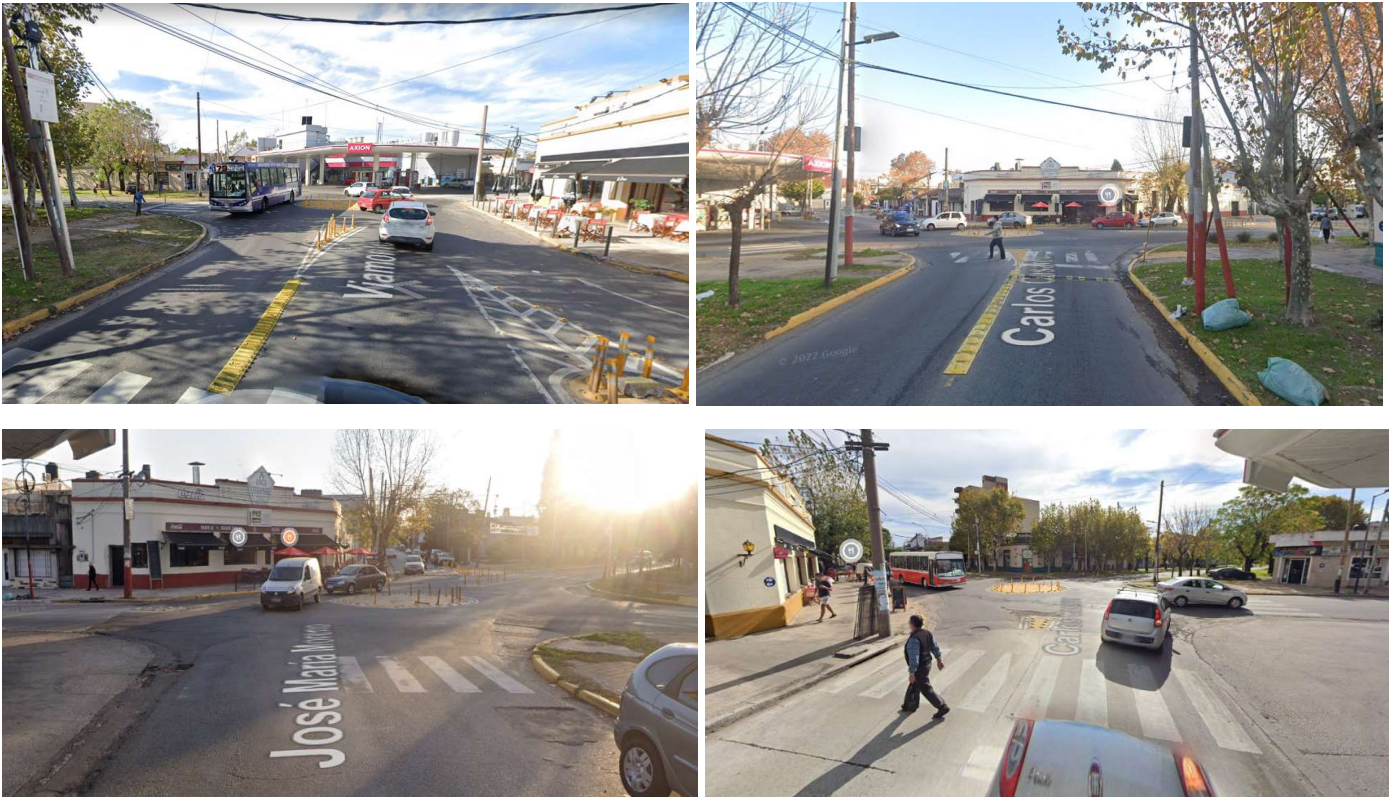


## 5. Situación Problema Tecnológica. “Problemática Abierta” (Metodología Projectual). \*

### “Sistema de semaforización triple”

En el municipio de Lanús en la provincia de Buenos Aires existe una intercepción de tres calles muy transitadas (José María Moreno, Viamonte y Carlos Casares). Los vecinos han juntado firmas para solicitar al municipio la semaforización de la intersección de estas calles debido a que se dificulta el cruce peatonal y ocasionalmente suceden accidentes.





Te invitamos a que diseñes y fabriques un prototipo maquetizado y automatizado de semaforización triple para colaborar con los vecinos de Lanús para resolver este problema de tránsito utilizando la metodología proyectual para resolver problemas tecnológicos:



### Posible Procedimiento

- **Diseño de la problemática**

La acción comienza pensando una situación problema para las y los estudiantes. En el caso de este ejemplo, les dará respuesta a los vecinos de un barrio de Lanús que solicitan al municipio la semaforización del cruce de tres calles (José María Moreno, Viamonte y Carlos Casares).



Las y los docentes involucrados se reúnen para pensar una situación problema para plantearle a sus estudiantes. Para su diseño deben tener en cuenta los ejes estructurantes, los contenidos y las capacidades que sus estudiantes deben desarrollar. Esta es una instancia en donde se articulan los espacios o módulos que conforman el taller. Para su planteo deben pensar un enunciado que contextualice una situación real que las y los jóvenes consideren útil para ellos o para la comunidad.

- **Planteo de la Problemática**

Una vez diseñada la problemática, las o los docentes deben planteárselas a sus estudiantes de la forma más abierta posible. La descripción del problema será realizada de modo tal que pueda ser considerada desde diferentes perspectivas. Esto se refiere a que los y las estudiantes tengan una percepción global en relación al problema como etapa previa a la búsqueda de una solución.



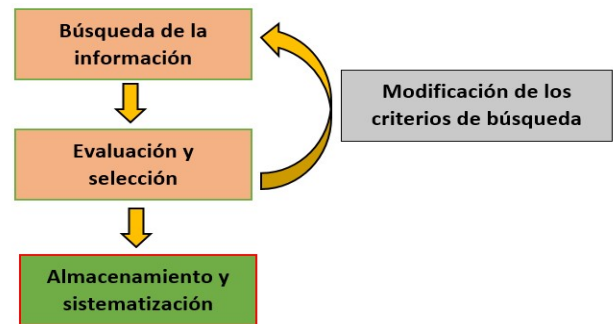
El planteo de la situación previamente diseñada por los docentes, puede estar acompañada por documentos, imágenes, artículos periodísticos, videos y todo tipo de información que sirva para contextualizar la situación problema que deberán resolver.

- **Búsqueda de alternativas de solución**

Si bien los docentes deben hacer un andamiaje durante todo el proceso. En esta etapa será de suma importancia sus estrategias de mediación para lograr que los jóvenes puedan consensuar en una propuesta de solución lo más colectiva posible y que este dentro de su alcance.

Esta etapa del método proyectual implica la necesidad, que los estudiantes adquieran saberes relacionados con la búsqueda y análisis de información. Los procesos de búsqueda de información son complejos y cíclicos, e implican una serie de actividades tales como:

- a) Búsqueda, evaluación y selección de la información.
- b) Almacenamiento de resultados parciales.
- c) Comparación y análisis de la información obtenida.
- d) Modificación de los criterios de búsqueda: ampliar, especificar o redefinir los criterios.



- **Selección de la solución adecuada**

En esta instancia los estudiantes tendrán que seleccionar la solución que mejor cumpla con la problemática planteada por el o los docentes.

En esta fase de la metodología proyectual debe predominar la ubicuidad. Estamos hablando de la búsqueda de la solución que pueden apropiarse. La solución debe ser posible de realizar con los recursos que tienen o que pueden llegar a tener.

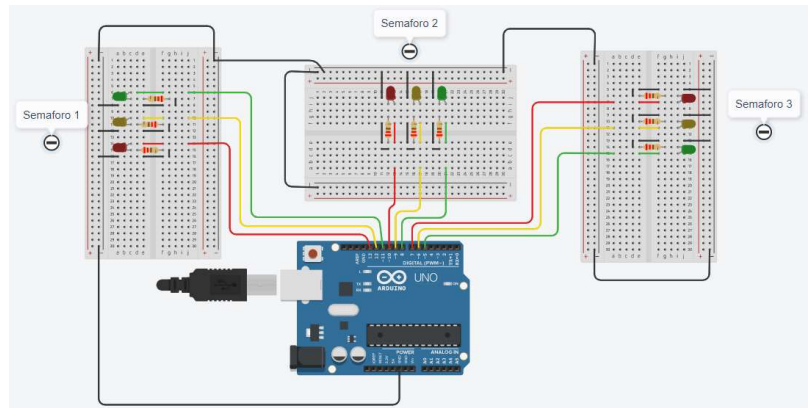
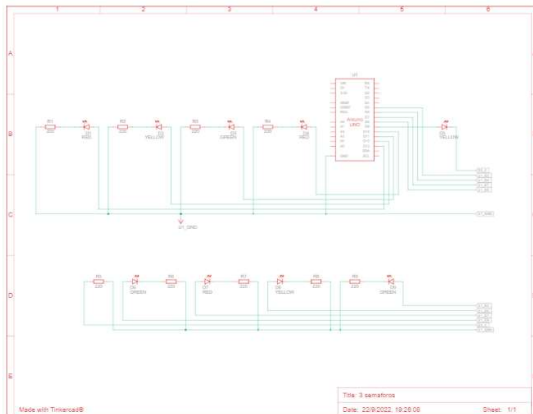


En la selección de la solución, los docentes deberán realizar formulaciones a través de diálogos, realizando preguntas al grupo en relación a las características del prototipo que van a fabricar para dar respuesta al problema planteado.

- **La etapa de diseño**

El diseño es la parte creativa del proyecto tecnológico, porque es aquí donde, a partir de la información que las y los jóvenes han consultado y del conocimiento de la situación en la realidad, realizan su propia respuesta al problema.

Involucra aprendizajes personales y colectivos y requiere de métodos tales como: croquis, bocetos y planos; diseñar el circuito y la programación, cálculos y estimaciones (de costos, de magnitudes físicas, etc.); la elaboración de planes de acción, etc.



- **Organización del trabajo**

Una vez que las y los estudiantes han optado por una alternativa de solución y realizado el diseño, y antes de comenzar a trabajar en la fabricación, es conveniente que dediquen un tiempo a pensar en cómo organizar y sistematizar las tareas que les permitirán llevar adelante la solución seleccionada.



Esto quiere decir, planificar el modo que se llevara a cabo la fabricación del prototipo, detallando todas las tareas a realizar, su secuencia, el tiempo estimado, previendo los recursos necesarios para la ejecución del proyecto y la distribución y asignación de estas tareas entre los integrantes del equipo de trabajo.



- **Construcción de modelos**

Durante esta etapa las y los estudiantes deberán seleccionar y utilizar los materiales, herramientas, aplicar los sistemas involucrados, utilizar máquinas, instrumentos y sus distintos procedimientos, planificando su aplicación eficiente, aplicar técnicas manuales y técnicas digitales de fabricación, explicar a terceros cómo se desarrolla el trabajo. Seleccionar caminos alternativos cuando aparezcan dificultades y solicitar ayuda cuando la necesiten.

En esta instancia las y los jóvenes aplicaran saberes de medición, trazado, corte, plegado, agujereado, diseño y fabricación digital, técnicas de unión y ensamblado, aplicaran saberes de electrónica y aplicaran la programación del sistema previamente diseñada, y todos los procedimientos para la fabricación del prototipo.





- **La etapa de evaluación y perfeccionamiento**

En esta etapa, se revisara todo el proceso que llevo a la fabricación del prototipo, se compararan el resultado obtenido con los objetivos iniciales. Probaran su funcionamiento e incluso podrán sugerir cambios y mejoras en cada una de las fases anteriores. Analizaran las consecuencias deseadas y las no deseadas.

