***Traductor De Código Morse***

*Integrantes del proyecto: Tomás Fleitas, Benjamín Korstanje,*

*Brunella Figallo, Martín Romero, Kiara Hernandez*

*Docente: Gonzalo Consorti*

*Curso: 4to 2da*

******

***Punto de vista inicial y Consignas:***

*La idea es que dos arduinos se puedan comunicar entre sí por código morse. Escribimos una palabra o texto por el monitor serial y esta se envía en formato morse y se reproduce mediante un buzzer en el otro arduino. También debemos lograr que el arduino interprete las pulsaciones de un botón como código morse y lo pase a texto y mostrarlo por una pantalla LCD, es decir en viceversa.*

***División de Tareas:***

*Ahora tenemos que decidir qué parte hace cada uno primero como explicación en código y conexiones, tenemos que conectar un botón y usar el comando “Serial.read()” para enviar datos al arduino y que éste los interprete como información para traducirlos como morse mediante el uso de vectores.*

*Discutimos por un par de minutos (para ser exacto 20 minutos) quienes harían que cosa. Tuvimos en cuenta más que nada la primera voluntad. Es decir, nombramos tareas al azar y los primeros en ofrecerse para tal trabajo. Concretarían la tarea. Por lo que con ese modus operandi, la repartición de tareas quedó algo así:*

***Fleitas: Conectar los cables.***

***Krawczyk: Traducción de pulsaciones de un botón a código morse y mostrar por LCD***

***Brunella:Traducción de pulsaciones de un botón a código morse y mostrar por LCD***

***Hernandez:Asistencias en distintos ámbitos del proyecto. (polivalente)***

***Korstanje: Asistencias en distintos ámbitos del proyecto. (polivalente)***

*Mi rol sería algo que describí como polivalencia. Cualquier tarea que me den yo la concretaría. Osea que si me piden que ayude a Fleitas, mi compañero, a conectar cables. Yo lo hago. Si me piden probar un código, yo lo hago y si necesitan que investigue sobre algo en especifico. Yo lo hago*

*Brunella y Romero tratarían de ver cómo encarar la programación del código para conseguir que el arduino traduzca por pulsaciones. Pues decidieron comenzar ellos mismos con esa primera parte del arduino y luego comenzar a diseñar el código para que se traduzca mediante una luz LED.*

*Fleitas se encargaría de conectar los cables en primera instancia, por lo que inicie un poco al principio con él en TiinkerCad en busca de como sería.*

*Hernandez cumpliría mi mismo papel, aunque desconozco que tarea le encargaron mis compañeros aún.*

***Elementos a Utilizar:***

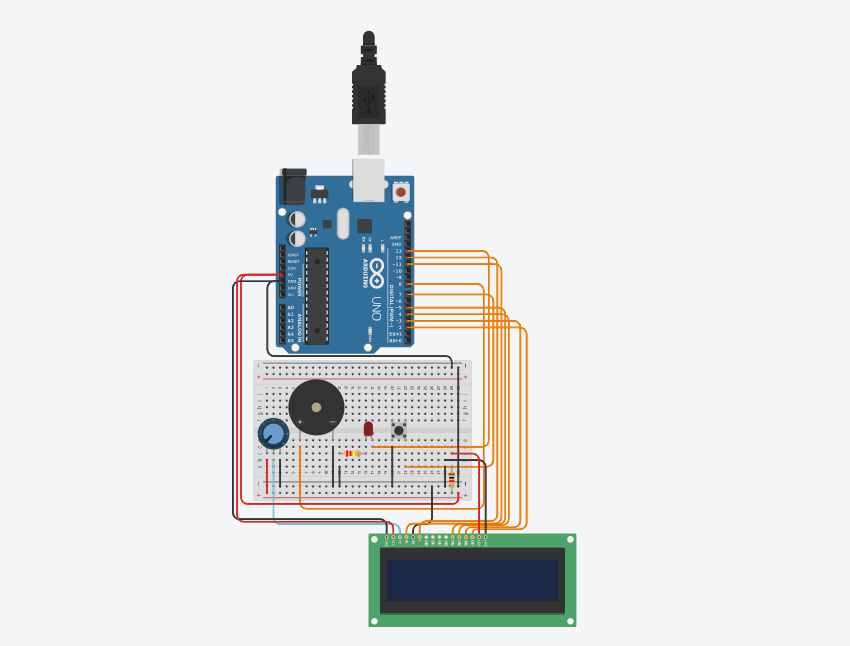
1. *Buzzer (1)*
2. *Resistencias (2)*
3. *LED (1)*
4. *Pantalla LCD 16x2 (1)*
5. *Botón (1)*
6. *Potenciómetro (1)*
7. *Cables (Alrededor de 25)*

*Lo siguiente Fue comenzar a plantear como sería la conexión de cables, con los materiales necesarios ya planteados.*

*El botón cumpliría la función de dar pulsaciones para que luego el monitor serial traduzca. Ahorré resistencias colocando simplemente una al final de las conexiones negativas.*

*Originalmente solo usamos el Led para la otra fase del proyecto. Pero añadimos un Buzzer por las dudas. (Solo quería poner un buzzer para que quede más fachero). Eso sería lo básico para el proyecto entre otros componentes.*

***Planteamiento Inicial:***

******

*Armado de cables y componentes inicialmente planteado por Korstanje, Fleitas y Hernandez*

*Más tarde se nos unió la compañera Hernandez al cableado. Cómo Fleitas y yo ya habíamos conectado la inmensa mayoría. Dejamos que ella conectará la pantalla LCD y les diera el color correspondiente a cada cable para asegurarnos de que esté de forma ordenada.*

*Este fue el planteamiento inicial del código. proporcionado por Figallo y Romero. Y hablado conmigo también. Ya que ellos lo escribieron pero previamente hablamos entre todos más o menos como plantear lo básico del código, que tras un tiempo de investigación. Terminó con la conclusión de que ese String sería la forma correcta de comenzar.*

| *String letrasMor[] =  {".-","-...","-.-.","-..",".","..-.","--.","....",".." // A-I ,".---","-.-",".-..","--","-.","---",".--.","--.-",".-." // J-R ,"...","-..-","...-",".--","-..-","-.--","--.."}; // S-Z   String numerosMor[] =  {"-----",".----","..---","...--","....,",".....","-...." // 0-6 ,"--...","---..","----."}; // 7-9* |
| --- |

*En cuanto al código la complicación inicial fue hacer que las funciones sean recursivas. Eso y configurar el abecedario como los puntos y guiones del código morse basado en secciones (desde cierto vector representa cierto carácter y en base a ese cada vector tiene una traducción única) por lo que optamos por usar una sola sección grande que cumpla esa función. Esto hablando desde un punto de vista del planteamiento, pues hasta este punto solo hablamos entre nosotros para saber cómo sería.*

*El planteamiento dicho se realizó puesto que algunos de nosotros sintieron que sería mucho más sencillo de esa manera y se evitarían contratiempos a la hora de hacerlo todo de corrido.*

*Una vez aplicado lo anterior mencionado, el string quedo asi:*

| *String letras[] = { "a" , "b" , "c" , "d" , "e" , "f" , "g" , "h" , "i" , "j" , "k" , "l" , "m" , "n" , "o" , "p" , "q" , "r" , "s" , "t" , "u" , "v" , "w" , "x" , "y" , "z" }; String codigo[] = { ".-", "-..." , "-.-." , "-.." , "." , "..-." , "--." , "...." , ".." , ".---" , "-.-" , ".-.." , "--" , "-." , "---" , ".--." , "--.-" , ".-." , "..." , "-" , "..-" , "...-" , ".--" , "-..-" , "-.--" , "--.." };*  *int numeros[] = { 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 }; String codigoNum[] = { "-----" , ".----" , "..---" , "...--" , "....-" , "....." , "-...." , "--..." , "---.." , "----." };  ".--/-...//-.-"* |
| --- |

*Por parte de mi compañero Martín, ha compartido el avance de su código. Ya que habíamos comentado un poco como continuar a partir del String, cosas como en que pin va cada conexión. Usamos lo planteado por Fleitas, Hernandez y yo como referencia para esas cosas. Por lo que tras tener unos minutos de charla para desarrollar vagamente el código. Romero terminó el código en privado y me lo envió con la petición de que lo testeara, usando la base de TinkerCad anteriormente mencionada.*



*Tras testearlo exitosamente. Comenzamos a pasar a otra etapa del proyecto. Repartir tareas tras la concreción de la conexión de cables.*

*Se les dieron otros trabajos Fleitas y Hernandez, no se cuales fueron ya que Romero se los comentó.*

*Por mi parte, se me encargó realizar e investigar sobre un código mediante el cual ingresando palabras en el monitor serial, a través de un LED, se traduzca en código morse (es decir parpadee según el lenguaje morse).*

*Rápidamente mi investigación se trasladó hacia Youtube, luego de haber mendigado por algunos sitios de algunas universidades en busca de tal proyecto, el cuál no logré avistar.*

*Sin embargo mi investigación en Youtube fue próspera y haye el siguiente video:*

*https://www.youtube.com/watch?v=hTC7e5rX1tk&t=8s*

*Ostentaba exactamente lo que necesitaba para continuar con la siguiente etapa de mi trabajo. Las bases para que funcionara el LED. Confirmaba nuestro planteo inicial del String y me ayudó a comprender de manera más simple la cadena de If´s, debido a la naturaleza sencilla del código que encontré en el video, también contenía formas de identificar los caracteres necesarios, cosa que había charlado con Romero con anterioridad por lo que ya tenía una idea .*

*Dejaré un poco del código para que vean de lo que hablo:Si bien logra prácticamente su cometido, no tiene definido el LED. Por eso es la base.*

| *if (ch >= 'a' && ch <= 'z') { flashSequence(letters[ch - 'a']); } else if (ch >= 'A' && ch <= 'Z') {* |
| --- |

*Mostré una pequeña parte de como funciona, de esa forma con los String anteriormente mostrado, se identifica una letra para luego realizar tal acción.*

*Una vez definido el LED me aseguré que funcionase y así lo hace, sin embargo todo esto lo testee en el planteamiento inicial del cableado, ya usado con anterioridad para testear el código proporcionado por Romero. Por ende… me sobraban muchísimos componentes, entre ellos: un buzzer, un botón, un potenciómetro y una resistencia. Los cuales no había usado para absolutamente nada.*

*Para hacer uso de más componentes me tomé la libertad y añadí un buzzer al código (usando así un buzzer y una resistencia que sobraban). Además de agregar un “Serial.print” para que el monitor serial diga: “Ingrese su palabra”.*

*Me costó alrededor de media hora lo del buzzer, ya que se me complicaba la posición a la hora de usar el Tabulador. Me daba error por alguna razón, hasta que a través de muchos intentos de diferentes maneras. Funcionó.*

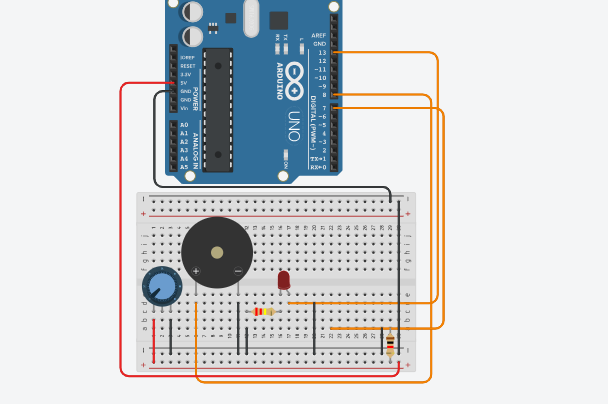
*Tras esos contratiempos y la adaptación completa del código. Terminé de formularlo y repito, lo testee y funciona. Este es el inicio del código, con el buzzer agregado:*

| *int ledPin = 13; int buzzerPin = 8;   char\* letters[] = {  ".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..",  ".--", "-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.",  "...", "-", "..-", "...-", ".--", "-..-", "-.--", "--.." };* |
| --- |
|  |

*Me asegure de definir claramente cada elemento a usar, siendo estos únicamente el LED y el buzzer, el resto se compone exactamente como mostré con anterioridad, con una cadena de If´s.*

*Cree un TinkerCad aparte para este código y testearlo. He aquí la lista de los componentes que usé:*

1. *Buzzer (1)*
2. *Resistencias (2)*
3. *LED (1)*
4. *Potenciómetro (1)*
5. *Cables (Alrededor de 11)*

**

*Así quedaría más o menos por mi parte, le quite la pantalla LCD y el botón en pos del trabajo que se me fue otorgado.*

*He aquí otro sitio que frecuente para tratar de instruirme más en el tema. Producto de mi investigación del código mencionado anteriormente:*

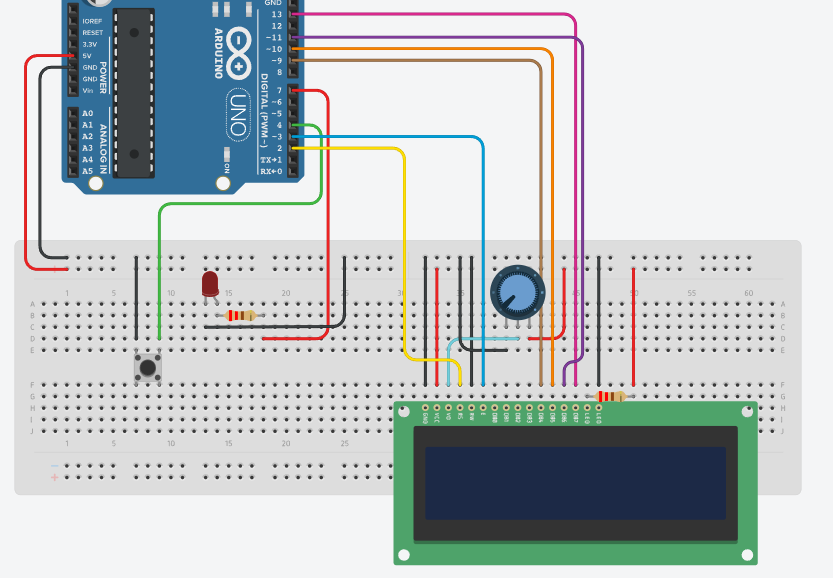
[*https://blogs.etsii.urjc.es/dseytr/traductor-bidireccional-espanol-morse/*](https://blogs.etsii.urjc.es/dseytr/traductor-bidireccional-espanol-morse/)

*Esta página de la Universidad Rey Juan Carlos habla detalladamente sobre el proceso, el hardware que se necesita, planteamientos, etc. Por lo que fue realmente útil en varios momentos a la hora de armar el TinkerCad en general, también me entretuve un poco escuchando música mientras leía el artículo.*

*Con eso de momento, he concluído mi investigación la cual duró alrededor de 3 horas entre sitios web y videos de Youtube, culminando en la integración y adaptación del código.*

*Lo siguiente fue la combinación luego de charlarlo en grupo cada uno le proporcionó compañero que se ofreció a unir todo, más bien estaba ya reuniendo todo. Romero. Por lo que le cedimos nuestras partes, en mi caso, el código mostrado hasta el momento junto a el diseño de cables realizado con Fleitas.*

*Con la fusión de las partes de todos (Figallo, Hernandez, Romero, Fleitas y Yo), la conexión de cables terminó de forma eficiente, para ser más exactos. De la forma que se puede ver por debajo de esta página:*

******

*Me siento ofendido porque quitaron el buzzer que había propuesto.*

*Parece que consideraron que sería demasiado molesto testearlo o ponerlo en práctica. por lo que simplemente lo sacaron.*

*De momento esta es la lista de tareas por hacer:*

1. *Modelo 3D (Impresión a fisico tambien)*
2. *PowerPoint y/o Canvas (Presentación)*

*Ya nos repartimos las tareas desde ese momento y nos propusimos lo siguiente:*

*Romero Y Figallo se encargaría del Modelo 3D. Y Fleitas, Hernandez y yo. De completar la presentación. Por lo que debo dar inicio a una de las últimas etapas del proyecto.*

*La de comenzar la presentación. Me dirigí a Canvas con mis compañeros y tratamos de escoger un tema, el cual se titula “Presentación de proyecto divertida azul”, esto puesto es mucho más fácil usar plantillas ya construidas a la hora de diseñar.*

*Por lo que planteamos varias en el momento en pos de buscar la más ilustrativa para el proyecto. Proceso del cual se encargó en su mayor parte la compañera Kiara Hernandez.*

**

*Esta es la plantilla en cuestión. La eligió Hernandez porque consideró que sería agradable visualmente para el espectador*

*Yo le dí el visto bueno ya que considere que los colores pasteles podrían servir para darle un buen atractivo visual, finalmente me decanté más por los colores oscuros pero el formato de la hoja en la diapositiva me pareció lo bastante bueno como para mantenerlo, lo mismo me pasó con el formato del título.*

*Comencé con lo que sería la carátula de la presentación, pensando en algo no muy ostentoso ya que sería la primera impresión.*

**

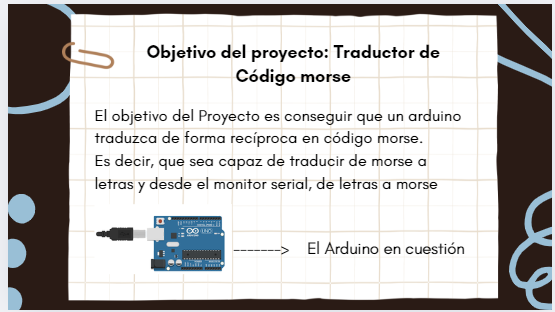
*puse un fondo negro, eliminando las comillas color pastel que adornaban el título y escribí tanto el título como los integrantes del grupo.*

*La fuente del título:*

*Es un LazyDog de color pastel, con nivel de fuente 92.*

*Mientras que el texto de los integrantes:*

*su fuente consta de un Glacial Indifference color negro simple en nivel de fuente 52. Además de eso, planeo meterle animación más adelante.*

**

*Durante la siguiente página trate de resumir el objetivo del proyecto, así como, de qué manera debería funcionar para ser considerado por los estándares del trabajo como válido.*

*También me asegure de añadir una imagen del arduino que utilizamos a lo largo del proyecto, puesto me parecería correcto mostrar de lo que estamos hablando.*

*Habiendo juntando objetivos, una ilustración sobre el componente principal del que trata el proyecto y una explicación resumida y entendible del proyecto. Creo haber conseguido una introducción satisfactoria.*

*Fuente del título:*

*Glacial Indifference color negro oscurecido. Nivel de fuente: 46,9.*

*Fuente del texto explicativo:*

*Glacial Indifference color negro simple. Nivel de fuente: 43,2.*

*Por cierto la fuente de la foto utilizada en esta página es la siguiente:*

[*https://www.google.com/imgres?q=foto%20de%20arduino%20png&imgurl=https%3A%2F%2Fupload.wikimedia.org%2Fwikipedia%2Fcommons%2F7%2F7c%2FArduino\_Uno\_%2528Versi%25C3%25B3n\_Inform%25C3%25A1tica%2529.png&imgrefurl=https%3A%2F%2Fes.m.wikipedia.org%2Fwiki%2FArchivo%3AArduino\_Uno\_(Versi%25C3%25B3n\_Inform%25C3%25A1tica).png&docid=MuP9mopYY1kneM&tbnid=yD5tR1nN9oYLMM&vet=12ahUKEwi4u6K44f-JAxWYGLkGHRXEKOUQM3oECGQQAA..i&w=1629&h=830&hcb=2&ved=2ahUKEwi4u6K44f-JAxWYGLkGHRXEKOUQM3oECGQQAA*](https://www.google.com/imgres?q=foto%20de%20arduino%20png&imgurl=https%3A%2F%2Fupload.wikimedia.org%2Fwikipedia%2Fcommons%2F7%2F7c%2FArduino_Uno_%2528Versi%25C3%25B3n_Inform%25C3%25A1tica%2529.png&imgrefurl=https%3A%2F%2Fes.m.wikipedia.org%2Fwiki%2FArchivo%3AArduino_Uno_(Versi%25C3%25B3n_Inform%25C3%25A1tica).png&docid=MuP9mopYY1kneM&tbnid=yD5tR1nN9oYLMM&vet=12ahUKEwi4u6K44f-JAxWYGLkGHRXEKOUQM3oECGQQAA..i&w=1629&h=830&hcb=2&ved=2ahUKEwi4u6K44f-JAxWYGLkGHRXEKOUQM3oECGQQAA)

*Lo cierto es que no se me ocurre mucho que poner en la siguiente página, por lo que*

*creo que pondré algo del proceso realizado en el código, sobre como lo planteamos,*

*etc.*



*En la tercera me decanté por el reparto de tareas, para que se supiera más o menos cómo y en qué trabajo cada uno.*

*De esa forma cumpliría una de las condiciones que recuerdo haber leído sobre la presentación. La organización de tareas.*

*En base a eso me pareció acorde mostrar la organización de tareas inicial.*

*Permítame recordar cómo se organizaron en cada tarea:*

***Fleitas: Conectar los cables.***

***Krawczyk: Traducción de pulsaciones de un botón a código morse y mostrar por LCD***

***Brunella:Traducción de pulsaciones de un botón a código morse y mostrar por LCD***

***Hernandez:Asistencias en distintos ámbitos del proyecto. (polivalente)***

***Korstanje: Asistencias en distintos ámbitos del proyecto. (polivalente)***

*Por lo que prácticamente calque, aunque aparezca que estoy en conexión de cables, fue un error y ya estoy colocado en polivalente.*

*La fuente del título es:*

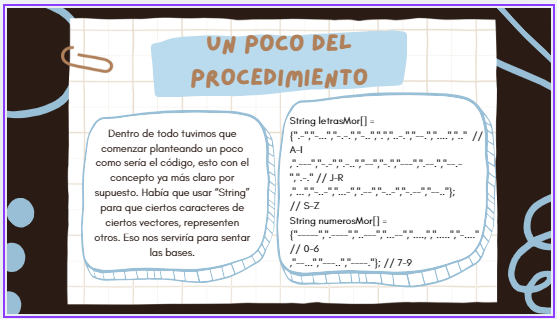
*LazyDog 92, de color pastel.*

*Fuente del texto del medio:*

*Glacial Indifference 30,8 color negro simple. Tuve complicaciones para que este pedazo de texto entrara, pero con paciencia logré que entrara.*

*Fuente del texto de abajo:*

*Glacial Indifference, todo el texto de abajo tiene como nivel de fuente 32. Color negro oscuro opté por copiar el fondo del título, ese especie de tinte de marcador y lo torne de un marrón para tratar de darle un efecto más agradable visualmente lo que me ayudo a contrastar el bordeado negro que le puse.*

**

*En la siguiente página cumplí mi palabra y comencé con las bases del procedimiento, dando a entender de forma breve en el cuadro de la izquierda como funcionaba el concepto básico de nuestro código, es decir, el String. Explicando que cada ciertos caracteres de ciertos vectores, representan otros caracteres del otro String.*

*Del lado derecho decidí citar el planteamiento en cuestión. Mostrando en bruto el código del String, código que vuelvo a citar:*

| *String letrasMor[] =*  *{".-","-...","-.-.","-..",".","..-.","--.","....",".." // A-I ,".---","-.-",".-..","--","-.","---",".--.","--.-",".-." // J-R ,"...","-..-","...-",".--","-..-","-.--","--.."}; // S-Z*  *String numerosMor[] =*  *{"-----",".----","..---","...--","....,",".....","-...." // 0-6 ,"--...","---..","----."}; // 7-9* |
| --- |

*Fuente del título:*

*LazyDog 63,6. Color pastel.*

*Fuente del cuadro de la izquierda:*

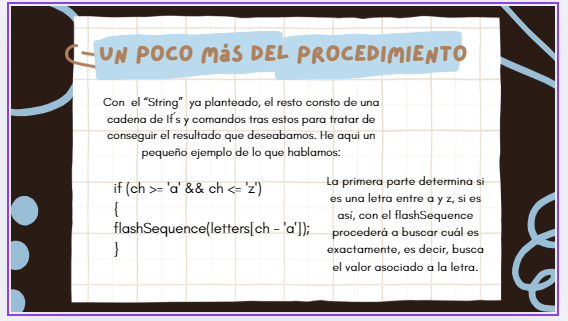
*Glacial Indifference 28,5. Color negro simple.*

*Fuente del cuadro de la derecha:*

*Glacial Indifference 27,3. Color negro simple.*

*Aún así no sabría si solo eso sería explicación suficiente como para que se hagan una idea de cómo funciona el código. Por lo que decidí agregar una diapositiva con una explicación extra.*

*Después de todo me pareció correcto que habiendo explicado ya las bases. Podría proceder como se desarrolló lo básico.*

**

*Busque una secuencia pequeña, para ser exacto. El primer If del código para tomarlo de ejemplo y mostrar cómo se identifican las letras en los vectores del string.*

*De esa forma se abren los caminos a ambos objetivos del proyecto. Se nos permite identificar los valores para ya sea traducirlos a morse a través del LED o para traducirlos a letras a través del botón.*

*Por eso me pareció buena idea que aparezca para que se vea de qué forma más o menos encaramos el proyecto.*

*Esta vez tuve que duplicar el efecto de resaltador sobre el título, ya que aunque era bastante apropiado para lo que trata la diapositiva, era demasiado largo como para estirar el que había ya. Por lo que quedó un efecto de doble resaltador que en mi opinión, quedo mejor que uno solo.*

*Fuente del título:*

*LazyDog 61. Color pastel.*

*Fuente del texto introductorio:*

*Glacial Indifference 32. Color negro simple.*

*Fuente del texto del código:*

*Glacial Indifference 38,5. Color negro simple.*

*Fuente del texto explicativo:*

*Glacial Indifference 32,9. Color negro simple.*

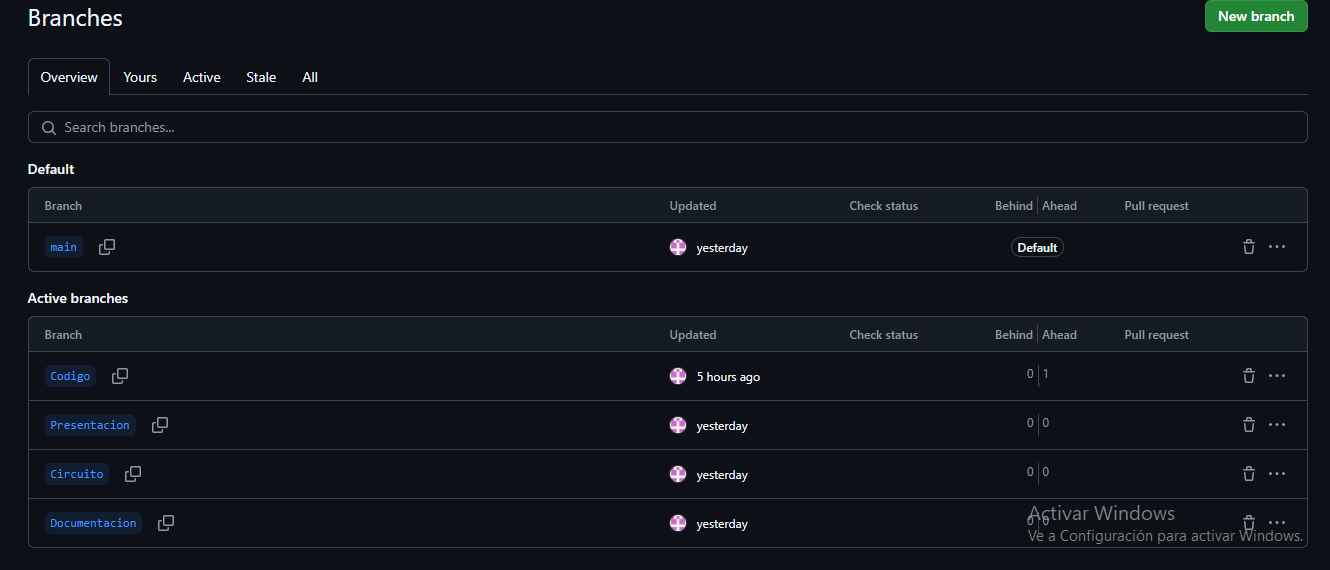
*Ciertamente no sabría qué más poner en la presentación ya que está algo corta de momento. No se si más explicación del código, probablemente sea algo relacionado con el arduino ensi.*

*Siento que eso sería más acorde a una explicación sencilla, cosa que siento debe ser una presentación.*

*Terminó habiendo un pequeño malentendido, resulta. Que el fragmento de código que utilice en la diapositiva de “Un poco más del procedimiento”. No la usó mi compañero Romero en la adaptación final del código.*

*Debido a que la adaptación del código me la compartió post realizada esa diapositiva, es que se dió tal error. Después de todo se me encargó la presentación antes de la culminación de la adaptación del código, hecho que marcó el traspié.*

*Más tarde recibí un mensaje de Romero, había creado un Github del proyecto (requisito para aprobar). Gracias a eso me dijo que me pasara por el mismo para poder ver el código final en su totalidad, por lo que me asegure de chequear la rama llamada “Código”:*

**

*Solo entre a la rama del código, en la parte de cambios realizados para encontrarme el código final completo. El cual dejare aquí abajo para dar una pequeña idea de como formuló el código mi compañero, mejor dicho, como lo adaptó:*

| *LiquidCrystal LCD(2, 3, 9, 10, 11, 13);  #define boton 4 #define led 7* |
| --- |

*Esta parte del código comienza con la definición de las variables.*

*En este caso;*

1. *2, 3 , 8, 10, 11, y 13 para la pantalla LCD.*
2. *4 para el botón*
3. *Y 7 para el LED*

*todo eso teniendo en cuenta la conexión final claramente.*

| *pinMode(boton, INPUT\_PULLUP); pinMode(led, OUTPUT);* |
| --- |

*Durante está parte configuramos el botón como INPUT\_PULLUP y el LED como OUTPUT, básicamente como entrada y salida, que uno reciba y el otro muestre lo que tiene para ofrecer cada uno.*

| *Serial.begin(9600); LCD.begin(16, 2);* |
| --- |

*También se asegura de configurar el inicio de la comunicación serial en 9600 milisegundos y establecer el tamaño de la pantalla LCD en 16 columnas y 2 filas.*

*(quitando el String anteriormente mostrado, esta es la configuración inicial del código. Por lo que ya se empieza a trabajar tras eso)*

| *String lectura() {  String a = "";  while (Serial.available()) {   char caracter = Serial.read();  a += caracter;  delay(2);  }  Serial.println(a);  return a; }* |
| --- |

*Esta función lee los datos del puerto serial, mientras haya caracteres disponibles. Los agrega a la variable a y luego la devuelve.*

*Como su nombre lo indica, sirve para la lectura.*

| *bool buscar(String string, int hastaDonde, String vectorComparar[], String vectorImprimir[]) {  bool encontrado = false;  for (int n = 0; n < hastaDonde; n++) {  if (string == vectorComparar[n]) {  LCD.print(vectorImprimir[n]);  encontrado = true;  break;  }  }  return encontrado; }* |
| --- |

*Como su nombre lo indica, busca una cadena en un arreglo de cadenas. Si encuentra la coincidencia, imprime el valor correspondiente en la pantalla LCD y devuelve True.*

*De no encontrarse dicho valor, no encuentra la coincidencia por ende el valor, devuelve False.*

*Es decir, que la función recorre todos los vectores hasta el índice: “HastaDonde” y de encontrarse una coincidencia en alguno de los mismos, se imprime dicho valor en la pantalla LCD y devuelve True. En caso contrario False.*

| *String buscarMorse(String caracter, int hastaDonde, String vectorComparar[], String vectorImprimir[]) {  String morse = "";  for (int n = 0; n < hastaDonde; n++) {  if (caracter == vectorComparar[n]) {  morse = vectorImprimir[n];  break;  }  }  return morse; }* |
| --- |

*Esta función funciona bastante similar a la anterior, nomás que el resultado es algo distinto. Permítanme explayarme:*

*La función busca un carácter específico en un arreglo de caracteres, y si lo encuentra, devuelve el equivalente en código Morse desde otro arreglo. De no encontrarse el carácter, devuelve una cadena vacía.*

*Yendo a lo literal al vaivén sería algo así: Si el valor de caracter coincide con el valor en VectorComparar, la función asigna a la variable morse el valor correspondiente en VectorImprimir.*

| *void principal(){  LCD.clear();  LCD.setCursor(0,0);  LCD.print("1 - Boton");  LCD.setCursor(0,1);  LCD.print("2 - Texto");  LCD.setCursor(0,0);* |
| --- |

*Esta función es bastante simple y en sí, como su nombre lo indica, la principal. Resumiendo plenamente, muestra las opciones en la pantalla LCD las cuales son:*

1. *Texto*
2. *Botón*

*Por lo que con el comando .available usado previamente, funcionará de forma correcta y sin errores.*

*Pasando por eso llegó la hora de otra de las funciones no principales porque esta lo es, pero sin dudas de las más importantes ya que son el objetivo, es decir, una de las finalidades de proyecto.*

| *void MorseATXT(){  while(!Serial.available()){  }  if (Serial.available()) {  LCD.clear();  String caracter = "";  contPos = 0;  String texto = lectura();  int longTexto = texto.length();* |
| --- |

*La función se llama MorseATXT (morse a texto) y cumple con su cometido. Primero espera los datos ingresados en el monitor serial y lee el mensaje que le llega ya traducido a código morse.*

*Luego de tal suceso, procede a escribir el texto en la pantalla LCD, mostrando así el resultado de la traducción.*

*Esta sería una función muy importante ya que recordemos que el objetivo del proyecto es hacer que traduzca de morse a texto y de texto a morse. Por lo que una función así equivale al 50 por ciento del objetivo, lo que nos lleva a la siguiente función:*

| *void TXTAMorse(){  while(!Serial.available()){  }  if (Serial.available()) {  LCD.clear();  contPos = 0;  int longMorse = 0;  String letraActual;  String texto = lectura();  texto.toUpperCase();* |
| --- |

*Si MorseATXT traducía del texto ingresado a morse, esta función es exactamente la reversa. Ahora mismo explico como funciona:*

*Convierte un texto en caracteres normales a código Morse. Luego convierte cada letra a mayúsculas, recorre el texto y convierte cada letra en Morse, mostrando el resultado en la pantalla LCD y parpadeando el LED según el código Morse traducido en el proceso.*

*Con esto ya tenemos prácticamente todo resuelto, aunque nos estaría faltando un punto clave. El botón. Después de todo las pulsaciones de un botón también es uno de los puntos claves del proyecto.*

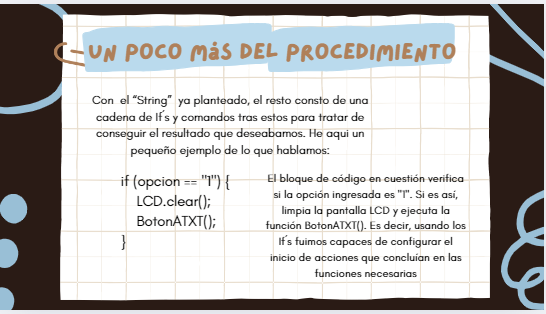
| *void BotonATXT(){  char caracter;  String texto = "";  String palabra = "";  while(true){  caracter;  float valor = pulseIn(boton, LOW, 3000000);* |
| --- |

*Esta función utiliza el estado más puro del lenguaje morse: las pulsaciones. El botón envía señales cada vez que es pulsado, estas pulsaciones son medidas en tiempo cuales son traducidas a puntos (.) o raya (-) las letras del morse.*

*Estas mismas rayas son comparadas en los String con las letras de la misma forma que se mostró con anterioridad, por lo que son traducidas y enviadas a la pantalla LCD para que muestre el resultado.*

*Ese sería el código completo, claro exceptuando algunos detalles. Solo me centré en mostrar lo más importante del código final. Pasando desde la configuración inicial, hasta las funciones principales las cuales dan vida al proyecto.*

*Cada parte del código tiene un propósito claro, ya sea para leer la entrada serial, convertir texto a Morse, o manejar las señales del botón, todo concluye en las tareas del proyecto.*

**

*La diapositiva actualizada quedó así. Utilice la siguiente linea de codigo como ejemplo:*

| *if (opcion == "1") { LCD.clear(); BotonATXT(); }* |
| --- |

*Me pareció apropiada debido a lo básico que resultaba. Con lo acumulado se entiende que el proyecto se basó en una series de comandos definidos por If´s para accionarse y conseguir lo que desearamos y un String para definir los valores de cada letra.*

*Por supuesto al cambiar partes del texto. Sus fuentes también cambiaron. Por lo que les mostraré en este momento, las nuevas fuentes de los textos trastocados durante este error.*

*Texto del código:*

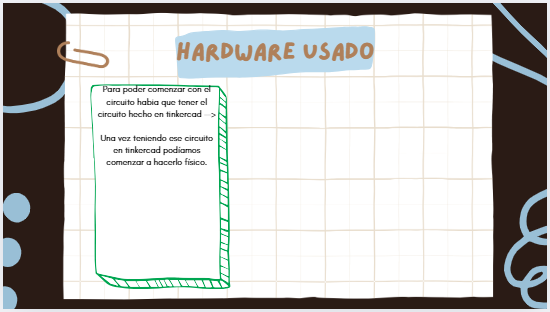
*Fuente Glacial Indifference 38,5. Color negro simple.*

*Texto explicativo sobre el código:*

*Fuente Glacial Indifference 30,1. Color negro simple.*

*Luego continúe con la idea de otra diapositiva, pensé por un rato pero llegué a una única conclusión. La siguiente debería ser sobre los componentes utilizados. Por razones de tiempo y comodidad no pude desarrollarlo, solo llegué a colocar el título con la idea.*

*Supe que Fleitas lo encontró y se dispuso a trabajarlo, recuerdo haber leído mensajes suyos al respecto. Al chequear su trabajo encontré lo siguiente:*

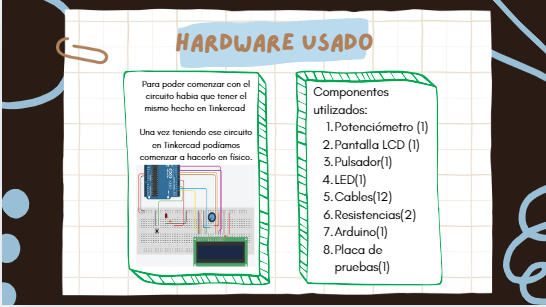
**

*Me pareció francamente bastante escaso para lo que podría ser. Solo un texto con la siguiente fuente:*

*Glacial Indifference 23,5. Color negro simple.*

*No me parecía mal para comenzar, pero sinceramente una breve explicación del porqué del planteo en Tinkercad no me parecía tan completo. Ademas de que note algunos errores en la ortografía, relacionados a la repetición constante de la palabra “circuito” en el texto de la diapositiva*

*Haciéndola corta, lo complete un poco.*

**

*Me centré en dar una demostración básica de cómo terminó el modelo del Tinkercad. Solo achique una imagen del mismo y la coloque abajo para mostrarla.*

*También sume una lista de los componentes usados en el mismo Tinkercad, así como la cantidad de los mismos entre paréntesis. Me acordé de los profesores de física preguntando. 300 que? y pensé que se me preguntaría. ¿Cuántos cables? razón por la que agregue la cantidad. Esto lo puse dentro de un recuadro que es el mismo que el de la izquierda, esto se debe a que solo lo copie y lo di vuelta para que no quedara medio croto.*

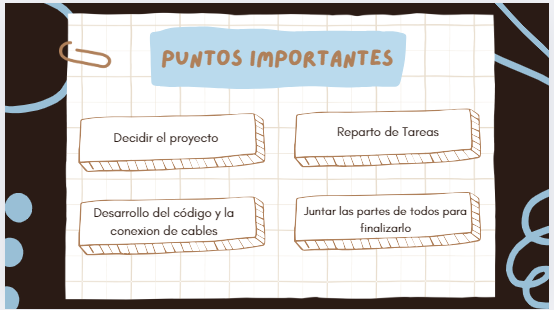
*También me tomé la molestia de corregir los errores de ortografía. Sustituyendo la palabra “circuito” hasta en dos ocasiones.*

*Fuente de la lista de componentes utilizados:*

*Glacial Indifference 33,3.. Color negro simple.*

*Trate de que en todo momento se entienda el Hardware que utilizamos sin complicar mucho al espectador, pues nombrar todas las utilidades de cada uno podría ser agobiante para el mismo, razón por la que trate de que sea mucho más ilustrativo. Cosa que una lista en tal formato representaría muy bien.*

*Lo siguiente fue resaltar los puntos importantes del proyecto. Me salté dos diapositivas (una para el modelo 3d) y otra que todavía no sé para qué voy a usar). Pero como ya venía predeterminado, me pareció correcta una diapositiva resaltando los puntos importantes.*

**

*Use los siguientes puntos para tener en cuenta:*

1. *Decidir el proyecto (en nuestro caso fue el traductor de código morse)*
2. *Reparto de Tareas (lo básico de la organización)*
3. *Desarrollo del código y conexión de cables (sería todo lo que rodea al proceso del proyecto)*
4. *Juntar las partes de todos para finalizarlo (incluye la creación del Github y la adaptación del modelo al juntar las partes de todos)*

*Fuente del título:*

*Lazy Dog 59. Color pastel.*

*Fuente del cuadro de arriba a la izquierda:*

*Glacial Indifference 34. Color negro simple.*

*Fuente del cuadro arriba a la derecha:*

*Glacial Indifference 34. Color negro simple.*

*Fuente del cuadro abajo a la izquierda:*

*Glacial Indifference 34. Color negro simple.*

*Fuente del cuadro abajo a la derecha:*

*Glacial Indifference 32,2. Color negro simple.*

*Todos los textos están rodeados por cuadros que ya venían puestos en la plantilla, por lo que solo bastó con que colocará el texto en cada uno para finalizar esta diapositiva sin problemas*

*Luego de un rato de pensar cómo rellenar la diapositiva restante. Tuve una idea: las posibles situaciones donde poner en práctica el proyecto. Es decir, las posibles aplicaciones del mismo.*

*Ya tenía algunas ideas pero aún así simplemente decidí googlear cuales serían las posibles aplicaciones del lenguaje morse en situaciones reales, también use ChatGPT para dicha tarea, puesto que me resultaría mucho más fácil recabar información de dicha manera.*

*Puse en común lo recabado por ChatGPT y por mí durante la recolecta de información para decidirme por las siguientes tres aplicaciones: Temas de discapacidad y problemas de comunicación en lugares desolados u hostiles.*

*Llegué a tres conclusiones con esos dos temas:*

1. *Tormentas marítimas (En caso de estar en medio del mar y en una tormenta. las ondas de radio no actúan con normalidad, permitiendo mayoritariamente solo pitidos, mismos que servirían para comunicarse a través del lenguaje morse)*
2. *Problemas de comunicación (resulta que para algunos discapacitados con problemas de audición o habla, se usa mucho el lenguaje morse como alternativa. Por lo que un traductor de tal lenguaje, sería más que recibido entre esta comunidad)*
3. *Errores en la comunicación espacial (Suponiendo de que el equipamiento de alguna estación espacial sufra desperfectos, el lenguaje morse sería una solución simple y efectiva para estos casos)*

**

*Fuente del título:*

*LazyDog 92. Color pastel.*

*Fuente del cuadro de la izquierda:*

*Glacial Indifference 24,5. Color negro simple.*

*Fuente del cuadro del medio:*

*Glacial Indifference 25. Color negro simple.*

*Fuente del cuadro de la derecha:*

*Glacial Indifference 24,5. Color negro simple.*

*Fuente de la imagen del barco:*

[*https://www.google.com/imgres?q=barco%20en%20una%20tormenta%20en%20el%20mar&imgurl=https%3A%2F%2Fimg.freepik.com%2Ffotos-premium%2Fbarco-medieval-lucha-contra-feroz-tormenta-mar-olas-rompiendo-relampagos-medio-dar\_434911-89.jpg%3Fsemt%3Dais\_hybrid&imgrefurl=https%3A%2F%2Fwww.freepik.es%2Ffotos-vectores-gratis%2Fbarco-tormenta&docid=UhmaeS85g7tl4M&tbnid=wi-EVKR7NqrVqM&vet=12ahUKEwjOidm8qIeKAxVtppUCHbVEH6gQM3oECBsQAA..i&w=626&h=417&hcb=2&ved=2ahUKEwjOidm8qIeKAxVtppUCHbVEH6gQM3oECBsQAA*](https://www.google.com/imgres?q=barco%20en%20una%20tormenta%20en%20el%20mar&imgurl=https%3A%2F%2Fimg.freepik.com%2Ffotos-premium%2Fbarco-medieval-lucha-contra-feroz-tormenta-mar-olas-rompiendo-relampagos-medio-dar_434911-89.jpg%3Fsemt%3Dais_hybrid&imgrefurl=https%3A%2F%2Fwww.freepik.es%2Ffotos-vectores-gratis%2Fbarco-tormenta&docid=UhmaeS85g7tl4M&tbnid=wi-EVKR7NqrVqM&vet=12ahUKEwjOidm8qIeKAxVtppUCHbVEH6gQM3oECBsQAA..i&w=626&h=417&hcb=2&ved=2ahUKEwjOidm8qIeKAxVtppUCHbVEH6gQM3oECBsQAA)

*Fuente del icono de discapacidad auditiva:*

[*https://www.google.com/imgres?q=discapacidad%20auditiva%20de%20comunicacion%20icono&imgurl=https%3A%2F%2Fcdn-icons-png.flaticon.com%2F256%2F8984%2F8984822.png&imgrefurl=https%3A%2F%2Fwww.flaticon.es%2Ficono-gratis%2Fla-discapacidad-auditiva\_8984822&docid=\_fWFDmlMPbYtAM&tbnid=5L1oCHlYlItgZM&vet=12ahUKEwj0g6ilqYeKAxXsLbkGHS1PEMgQM3oECBgQAA..i&w=256&h=256&hcb=2&ved=2ahUKEwj0g6ilqYeKAxXsLbkGHS1PEMgQM3oECBgQAA*](https://www.google.com/imgres?q=discapacidad%20auditiva%20de%20comunicacion%20icono&imgurl=https%3A%2F%2Fcdn-icons-png.flaticon.com%2F256%2F8984%2F8984822.png&imgrefurl=https%3A%2F%2Fwww.flaticon.es%2Ficono-gratis%2Fla-discapacidad-auditiva_8984822&docid=_fWFDmlMPbYtAM&tbnid=5L1oCHlYlItgZM&vet=12ahUKEwj0g6ilqYeKAxXsLbkGHS1PEMgQM3oECBgQAA..i&w=256&h=256&hcb=2&ved=2ahUKEwj0g6ilqYeKAxXsLbkGHS1PEMgQM3oECBgQAA)

*Fuente de la foto de la estación espacial:*

[*https://www.google.com/imgres?q=problemas%20de%20comunicacion%20en%20una%20estacion%20espacial%20icono&imgurl=https%3A%2F%2Fcloudfront-us-east-1.images.arcpublishing.com%2Fartear%2FKATCPVJJC4RLYNXIJTV2FPQ7HY.jpg&imgrefurl=https%3A%2F%2Ftn.com.ar%2Ftecno%2Fciencia%2F2023%2F07%2F27%2Fla-estacion-espacial-internacional-sufrio-una-perdida-total-de-comunicacion-con-la-tierra%2F&docid=0ZvkL-qS9M8q6M&tbnid=0lFz26GgrKWtjM&vet=12ahUKEwjZjaaiqoeKAxXqr5UCHZvONMwQM3oECCwQAA..i&w=1366&h=764&hcb=2&ved=2ahUKEwjZjaaiqoeKAxXqr5UCHZvONMwQM3oECCwQAA*](https://www.google.com/imgres?q=problemas%20de%20comunicacion%20en%20una%20estacion%20espacial%20icono&imgurl=https%3A%2F%2Fcloudfront-us-east-1.images.arcpublishing.com%2Fartear%2FKATCPVJJC4RLYNXIJTV2FPQ7HY.jpg&imgrefurl=https%3A%2F%2Ftn.com.ar%2Ftecno%2Fciencia%2F2023%2F07%2F27%2Fla-estacion-espacial-internacional-sufrio-una-perdida-total-de-comunicacion-con-la-tierra%2F&docid=0ZvkL-qS9M8q6M&tbnid=0lFz26GgrKWtjM&vet=12ahUKEwjZjaaiqoeKAxXqr5UCHZvONMwQM3oECCwQAA..i&w=1366&h=764&hcb=2&ved=2ahUKEwjZjaaiqoeKAxXqr5UCHZvONMwQM3oECCwQAA)

*Solo resta que me pasen el modelo 3D del proyecto para añadirlo a la diapositiva restante.*

*Por mi parte de momento realizaré el final de la presentación. Con eso debería haberla finalizado, por supuesto les pediré opinión a mis compañeros al respecto. Puesto que no quiero haberme equivocado al respecto y entregarlo sin miramientos.*

*Antes de esa tarea les agregue animación a todas las hojas de las diapositivas: Paneo al entrar y al salir.*

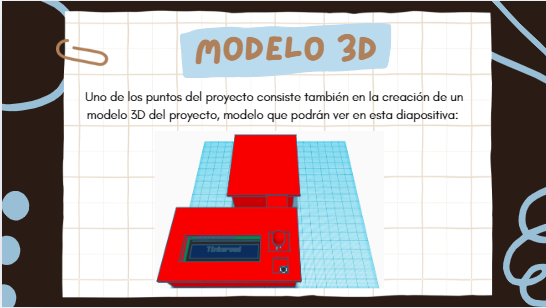
**

*Luego de diseñar el final por un tiempo así quedó. Exceptuando el título, todo está hecho con la sección “elementos” de Canva.*

*Fuente del título:*

*LazyDog 124. Color pastel*

*Mi compañera, Figallo, ya me pasó el modelo 3d, por lo que procederé a realizar la última diapositiva.*

**

*Un pequeño texto describiendo el porqué de tal punto, adjunto de una foto .PNG del modelo 3D se logran ver en esta diapositiva, la cual, era la última que faltaba completar.*

*Fuente el título:*

*LazyDog 92. Color pastel.*

*Fuente del texto introductorio:*

*Glacial Indifference 33,4. Color negro simple.*