**Proceso instalación Visual Studio Code**

**Paso 1:** Escribimos en Google Visual Studio Code y seleccionamos donde dice “Download”.



**Paso 2:** Seleccionamos el sistema operativo que tenemos y lo descargamos.



**Paso 3:** Al darle clic nos descargará un .exe, al cual le daremos clic encima.



**Paso 4:** Lee y acepta el acuerdo de licencia. Haz clic en Next para continuar.



**Paso 5:** Puedes cambiar la ubicación de la carpeta de instalación o mantener la configuración predeterminada. Haz clic en Next para continuar.



**Paso 6:** Elige si deseas cambiar el nombre de la carpeta de accesos directos en el menú Inicio o si no deseas instalar accesos directos en absoluto. Haz clic en Next.



**Paso 7:** Selecciona las tareas adicionales, por ej. crear un icono en el escritorio o añadir opciones al menú contextual de Windows Explorer. Haz clic en Next.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 8:** Haz clic en Install para iniciar la instalación.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 9:** El programa está instalado y listo para usar. Haz clic en Finish para finalizar la instalación y lanzar el programa.



**Creación de archivos y carpetas**

**Paso 1:** Creamos una carpeta raíz. En nuestro caso la llamaremos “iatriki”.

Texto

Descripción generada automáticamente

**Paso 2:** Dentro de esta, crearemos otras dos carpetas llamadas “css”, la cual contendrá todo lo estético de nuestra página y otra llamada “js” la cual contendrá la lógica de nuestra página.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Tarjeta**

**Paso 1:** Para abrir el Visual Studio Code, haremos lo siguiente: Dentro de la carpeta raíz, en la barra superior, escribiremos cmd y le damos enter.

Interfaz de usuario gráfica

Descripción generada automáticamente

**Paso 2:** Eso nos abrirá una terminal, solo tendremos que escribir “code .”.

Captura de pantalla de computadora

Descripción generada automáticamente

**Paso 3:** Esto nos abrirá el visual studio code. Ahora, ya que estamos dentro, crearemos un archivo llamado “index.html” danle click derecho en la barra de la izquierda y luego dale “new file” o “nuevo archivo”

Captura de pantalla de un celular

Descripción generada automáticamente

**Paso 4:** Llamaremos el archivo “index.html”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 5:** Una vez en el archivo recién creado, colocamos el código “!” y escogemos la primera opción.

Interfaz de usuario gráfica

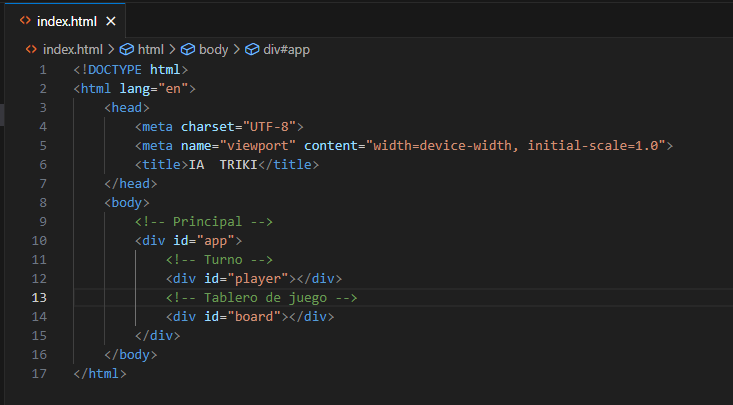
Descripción generada automáticamente

**Paso 6:** Una vez colocada la estructura, cambiaremos el titulo (línea 6)

Captura de pantalla de un celular

Descripción generada automáticamente

**Paso 7:** Ahora dentro del body crearemos 1 div principal para nuestra página (Linea 10) le pondremos un id llamado “app”, y además crearemos otro div dentro de ese mismo div (línea 12) y le pondremos el id “player” ya que será el que nos diga de quien es turno, y por ultimo otro div con el id “board” (línea 14), el cual tendrá nuestro tablero.



**Paso 8:** Dentro la carpeta “js” vamos a crear un nuevo archivo llamado “app.js”

Pantalla de computadora con letras

Descripción generada automáticamente con confianza mediaCaptura de pantalla de un celular

Descripción generada automáticamente

**Paso 9:** Ahora vamos a vincular nuestro javascript en el Html (línea 18)



**Paso 10:** Una vez vinculado el javascript iremos a nuestro archivo “app.js” y crearemos un arreglo para el tablero (línea 2 hasta la 6) y crearemos otra variable llamada “turn” el cual servirá para saber de quién es el turno

Pantalla de un video juego

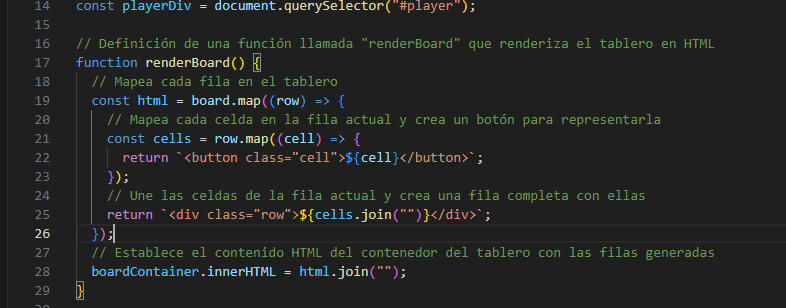
Descripción generada automáticamente con confianza media

**Paso 11:** Ahora vamos a seleccionar los elementos de nuestro html y los almacenaremos en variables con el siguiente código: const (nombre de la constante) = document.querySelector(‘#nombre de id’);. Usamos el “#” porque los estamos capturando de un “id” capturamos el tablero (línea 12) y también el turno del jugador (línea 14)

Texto

Descripción generada automáticamente

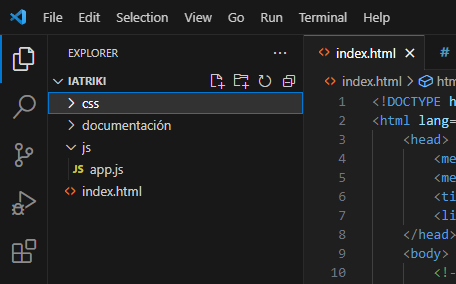
**Paso 12:** Ahora vamos a crear una función para renderizar el tablero. La llamaremos “renderBoard”, y vamos a mapear las celdas(línea 19) y cear un botón por cada celda (línea 21 a 23) y vamos a guardar esas celdas en una constante llamada “cells” después de crear los botones para cada celda en la fila, se utiliza el método join("") para unir todos los botones en un solo string (línea 25), y por ultimo actualizamos nuestro tablero del html con la información que le acabamos de poner. (línea 28).



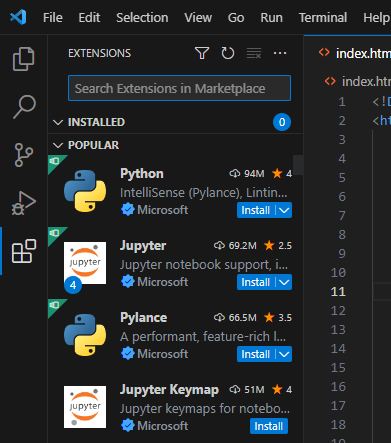
**Paso 13:** Ahora llamamos la función que acabamos de crear.

Interfaz de usuario gráfica, Aplicación

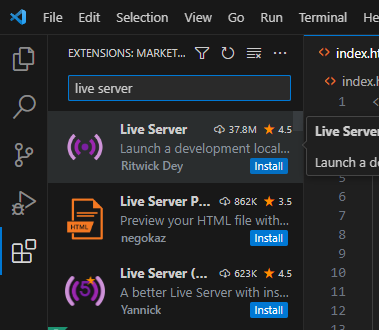
Descripción generada automáticamente

**Paso 14:** Vamos a visualizar cómo está quedando para esto necesitamos ir a la opción de abajo a la izquierda para instalar una extensión de el visual studio. 

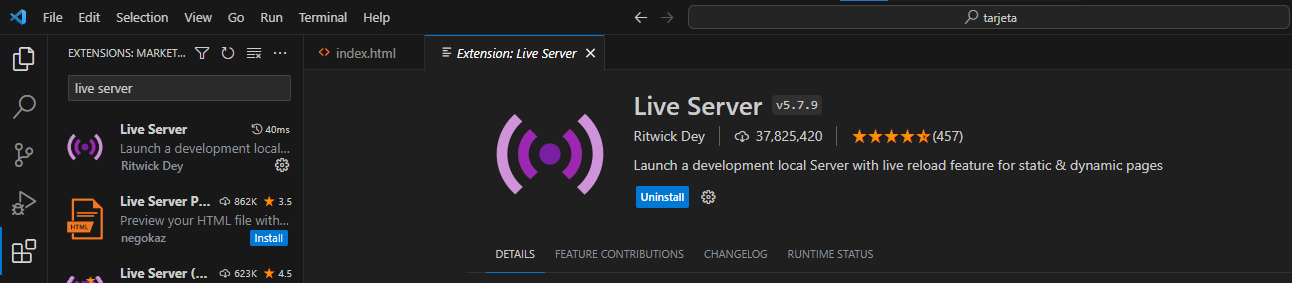
**Paso 15:** Nos saldrá esta pestaña y buscaremos live server, en la barra de búsqueda.

****

**Paso 16:** Ahora que encontramos lo que necesitamos, le damos click a “install”

****

**Paso 17:** Ahora que tenemos instalado el live server nos debería de aparecer algo así donde podremos volver a nuestro proyecto dando click al explorador.

****

**Paso 18:** Una vez te sale el explorador volvemos al Index.html

**Captura de pantalla de un celular

Descripción generada automáticamente**

**Paso 19:** Ahora vamos a dar click derecho en cualquier parte del código y damos click en la opción “Open with live server”

Texto

Descripción generada automáticamente

**Paso 20:** Así esta quedando nuestro pequeño proyecto, podemos mirar que ya esta generado el tablero. Ahora vamos a meterle diseño.

Forma, Rectángulo

Descripción generada automáticamente

**Paso 21:** Lo primero que haremos será volver a nuestro código y vamos a crear un nuevo archivo dentro de la carpeta css, lo llamaremos “style.css”

Captura de pantalla de computadora

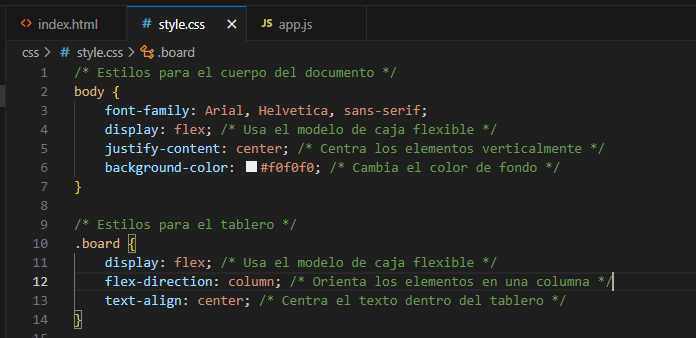
Descripción generada automáticamente

**Paso 22:** Una vez creado el archivo css vamos a vincularlo a nuestro html (línea 7).

Captura de pantalla de computadora

Descripción generada automáticamente

**Paso 23:** Comenzamos con el diseño del body para centrar su contenido y también le ponemos una fuente. También diseñamos el tablero con él “.board” centrando su texto.



**Paso 24:** Ahora ponemos el “display: flex;” a la filas del tablero, y también cambiamos el diseño de las celdas para ponerle un tamaño de fuente y ponerlo en negrita.

Texto

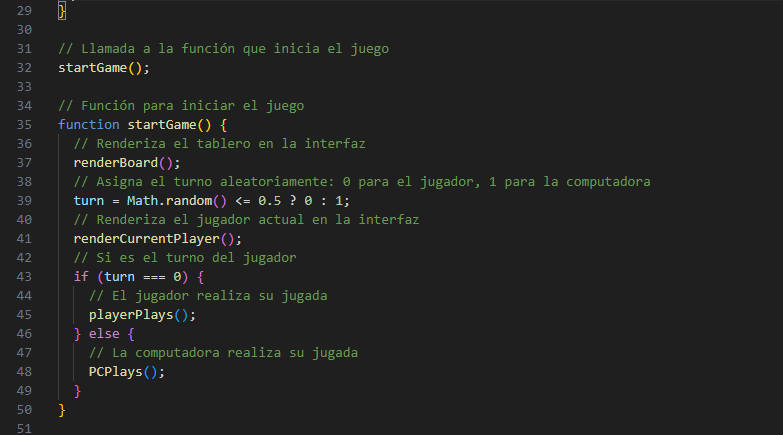
Descripción generada automáticamente

**Paso 25:** Ahora miramos como está quedando nuestro proyecto (paso 19).

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

**Paso 26:** Una vez miramos como está quedando nuestro proyecto vamos a crear una nueva función en el archivo “app.js” y llamamos una función llamada “startGame” y la creamos además cuando llamamos la función “renderBoard” (Paso 13), vamos a meterlo en la nueva función, y llamamos la función ahí (línea 37), también llamaremos la constante “turn” y vamos a generar un numero aleatorio entre el 0 y el 1 para que se genere el turno (línea 39) y vamos a llamar una función llamada “renderCurrentPlayer” (línea 41) la cual va a mostrar arriba del tablero de quien es turno, y por ultimo crearemos una condicional (línea 43) la cual en caso de que la constante “turn” sea igual a 0 llame a la función para que el jugador pueda jugar, de lo contrario llama a la función para que la computadora juegue.



**Paso 27:** Ahora vamos a crear la función que mencione anteriormente llamada “renderCurrentPlayer” la cual lo único que hará será cambiar nuestro texto de arriba del tablero para que muestre de quien es turno (línea 55)

Texto

Descripción generada automáticamente

**Paso 28:** Ahora aleatoriamente al cargar la página (paso 19) va a colocar arriba quien comienza a jugar.

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

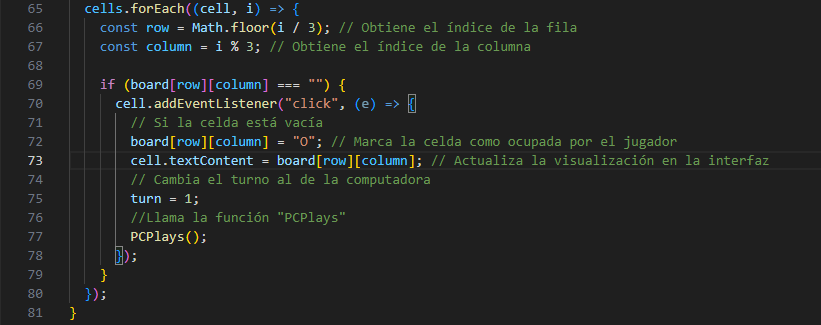
**Paso 29:** Ahora crearemos la función del jugador.

Lo primero será crear la constante “cells” y agarrará todos los elementos con la clase.” cell” dentro del tablero (línea 63), ahora creamos un foreach que calcule la fila y la celda actual basándose en su índice (línea 65 a 69).

Texto

Descripción generada automáticamente

**Paso 30:** Ahora vamos a crear una condicional de que en el caso de que la columna y la fila este vacia, añada un evento “click” y que coloque un “O” y que se muestre en el tablero (línea 73), cambiamos la constante “turn” a 1 y llamamos la función que hace que pueda jugar el computador



**Paso 31:** Ahora podremos mirar que podemos ir añadiendo nuestros símbolos en nuestro turno.

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

**Paso 32:** Ahora creamos una función para que en caso de empate se muestre.

Texto

Descripción generada automáticamente

**Paso 33:** Ahora vamos a comenzaremos a crear la inteligencia de nuestra máquina.

Comenzamos creando una función llamada “checkIfCanWin” y crearemos una constante llamada “arr” que usando el método JSON.parse(), el cual se utiliza para analizar una cadena JSON y crear un objeto JavaScript a partir de ella. Y dentro usamos el método “JSON.stringify” el cual se utiliza para convertir un objeto (en este caso, el arreglo board) en una cadena JSON. Esto toma todos los elementos del arreglo, los convierte en su representación JSON y los concatena en una cadena.

Texto

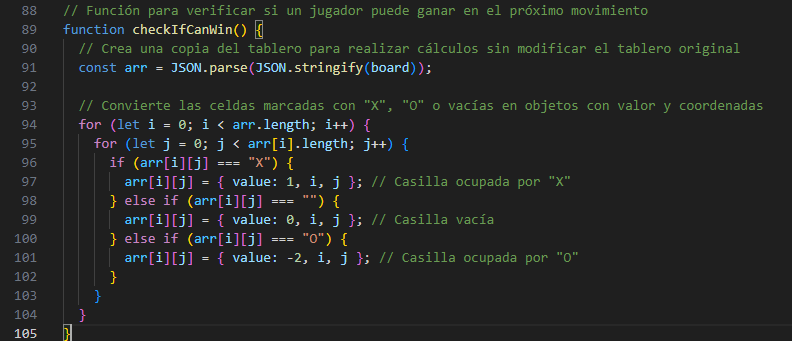
Descripción generada automáticamente

**Paso 34:** Ahora crearemos un for para que itere por todas las casillas y les de un valor según si la casilla tiene una “X”, “O” o este vacia.

En caso de que la casilla tenga una “X” va a darle un valor de 1. (Línea 96 y 97)

En caso de que la casilla este vacía va a darle un valor de 0. (Línea 98 y 99)

En caso de que la casilla tenga una “O” va a darle un valor de -2. (Línea 100 y 101)



**Paso 35:** Ahora crearemos constantes para saber la ubicación de cada casilla del tablero.

Pantalla de computadora con letras

Descripción generada automáticamente

**Paso 36:** Crearemos arrays para cada forma de ganar en el juego.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

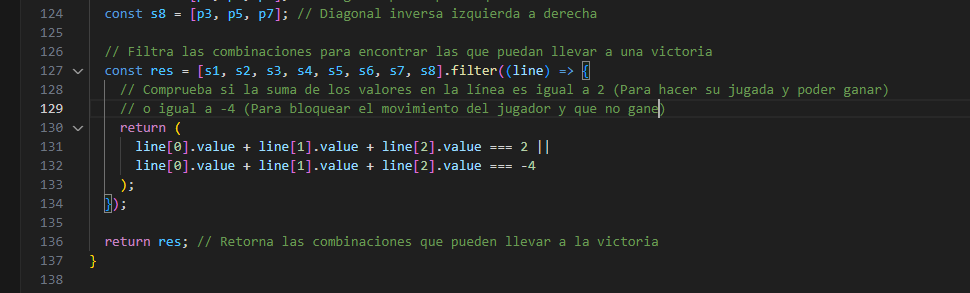
**Paso 37:** Así debe de estar.

Texto

Descripción generada automáticamente

**Paso 38:** Ahora crearemos una constante que mire que filtre todas las combinaciones para encontrar las que llevan a la victoria (línea 127) y va a sumar los valores y que en caso de que el valor sumado sea “2” significa que es 1 + 1 (Paso 34) y significa que podrá ganar y pondrá el movimiento en la casilla para ganar (línea 131) o si por el contrario la suma es “-4” ósea -2 -2, significa que el jugador va a ganar y por ello la maquina lo bloqueará (línea 132)

Y por último retorna las combinaciones (línea 136)



**Paso 39:** Ahora comenzamos con la función del movimiento de la computadora con una llamándola “PCPlays” y llamaremos la función “renderCurrentPlayer” para que se muestre que es turno de la máquina.

Y vamos a colocar un “setTimeOut” para el tiempo que la máquina tendrá para jugar.

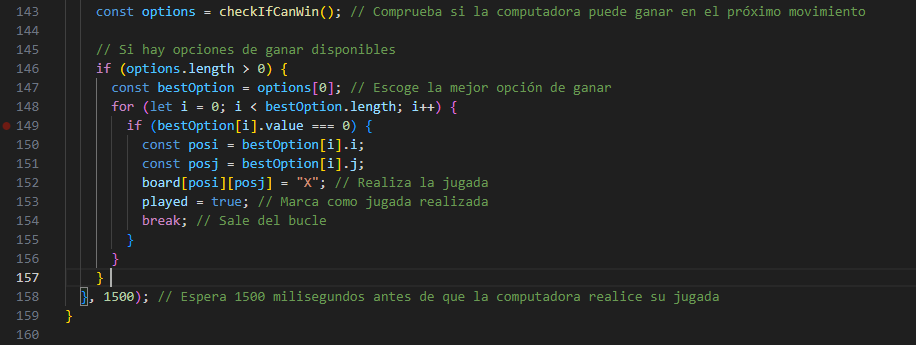
Dentro colocaremos una variable que controla si la computadora ya jugó o no (línea 142) y otra para comprobar si la computador puede ganar en su próximo movimiento gracias a la función que antes creamos (línea 143)

Texto

Descripción generada automáticamente

**Paso 40:** Ahora crearemos una condicional (línea 146) y mirará si hay opciones para ganar. En caso de que haya va a escoger la mejor opción (línea 147)

Creando un “for” (línea 148) que en caso de que el valor de la mejor opción sea “0” (paso 34) ósea este vacío, ubique su posición en la fila (línea 150) y en la columna (línea 151) y coloque una “X” (línea 152) y también cambiará la constante (played) a true para que se sepa que ya la maquina jugó(línea 153), además saldrá del bucle (línea 154) y por último, colocamos lo que tardará la maquina en hacer todo este proceso (línea 158)



**Paso 41:** De lo contrario si no hay opción buena realiza una jugada aleatoria y va a colocar una “X” y cambiará su estado a que ya jugo (línea 163)

Texto

Descripción generada automáticamente

**Paso 42:** Ahora vamos a cambiar la variable “turn” a 0 para que se sepa que es el turno del jugador (línea 169) y llamamos la función “renderBoard”(línea 170) además actualizamos lo que se muestra de los turnos llamando la función “renderCurrentPlayer” (línea 171) y por último llamamos la función para que pueda jugar el jugador “playerPlays” (línea 172)

**Texto

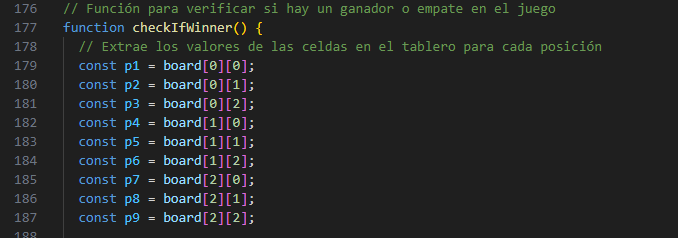
Descripción generada automáticamente**

**Paso 43:** Ahora podremos mirar nuestro proyecto y vemos que está funcionando bien cambiando de turno y la maquina con cierta inteligencia. (Paso 19)

**Imagen que contiene Gráfico

Descripción generada automáticamente**

**Paso 44:** Ahora vamos a crear una función para saber si alguien ya gano creando variables para saber las ubicaciones del tablero. (línea 179 a 187)

****

**Paso 45:** También vamos a definir los patrones de las victorias. (línea 190 a 199)

**Texto

Descripción generada automáticamente**

**Paso 46:** Ahora vamos a iterar en los patrones de victoria para ver si alguno se cumple (línea 202) y en caso de que alguno se cumpla, verifica si todas las casillas en el patrón son “X” (línea 204) y en caso de que así sea va a actualizar el “playerDiv” para que muestre un mensaje que diga que gano el computador (línea 205) y retorna un string (línea 206).

De lo contrario si todas las casillas en el patrón son “O” (línea 209) actualiza el “playerDiv” a que gano el jugador (línea 210).

**Pantalla de computadora con letras

Descripción generada automáticamente con confianza media**

**Paso 47:** Y ahora crearemos una nueva condicional para saber si todas las casillas están llenas y no hay ningún patrón que se cumpla (línea 216) en caso de que así sea llama a la función “renderDraw” para que muestre un empate(línea 217) y retornará un string “draw” (línea 218) y por último fuera de esa condicional va a retornar un “none” en caso de que aun no haya empate ni ganador aún

**Pantalla de computadora con letras

Descripción generada automáticamente con confianza media**

**Paso 48:** Ahora iremos a la función “PCPlays” (Paso 42) borraremos donde llamamos la función “playerPlays” y crearemos una constante llamada “won” y comprobará si hay un ganador llamando la función “checkIfWinner” (línea 173) y en caso de que lo que se le retorno es “none” (línea 175) significa que aún no hay ganador ni empate, por lo tanto llama a la función para que pueda jugar el jugador que borramos arriba (línea 176)y en caso de que se le retorno el string “draw” (línea 180) significa que ya se empato y llama a la función “renderDraw” (línea 181)

**Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media**

**Paso 49:** Ahora iremos a la función “playerPlays” (paso 30) y vamos a borrar donde llamaos la función “PCPlays” y crearemos una constante llamada “won” y comprobará si hay un ganador llamando la función “checkIfWinner” (línea 77) y en caso de que lo que se le retorno es “none” (línea 79) significa que aún no hay ganador ni empate, por lo tanto llama a la función para que pueda jugar el computador que borramos arriba (línea 81)

**Texto

Descripción generada automáticamente**

**Paso 50:** Por último creamos otra condicional de que en caso de que “won” sea igual a “draw” significa que hay un empate (línea 86) llamando a la función “renderDraw” (línea 88) y remueve el evento clic para que el jugador ya no pueda hacer más jugadas (línea 90 a 93)

**Captura de pantalla de computadora

Descripción generada automáticamente**

**Paso 51:** Por último, así quedo nuestro proyecto, con nuestra maquina teniendo cierta inteligencia, y sabiendo si es un empate o una victoria para el jugador/maquina

**Imagen que contiene Rectángulo

Descripción generada automáticamenteImagen que contiene Diagrama

Descripción generada automáticamenteImagen que contiene Gráfico

Descripción generada automáticamente**