Vamos a crear un juego de memoria.

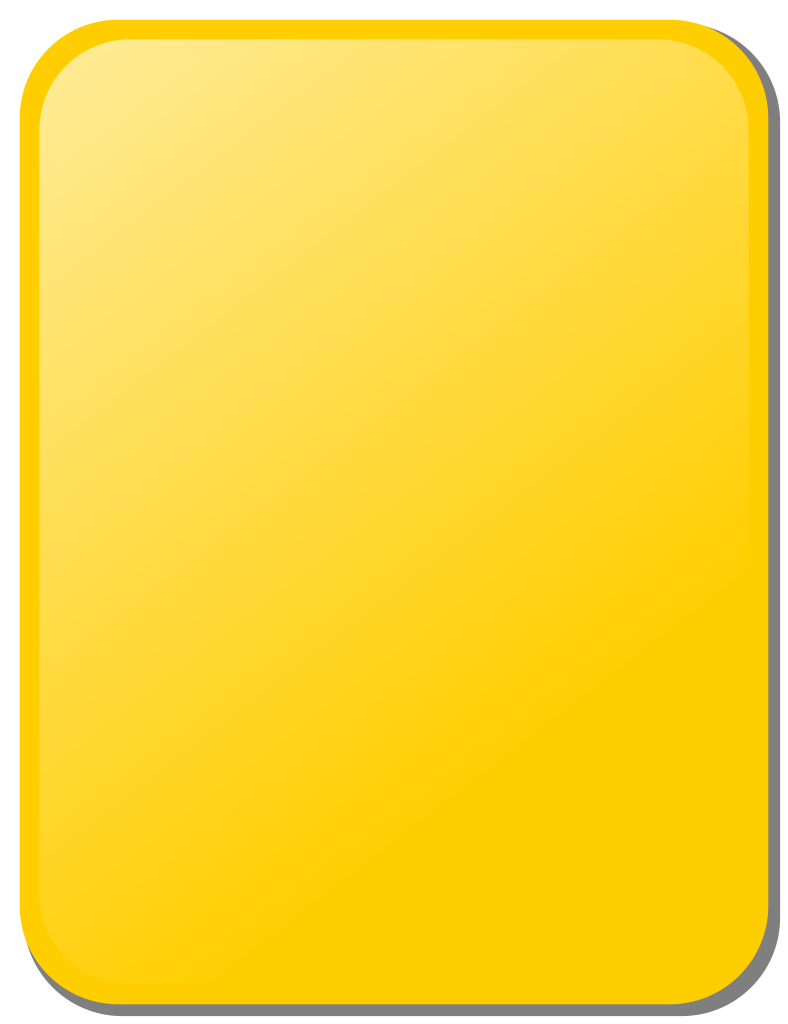
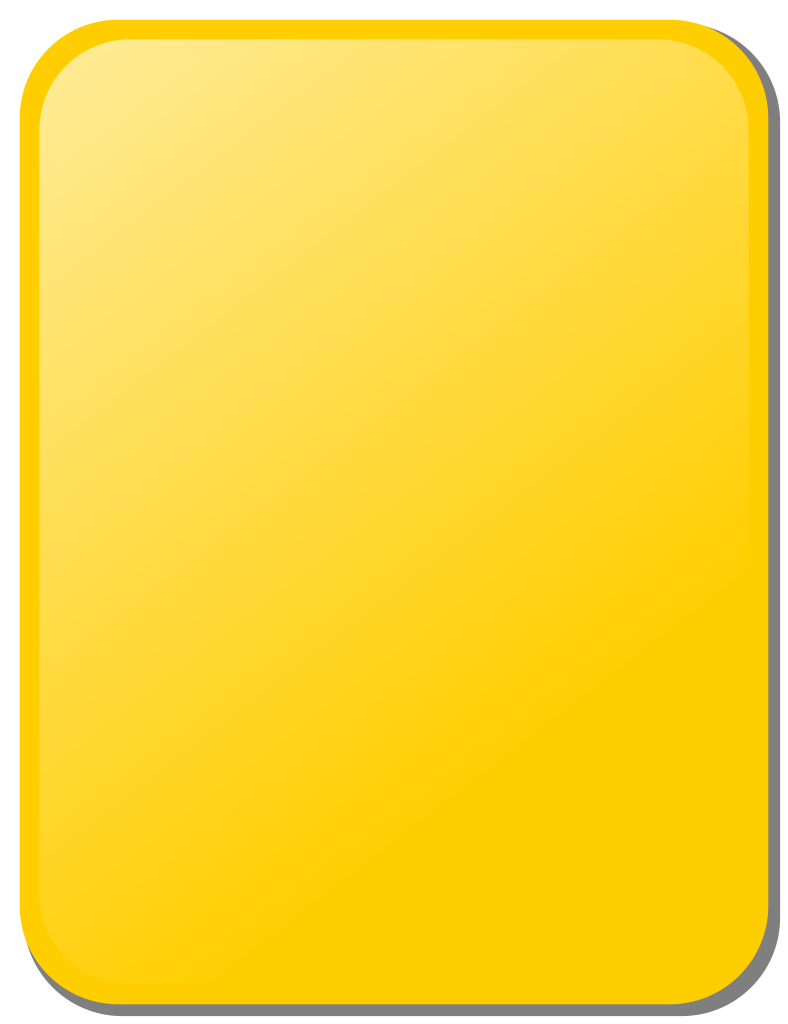
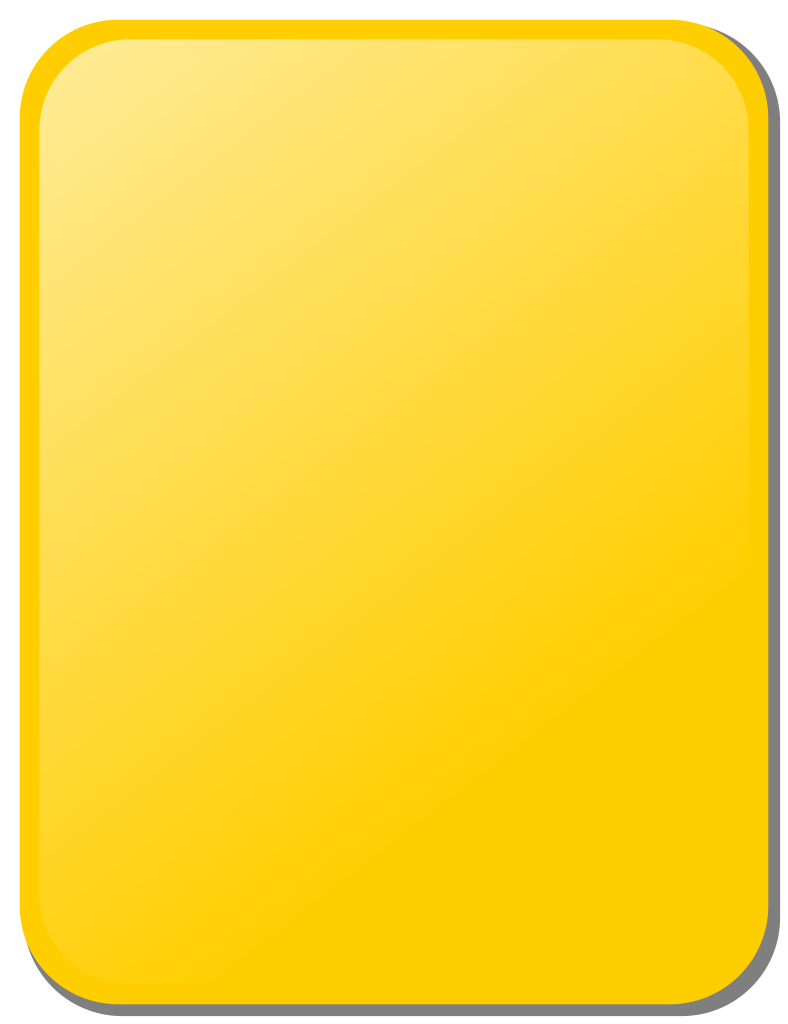
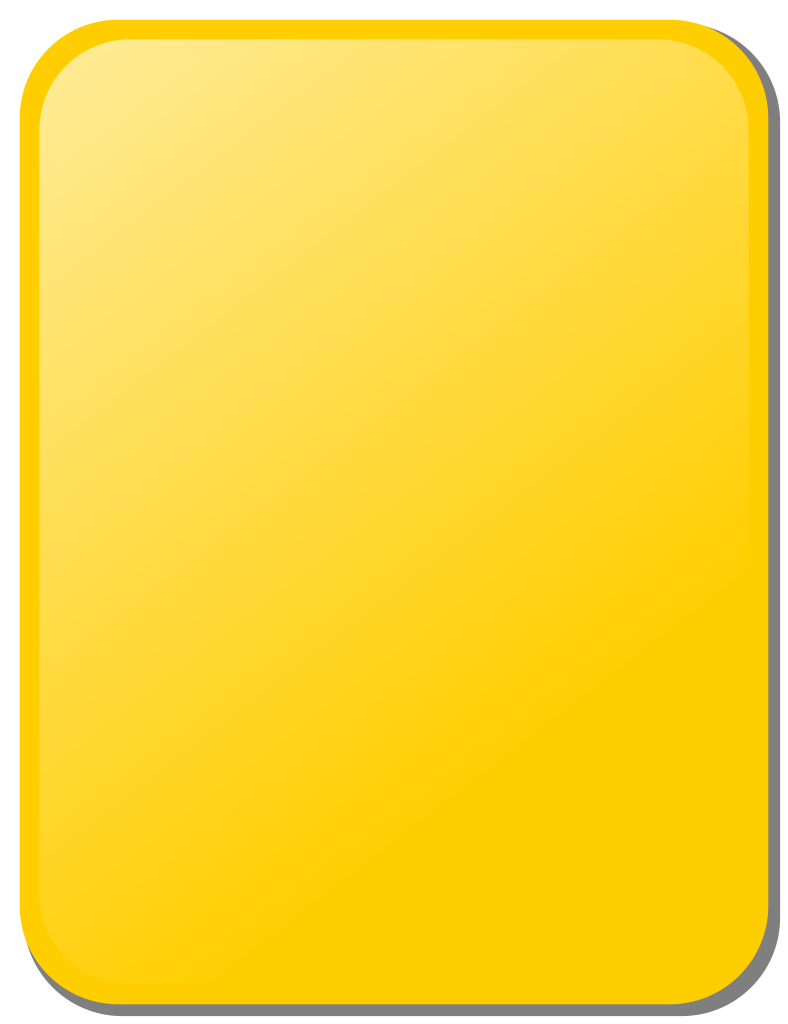
* El juego debe ser adaptado a móviles para ello:
  + Debe usarse un diseño responsive
  + Debe usar únicamente tecnologías **HTML** y **JAVASCRIPT/YPESCRIPT**

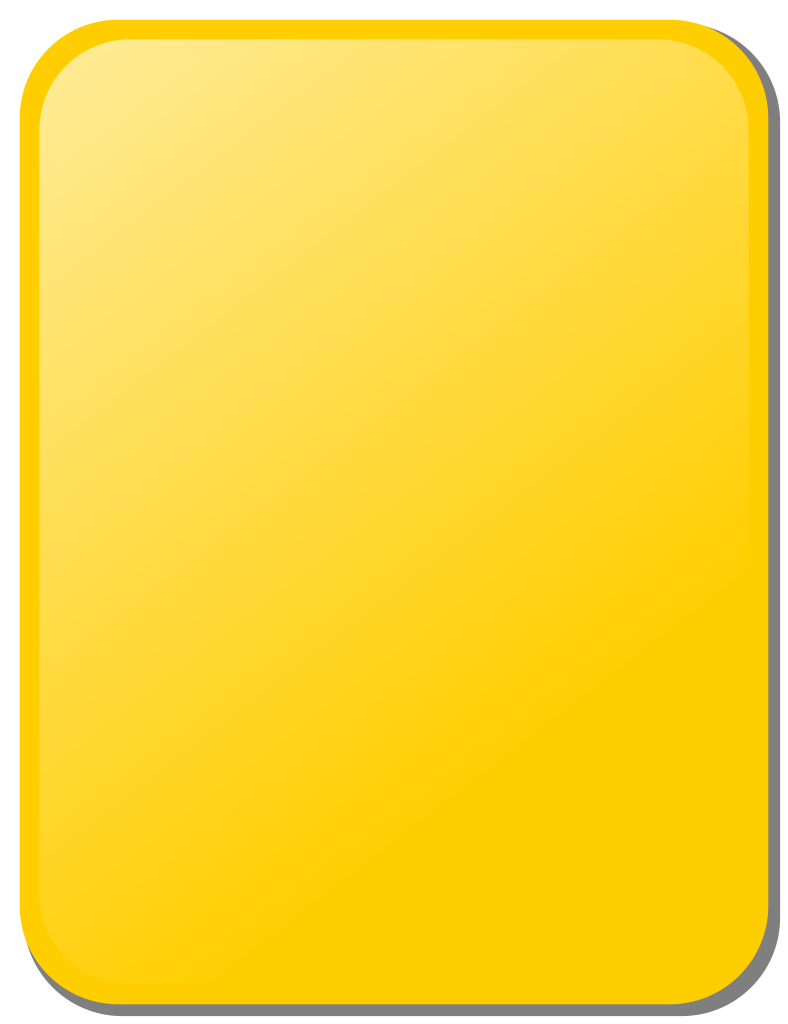
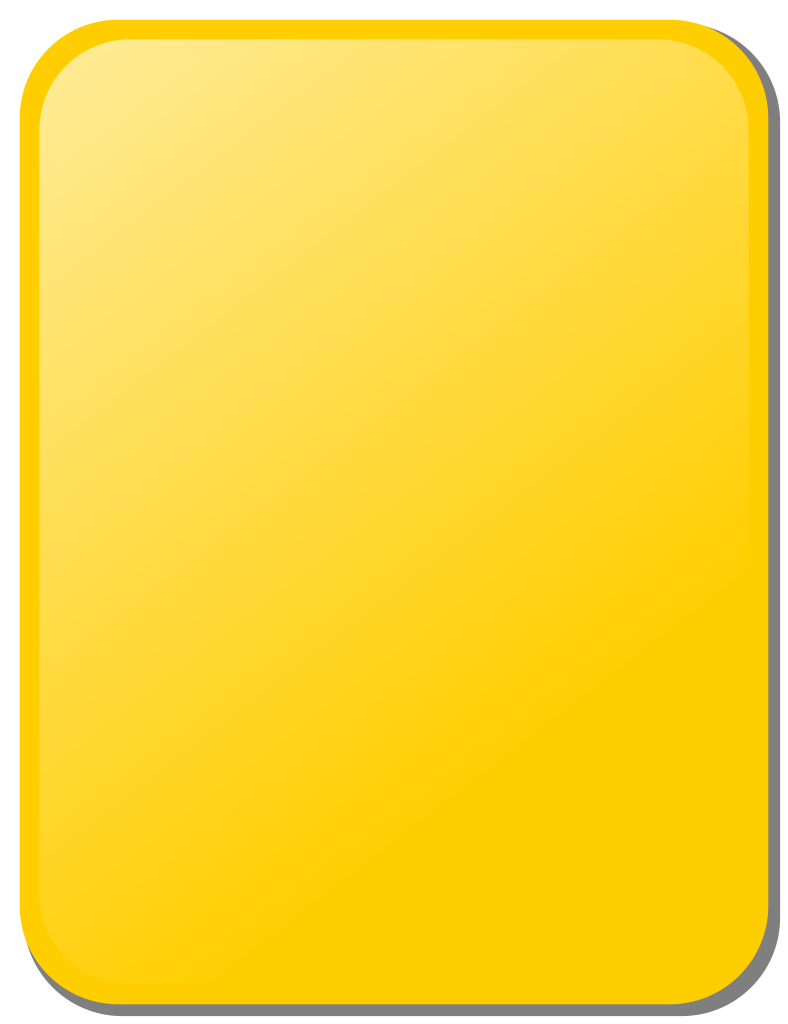
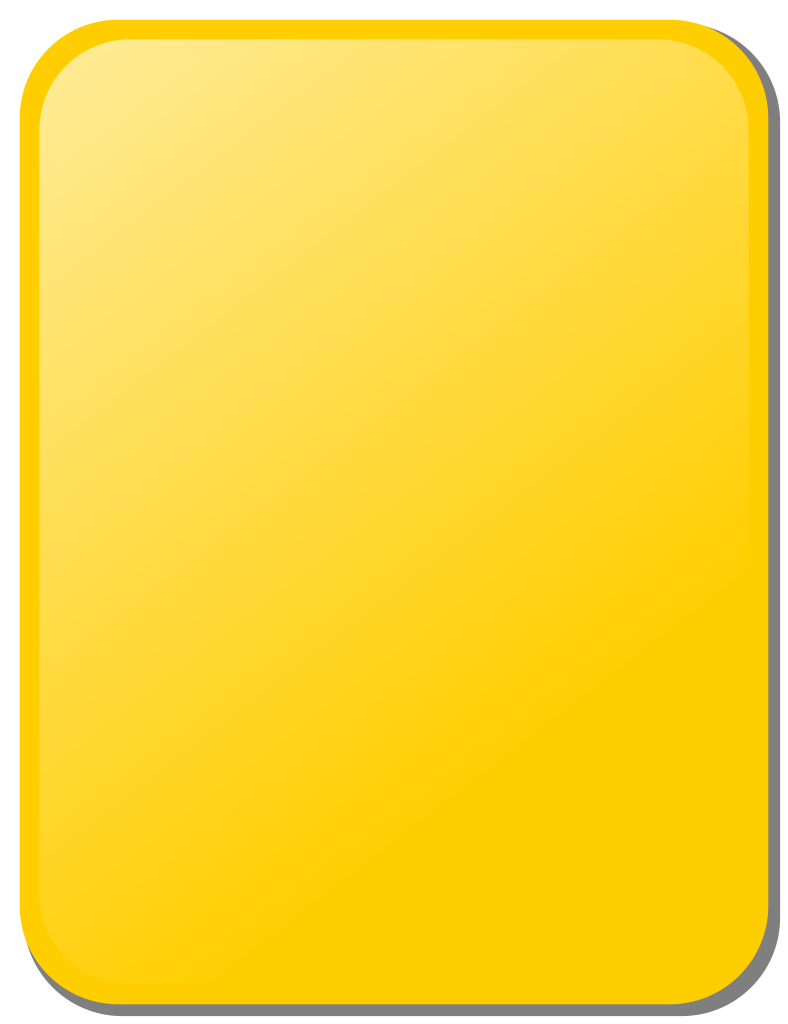
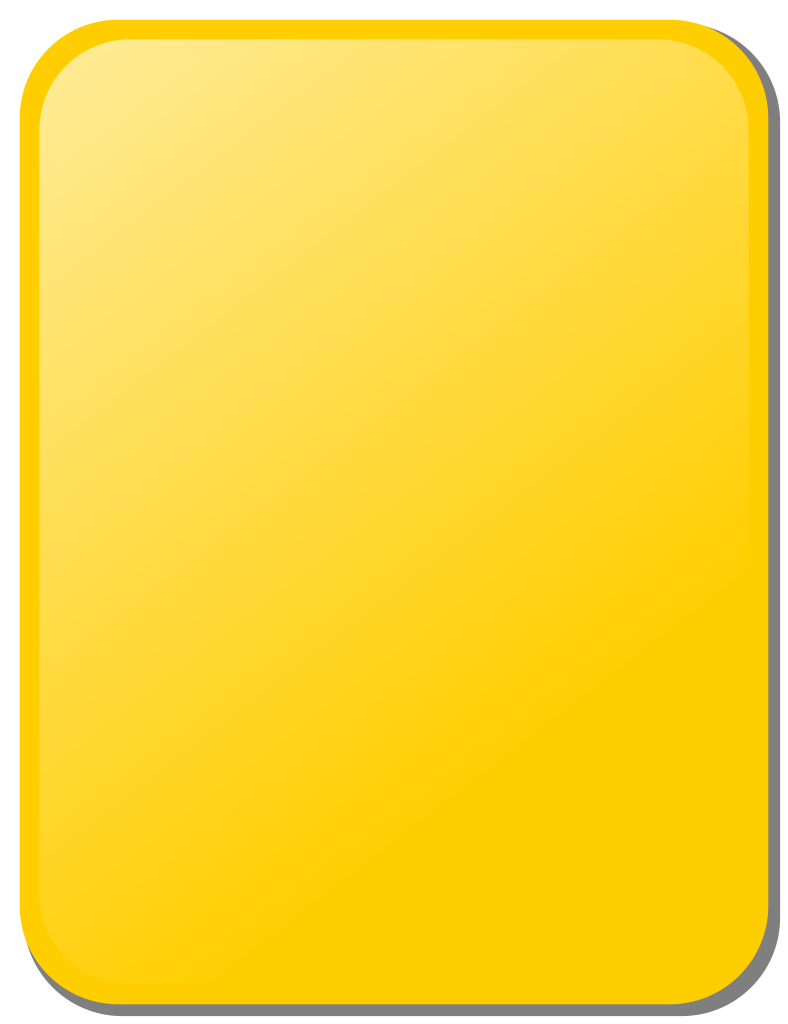
**STORY 0: CALIDAD DE CÓDIGO**

* Código comentado, explicando bien que hace cada función +1 punto por story
* Optimización de código: + 3 puntos por story
  + Se debe seguir las buenas prácticas de javascript indicadas en la siguiente página:
  + <https://dev.opera.com/articles/javascript-best-practices/>
  + Además se debe tener en cuenta que:
    - No haya variables no declaradas
    - Variables globales bien usadas, no usar si no son necesarias.
    - Uso de funciones: código separado en funciones
    - Código limpio, sin variables que no se usan, ni funciones o código comentado
* Uso de Typescript: + 1 punto por story

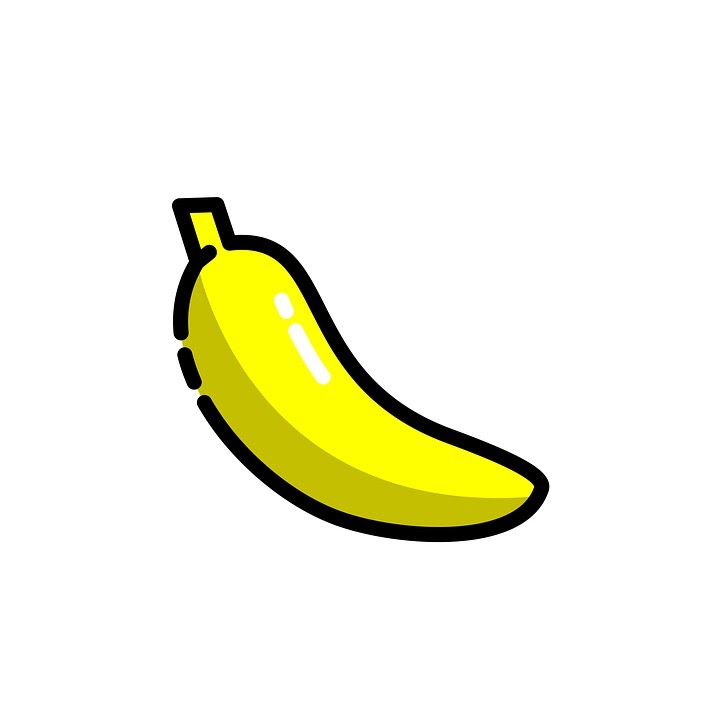
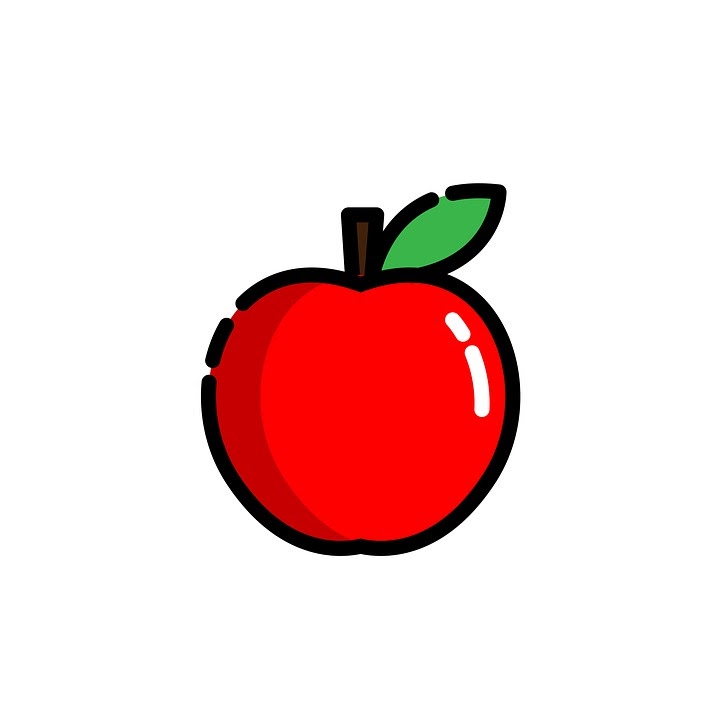
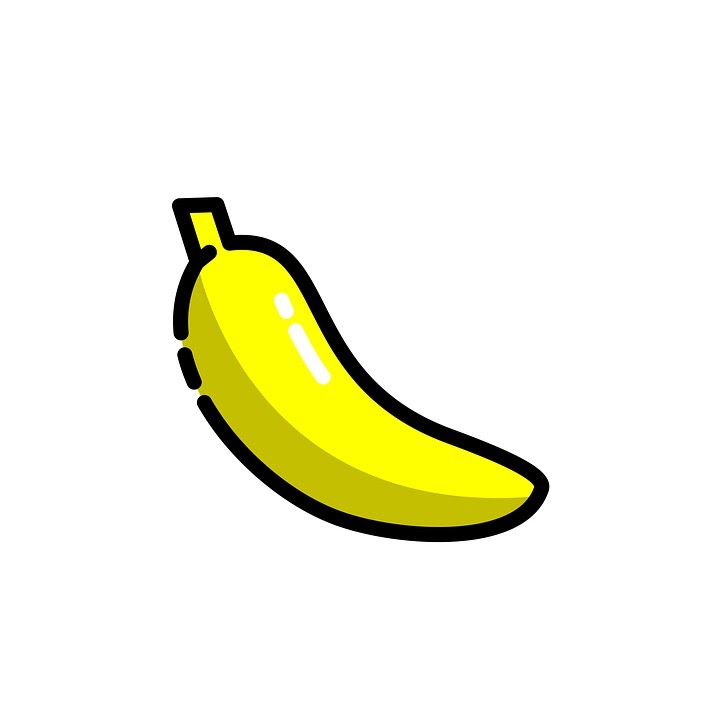
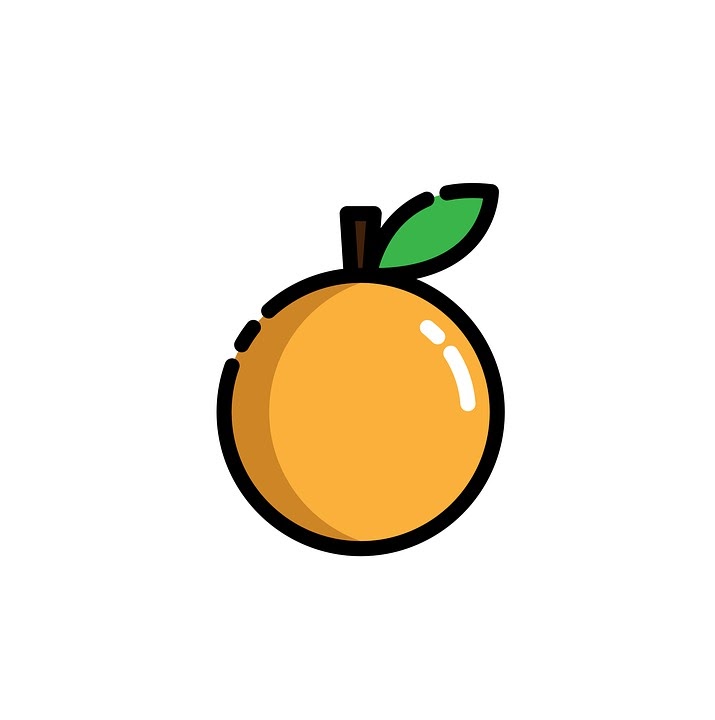
**STORY 1: PÁGINA DE INICIO DEL JUEGO. RESPONSIVE - 6 PUNTOS**

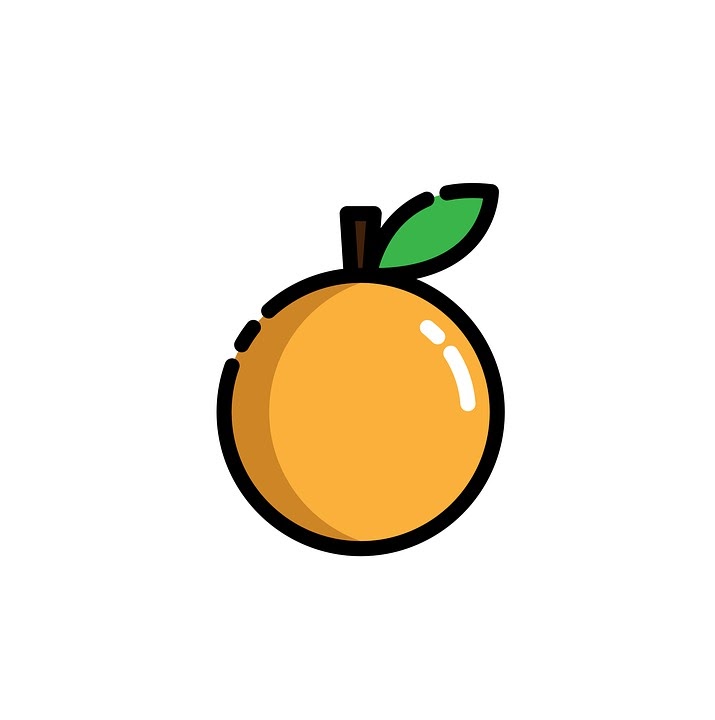
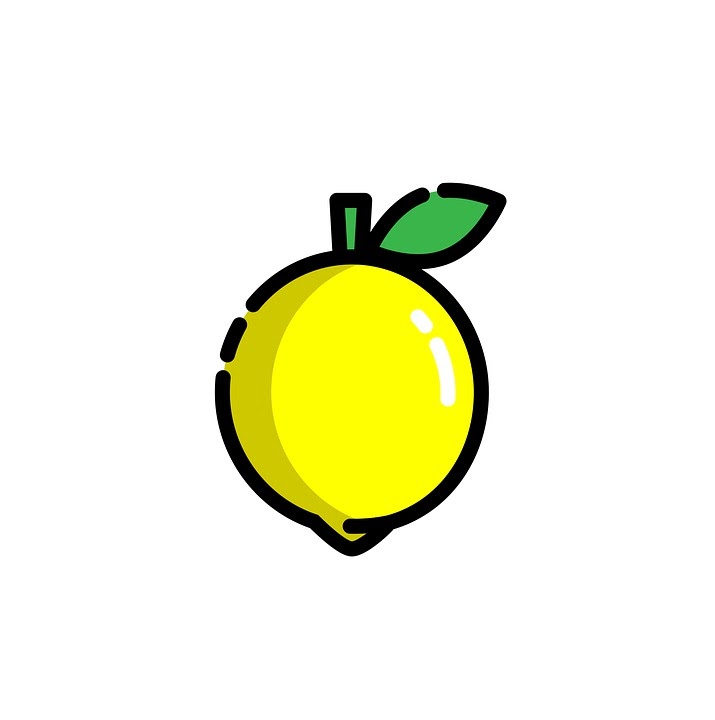
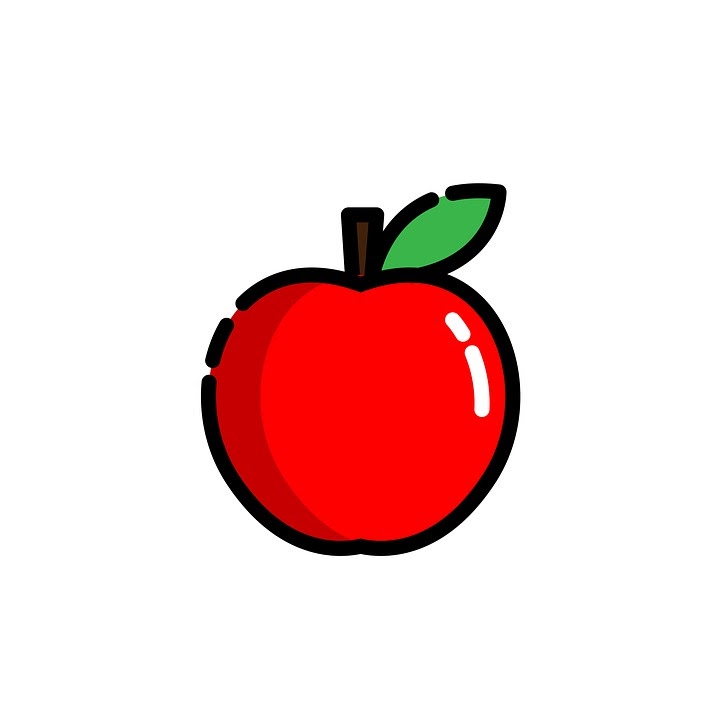
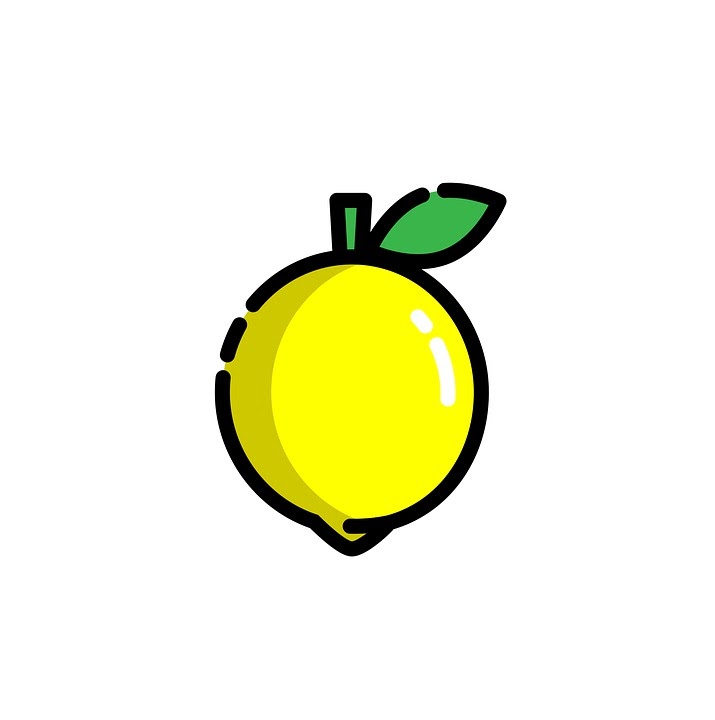
Inicialmente, se mostrará una página con 8 imágenes, que serán el equivalente a las cartas mostradas boca abajo:





A continuación, en la misma página habrán 4 pares de imágenes , ocultas inicialmente, que será el equivalente a las cartas mostradas boca arriba:





**STORY 2: MOSTRAR CARTA - 6 PUNTOS**

Función “Mostrar Carta”

Crea una función mostrar carta, que sea llamada cuando el usuario haga click sobre una de las cartas boca abajo. Esta función ocultará la imagen de carta boca abajo y mostrará la imagen de la carta boca arriba que está en su posición.

**STORY 3: OCULTAR CARTAS - 6 PUNTOS**

Función ocultar carta

Crea una función ocultar carta, que cuando se la llama, oculte la imagen de carta boca arriba de la posición que se le ha pasado y muestre la imagen de carta boca abajo.

**STORY 4: COMPROBAR CARTAS - 10 PUNTOS**

Función Comprobar cartas:

Cuando se han mostrado dos cartas, se debe comprobar si las dos cartas boca arriba son iguales, si son iguales, se dejan ambas boca arriba y se suma un punto al jugador. Si no son iguales, deben volver a ocultarse ambas (llamando a ocultar carta)

**STORY 5: COMPROBAR JUEGO FINALIZADO - 6 PUNTOS**

Función comprobar juego finalizado:

Si el juego se ha finalizado, es decir, el jugador tiene los 4 puntos, se deberá mostrar el mensaje, enhorabuena, ha ganado el juego.

**STORY 6: INICIO JUEGO - 6 PUNTOS**

Función iniciar juego:

Debe mostrarse un botón iniciar juego, que volverá a mostrar todas las cartas boca abajo y reiniciar el marcador.

**STORY 7: CALCULAR TIEMPO - 10 PUNTOS**

Función calcular tiempo:

En la función iniciar juego, se debe guardar la hora de inicio del juego.

En la función comprobar juego finalizado, si se ha finalizado, se debe guardar la hora de fin del juego.

En la función calcular tiempo, se debe calcular el tiempo trascurrido desde el inicio hasta el fin del juego, y devuelva el tiempo que ha tardado el jugador en finalizar el juego.

Modifica la función comprobar juego finalizado, para que además de mostrar el mensaje de enhorabuena, muestre el tiempo que ha tardado en finalizar el juego.

**STORY 8: PUBLICAR JUEGO - 6 PUNTOS**

Publicar Juego HTML

Usa google drive para publicar el juego y poder acceder a él desde cualquier ordenador o móvil. Para ello sigue los pasos indicados en la siguiente WEB:

<https://www.eldiario.es/tecnologia/diario-turing/google-drive-pagina_1_5579850.html>

**STORY 9: AÑADE ANIMACIÓN, EFECTO AL GIRAR LA CARTA - 10 PUNTOS**

Sigue los pasos del siguiente tutorial, para que con css añadas efecto de girar la carta:

<https://codepen.io/simplicitytree/pen/LvvzPq>

**RECURSOS Y EJEMPLOS:**

WEB RESPONSIVE:

Ejemplo de mostrar elementos de forma responsive, usando FLEXBOX de CSS3

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  // Habrá un elemento contenedor, que contendría las imágenes, se le debe indicar display:flex y propiedad flex-wrap para que se adapte al tamaño de la pantalla.    .flex-container {  display: flex;  flex-wrap: wrap;    background-color: DodgerBlue;  }    li {  list-style-type: none;  background-color: #f1f1f1;  width: 100px;  margin: 10px;  text-align: center;  line-height: 75px;  font-size: 30px;  }  </style>  </head>  <body>  <h1>Flexible Boxes</h1>    <ul class="flex-container">  <li>1</li>  <li>2</li>  <li>3</li>  <li>4</li>  <li>5</li>  <li>6</li>  <li>7</li>  <li>8</li>  </ul>    </body>  </html> |

USO DE TYPESCRIPT:

<https://medium.com/javascript-comunidad/typescript-fundamentos-y-ejemplos-b%C3%A1sicos-efd7ddcea90d>

COMPARATIVA, PASAR DE JAVASCRIPT A TYPESCRIPT CON EJEMPLOS:

<https://www.adictosaltrabajo.com/2018/08/01/typescript-vs-javascript/>