

# UNIVERSIDAD POLITÉCNICA DE TECÁMAC.

**Docente:** Emmanuel Torres Servín.

**Integrantes:**

- Alemán Pérez Natali Joselin.
- Ángel Velasco Marco Joel.
- De la Cruz Medina José Eduardo.
- Lopez Castillo Catherine Danae.
- Díaz García Karla Faviola.

**Matriculas:**

1321124050  
1321124015  
1321124069  
1321124020  
1321124001

**Grupo:** 5322IS.

## Hotwheels.

## Índice:

|   |   |
|---|---|
| Explicación de la maqueta mockups y su importancia.....     | 3 |
| Mockups (Hotwheels) .....                                   | 4 |
| Lista de componentes visuales y no visuales.....            | 5 |
| Mapa conceptual de componentes visuales y no visuales. .... | 6 |
| Juego Hotwheels.....  | 7 |
| Reglas del juego:.....                                      | 7 |
| Documentación.....  | 8 |
| Descripción del funcionamiento .....                        | 8 |

## Explicación de la maqueta mockups y su importancia.

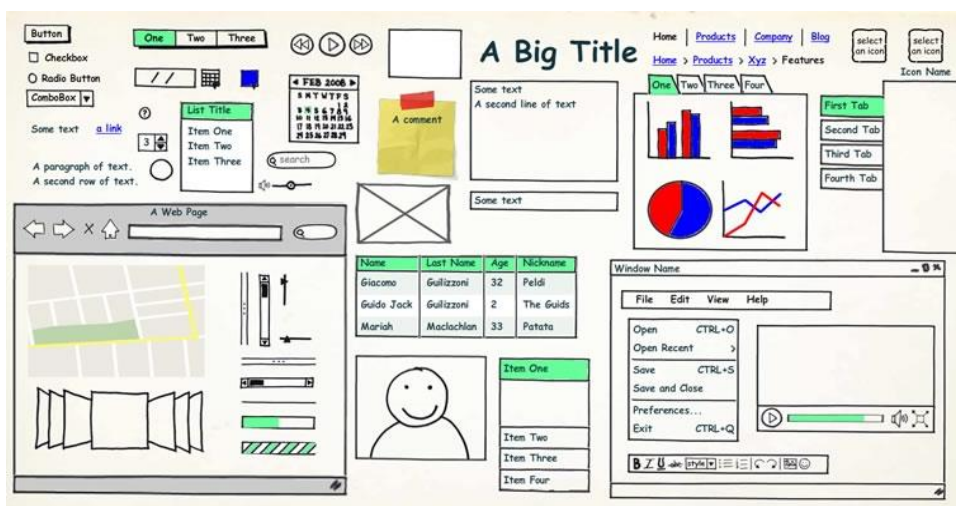
Un mock-up es un prototipo hecho antes del desarrollo del trabajo. Sirve para transformar ideas en funcionalidades y ayuda al cliente a exteriorizar y comprender lo que necesitamos, dicho de otra manera un mockup no es más que **un montaje que realizan los diseñadores gráficos y diseñadores web**, para mostrar a sus clientes cómo quedarán sus diseños impresos en alguna superficie.

Al trabajar con un mock-up, el cliente expone ante el profesional qué es lo que pasa por su cabeza, y consigue comprender mejor el nivel de complejidad del trabajo, lo cual también facilita la comunicación y desarrollo del proyecto luego de la contratación, además permite que el producto sea testeado a un costo mucho menor.

Es una pieza fundamental porque al momento de reunirnos con el cliente realizamos la estructura del diseño del sitio o aplicación web, permitiendo tener una idea más clara de cómo van a estar distribuidos los elementos, como la barra de navegación, los titulares, las imágenes, los párrafos. etc.

El valor y tiempo para ajustes en esa etapa es enorme, y esto puede evitarse pasando por una fase de test a través del mock-up.

### Ejemplo de un Mock-up:

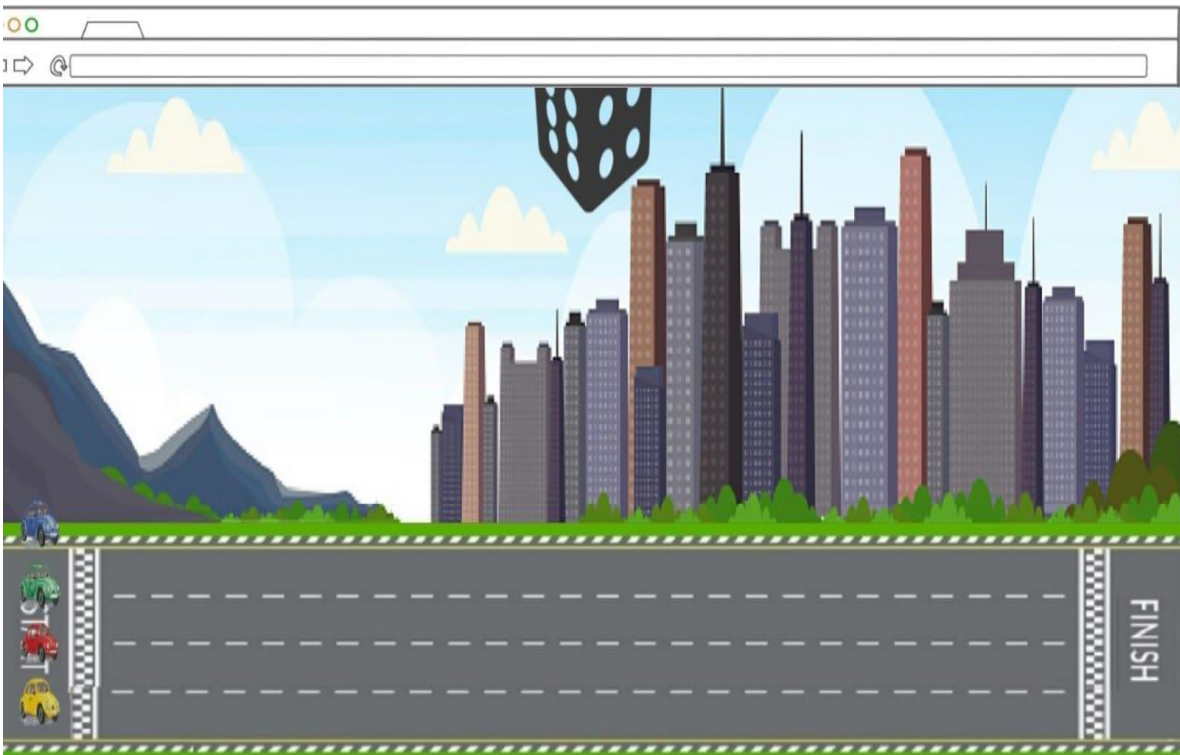


## Mockups (Hotwheels)

Interfaz de inicio del juego.



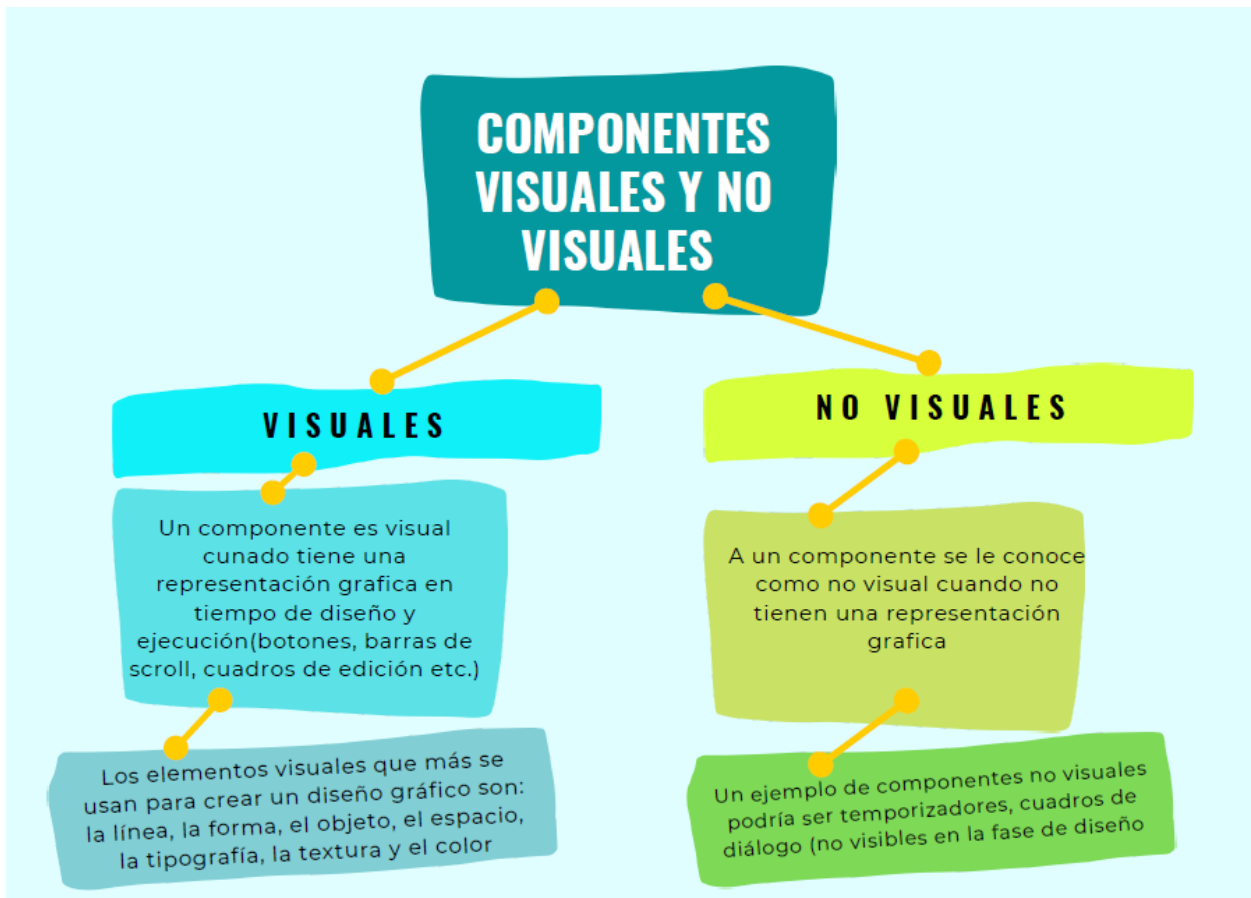
Interfaz de juego.



## Lista de componentes visuales y no visuales.

| Componentes          |                         |
|----------------------|-------------------------|
| Componentes visuales | Componentes no visuales |
| Botones              | Tamaño de los paneles   |
| Etiquetas            |                         |
| Imágenes             | Color de los paneles    |
| Iconos               |                         |
| Barras de scroll     |                         |

## Mapa conceptual de componentes visuales y no visuales.



# Juego Hotwheels.

## Reglas del juego:

- El juego consta de 4 jugadores que serán representados con automóviles.
- El turno de los jugadores dependerá de un dado, el cual se definirá conforme a la máxima puntuación.
- Ganara quien llegue a la meta.
- El podio se definirá conforme a la posición de los automóviles.
- Los carros avanzaran dependiendo el número generado por el dado.

# Documentación.

## Descripción del funcionamiento

En esta parte se dará una breve descripción del funcionamiento del código para tratar de llegar a un nivel de entendimiento mejor.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Hotwheels</title>
</head>
<!--Fondo de la pagina-->
<body style="background-image: url(carritos/fondo2.jpeg); background-repeat: no-repeat; background-size: cover; background-color: #008000;">
  <div style="
    background-image: url(carritos/pista.jpeg);
    width: 100%;
    background-size: cover;
    background-repeat: no-repeat;
    background-position: center;
    background-color: #008000;
  "></div>
```

En esta parte lo que se hace es establecer el fondo. Nosotros buscamos colocar un fondo verde que simulara el pasto de una pista.

La colocación del fondo empieza desde el <body style seguido de la URL del fondo asignándole que se centre y que cubra el 100% de la pantalla.

Después se asigna el segundo fondo que sería la pista por donde correrán los carros la cual empieza desde el <div style asignándole un div para mejor manejo del fondo le igual manera se le asigno la posición pero esta vez se le asigno la posición de centrado en la parte de debajo de la pantalla. Una vez colocado exitosamente se vería algo así.





```

<!--Diseño de los carros-->
<div style="position: absolute; top: 600px">
  <button id="carro1" style="
    background-image: url('carritos/azul.png');
    background-color: transparent;
    border: transparent;
    background-position: center;
    background-size: 100% 100%;
    cursor: pointer;
    top: 0px;
    left: 20px;
    width: 130px;
    height: 110px;
    position: relative;
  "></button>
<br />
  <button id="carro2" style="
    background-image: url('carritos/verde.png');
    background-color: transparent;
    border: transparent;
    background-position: center;
    background-size: 100% 100%;
    cursor: pointer;
    top: 0px;
    left: 20px;
    width: 130px;
    height: 110px;
    position: relative;
  "></button>
<br />
  <button id="carro3" style="
    background-image: url('carritos/rojo.png');
    background-color: transparent;
    border: transparent;
    background-position: center;
    background-size: 100% 100%;
    cursor: pointer;
    top: 0px;
    left: 20px;
    width: 140px;
    height: 120px;
    position: relative;
  "></button>
<br />
  <button id="carro4" style="
    background-image: url('carritos/amarillo.png');
    background-color: transparent;
    border: transparent;
    background-position: center;
    background-size: 100% 100%;
    cursor: pointer;
    top: 0px;
    left: 20px;
    width: 130px;
    height: 110px;
    position: relative;
  "></button>
</div>

```

Para el diseño de los carros se les asigno un <div> para un mejor manejo dentro del código, se les asigno una posición absoluta y un tamaño de 60 pixeles.

Los carros se asignaron como botones ya que después los jugadores tendrán que dar un clic encima de ellos para elegir con que carro participar.

Se les asigno un background(fondo) transparente una posición centrada abajo izquierda.

```

<!--Diseño del dado-->
<div style="text-align: center">
  <button id="dado" style="
    background-image: url('carritos/dado1.png');
    background-color: transparent;
    border: transparent;
    background-position: center;
    background-size: 100% 100%;
    cursor: pointer;
    top: 0px;
    left: 0px;
    width: 200px;
    height: 200px;
    position: relative;
  "></button>
<br>

```

Para darle diseño al dado se asignó un botón con la imagen de un lado con borde transparente y asignándolo al centro arriba de la pantalla.

```
<!--Diseño de texto que se mostrara abajo del dado al tirarlo-->
<font id="etiqueta" color="white" ; face="monospace" ; size=6;"></font>
</div>
```

Se hace el diseño del texto que aparecerá debajo del lado cuando los jugadores le den clic.

```
<script>
  var avanzar = "derecha";
  var inicio = 0;
  var velocidad = 60;
  var total_turno = 4;
  var turno=1;
  var posicion = [20, 20, 20, 20];
  var jugadores=[];
  var estado=0;

  var indiceJugador = 0;
  let dado = document.getElementById("dado");
```

### **Declaración de las variables a utilizar para el desarrollo de este videojuego.**

Las variables `let dado= document.getElementById("dado");` se le colocó el `document.getElementById` para que nos devuelva una referencia del elemento por su ID.

```
//Pedir nombres
function pedirNombres() {
  for (let i = 0; i < 4; i++) {
    jugadores.push({
      nombre:prompt(`Ingresa el nombre del Jugador ${i+1}`)
    });
  }
}
```

La función `pedirNombre` lo que hace es que con el `for` hace que se repita la frase ingresa nombre del jugador hasta que sea igual a 4 y que se guarde en `jugadores` con el `.push`.

La función `asignarTurnos` les explica las reglas del videojuego después les dice que cada jugador presione el dado para que salga un número al lanzar y el jugador con el número más alto es el que primero tire.

```
function asignarTurnos() {
  //Llamar la función pedirNombre();
  pedirNombres();
  let aux;
  alert(`Instrucciones\n1.Los turnos serán asignados conforme la máxima puntuación de cada jugador\n2.Para tirar el dado haz click sobre
`);
}
```

```

//Funcion para elegir los carros
function elegirCarros() {
  //Si estado es 0, entonces seleccionar los carros y asignar turnos
  if(estado==0){
    for (let i = 0; i < jugadores.length; i++) {
      //Si da click en en dado, entonces tirar el dado
      document
        .getElementById("carro" + (i + 1))
        .addEventListener("click", () => {
          alert(jugadores[indiceJugador].nombre + " eligió el coche " + i);
          jugadores[indiceJugador]={
            nombre:jugadores[indiceJugador].nombre,
            turno:jugadores[indiceJugador].turno,
            coche:document.getElementById("carro" + (i + 1))
          };
          //Asignar el turno al jugador para elegir el carro
          if(indiceJugador<3){
            alert(`${jugadores[indiceJugador+1].nombre} elige tu carro`);
          }
          //Si se terminaron de elegri los carros
          if(indiceJugador<4){
            estado=1;
          }
          indiceJugador++;
        });
    }

    alert(`${jugadores[0].nombre} elige tu carro`);
  }
}

```

Función elegirCarros se encarga que después de tirar el lado para asignar los tiros le haga elegir a cada jugador que carro quiere utilizar si alguno de los jugadores llega a elegir un carro que ya este ocupado le dirá que elija otro carro, si no está ocupado le dirá (jugador eligió el carro N).

```

//Funcion para mover los carros
function moverCarrito(carro) {
  if (avanzar == "derecha") {
    inicio += velocidad;
  } else {
    inicio -= velocidad;
  }
  //
  carro.style.left = inicio + "px";
  this.posicion[turno - 1] = inicio
  let ancho=window.innerWidth-100;
  //Si el carro llega al final de la pantalla, entonces finalizar juego
  if(inicio>=ancho){
    alert("Gano el jugador "+jugadores[this.turno-1].nombre);
    estado=2;
    return;
  }
}

```

La función moverCarrito lo que hace es que asigna que los carros solo se pueden mover a la derecha y se colocó el window.innerWidth para que os carros no pudieran sobre salir de la pantalla.

También se asigna que si alguno de los carros llega a cruzar la meta la variable estado se igualara a 2 y esto significara que alguno de los 4 jugadores ya abra ganado dejando sin poder tirar a los otros 3 jugadores.

```
//Si se da click en el dado, entonces tirar el dado
dado.addEventListener("click", () => {
  if(estado==2){
    alert("Juego terminado");
    return;
  }
  //Asignar el turno al jugador para tirar el dado
  let num;
  dado.style.backgroundImage = "url('carritos/dado.gif')";
  document.getElementById("etiqueta").innerHTML = "Tirando Dado...";

  setTimeout(() => {
    num = Math.round(Math.random() * (6 - 1) + 1);
    //Si el numero es 1, ejecutar proceso de asignar turnos
    if(estado==1){
      inicio = this.posicion[this.turno - 1];
      dado.style.backgroundImage = "url('carritos/dado1.png')";
      document.getElementById("etiqueta").innerHTML = "";
      alert("Tu número es " + num);
      let cont = 1;
      let idIntervalo = setInterval(() => {
        moverCarrito(jugadores[(this.turno-1)].coche);

        if (cont >= num) {
          clearInterval(idIntervalo);

          if (this.turno >= this.total_turno) {
            this.turno = 1;
          } else {
            this.turno++;
          }
        }
      }, 1000);
    }
  }, 1000);
});
```

Si se llega a dar clic cuando uno de los 4 jugadores ya allá cruzado la línea de meta entonces no dejará realizar más turnos y el juego se dará por terminado.

Por otra parte, también se realiza el texto y el cambio de imagen cuando se presiona el dado asiéndolo un poco más visual.

También se avisará con su nombre al jugador que le toque tirar el dado.

