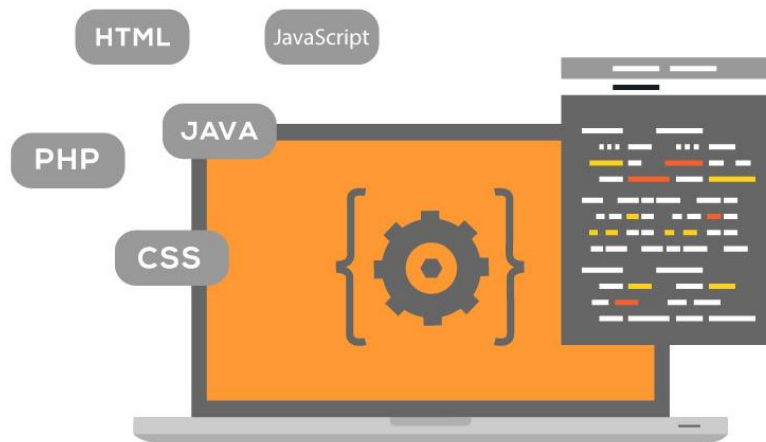




**UNIVERSIDAD AUTONOMA DE
CHIHUAHUA
FACULTAD DE INGENIERIA**



DESARROLLO BASADO EN PLATAFORMAS



REPORTE TÉCNICO FINAL

JORGE AARÓN HERNÁNDEZ BUSTILLOS-329747

HUGO EDIBRAY BECERRA GÁNDARA-329532

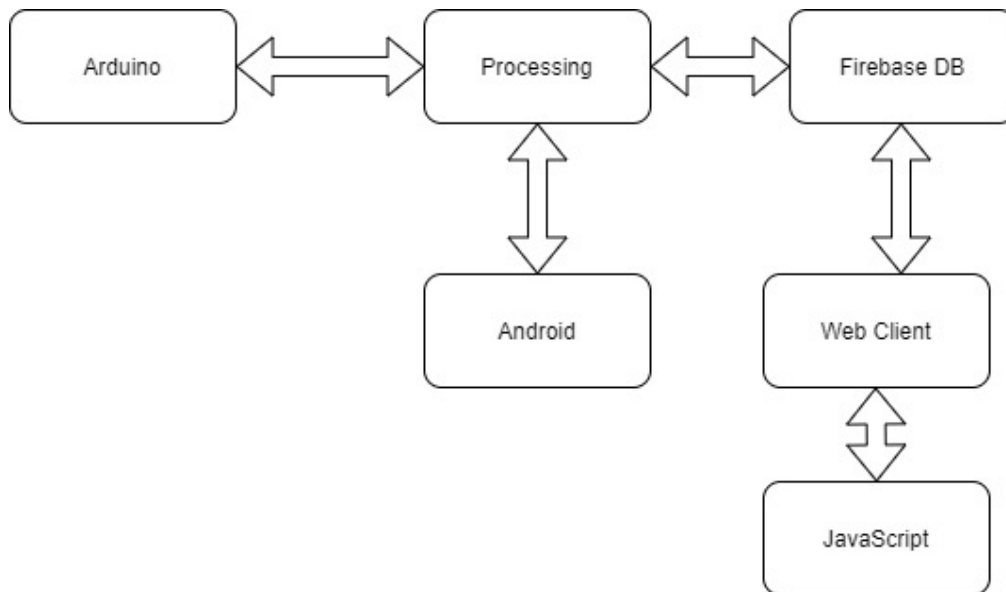
MAESTRO: FERNANDO MARTINEZ REYES

24/05/2021

INTRODUCCIÓN

El objetivo general de este proyecto es poder implementar todos los temas, tecnologías y conocimientos adquiridos a lo largo de este semestre. Además de poner a prueba nuestra creatividad con un proyecto que no sea parecido a los ya realizados anteriormente. Y claro que, uno de los mayores propósitos del proyecto y también una de las grandes dificultades presentadas en la elaboración y planificación de este proyecto, fue la implementación de lo ya mencionado anteriormente de una forma coherente y que tenga sentido para el desarrollo de la aplicación y que no se sienta implementado a lo fuerza o fuera de lugar.

Diagrama de Bloques



Arduino

La implementación del Arduino la consideramos cómo una de las más “sencillas” o al menos de las menos complicadas, pues está basada directamente en una de las prácticas vistas en clase. En nuestro proyecto, la función del Arduino es cambiar mediante los sensores de temperatura y de luz, el fondo de la página de html y la imagen del estadio del juego, respectivamente. Si el sensor de temperatura casi no detecta temperatura el fondo será de nieve y en caso contrario habrá sol. Mientras que, en el estadio, si no detecta luz, el fondo será de un estadio con las luces prendidas y en caso de que si detecte luz la imagen del estadio será de día.

```
Arduino
#include <TimerOne.h>
#include <Wire.h>
#include <MultiFuncShield.h>

//Y que pin analógico conectamos el LDR
int valorLDR;
int pinLDR = A5;
float temp;
void setup() {
  Timer1.initialize();
  MFS.initialize(&Timer1);
  MFS.initLM35(SMOOTHING_MODERATE);
  Serial.begin(9600);
}

void loop() {
  int tempCentigrade = MFS.getLM35Data(); // get centigrade in 1/10 of degree.
  temp = (float)tempCentigrade/10;
  //MFS.write((float)tempCentigrade / 10, 1); // display temp to 1 decimal place.
  //Serial.write(tempCentigrade);

  valorLDR = analogRead(pinLDR);
  MFS.write(valorLDR);
  delay(100);
  Serial.print(temp);
  Serial.print(valorLDR);
  Serial.print('%');
  //int arreglo[] = {tempCentigrade, valorLDR};
  //Serial.write(arreglo[0]);
  //Serial.write(arreglo[1]);
  //delay(100);
}
```

Processing

Processing en este proyecto fue implementado al momento de realizar, la conexión Wifi entre nuestro dispositivo Android, la computadora y Firebase; Processing de hecho cumple un papel fundamental en el desarrollo de este proyecto, pues en él fue desarrollada la aplicación que después lanzaríamos desde nuestro dispositivo Android, a la vez que es la encargada de la aplicación de la pc, la cuál manda la información de la pc a la base de datos(más adelante veremos cuál es el tipo de información enviada)

```
}  
void draw(){  
  background(0, 255, 0);  
  /*if(puerto.available() > 0){  
    port = puerto.readStringUntil('%');  
  }  
  if(port!=null){  
    fire.setValue("Temp", port);  
  }*/  
  fire.setValue("Jugador2/Nombre", name);  
  s = Float.toString(skin);  
  fire.setValue("Jugador2/Skin", s);  
  if(Ready==1.0){  
    ready="LISTO";  
    fire.setValue("Jugador2/Ready", "1");  
  }  
  if(Ready==0.0){  
    ready="NO LISTO";  
    fire.setValue("Jugador2/Ready", "0");  
  }  
  text("Jugador listo: " + ready, width/2, height/2);  
  m = Float.toString(move);  
  fire.setValue("Jugador2/posMono", m);  
  j = Float.toString(jump);  
  fire.setValue("Jugador2/Salto", j);  
}
```

Parte del código de la computadora

```
void draw() {  
  //background(0);  
  
  //image(fondo, 0, 0, width, height);  
  if(menu == 0){  
    image(fondo, 0, 0, width, height);  
    image(logo, width/2-((width/2)/2), 80, width/2, height/3);  
    //fill(255);  
    //text("HEAD SOCCER", width/2, (height/2)-200);  
    fill(0);  
    text("Nombre: " + typing, width/2, height/2);  
    fill(255);  
    rect(width/2, height/2+300, width/5, height/6, 40);  
    fill(0);  
    text("Play", width/2, height/2+325);  
  }  
  if(menu == 1){  
    image(fondo, 0, 0, width, height);  
    image(logo, width/2-((width/2)/2), 80, width/2, height/3);  
    fill(255);  
    text("Choose a player", width/2, height/2-100);  
    image(skin1, width/6-100, height/2, 200, 400);  
    image(skin2, (width/6)*2-100, height/2, 200, 400);  
    image(skin3, (width/6)*3-100, height/2, 200, 400);  
    image(skin4, (width/6)*4-100, height/2, 200, 400);  
    image(skin5, (width/6)*5-100, height/2, 200, 400);  
  }  
  if(menu == 2){  
    background(0, 255, 0);  
    textSize(60);  
    if(ready == 0.0){  
      fill(0);  
      rect(x,y,w,h,30);  
      fill(255);  
      text("NOT READY",x,y+20);  
    }  
  }  
}
```

Parte del código Android

Android

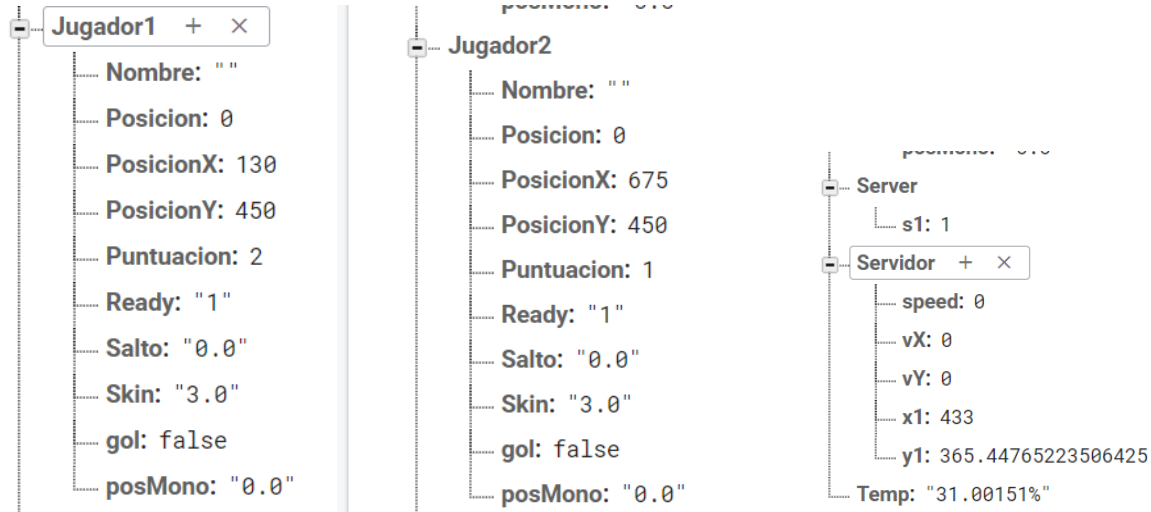
Podemos decir que nuestro dispositivo Android es nuestro control, pues con el somos capaces, de seleccionar nuestro jugador favorito, indicar que estamos listos para el partido y mover a nuestro jugador antes seleccionado



Menú de selección de personaje corriendo en Android

Firestore DB

Nuestra base de datos es la encargada de almacenar los siguientes datos:

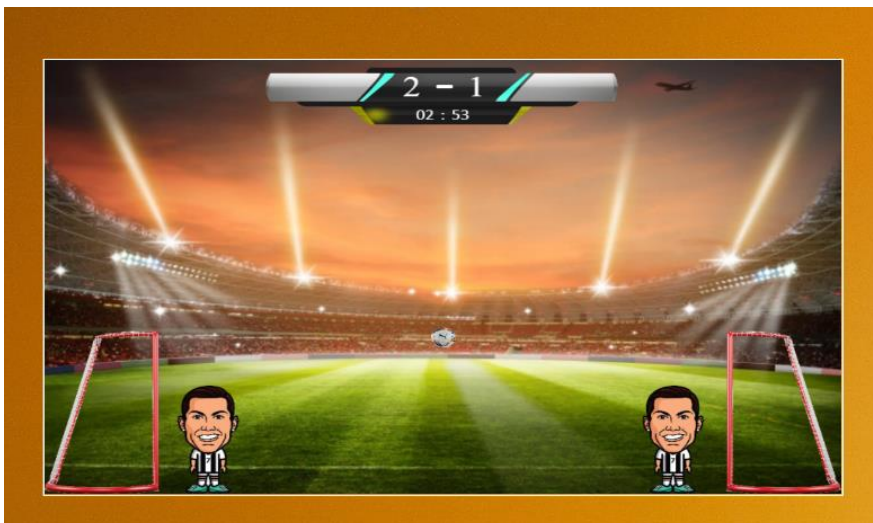


Cabe mencionar que el nombre se encuentra así, pues ese es insertado desde el dispositivo Android

Dependiendo de la skin que seleccionemos, en processing se va a mandar un numero a nuestra variable skin en Firebase para poder desplegar la imagen que también está almacenada en nuestra base de datos.

Web Client & JavaScript

No podemos mencionar Web Client (Html y Css) sin mencionar JavaScript, pues van de la mano. Web Client es la base de nuestro videojuego, pues nuestro Web Client es el encargado de mostrar el videojuego; JavaScript es el encargado de implementar nuestras físicas y todo lo relacionado a la lógica de nuestro videojuego



Página principal del juego (HTML, JS Y CSS)

Servidor/Cliente en JS

Como podemos ver, el objeto balón tiene el atributo server, el cual en un inicio es 0; este valor cambiará a 1, dependiendo de quien sea el primero que entre a la página

```
const ball = {  
  //y : canvas.height/4,  
  x : (canvas.width/2)-17,  
  y : (canvas.height/2)-17,  
  radius : 17,  
  w: 34,  
  h: 34,  
  velocityX : 0,  
  velocityY : 5,  
  speed : 20,  
  server:0 //guarda el dato de si eres servidor o cliente  
}
```

Si fuiste el primero en entrar a la página, serás tu valor en server será 1, por lo cuál tu serás el encargado de realizar todos los cálculos del juego

```
function update(){  
  /*Servidor  
  Se comprueba si la página en la que se está corriendo el juego es servidor o no*/  
  if(ball.server == 1){  
    movements(user)//se llama a los movimientos de los jugadores  
    movements(user2)  
    if(user.x+user.w >= user2.x){//se comprueba si los jugadores están chocando, dando un efecto de ser cuerpos físicos  
      user.x = user2.x-user.w  
      user2.x = user.x+user.w  
    }  
    if(ball.velocityY < 0){//Si el balón está yendo hacia arriba se le aplica gravedad  
      ball.velocityY *= 0.88  
      if(ball.velocityY >= -0.4)//si la velocidad es muy pequeña, se cambia su dirección  
        ball.velocityY = 0.4  
    }  
    else  
      if(ball.velocityY < 20){//Si el balón está cayendo, se agrega gravedad para aumentar su velocidad  
        ball.velocityY *= 1.15  
      }  
    if(ball.y+ball.h > canvas.height){//si el balón choca con el piso, se reinicia su velocidad y rebota  
      ball.velocityY = 25  
      ball.velocityY *= -1  
    }  
    if(ball.x < 0){//si el balón choca con alguna de las paredes, rebota  
      ball.velocityX *= -1  
    }  
    if(ball.x+ball.w > canvas.width)  
      ball.velocityX *= -1  
    ball.x += ball.velocityX;//movimientos del balón en "x" y en "y"  
    ball.y += ball.velocityY;  
    let player = (ball.x < user.x+user.w) ? user : user2;//se usan variables para saber que portería o que jugador chocó con el balón.  
    let arco = (ball.x < canvas.width/2) ? arco : arco2;  
    /*Colisiones con las porterías  
    Se chequea si el balón chocó con alguna de las porterías*/  
    if(collision(ball, arco)){  
      collidePoint = ((ball.y+ball.radius) - (arco.y + 5));//para comprobar si el balón está por encima o por debajo del travesaño  
      if(collidePoint < -6){//si el balón choca con la parte de arriba de la portería, rebota  
        ball.velocityY = 15  
      }  
    }  
  }  
}
```

Esto es solo una pequeña parte de los cálculos que se realizan siendo el servidor, pero van desde la línea 430 hasta la 526 del archivo fut.js disponible en el GitHub (al igual que todos los archivos)

En caso de haber obtenido el valor 0, serás el cliente

```
527  /*Cliente
528  Si en la carga de la página obtienes el valor de 0 significa que serás el cliente
529  por lo tanto, no son necesarios los cálculos
530  simplemente se obtienen las posiciones del balón, usuarios y puntajes para luego dibujarlos dentro de la página.
531  Como el sonido de gol solo se está ejecutando en el servidor, es necesario obtener el dato de si se hizo un gol o no
532  para ejecutar el sonido también del lado del cliente*/
533  else{
534      servidor.child("x1").on('value',function(snap){
535          ball.x = snap.val();
536      });
537      servidor.child("y1").on('value',function(snap){
538          ball.y = snap.val();
539      });
540      Puntuacion.child("PosicionX").on('value', function(snap){
541          user.x = snap.val();
542      });
543      Puntuacion2.child("PosicionX").on('value', function(snap){
544          user2.x = snap.val();
545      });
546      Puntuacion.child("PosicionY").on('value', function(snap){
547          user.y = snap.val();
548      });
549      Puntuacion2.child("PosicionY").on('value', function(snap){
550          user2.y = snap.val();
551      });
552      Puntuacion.child("Puntuacion").on('value', function(snap){
553          user.score = snap.val();
554      });
555      Puntuacion2.child("Puntuacion").on('value', function(snap){
556          user2.score = snap.val();
557      });
558      Puntuacion.child("gol").on('value', function(snap){
559          gol = snap.val();
560          if(gol){
561              goal.play()
562          }
563      });
564      Puntuacion2.child("gol").on('value', function(snap){
565          gol = snap.val();
566          if(gol){
567              goal.play()
```

Este apartado de código no es tanto, pues no se realizan cálculos, sólo obtenemos las posiciones y puntajes. Aunque igual si se quiere indagar más en este apartado de código va de la línea 527 a la 578.