

## CAPÍTULO III: DESARROLLO

### 3.1 Capturas de la Aplicación (Documentación completa del desarrollo, Scripts, Sprites, Prefabs e imágenes)

```
C# Barra.cs X
Assets > Codigo > C# Barra.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class Barra : MonoBehaviour {
5
6      public float velocidad = 0.4f;
7
8      Vector3 posicionInicial;
9
10     // Use this for initialization
11     void Start () {
12         posicionInicial = transform.position;
13     }
14
15     public void Reset()
16     {
17         transform.position = posicionInicial;
18     }
19
20     // Update is called once per frame
21     void Update () {
22         float tecladoHorizontal = Input.GetAxisRaw("Horizontal");
23         float posX = transform.position.x + (tecladoHorizontal * velocidad * Time.deltaTime);
24         transform.position = new Vector3(Mathf.Clamp(posX, -8, 8), transform.position.y, transform.position.z);
25     }
26 }
27
```

```
C# Bloque.cs X
Assets > Codigo > C# Bloque.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class Bloque : MonoBehaviour {
5
6      public GameObject efectoParticulas;
7      public Puntos puntos;
8
9      // Is Trigger DESACTIVADO
10     void OnCollisionEnter()
11     {
12         Instantiate(efectoParticulas, transform.position, Quaternion.identity);
13         Destroy(gameObject);
14         transform.SetParent(null);
15         puntos.GanarPunto();
16     }
17
18 }
```

```

C# BotonSalir.cs X
Assets > Codigo > C# BotonSalir.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class BotonSalir : MonoBehaviour {
5
6      public bool salir;
7
8      // Update is called once per frame
9      void Update () {
10         if (Input.GetKeyDown(KeyCode.Escape))
11         {
12             if (salir)
13             {
14                 Debug.Log("Salimos del juego");
15                 Application.Quit();
16             }
17             else
18             {
19                 Application.LoadLevel("Portada");
20             }
21         }
22     }
23 }
24

```

```

C# EmpezarPartida.cs X
Assets > Codigo > C# EmpezarPartida.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class EmpezarPartida : MonoBehaviour {
5
6      // Update is called once per frame
7      void Update () {
8         if (Input.GetButtonDown("Fire1"))
9         {
10             Puntos.puntos = 0;
11             Vidas.vidas = 3;
12             Application.LoadLevel("Nivel01");
13         }
14     }
15 }
16

```

C# Pelota.cs X

Assets > Codigo > C# Pelota.cs

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class Pelota : MonoBehaviour {
5
6      public float velocidadInicial = 600f;
7
8      public Rigidbody rig;
9
10     bool enJuego;
11
12     Vector3 posicionInicial;
13
14     public Transform barra;
15
16     // Use this for initialization
17     void Start () {
18         posicionInicial = transform.position;
19     }
20
21     public void Reset()
22     {
23         transform.position = posicionInicial;
24         transform.SetParent(barra);
25         enJuego = false;
26         DetenerMovimiento();
27     }
28
29     public void DetenerMovimiento()
30     {
31         rig.isKinematic = true;
32         rig.velocity = Vector3.zero;
33     }
34
35     // Update is called once per frame
36     void Update () {
37         if(!enJuego && Input.GetButtonDown("Fire1"))
38         {
39             enJuego = true;
40             transform.SetParent(null);
41             rig.isKinematic = false;
42             rig.AddForce(new Vector3(velocidadInicial, velocidadInicial, 0));
43         }
44     }
45 }
46
```

C# Puntos.cs X

Assets > Codigo > C# Puntos.cs

```
1  using UnityEngine;
2  using System.Collections;
3  using UnityEngine.UI;
4
5  public class Puntos : MonoBehaviour {
6
7      public static int puntos = 0;
8      public Text textoPuntos;
9
10     public GameObject nivelCompletado;
11     public GameObject juegoCompletado;
12
13     public SiguienteNivel siguienteNivel;
14
15     public Pelota pelota;
16     public Barra barra;
17
18     public Transform contenedorBloques;
19
20     public SonidosFinPartida sonidosFinPartida;
21
22     // Use this for initialization
23     void Start () {
24         ActualizarMarcadorPuntos();
25     }
26
27     void ActualizarMarcadorPuntos()
28     {
29         textoPuntos.text = "Puntos: " + Puntos.puntos;
30     }
31
32     public void GanarPunto()
33     {
34         Puntos.puntos++;
35         ActualizarMarcadorPuntos();
36
37         if (contenedorBloques.childCount <= 0)
38         {
39             pelota.DetenerMovimiento();
40             barra.enabled = false;
41
42             if (siguienteNivel.EsUltimoNivel())
43             {
44                 juegoCompletado.SetActive(true);
45             }
46             else
47             {
48                 nivelCompletado.SetActive(true);
49             }
50
51             sonidosFinPartida.NivelCompletado();
52
53             siguienteNivel.ActivarCarga();
54         }
55     }
56 }
```

```

C# SiguienteNivel.cs X
Assets > Codigo > C# SiguienteNivel.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class SiguienteNivel : MonoBehaviour {
5
6      public string nivelACargar;
7      public float retraso;
8
9      [ContextMenu("Activar Carga")]
10     public void ActivarCarga()
11     {
12         Invoke("CargarNivel", retraso);
13     }
14
15     void CargarNivel()
16     {
17         if (!EsUltimoNivel())
18         {
19             Vidas.vidas++;
20         }
21         Application.LoadLevel(nivelACargar);
22     }
23
24     public bool EsUltimoNivel()
25     {
26         return nivelACargar == "Portada";
27     }
28
29 }
30

```

```

C# SonidosFinPartida.cs X
Assets > Codigo > C# SonidosFinPartida.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class SonidosFinPartida : MonoBehaviour {
5
6      public AudioSource audioSource;
7      public AudioClip completado;
8      public AudioClip gameOver;
9
10     public void GameOver()
11     {
12         ReproduceSonido(gameOver);
13     }
14
15     public void NivelCompletado()
16     {
17         ReproduceSonido(completado);
18     }
19
20     void ReproduceSonido(AudioClip sonido)
21     {
22         audioSource.clip = sonido;
23         audioSource.loop = false;
24         audioSource.Play();
25     }
26 }

```

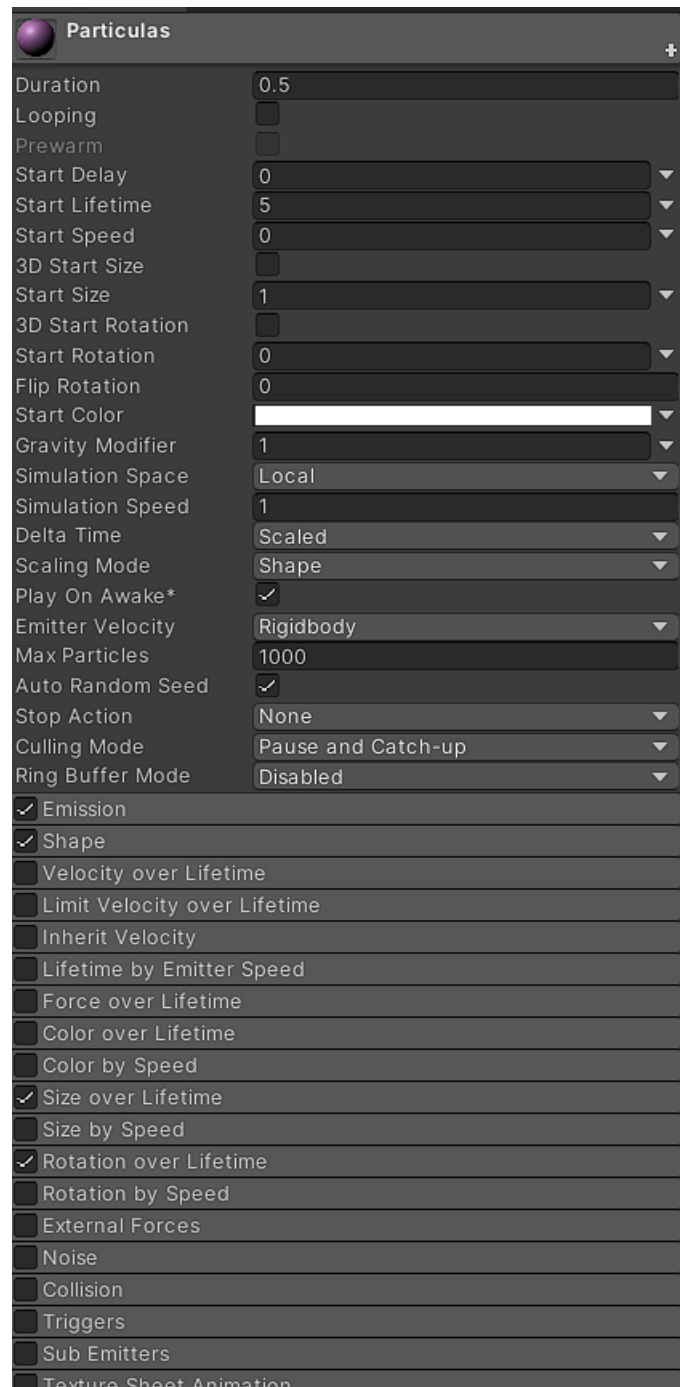
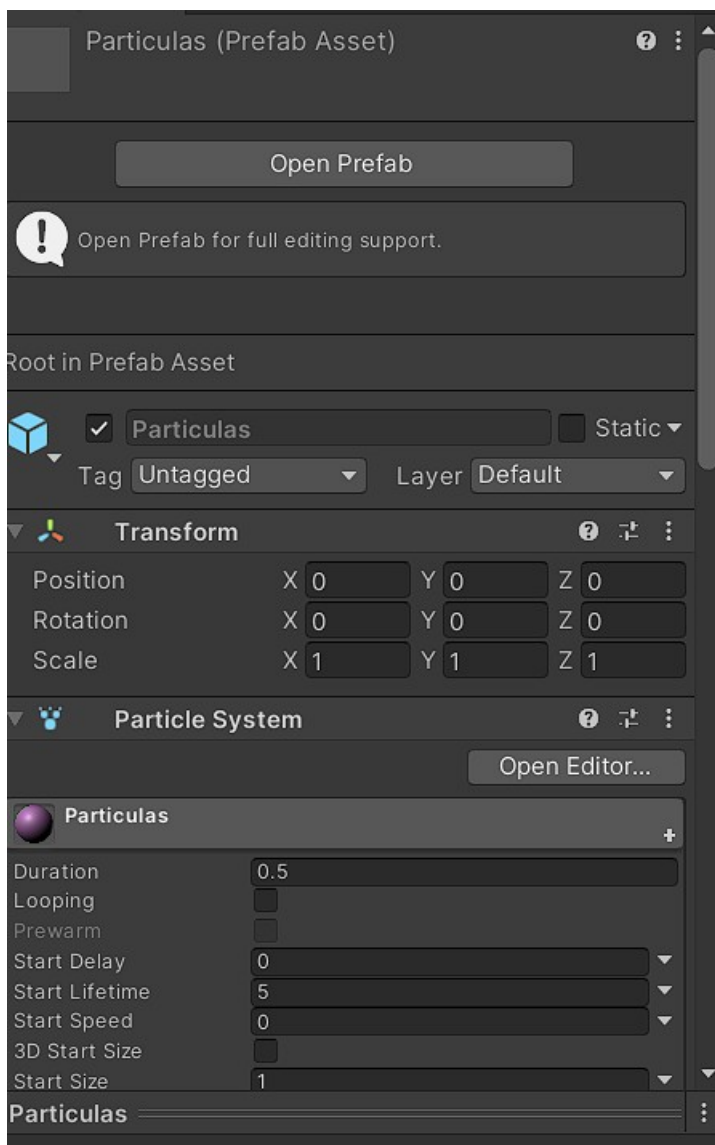
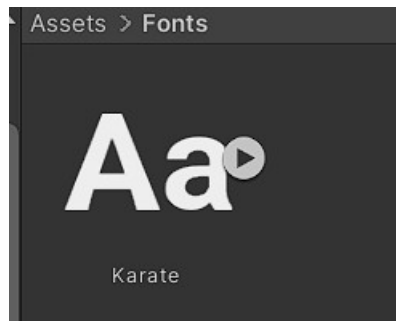
```
C# SonidosPelota.cs X
Assets > Codigo > C# SonidosPelota.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class SonidosPelota : MonoBehaviour {
5
6      public AudioSource rebote;
7      public AudioSource punto;
8
9      void OnCollisionEnter(Collision otro)
10     {
11         if (otro.gameObject.CompareTag("Bloque"))
12         {
13             punto.Play();
14         }
15         else
16         {
17             rebote.Play();
18         }
19     }
20
21 }
```

```
C# Suelo.cs X
Assets > Codigo > C# Suelo.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class Suelo : MonoBehaviour {
5
6      public Vidas vidas;
7
8      void OnTriggerEnter()
9      {
10         vidas.PerderVida();
11     }
12 }
13
```

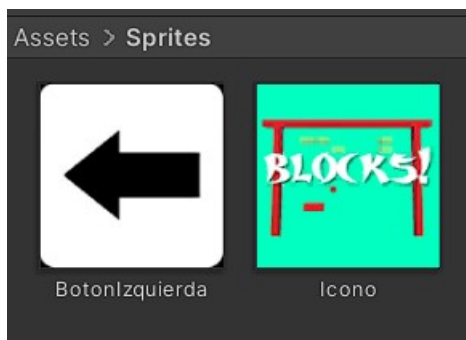
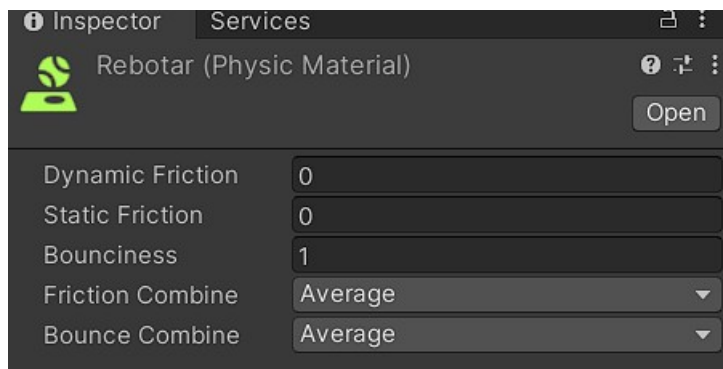
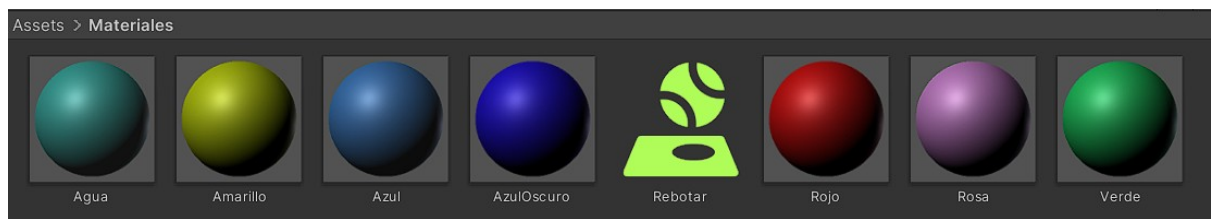
C# Vidas.cs X

Assets > Codigo > C# Vidas.cs

```
1  using UnityEngine;
2  using System.Collections;
3  using UnityEngine.UI;
4
5  public class Vidas : MonoBehaviour {
6
7      public static int vidas = 3;
8
9      public Text textoVidas;
10
11     public Pelota pelota;
12     public Barra barra;
13
14     public GameObject gameOver;
15     public SiguienteNivel siguienteNivel;
16
17     public SonidosFinPartida sonidosFinPartida;
18
19     // Use this for initialization
20     void Start () {
21         ActualizarMarcadorVidas();
22     }
23
24     void ActualizarMarcadorVidas()
25     {
26         textoVidas.text = "Vidas: " + Vidas.vidas;
27     }
28
29     public void PerderVida()
30     {
31         if (vidas <= 0) return;
32
33         Vidas.vidas--;
34         ActualizarMarcadorVidas();
35
36         if (vidas <= 0)
37         {
38             sonidosFinPartida.GameOver();
39             // Mostraremos GameOver
40             gameOver.SetActive(true);
41             pelota.DetenerMovimiento();
42             barra.enabled = false;
43
44             siguienteNivel.nivelACargar = "Portada";
45             siguienteNivel.ActivarCarga();
46
47         }
48         else
49         {
50             barra.Reset();
51             pelota.Reset();
52         }
53     }
54 }
```







### 3.2 Prototipos

El juego tendrá dos prototipos:

- El primero se vio en la entrega del segundo parcial de la materia de programación de videojuegos el cual tenía una cantidad limitada de niveles y funcionalidad.
- Para el segundo prototipo se contara con un nivel mas de dificultad y con mas mecánicas de juego en la fecha del tercer parcial de programación de videojuegos.

### 3.3 Perfiles de Usuarios

El juego al ser del genero arcade no tendrá tipos de usuario solo jugadores sin hacer distinción entre ellos.

Solo iniciar el juego y ganar o perder.

### 3.4 Usabilidad

El juego no tendrá muchas opciones, por lo tanto no sera complicado de entender y podra ser usado incluso por niños pequeños.

### 3.5 Test

Sexo	Mujer
Edad	8
Nivel de estudios	primaria
Aficiones	Jugar videojuegos
<b>Resultados</b>	
Tareas	Puntuación
Jugabilidad	7
Dificultad	8
Control del personaje	8
Guía de usuario	3
Información proporcionada por el juego	2
Diseño visual	7
Coherencia	8

Sexo	Hombre
Edad	19
Nivel de estudios	Universitarios
Aficiones	Jugar videojuegos

Resultados	
Tareas	Puntuación
Jugabilidad	9
Dificultad	5
Control del personaje	6
Guía de usuario	1
Información proporcionada por el juego	1
Diseño visual	6
Coherencia	4

Sexo	Mujer
Edad	15
Nivel de estudios	Secundaria
Aficiones	Ver películas, series, anime, leer manga

Resultados	
Tareas	Puntuación
Jugabilidad	7
Dificultad	5
Control del personaje	6
Guía de usuario	0
Información proporcionada por el juego	0
Diseño visual	8
Coherencia	5

La aplicación va a tener tres versiones:

1. Una versión web que estará publicada en ITCH.io y en Github.
2. Una versión para Windows que se podrá descargar desde ITCH.io.
3. Una versión para Mac que se podrá descargar desde ITCH.io.

**Enlace de github:**

**[https://github.com/jordaly/tarea\\_semana\\_11\\_videojuegos](https://github.com/jordaly/tarea_semana_11_videojuegos)**