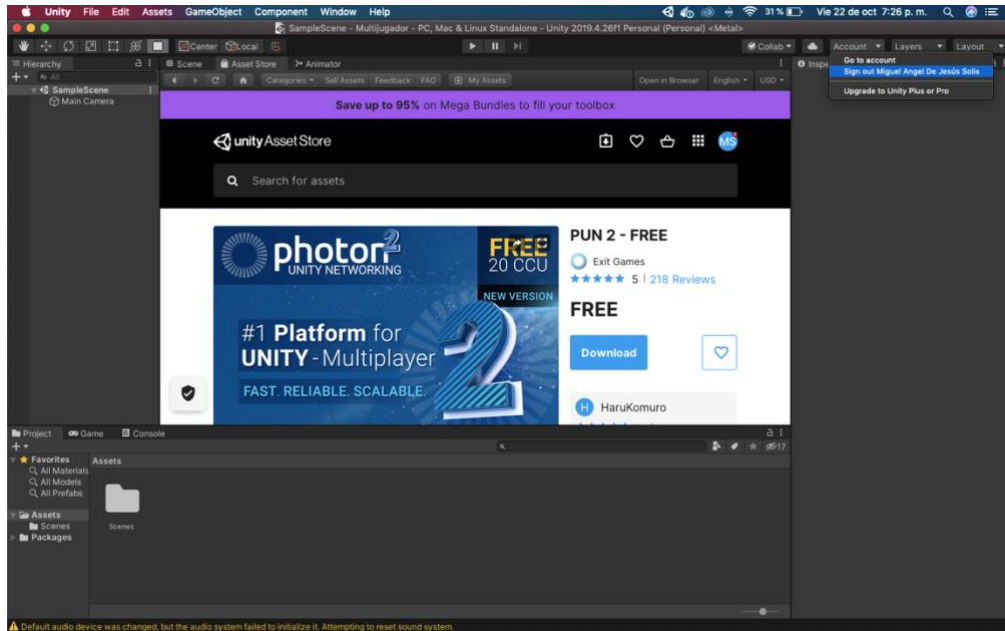
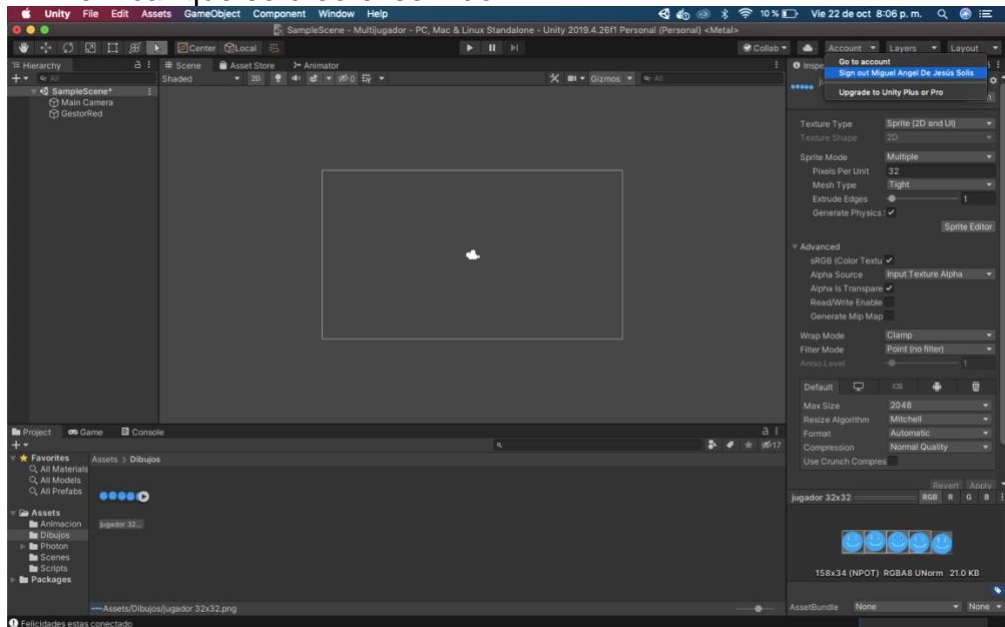


Evidencia Unity juego Multijugador Online 2D

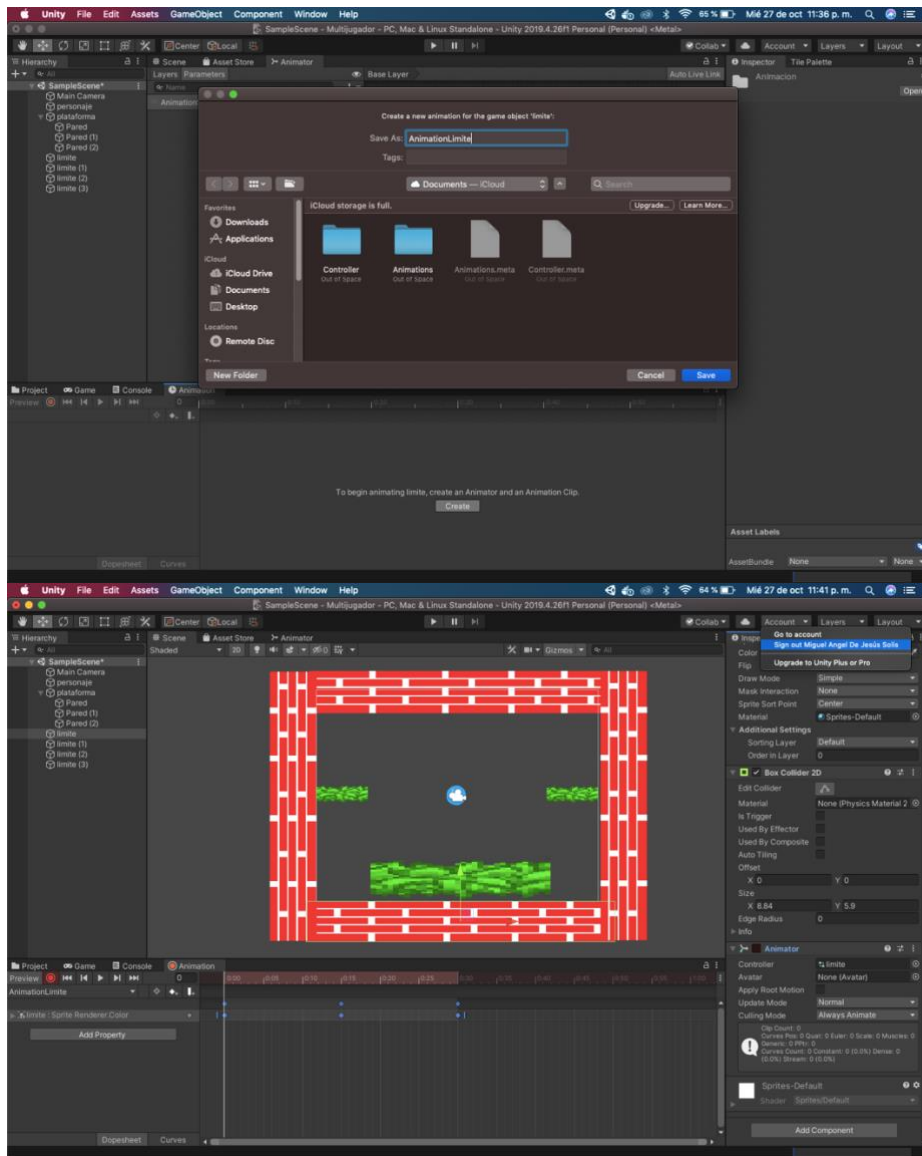
- Creación del servidor



- Verificar que se creo el servidor



- Creación de animación del personaje



- Script de Lógica del Juego el movimiento del personaje

The screenshot shows the Visual Studio IDE with the Explorer pane on the left displaying a project structure. The 'Assets' folder is expanded, showing 'Scripts' and 'LogicaJugador.cs'. The main editor window displays the code for 'LogicaJugador.cs'. The code includes using statements for 'System.Collections', 'System.Collections.Generic', and 'UnityEngine'. It defines a 'public class LogicaJugador : MonoBehaviour' with properties for 'velocidad', 'fuerzaSalto', and 'posX'. It also includes a 'Rigidbody2D' component and a 'Start()' method. The 'Update()' method contains logic for movement and jumping. A 'OnCollisionEnter2D()' method is also present.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 0 references
6 public class LogicaJugador : MonoBehaviour
7 {
8     1 reference
9     public float velocidad;
10    1 reference
11    public float fuerzaSalto;
12    2 references
13    public float posX;
14    1 reference
15    public Rigidbody2D rb;
16    // Start is called before the first frame update
17    0 references
18    void Start()
19    {
20    }
21
22    // Update is called once per frame
23    0 references
24    void Update()
25    {
26        posX = Input.GetAxis("Horizontal");
27        transform.position += (Vector3)new Vector2(posX * velocidad * Time.deltaTime, 0f);
28        if(Input.GetKeyDown(KeyCode.Space)){
29            rb.AddForce(transform.up * fuerzaSalto, ForceMode2D.Impulse);
30        }
31    }
32
33    0 references
34    public void OnCollisionEnter2D(Collision2D collision){
35        if (collectionExtensions.gameObject.tag == "Limite" ){
36        }
37    }
38
39 0 references
```

- Script de objeto limite ya que va cambiando su tamaño

The screenshot shows the Visual Studio IDE with the Explorer pane on the left displaying a project structure. The 'Assets' folder is expanded, showing 'Scripts' and 'CambiarEscala.cs'. The main editor window displays the code for 'CambiarEscala.cs'. The code includes using statements for 'System.Collections', 'System.Collections.Generic', and 'UnityEngine'. It defines a 'public class CambiarEscala : MonoBehaviour' with a property for 'aumentarTamaño'. It includes a 'Start()' method and an 'Update()' method that increases the local scale of the object over time.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 0 references
6 public class CambiarEscala : MonoBehaviour
7 {
8     2 references
9     public float aumentarTamaño;
10    // Start is called before the first frame update
11    0 references
12    void Start()
13    {
14    }
15
16    // Update is called once per frame
17    0 references
18    void Update()
19    {
20        transform.localScale += new Vector3(aumentarTamaño, aumentarTamaño, 1) * Time.deltaTime;
21    }
22
23 0 references
```

- Script Gestor de photon es en donde se pone en uso

The screenshot shows the Visual Studio IDE with the Explorer pane on the left displaying a project structure for a multi-player game. The main editor window shows the `GestorPhoton.cs` script. The script includes the following code:

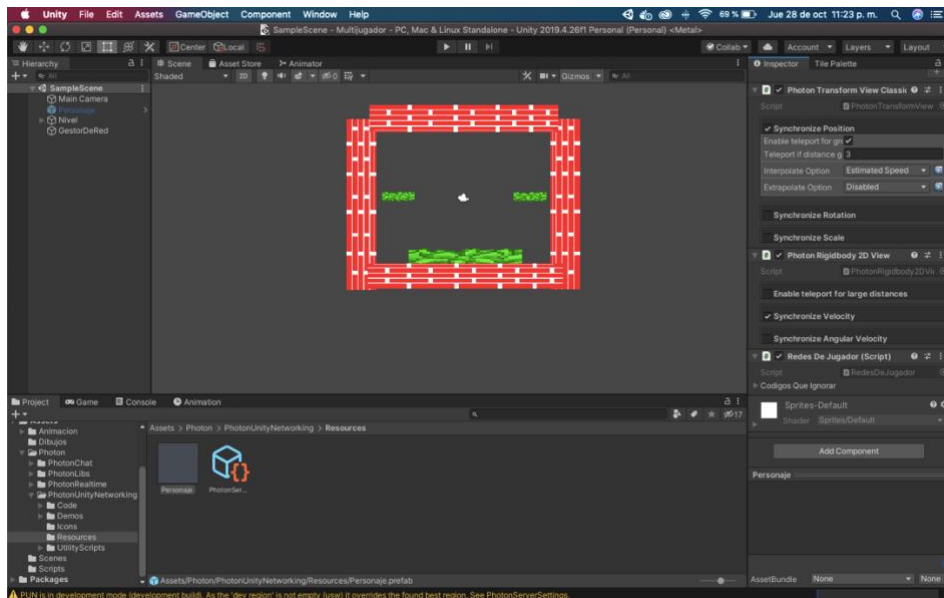
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 using Photon.Pun;
6 using Photon.Realtime;
7
8
9 0 references
10 public class GestorPhoton : MonoBehaviourPunCallbacks {
11     // Start is called before the first frame update
12     0 references
13     void Start()
14     {
15         PhotonNetwork.ConnectUsingSettings();
16     }
17
18     // Update is called once per frame
19     0 references
20     void Update()
21     {
22     }
23
24     0 references
25     public override void OnConnectedToMaster()
26     {
27         PhotonNetwork.JoinLobby();
28     }
29
30     0 references
31     public override void OnJoinedLobby()
32     {
33         PhotonNetwork.JoinOrCreateRoom("Cuarto", new RoomOptions {MaxPlayers = 2}, TypedLobby.Default);
34     }
35
36     0 references
37     public override void OnJoinedRoom()
38     {
39         PhotonNetwork.Instantiate("Personaje", new Vector2(Random.Range(-7f,7f),2), Quaternion.Identity);
40     }
41 }
```

- Script Redes de Jugador

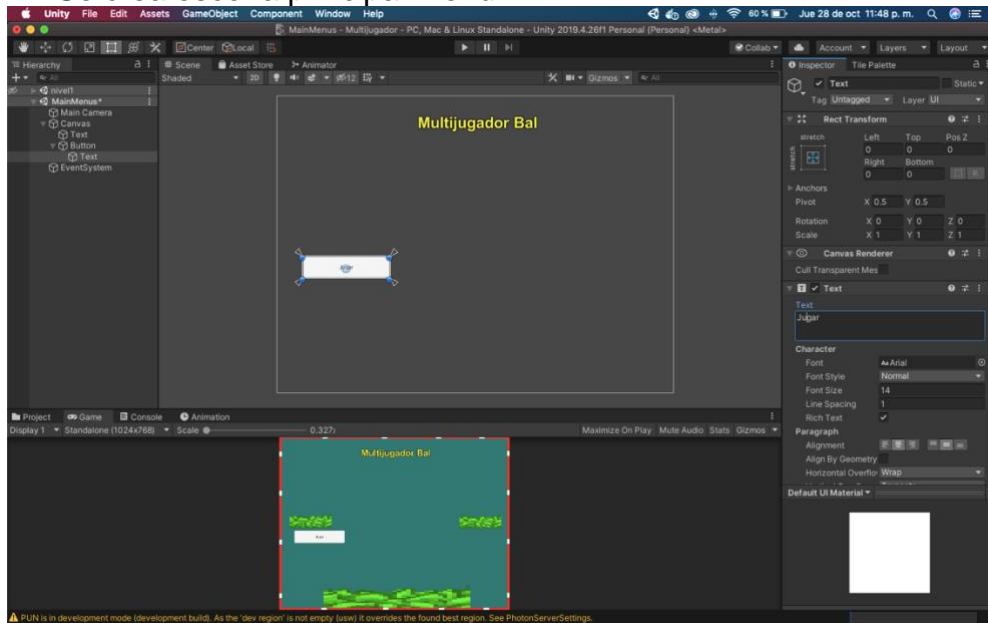
The screenshot shows the Visual Studio IDE with the Explorer pane on the left displaying a project structure. The main editor window shows the `RedesDeJugador.cs` script. The script includes the following code:

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using Photon.Pun;
5
6 0 references
7 public class RedesDeJugador : MonoBehaviour
8 {
9     1 reference
10     public MonoBehaviour[] codigosQueIgnorar;
11
12     2 references
13     private PhotonView photonView;
14     // Start is called before the first frame update
15     0 references
16     void Start()
17     {
18         photonView = GetComponent<PhotonView>();
19         if(!photonView.IsMine){
20             foreach (var codigo in codigosQueIgnorar){
21                 codigo.enabled=false;
22             }
23         }
24
25         // Update is called once per frame
26         0 references
27         void Update()
28         {
29         }
30     }
31 }
```

- El codigo copia de los scrips



• Se crea escena principal menú



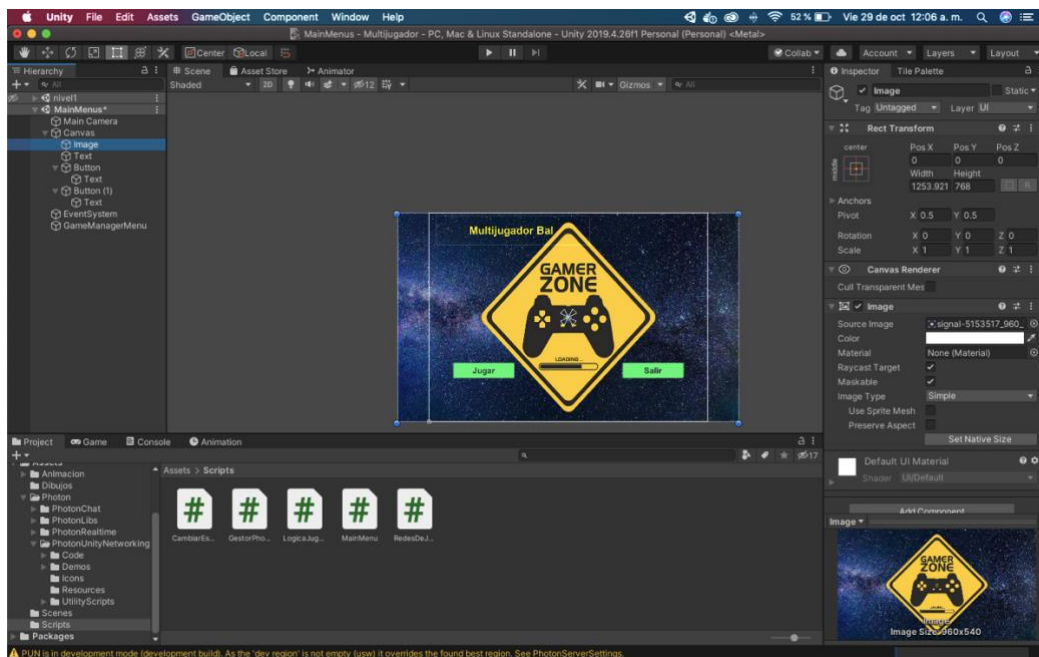
- Script del menú

```

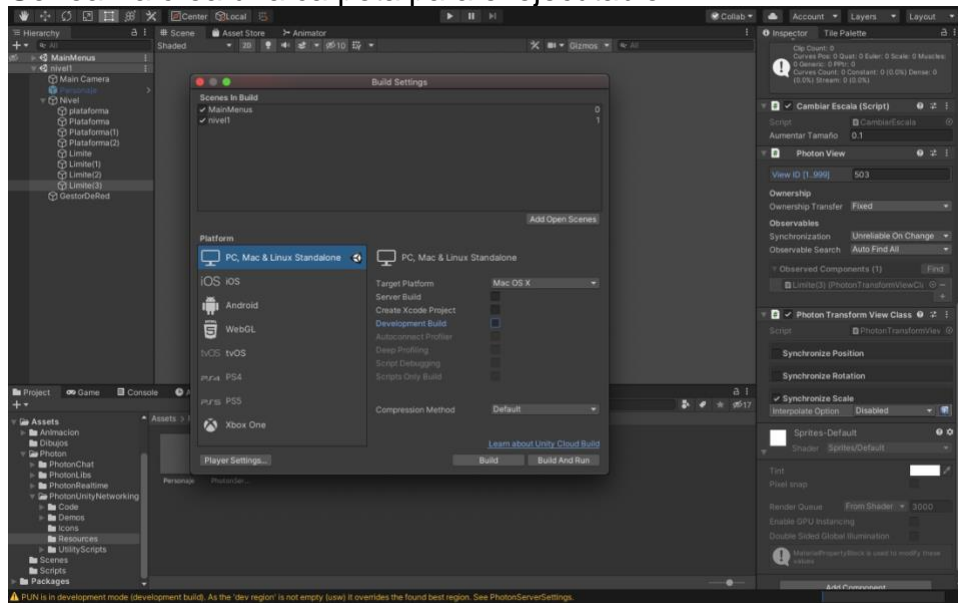
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class MainMenu : MonoBehaviour
7  {
8      // Start is called before the first frame update
9      void Start()
10     {
11     }
12
13     // Update is called once per frame
14     void Update()
15     {
16     }
17
18     public void EscenaJuego(){
19         SceneManager.LoadScene("nivel1");
20     }
21
22     public void Salir(){
23         Application.Quit();
24     }
25
26
27

```

- La escena con imagen de fondo



- Se realiza crea una carpeta para el ejecutable



- Se muestra el ejecutable en la biblioteca



- Juego funcionando con el ejecutable

