

```

using System.Collections;
using System.Collections.Generic;
using System.Text.RegularExpressions;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
using Photon.Pun;
using PlayFab;
using PlayFab.ClientModels;

```

```

public class ControlMenus : MonoBehaviourPunCallbacks

```

```

{
    private bool todoOk;
    public Text feedback;
    public Text textoBoton;
    public Button botonRegistrar;
    //public Text userName, userEmail, userPassword, userRepeatPassword;
    public InputField userName, userEmail, userPassword, userRepeatPassword;
    public InputField nombreJugador;
    public GameObject panelNuevoNombre;
    public GameObject panelLeaderBoard;
    //=====ESCENA CREAR USUARIO=====
    public void ToInicioSesion()
    {
        PhotonNetwork.Disconnect();
        SceneManager.LoadScene("InicioSesion(1)");
    }

    public void ToCrearUsuario() {
        SceneManager.LoadScene("NuevoUsuario(0)");
    }

    public void ToLobby() {
        PhotonNetwork.LeaveRoom();
        SceneManager.LoadScene("Rooms(3)");
    }

    public void ComprobarDatos()
    {
        feedback.gameObject.SetActive(false);
        todoOk = true;
        //PASSWORD
        if (userPassword.text.Length < 8)
        {
            if (userPassword.text.Length > 0)
            {
                userPassword.transform.GetChild(1).GetComponent<Text>().color = Color.red;
                todoOk = false;
                MostrarFeedback("La contrase❖a debe tener m❖s de 8 caracteres");
            }
        }
        else if (!userPassword.text.Equals(userRepeatPassword.text))
        {
            if (userPassword.text.Length > 0 && userRepeatPassword.text.Length > 0)
            {
                userPassword.transform.GetChild(1).GetComponent<Text>().color = Color.red;
                userRepeatPassword.transform.GetChild(1).GetComponent<Text>().color = Color.red;
                todoOk = false;
                MostrarFeedback("Las contrase❖as no coinciden");
            }
        }
        else
        {

```

```

userPassword.transform.GetChild(1).GetComponent<Text>().color = Color.black;
userRepeatPassword.transform.GetChild(1).GetComponent<Text>().color = Color.black;
}

```

//CORREO

```

Regex regex = new Regex(@"^([w\.-]+)@([w\.-]+)(\.[w]{2,3})+$");
Match match = regex.Match(userEmail.text);
if (!match.Success)
{
    if (userEmail.text.Length > 0)
    {
        userEmail.transform.GetChild(1).GetComponent<Text>().color = Color.red;
        todoOk = false;
        MostrarFeedback("Introduce un correo v❌lido");
    }
}
else
{
    userEmail.transform.GetChild(1).GetComponent<Text>().color = Color.black;
}

```

//NOMBRE

```

if (userName.text.Length < 4)
{
    if (userName.text.Length > 0)
    {
        userName.transform.GetChild(1).GetComponent<Text>().color = Color.red;
        todoOk = false;
        MostrarFeedback("El nombre debe de tener más de 4 caracteres");
    }
}
else
{
    userName.transform.GetChild(1).GetComponent<Text>().color = Color.black;
}
}

```

```

public void MostrarFeedback(string texto)
{
    feedback.gameObject.SetActive(true);
    feedback.text = texto;
}

```

```

public void SalirJuego() {
    Application.Quit();
}

```

```

/*
string Encryptar(string pass) {
    System.Security.Cryptography.MD5CryptoServiceProvider x = new
System.Security.Cryptography.MD5CryptoServiceProvider();
    byte[] bs = System.Text.Encoding.UTF8.GetBytes(pass);
    bs = x.ComputeHash(bs);
    System.Text.StringBuilder s = new System.Text.StringBuilder();
    foreach (byte b in bs) {
        s.Append(b.ToString("x2").ToLower());
    }
    return s.ToString();
}
*/

```

```

public void CrearUsuario()
{
    if (!userPassword.text.Equals("") && !userRepeatPassword.text.Equals("") && !userName.text.Equals("") &&
!userEmail.text.Equals(""))
    {
        var requestRegistro = new RegisterPlayFabUserRequest { Email = userEmail.text, Password = userPassword.text,

```

```

Username = userName.text };
    PlayFabClientAPI.RegisterPlayFabUser(requestRegistro, RegisterSuccess, RegisterError);
}
else {
    MostrarFeedback("Rellena todos los campos");
}
}

public void RegisterSuccess(RegisterPlayFabUserResult result) {
    ToInicioSesion();
}

public void RegisterError(PlayFabError error) {
    MostrarFeedback(error.GenerateErrorReport());
}

public void IniciarSesion() {
    var requestRegistro = new LoginWithEmailAddressRequest { Email = userEmail.text, Password = userPassword.text,
InfoRequestParameters = new GetPlayerCombinedInfoRequestParams {
    GetPlayerProfile = true
}
}};
    PlayFabClientAPI.LoginWithEmailAddress(requestRegistro, LoginSuccess, RegisterError);
}
public void LoginSuccess(LoginResult result)
{
    string nombre = null;
    if (result.InfoResultPayload.PlayerProfile.DisplayName != null)
    {
        nombre = result.InfoResultPayload.PlayerProfile.DisplayName;
        Debug.Log(nombre);
        OnClickConnect(nombre);
    }
    else {
        panelNuevoNombre.SetActive(true);
    }
}

public void OnClickConnect(string nombre)
{
    if (userEmail.text.Length > 0)
    {
        PhotonNetwork.NickName = nombre;
        textoBoton.text = "Conectando...";
        PhotonNetwork.AutomaticallySyncScene = true;
        PhotonNetwork.ConnectUsingSettings();
    }
}

public override void OnConnectedToMaster()
{
    SceneManager.LoadScene("Rooms(3)");
}

public void NombreEstablecido() {
    var request = new UpdateUserTitleDisplayNameRequest()
    {
        DisplayName = nombreJugador.text
    };
    PlayFabClientAPI.UpdateUserTitleDisplayName(request, OnDisplayNameUpdate, RegisterError);
}

void OnDisplayNameUpdate(UpdateUserTitleDisplayNameResult result) {
    OnClickConnect(result.DisplayName);
}

```

```
}

public void AbrirLeaderBoard() {
    panelLeaderBoard.SetActive(true);
}

public void ReiniciarPassword() {
    var request = new SendAccountRecoveryEmailRequest
    {
        Email = userEmail.text,
        TitleId = "DB976"
    };

    PlayFabClientAPI.SendAccountRecoveryEmail(request, OnPasswordReset, OnResetError);
}

void OnPasswordReset(SendAccountRecoveryEmailResult result) {
    MostrarFeedback("Revisa tu correo electrónico");
}

void OnResetError(PlayFabError error) {
    MostrarFeedback(error.GenerateErrorReport());
}
}
```