

SCRIPT NUMERO 1 (LENSMANAGER)

LIBRERÍAS UTILIZADAS

```
using UnityEngine;  
using UnityEngine.UI;
```

```
public class ManagerLens : MonoBehaviour //SE DECLARA LA CLASE
```

```
{  
    [SerializeField] private GameObject m_lente1 = null; //SE ASIGNA EL OBJETO (EL  
    MODELO DEL LENTE)
```

```
  
    [SerializeField] private GameObject m_lente1_menu = null; //SE ASIGNA EL OBJETO  
    (EL MENU DE LA INTERFAZ GRAFICA)
```

```
  
    [SerializeField] private GameObject m_lente1_but = null; //SE ASIGNA EL OBJETO (EL  
    BOTON QUE HARA LA FUNCION DE MOSTRAR EL MENU)
```

```
public void showlente1() //METODO PARA HACER EL CAMBIO DEL MODELO  
SELECCIONADO
```

```
{  
    m_lente1.SetActive(true); //SE ACTIVA EL MODELO  
    m_lente1_menu.SetActive(false); //SE DESACTIVA EL MENU CUANDO SE CAMBIE  
    DE MODELO  
    m_lente1_but.SetActive(true); //VUELVE A ACTIVARSE EL BOTON PARA ABRIR EL  
    MENU  
}
```

```
public void showlente1menu() //METODO PARA EL MENÚ CON LOS MODELOS
```

```
{  
    m_lente1.SetActive(true); //EL MODELO QUEDA ACTIVADO  
    m_lente1_menu.SetActive(true); //SE MUESTRA EL MENU CON LOS MODELOS  
    DISPONIBLES  
    m_lente1_but.SetActive(false); //SE OCULTA EL BOTON QUE ABRE EL MENU DE  
    LOS MODELOS  
}  
}
```

SCRIPT NUMERO 2 (UIMANAGER)

LIBRERÍAS UTILIZADAS

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using UnityEngine.SceneManagement; //USO PARA EL CAMBIO DE ESCENAS Y
ADMINISTRARLAS
```

```
public class UIManager : MonoBehaviour //SE DECLARA LA CLASE
```

```
{
```

```
    public int cambioEscena; //SE DECLARA EL CAMBIO DE ESCENA EN TIPO INT YA QUE
    ES EN BASE AL NUMERO DE ESCENAS
```

```
void Start() //SE ACTIVA EL METODO
```

```
{
```

```
    // GameManager.instance.OnMainMenu += ActivateMainMenu;
    // GameManager.instance.OnItemsMenu += ActivateItemsMenu;
    // GameManager.instance.OnAirPosition += ActivateArPosition;
```

```
}
```

```
public void CambiarEscena(int indice) //METODO PARA EL CAMBIO DE ESCENA (TIPO
ENTERO CON INDICES ASIGNADOS POR EL MISMO UNITY)
```

```
{
```

```
    SceneManager.LoadScene(cambioEscena); //EL SCENEMANAGER CARGA LA ESCENA
    QUE SE LE ESPECIFIQUE
```

```
}
```

```
}
```

SCRIPT NUMERO 3 (NETWORKSMANAGER)

LIBRERIAS UTILIZADAS

```
using UnityEngine;
using System;
using System.Collections;
```

```
public class NetworkManager : MonoBehaviour //SE DECLARA LA CLASE
{
    public void createUser(String email, String pass, Action<Response> response) //SE DECLARA EL METODO PARA CREAR LOS USUARIOS
    {
        StartCoroutine(CO_CreateUser(email, pass, response)); //SE DECLARAN LOS CAMPOS QUE UTILIZAREMOS PARA EL REGISTRO
    }
}
```

```
private IEnumerator CO_CreateUser(String email, String pass, Action<Response> response) //SE DECLARAN LOS TIPOS PARA LOS CAMPOS REQUERIDOS EN EL REGISTRO
{
    WWWForm form = new WWWForm(); //METODO PARA UTILIZARSE EN FORMULARIO WEB
    form.AddField("email", email); //SE DECLARA EL CAMPO PARA EL CORREO O USUARIO
    form.AddField("pass", pass); //SE DECLARA EL CAMPO PARA LA CONTRASEÑA (SE DELCARA POR AUTOMATICO QUE ES FORMATO DE PASSWORD)
}
```

```
WWW w = new WWW("http://localhost/lensuser/createUser.php", form); //ENLACE PARA LA BASE DE DATOS DONDE SE ALMACENARAN LOS USUARIOS REGISTRADOS
```

```
yield return w;
```

```
response(JsonUtility.FromJson<Response>(w.text)); //METODO RETURN QUE VERIFICA QUE EL USUARIO FUE REGISTRADO CORRECTAMENTE
```

```
}
```

```
}
```

```
[Serializable]
```

```
public class Response //CLASE QUE SE ENCARGARA PARA EL EVENTO DEL FORMULARIO
```

```
{
```

```
    public bool done = false;
```

```
    public String messege = ""; //MENSAJE DE AVISO AL USUARIO
```

```
}
```

SCRIPT NUMERO 4 (USERSMANAGER)

LIBRERIAS UTILIZADAS

```
using UnityEngine;  
using UnityEngine.UI;
```

```
public class UsersManager : MonoBehaviour //DECLARACION DE LA CLASE  
{  
    [SerializeField] private GameObject m_registerUI = null; //OBJETO QUE SERA LA  
INTERFAZ DE REGISTRO  
    [SerializeField] private GameObject m_loginUI = null; //OBJETO QUE SERA LA  
INTERFAZ DE LOGIN  
    [SerializeField] private InputField m_emailInput = null; //CAMPO DE ENTRADA DEL  
USUARIO  
    [SerializeField] private InputField m_passInput = null; //CAMPO DE ENTRADA DE LA  
PASSWORD DEL USUARIO  
    [SerializeField] private Text m_Text = null; //TEXTO DE AVISO AL USUARIO  
    [SerializeField] private Text m_Textt = null; //TEXTO DE AVISO AL USUARIO DE  
ALGUN ERROR  
  
    private NetworkManager m_NetworkManager = null; //SE DEDCLARA EL CAMPO  
PRIVADO DEL NETWORKMANAGER  
  
    private void Awake() //METODO DE AVISO  
    {  
        m_NetworkManager = GameObject.FindObjectOfType<NetworkManager>(); //SE  
DECLARA EL OBJETO A UTILIZAR PARA LLAMAR LOS METODOS  
CORRESPONDIENTES DEL SCRIPT  
    }  
  
    public void SubmitRegister() //METODO QUE CREA EL REGISTRO DEL USUARIO  
    {  
        if (m_emailInput.text == "" || m_passInput.text == "") //CONDICION DE QUE SI EL  
USUARIO O PASSWORD ESTA VACIO MANDE A LLAMAR UN MENSAJE  
        {  
            m_Text.text = "Favor de rellenar los campos"; //MENSAJE EN CASO DE QUE EL  
USUARIO OLVIDE LLENAR UN CAMPO  
            return;  
        }  
  
        m_NetworkManager.createUser(m_emailInput.text, m_passInput.text, delegate  
(Response response) //METODO QUE CREA EL USUARIO  
        {  
            m_Text.text = "registrado"; //UNA VEZ QUE EL USUARIO INGRESE SU USUARIO  
Y CONTRASEÑA DESEADA SALTARA UN MENSAJE DE REGISTRO  
            return;  
            m_Text.text = response.messege;
```

```

    });
}

public void SubmitLogin() //METODO PARA LOGEAR AL USUARIO
{
    if (m_emailInput.text == "" || m_passInput.text == "") //SI EL USUARIO OLVIDA
LLENAR ALGUN CAMPO SALTARA UN AVISO
    {
        m_Textt.text = "Favor de rellenar los campos"; //MENSAJE DE AVISO PARA QUE
EL USUARIO CORROBORE SUS DATOS CORRECTAMENTE
        return;
    }
}

public void showlogin() //METODO QUE CAMBIARA LA INTERFAZ A LA DE LOGIN Y
DESACTIVA LA DE REGISTRO
{
    m_registerUI.SetActive(false); //SE DESACTIVA LA UI DE REGISTRO
    m_loginUI.SetActive(true); //SE ACTIVA LA UI DE LOGIN
}

public void showRegister() //METODO QUE CAMBIARA LA INTERFAZ DE LOGIN A LA
DE REGISTRO
{
    m_registerUI.SetActive(true); //SE ACTIVA LA UI DE REGISTRO
    m_loginUI.SetActive(false); //SE DESACTIVA LA UI DE LOGIN
}
}

```