

Prototipo **DEL FILTRO**

Prototipo del filtro de dinosaurios en el
exterior



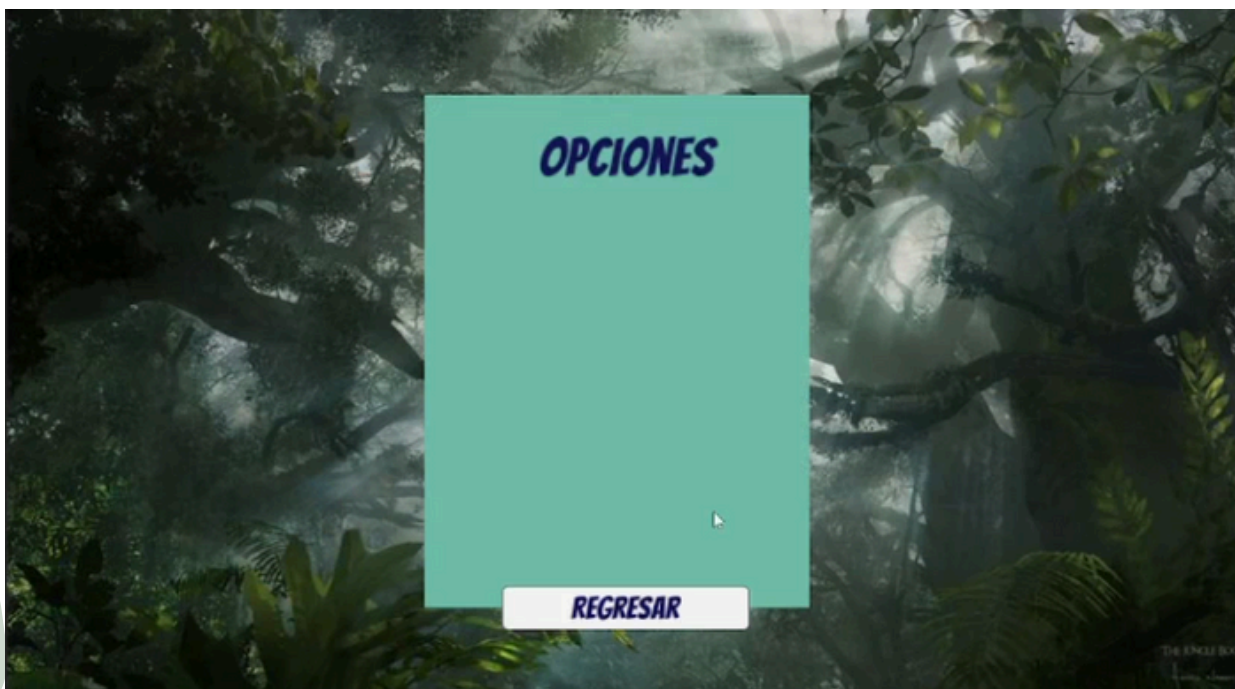
Lo primero que hicimos en este proyecto fue la interfaz de la aplicación.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class Opciones_menu : MonoBehaviour
7 {
8     public void EmpezarVisualización(string Visualizacion_dino)
9     {
10         SceneManager.LoadScene (Visualizacion_dino);
11     }
12
13     public void Salir()
14     {
15         Application.Quit();
16         Debug.Log("Gracias por visitarnos");
17     }
18 }
19
```

[22:19:55] Gracias por visitarnos
UnityEngine.Debug:Log (object)

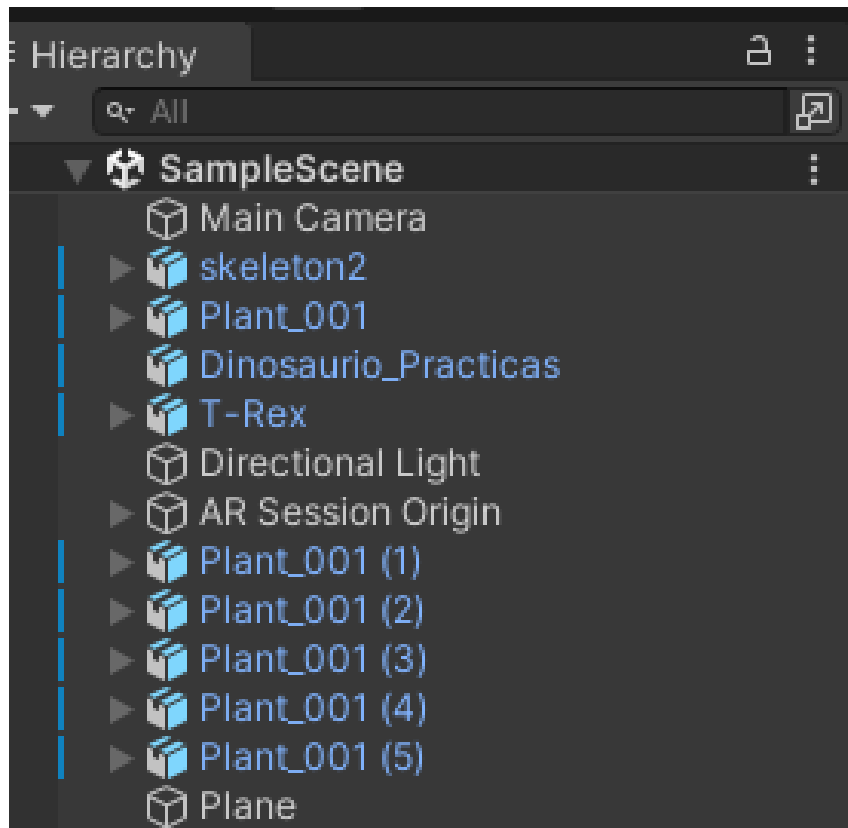
Salida

Para obtener un visual así:



Para este proyecto utilizamos, Unity, Visual Studio y ARCode.

Lo primero que hicimos fue adjuntar



AR Session Origin, y a este AR Camera.

Despues de esto añadimos los modelos 3D que usaremos para este filtro.

Preparado eso, lo que proseguimos a hacer fue el código, para programar que al ver una superficie plana y tocar la pantalla, salieran los modelos 3D.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.XR.ARSubsystems;

public class SurfaceDetectionController : MonoBehaviour
{
    public GameObject DinoToPlace; //el objeto 3D que quiero colocar en la superficie
    private ARRaycastManager raycastManager; // Manejador de rayos para detectar superficies
    private int layerMask; // Máscara de capa para raycast

    void Start()
    {
        // Obtener referencia al ARRaycastManager
        raycastManager = GetComponent<ARRaycastManager>();
        // Configurar la máscara de capa para raycast
        layerMask = LayerMask.GetMask("AR GameObject");
    }

    void Update()
    {
        // Si se toca la pantalla (o se hace clic con el ratón), intenta colocar el objeto en la superficie detectada
        if (Input.touchCount > 0 && Input.GetTouch(0).phase == TouchPhase.Began)
        {
            // Lanzar un rayo desde el punto del toque en la pantalla
            Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position);
            RaycastHit hit;

            // Si el rayo impacta con un plano detectable, coloca el objeto en esa posición
            if (Physics.Raycast(ray, out hit, Mathf.Infinity, layerMask))
            {
                // Intentar colocar el objeto en el plano detectado
                TryPlaceObject(ray, hit);
            }
        }
    }
}
```

```
37     }
38 }
39 }
40 }
41
42 // Método para intentar colocar el objeto en el plano detectado
43 void TryPlaceObject(Ray ray, RaycastHit hit)
44 {
45     // Lista para almacenar los resultados del raycast
46     List<ARRaycastHit> hits = new List<ARRaycastHit>();
47
48     // Lanzar un rayo desde la posición y dirección de la cámara al plano detectado
49     if (Physics.Raycast(ray, out hit, Mathf.Infinity, layerMask))
50     {
51         // Obtener el primer punto de impacto
52         var hitPose = hits[0].pose;
53
54         // Instanciar el objeto en la posición y rotación del punto de impacto
55         Instantiate(DinoToPlace, hitPose.position, hitPose.rotation);
56     }
57 }
58
59 }
```