

```
// Importamos las librerías necesarias para trabajar con colecciones y elementos visuales en Unity
```

```
using System.Collections;
```

```
using System.Collections.Generic;
```

```
using UnityEngine;
```

```
// Definimos una clase pública llamada "Grafdatosbolos" que hereda de MonoBehaviour
```

```
// Esto permite que el script se pueda asociar a un objeto dentro de la escena en Unity
```

```
public class Grafdatosbolos : MonoBehaviour
```

```
{
```

```
    // Declaramos tres variables públicas de tipo GameObject que serán referenciadas en el editor de Unity
```

```
    // Estas variables representan las gráficas que se mostrarán al usuario
```

```
    [SerializeField] private GameObject graficaMpuAbol; // Gráfica de los datos del sensor MPU para la opción "Abol"
```

```
    [SerializeField] private GameObject graficaMpuEbol; // Gráfica de los datos del sensor MPU para la opción "Ebol"
```

```
    [SerializeField] private GameObject graficaEMGbol; // Gráfica de los datos del sensor EMG para la opción "Bolos"
```

```
    // Declaramos una variable privada para llevar un registro de la gráfica actualmente activa
```

```
private GameObject pantallaActual;
```

```
// Método Start que se ejecuta automáticamente cuando la escena comienza o  
cuando el script se inicializa
```

```
private void Start()
```

```
{
```

```
// Establece el estado inicial de la pantalla activa, configurándola a la gráfica  
"MPU Ebol"
```

```
pantallaActual = graficaMpuEbol;
```

```
// Llama al método para validar que todas las referencias a las gráficas estén  
correctamente asignadas
```

```
ValidarReferencias();
```

```
}
```

```
// Método para validar que todas las referencias de las gráficas han sido asignadas  
en el inspector de Unity
```

```
private void ValidarReferencias()
```

```
{
```

```
// Si alguna de las referencias de gráficas está vacía (es decir, no se ha asignado  
en el inspector), se muestra un mensaje de error
```

```
if (graficaMpuAbol == null || graficaMpuEbol == null || graficaEMGbol == null)
```

```
{  
  
    Debug.LogError("Faltan referencias de gráficas en " + gameObject.name);  
  
}  
  
}
```

// Método privado para cambiar entre las gráficas. Se desactiva la gráfica actual y se activa la nueva

```
private void CambiarGrafica(GameObject graficaNueva)
```

```
{  
  
    // Si la gráfica nueva es nula, no hacemos nada
```

```
    if (graficaNueva == null) return;
```

```
    // Si ya hay una pantalla activa, la desactivamos
```

```
    if (pantallaActual != null)
```

```
{  
  
        pantallaActual.SetActive(false);
```

```
}
```

```
    // Activamos la nueva gráfica
```

```
graficaNueva.SetActive(true);
```

```
// Actualizamos la variable pantallaActual para que ahora apunte a la nueva  
gráfica activa
```

```
pantallaActual = graficaNueva;
```

```
}
```

```
// Métodos públicos que se llamarán desde los botones en la interfaz de usuario de  
Unity
```

```
// Cada uno de estos métodos cambia la gráfica que se muestra al usuario
```

```
// Método que cambia a la gráfica "MPU Abol"
```

```
public void MostrarMpuAbol()
```

```
{
```

```
    CambiarGrafica(graficaMpuAbol);
```

```
}
```

```
// Método que cambia a la gráfica "MPU Ebol"
```

```
public void MostrarMpuEbol()
```

```
{
```

```
CambiarGrafica(graficaMpuEbol);

}

// Método que cambia a la gráfica "EMG Bolos"

public void MostrarEMGbol()

{

    CambiarGrafica(graficaEMGbol);

}

}
```