

```
// Importa las librerías necesarias de Unity
```

```
using System.Collections;
```

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

```
using UnityEngine;
```

```
// Define una clase pública que hereda de MonoBehaviour
```

```
public class basededatosinterc : MonoBehaviour
```

```
{
```

```
    // Referencia al objeto que contiene la gráfica del sensor MPU en el antebrazo
```

```
    [SerializeField] private GameObject graficaMpuAbol;
```

```
    // Referencia al objeto que contiene la gráfica del sensor MPU en el  
    hombro/escápula
```

```
    [SerializeField] private GameObject graficaMpuEbol;
```

```
    // Referencia al objeto que contiene la gráfica del sensor EMG
```

```
    [SerializeField] private GameObject graficaEMGbol;
```

```
    // Referencia al objeto que contiene la pantalla de base de datos
```

```
    [SerializeField] private GameObject basededatos;
```

```
// Objeto actualmente visible en pantalla
```

```
private GameObject pantallaActual;
```

```
// Se ejecuta una vez al iniciar el script
```

```
private void Start()
```

```
{
```

```
    // Define que la pantalla visible al inicio será la de base de datos
```

```
    pantallaActual = basededatos;
```

```
    // Valida que todas las referencias necesarias estén correctamente asignadas
```

```
    ValidarReferencias();
```

```
}
```

```
// Método privado que verifica que todas las referencias hayan sido asignadas en  
el editor
```

```
private void ValidarReferencias()
```

```
{
```

```
    // Si falta alguna referencia, lanza un error en la consola
```

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

```
{
```

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

```
}
```

```
}
```

```
// Método privado para cambiar la pantalla actual por una nueva gráfica
```

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

```
{
```

```
    // Si la nueva gráfica es nula, salir del método sin hacer nada
```

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

```
    // Si hay una pantalla visible actualmente, desactívala
```

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

```
{
```

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

```
}
```

```
// Activa la nueva pantalla y actualiza el estado actual

graficaNueva.SetActive(true);

pantallaActual = graficaNueva;

}
```

// Método público que se puede llamar desde un botón para mostrar la gráfica del MPU en el antebrazo

```
public void MostrarMpuAbol()

{

    CambiarGrafica(graficaMpuAbol);

}
```

// Método público que se puede llamar desde un botón para mostrar la gráfica del MPU en el hombro/escápula

```
public void MostrarMpuEbol()

{

    CambiarGrafica(graficaMpuEbol);

}
```

// Método público que se puede llamar desde un botón para mostrar la gráfica del EMG

```
public void MostrarEMGbol()
```

```
{
```

```
    CambiarGrafica(graficaEMGbol);
```

```
}
```

// Método público que se puede llamar desde un botón para mostrar la pantalla de base de datos

```
public void Mostrarbase()
```

```
{
```

```
    CambiarGrafica(basededatos);
```

```
}
```

```
}
```