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

public abstract class Accion {

    public delegate void Act();

    public Stage stg = Stage.elegir;
    public Act ac;

    public static Accion CreateAccion(string accion,Monstruo target)
    {
        Type types = Type.GetType(accion);

        if (types == null)
            throw new InvalidOperationException("The given action does not have a Type associated with it.");

        return Activator.CreateInstance(types,target) as Accion;
    }
    public static Accion CreateAccion(string accion)
    {
        Type types = Type.GetType(accion);

        if (types == null)
            throw new InvalidOperationException("The given action does not have a Type associated with it.");

        return Activator.CreateInstance(types) as Accion;
    }
    public static Accion CreateAccion(string accion, Monstruo source, Monstruo target)
    {
        Type types = Type.GetType(accion);

        if (types == null)
            throw new InvalidOperationException("The given action does not have a Type associated with it.");

        return Activator.CreateInstance(types,source,target) as Accion;
    }
}
```