























Character.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 
5 // Clase genérica para los personajes del juego.
6 // HayeLi Jaqueline Padron Velazquez
7 public abstract class Character : MonoBehaviour
8 {
9     public HitPoints hitPoints; //Puntos actuales de jugador
10    public float maxHitPoints; //Máximos puntos a obtener
11 }
```

Player.cs

```
1 using UnityEngine;
2 // HayeLi Jaqueline Padron Velazquez
3 // Se ha usado Unity 10 referencias
4 public class Player : Character
5 {
6     public HealthBar healthBarPrefab; //Referencia HealthBar Prefab
7     private HealthBar healthBar; //Copia de referencia de HealthBar Prefab
8     // Mensaje de Unity [0 referencias]
9     void Start()
10    {
11         healthBar = Instantiate(healthBarPrefab); //Instanciar HealthBar
12         character = this; //Referencia del Player en HealthBar
13     }
14     // Mensaje de Unity [0 referencias]
15     public void OnTriggerEnter(Collider collision)
16     {
17         if (collision.gameObject.CompareTag("CanBePickedUp"))
18         {
19             Item hitObject = collision.gameObject.GetComponent<Consumable>().item;
20             if (hitObject != null)
21             {
22                 Debug.Log("Nombre: " + hitObject.objectName);
23                 bool shouldDisappear = false;
24                 switch (hitObject.itemType)
25                 {
26                     case Item.ItemType.COIN: //Moneda
27                         shouldDisappear = true;
28                         break;
29                     case Item.ItemType.HEALTH: //Barra de Salud
30                         Debug.Log("Cantidad a Incrementar: " + hitObject.quantity);
31                         shouldDisappear = AdjustHitPoints(hitObject.quantity);
32                         break;
33                     default:
34                         break;
35                 }
36                 if (shouldDisappear)
37                 {
38                     collision.gameObject.SetActive(false); //Desaparecer
39                 }
40             }
41         }
42     }
43     // Referencia
44     private bool AdjustHitPoints(int amount)
45     {
46         if (hitPoints.value > maxHitPoints) // no se puede exceder el máximo de puntos
47         {
48             hitPoints.value = hitPoints.value + amount;
49         }
50     }
51 }
```

HealthBar.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI; //Componentes Interfaz Gráfica

public class HealthBar : MonoBehaviour
{
    [HideInInspector]
    public Player character; //Referencia al jugador
    public float maxHitPoints; //Medidor Meter de la salud
    public Text hpText; //Texto en barra de salud
    //Mensaje de Unity (0 referencias)
    void Start()
    {
        character.hitPoints.value = 0;
    }
    //Mensaje de Unity (0 referencias)
    void Update()
    {
        if (character != null)
        {
            //Modifica barra de salud
            meterImage.fillAmount = character.hitPoints.value / character.maxHitPoints;
            //Texto a mostrar
            hpText.text = "HP: " + (meterImage.fillAmount * 100);
        }
    }
}
```



Player.cs

```
using UnityEngine;
//Hay que poner el Button Valores
public class Player : Character
{
    public HealthBar healthBarPrefab; //Referencia HealthBar Prefab
    private HealthBar healthBar; //Copia de referencia de HealthBar Prefab
    //Modificaciones
    //Mensaje de Unity (0 referencias)
    private void Start()
    {
        healthBar = Instantiate(healthBarPrefab); //Instanciar HealthBar
        healthBar.character = this; //Referencia del Player en HealthBar
    }
    //Mensaje de Unity (0 referencias)
    public void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.CompareTag("CanBePickedUp"))
        {
            Item hitObject = collision.gameObject.GetComponent<Consumable>().item;
            if (hitObject != null)
            {
                Debug.Log("Nombre: " + hitObject.objectName);
                bool shouldDisappear = false;
                switch (hitObject.itemType)
                {
                    case Item.ItemType.COIN: //Moneda
                        shouldDisappear = true;
                        break;
                    case Item.ItemType.HEALTH: //Berga de Salud
                        Debug.Log("Cantidad a Incrementar: " + hitObject.quantity);
                        shouldDisappear = AdjustHitPoints(hitObject.quantity);
                        break;
                }
                if (shouldDisappear)
                {
                    collision.gameObject.SetActive(false); //Desaparecer
                }
            }
        }
    }
    //referencia
    private bool AdjustHitPoints(int amount)
    {
        if (hitPoints.value < maxHitPoints) // no se puede exceder el máximo de puntos
        {

```