

---

# Diseño de Sistemas Interactivos

Curso 2023 - 2024

Profesor: José Manuel Velasco

Despacho 309, 3ª planta. Facultad de Informática.

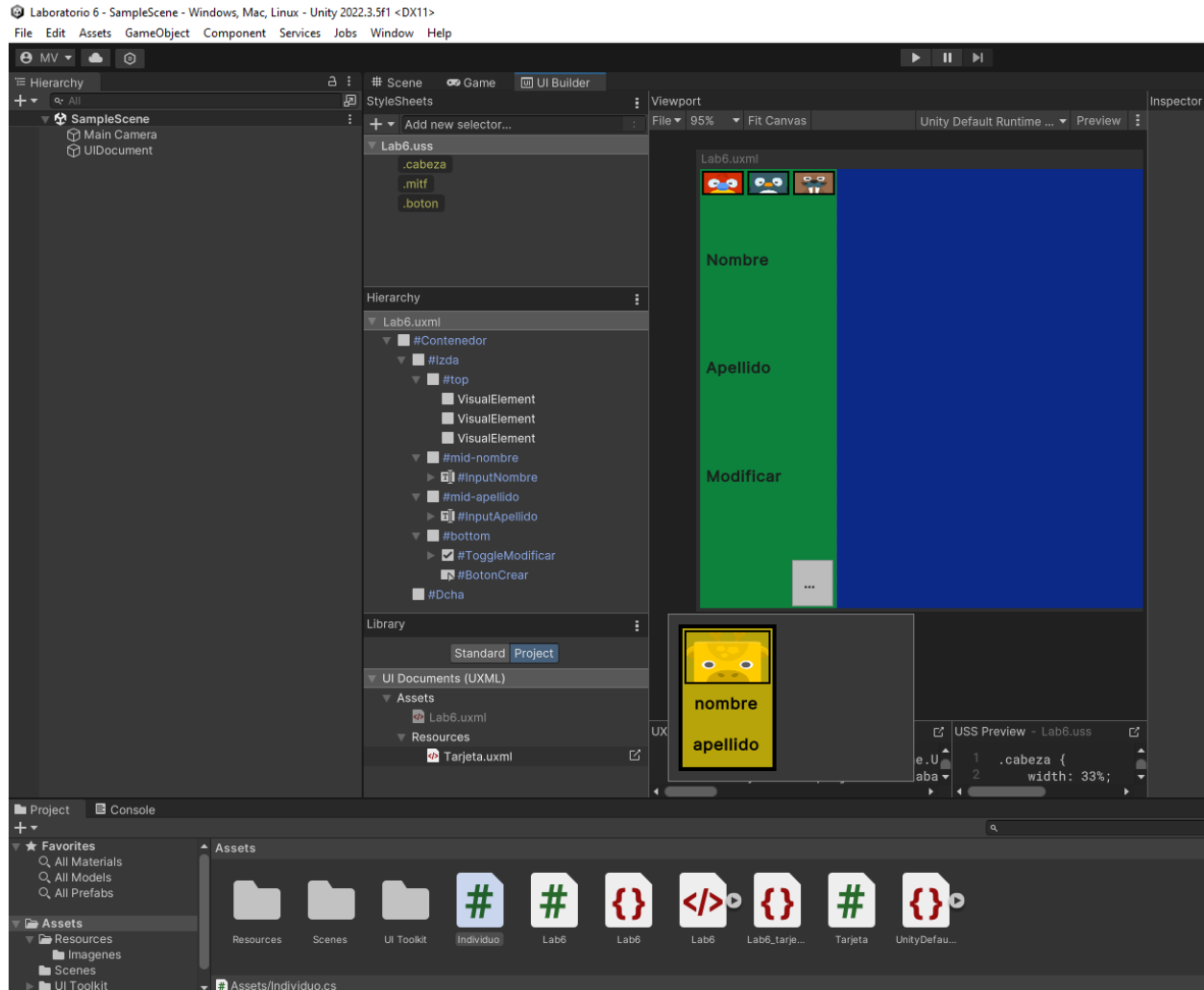
Laboratorio 6: Introducción a Data Binding en UI Toolkit, parte 2.

# Índice

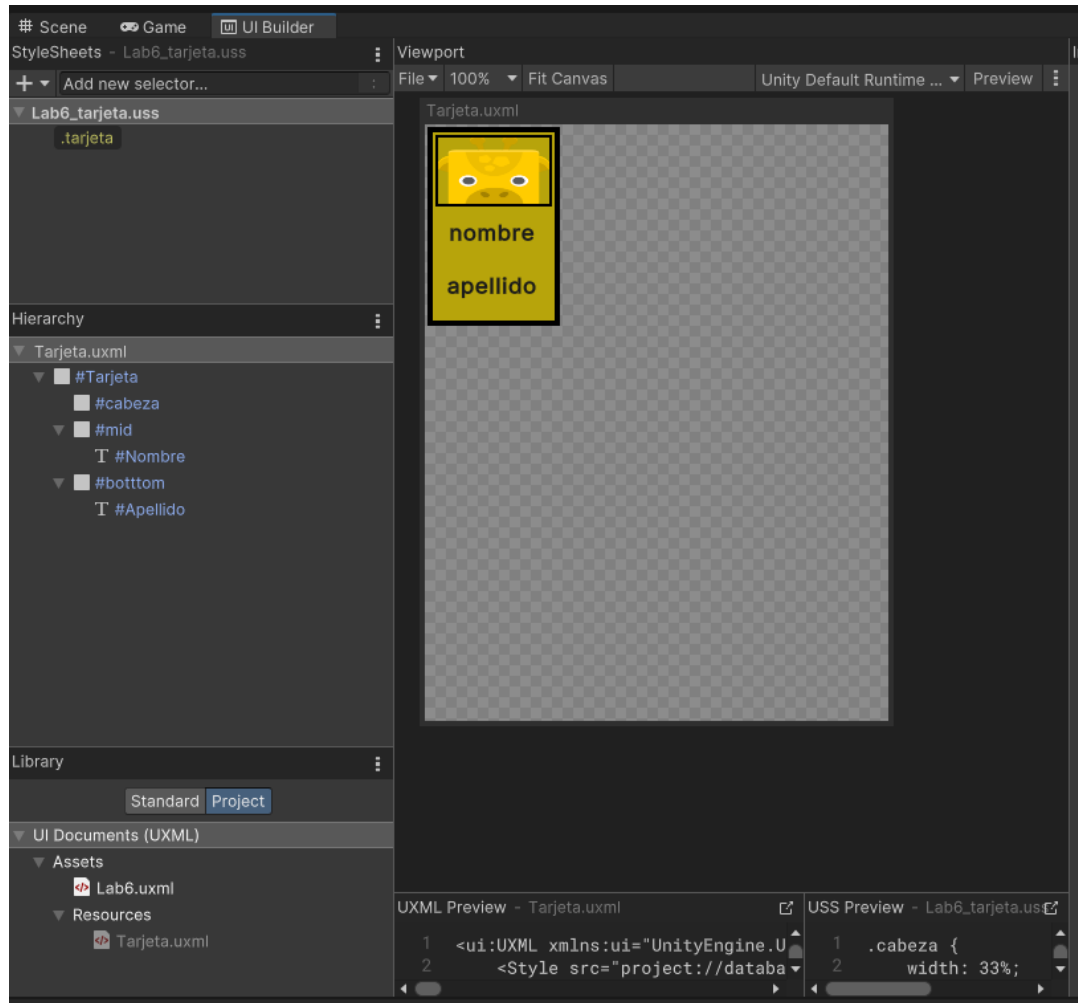
---

- **Creación de nuevos elementos**
- Modificación de elementos
- Modificación de atributos en el Template
- Serialización en formato Json

# Proyecto



# Template → Tarjeta



# Tarjeta

```
Lab6.cs  Tarjeta.cs x  Individuo.cs
Assets > Tarjeta.cs > Tarjeta > .ctor
1  using UnityEngine;
2  using UnityEngine.UIElements;
3
4  namespace Lab6_namespace
5  {
6      3 references
7      public class Tarjeta
8      {
9          4 references
10         Individuo miIndividuo;
11         1 reference
12         VisualElement tarjetaRoot;
13
14         2 references
15         Label nombreLabel;
16         2 references
17         Label apellidoLabel;
18
19         1 reference
20         public Tarjeta(VisualElement tarjetaRoot, Individuo individuo)
21         {
22             this.tarjetaRoot = tarjetaRoot;
23             this.miIndividuo = individuo;
24
25             nombreLabel = tarjetaRoot.Q<Label>("Nombre");
26             apellidoLabel = tarjetaRoot.Q<Label>("Apellido");
27
28             UpdateUI();
29
30             miIndividuo.Cambio += UpdateUI;
31
32         }
33
34         2 references
35         void UpdateUI()
36         {
37             nombreLabel.text = miIndividuo.Nombre;
38             apellidoLabel.text = miIndividuo.Apellido;
39         }
40     }
41 }
```

# Individuo

```
Assets > Individuo.cs > ...
1  using UnityEngine;
2  using System;
3
4  namespace Lab6_namespace
5  {
6      6 references
7      public class Individuo
8      {
9          3 references
10         public event Action Cambio;
11
12         4 references
13         private string nombre;
14         2 references
15         public string Nombre
16         {
17             get { return nombre; }
18             set
19             {
20                 if (value != nombre)
21                 {
22                     nombre = value;
23                     Cambio?.Invoke();
24                 }
25             }
26         }
27
28         4 references
29         private string apellido;
30         2 references
31         public string Apellido
32         {
33             get { return apellido; }
34             set
35             {
36                 if (value != apellido)
37                 {
38                     apellido = value;
39                     Cambio?.Invoke();
40                 }
41             }
42         }
43
44         1 reference
45         public Individuo(string nombre, string apellido)
46         {
47             this.nombre = nombre;
48             this.apellido = apellido;
49         }
50     }
51 }
```

# Lab6.cs

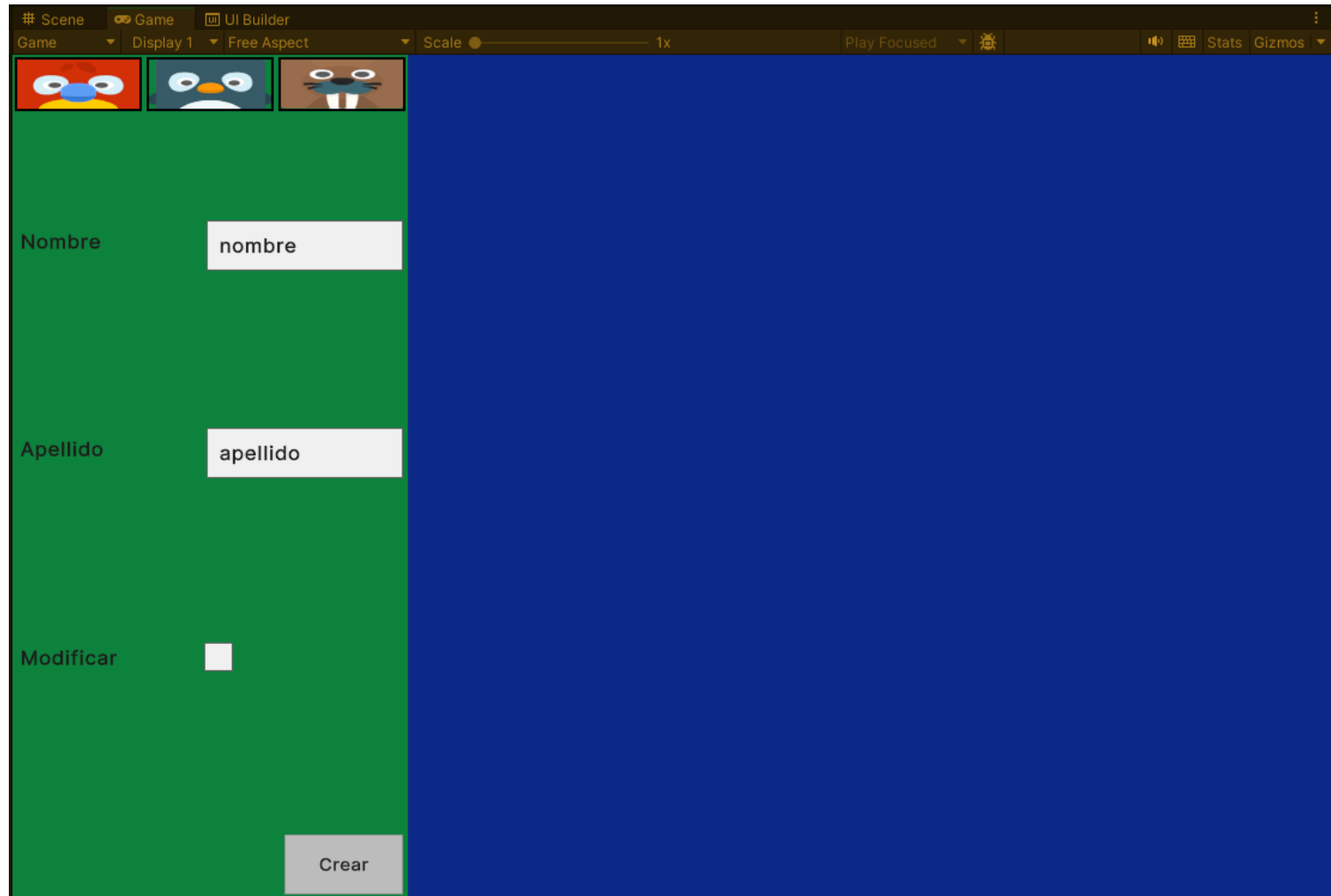
```
Assets > Lab6.cs > Lab6 > CambioNombre
1  using UnityEngine.UIElements;
2  using UnityEngine;
3  using System;
4
5  namespace Lab6_namespace
6  {
7      0 references
      public class Lab6 : MonoBehaviour
8      {
9          2 references
          VisualElement botonCrear;
10         3 references
          Toggle toggleModificar;
11         2 references
          VisualElement contendor_dcha;
12         3 references
          TextField input_nombre;
13         3 references
          TextField input_apellido;
14         3 references
          Individuo individuoSelec;
15         0 references
          private void OnEnable()
16         {
17             VisualElement root = GetComponent<UIDocument>().rootVisualElement;
18
19             contendor_dcha = root.Q<VisualElement>("Dcha");
20             input_nombre = root.Q<TextField>("InputNombre");
21             input_apellido = root.Q<TextField>("InputApellido");
22             botonCrear = root.Q<Button>("BotonCrear");
23             toggleModificar = root.Q<Toggle>("ToggleModificar");
24
25             botonCrear.RegisterCallback<ClickEvent>(NuevaTarjeta);
26             input_nombre.RegisterCallback<ChangeEvent<String>>(CambioNombre);
27             input_apellido.RegisterCallback<ChangeEvent<String>>(CambioApellido);
28         }
29     }
```

# Lab6

```
29
30 1 reference
31 void NuevaTarjeta(ClickEvent evt)
32 {
33     if (!toggleModificar.value)
34     {
35         VisualTreeAsset plantilla = Resources.Load<VisualTreeAsset>("Tarjeta");
36         VisualElement tarjetaPlantilla = plantilla.Instantiate();
37
38         contendor_dcha.Add(tarjetaPlantilla);
39
40         Individuo individuo = new Individuo(input_nombre.value, input_apellido.value);
41         Tarjeta tarjeta = new Tarjeta(tarjetaPlantilla, individuo);
42         individuoSelec = individuo;
43     }
44 }
45
46 1 reference
47 void CambioNombre(ChangeEvent<string> evt)
48 {
49     if (toggleModificar.value)
50     {
51         individuoSelec.Nombre = evt.newValue;
52     }
53 }
54
55 1 reference
56 void CambioApellido(ChangeEvent<string> evt)
57 {
58     if (toggleModificar.value)
59     {
60         individuoSelec.Apellido = evt.newValue;
61     }
62 }
63 }
```



# Funcionamiento



# Funcionamiento

The image displays a user interface with two main sections: a green form area on the left and a blue list area on the right.

**Form Section (Green Background):**

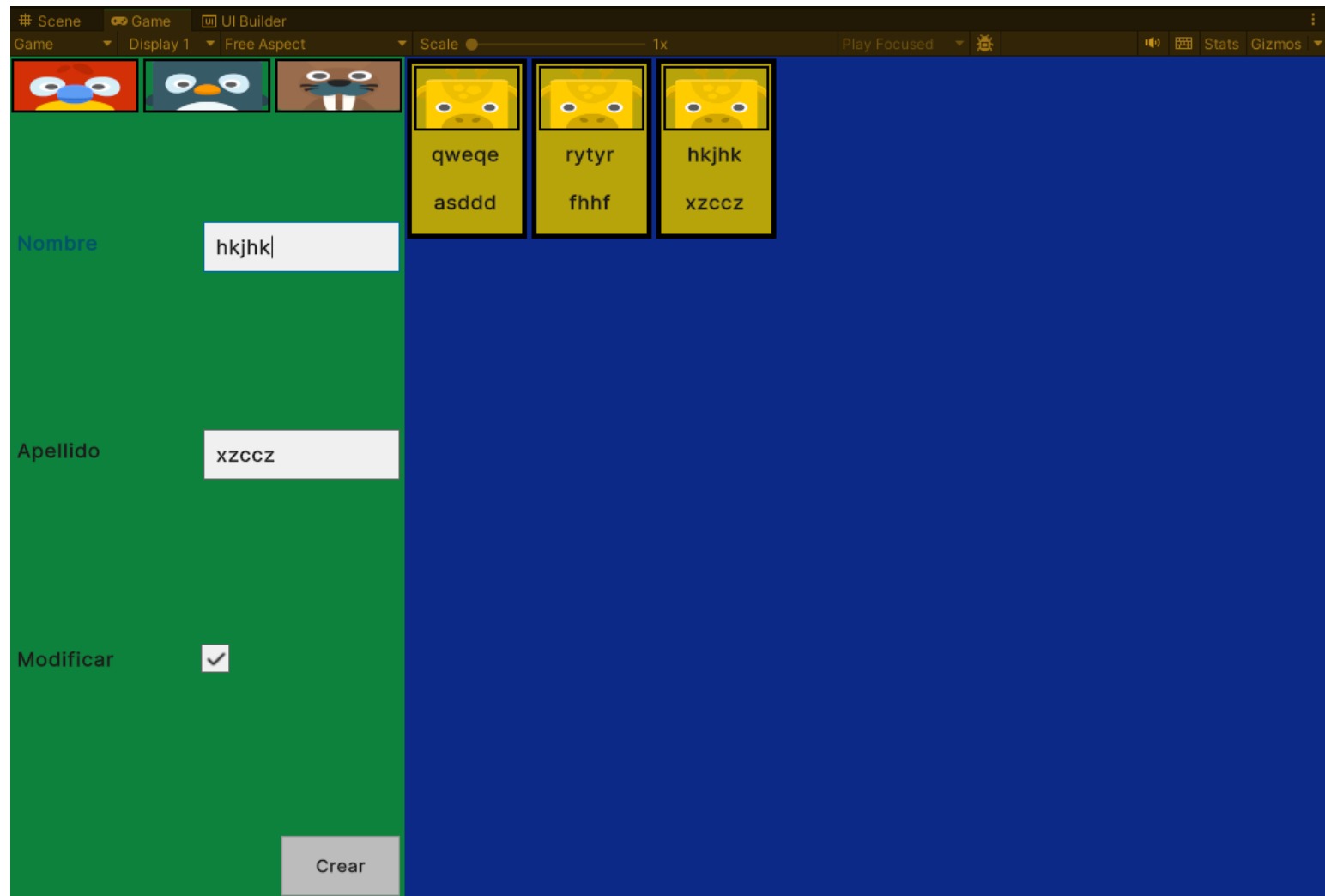
- Nombre:** A text input field containing the value "afaf".
- Apellido:** A text input field containing the value "xzccz".
- Modificar:** A checkbox that is currently unchecked.
- Crear:** A button located at the bottom right of the form area.

**List Section (Blue Background):**

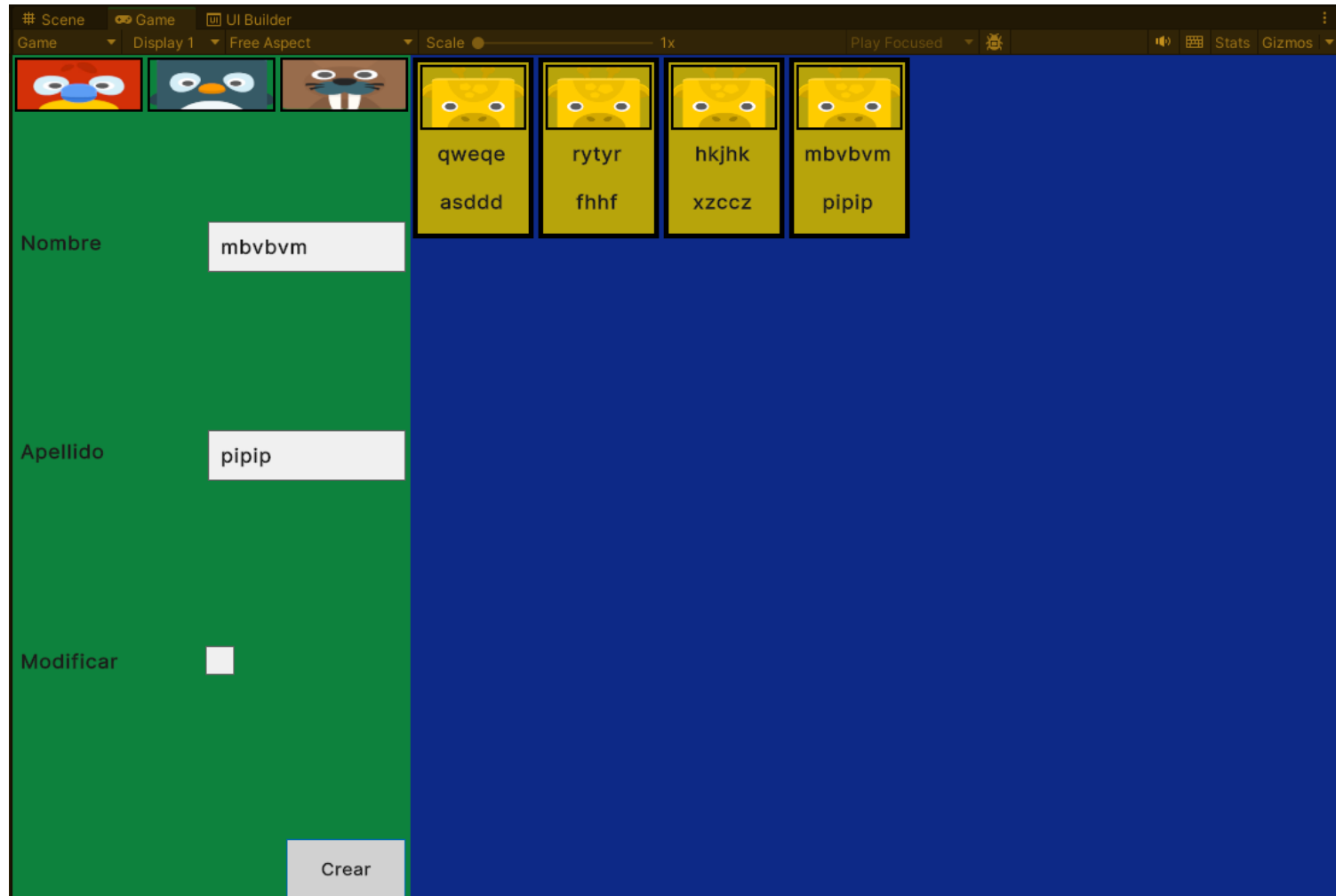
At the top of the list section, there are three yellow cat icons. Below each icon is a card containing two lines of text:

- Card 1: "qweqe" and "asddd"
- Card 2: "rytyr" and "fhhf"
- Card 3: "afaf" and "xzccz"

# Funcionamiento



# Funcionamiento



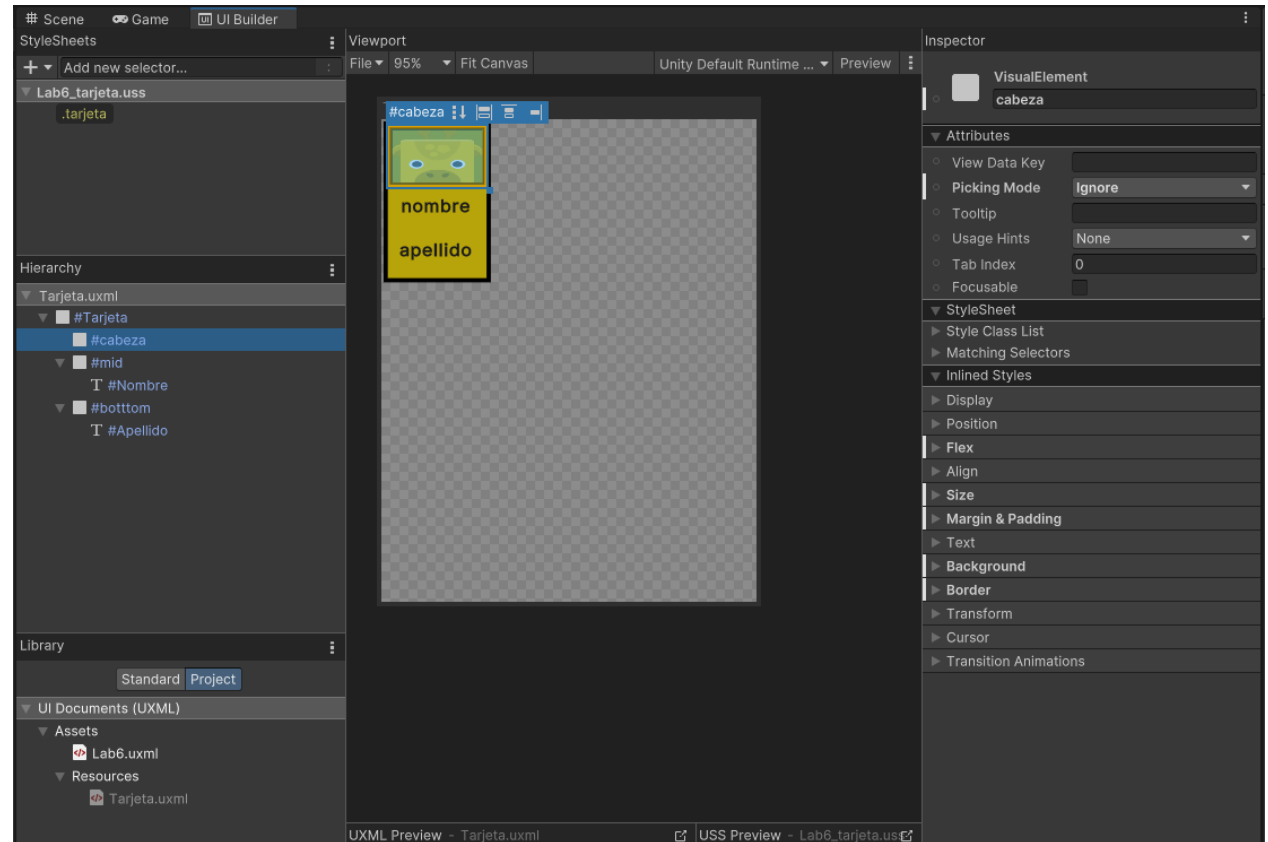
# Índice

---

- Creación de nuevos elementos
- **Modificación de elementos**
- Modificación de atributos en el Template
- Serialización en formato Json

# Picking Mode → Ignore

Todos los elementos del Template “Tarjeta” deben tener el atributo PickingMode → Ignore



# Tarjeta

```
Tarjeta.cs x Lab6.cs
Assets > Tarjeta.cs > Tarjeta > .ctor
1 using UnityEngine;
2 using UnityEngine.UIElements;
3
4 namespace Lab6_namespace
5 {
6     3 references
7     public class Tarjeta
8     {
9         5 references
10         Individuo miIndividuo;
11         1 reference
12         VisualElement tarjetaRoot;
13
14         2 references
15         Label nombreLabel;
16         2 references
17         Label apellidoLabel;
18
19         1 reference
20         public Tarjeta(VisualElement tarjetaRoot, Individuo individuo)
21         {
22             this.tarjetaRoot = tarjetaRoot;
23             this.miIndividuo = individuo;
24
25             nombreLabel = tarjetaRoot.Q<Label>("Nombre");
26             apellidoLabel = tarjetaRoot.Q<Label>("Apellido");
27             tarjetaRoot.userData = miIndividuo;
28
29             UpdateUI();
30
31             miIndividuo.Cambio += UpdateUI;
32
33         }
34
35         2 references
36         void UpdateUI()
37         {
38             nombreLabel.text = miIndividuo.Nombre;
39             apellidoLabel.text = miIndividuo.Apellido;
40         }
41     }
42 }
```

# Lab6

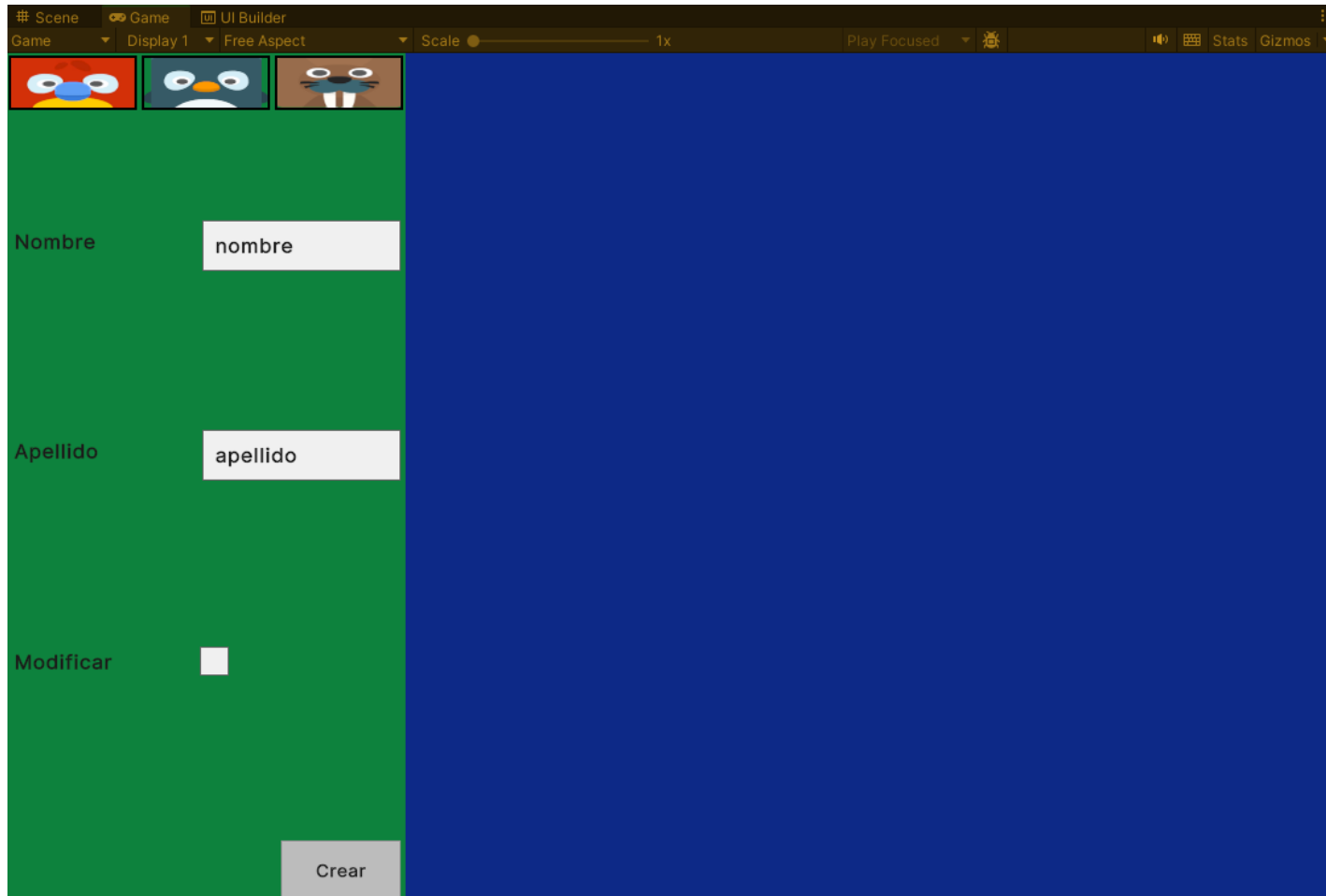
```
Tarjeta.cs  Lab6.cs  •
Assets > Lab6.cs > Lab6 > NuevaTarjeta
1  using UnityEngine.UIElements;
2  using UnityEngine;
3  using System;
4
5  namespace Lab6_namespace
6  {
7      0 references
      public class Lab6 : MonoBehaviour
8      {
9          2 references
          VisualElement botonCrear;
          5 references
10         Toggle toggleModificar;
          3 references
11         VisualElement contendor_dcha;
          4 references
12         TextField input_nombre;
          4 references
13         TextField input_apellido;
          6 references
14         Individuo individuoSelec;
          0 references
15         private void OnEnable()
16         {
17             VisualElement root = GetComponent<UIDocument>().rootVisualElement;
18
19             contendor_dcha = root.Q<VisualElement>("Dcha");
20             input_nombre = root.Q<TextField>("InputNombre");
21             input_apellido = root.Q<TextField>("InputApellido");
22             botonCrear = root.Q<Button>("BotonCrear");
23             toggleModificar = root.Q<Toggle>("ToggleModificar");
24
25             contendor_dcha.RegisterCallback<ClickEvent>(seleccionTarjeta);
26             botonCrear.RegisterCallback<ClickEvent>(NuevaTarjeta);
27             input_nombre.RegisterCallback<ChangeEvent<String>>(CambioNombre);
28             input_apellido.RegisterCallback<ChangeEvent<String>>(CambioApellido);
29         }
30     }
```



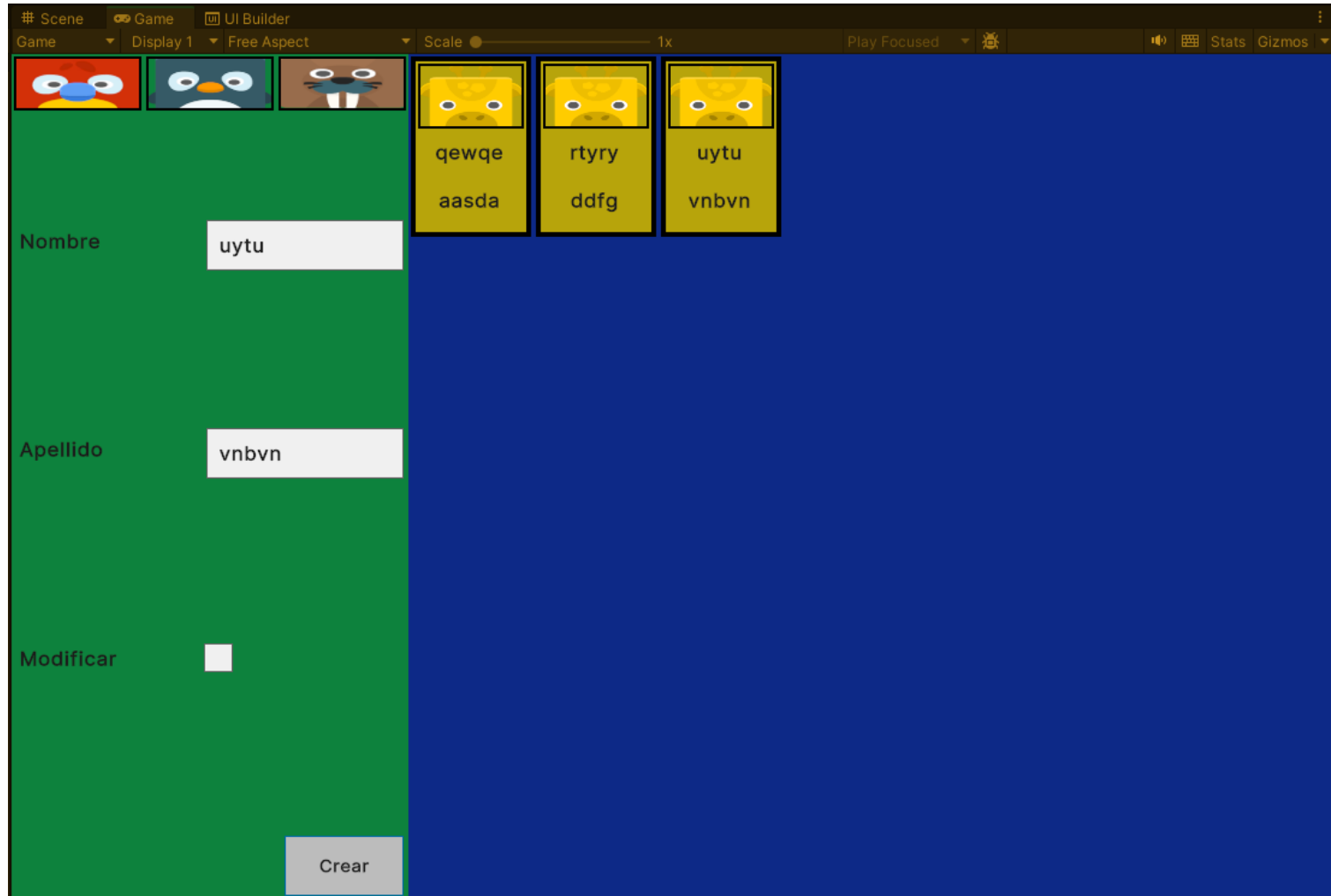
# Lab6

```
30
31 1 reference
32 void NuevaTarjeta(ClickEvent evt)
33 {
34     if (!toggleModificar.value)
35     {
36         VisualTreeAsset plantilla = Resources.Load<VisualTreeAsset>("Tarjeta");
37         VisualElement tarjetaPlantilla = plantilla.Instantiate();
38
39         contendor_dcha.Add(tarjetaPlantilla);
40
41         Individuo individuo = new Individuo(input_nombre.value, input_apellido.value);
42         Tarjeta tarjeta = new Tarjeta(tarjetaPlantilla, individuo);
43         individuoSelec = individuo;
44     }
45 }
46
47 1 reference
48 void seleccionTarjeta(ClickEvent e)
49 {
50     VisualElement miTarjeta = e.target as VisualElement;
51     individuoSelec = miTarjeta.userData as Individuo;
52
53     input_nombre.SetValueWithoutNotify(individuoSelec.Nombre);
54     input_apellido.SetValueWithoutNotify(individuoSelec.Apellido);
55     toggleModificar.value = true;
56 }
57
58 1 reference
59 void CambioNombre(ChangeEvent<string> evt)
60 {
61     if (toggleModificar.value)
62     {
63         individuoSelec.Nombre = evt.newValue;
64     }
65 }
66
67 1 reference
68 void CambioApellido(ChangeEvent<string> evt)
69 {
70     if (toggleModificar.value)
71     {
72         individuoSelec.Apellido = evt.newValue;
73     }
74 }
```

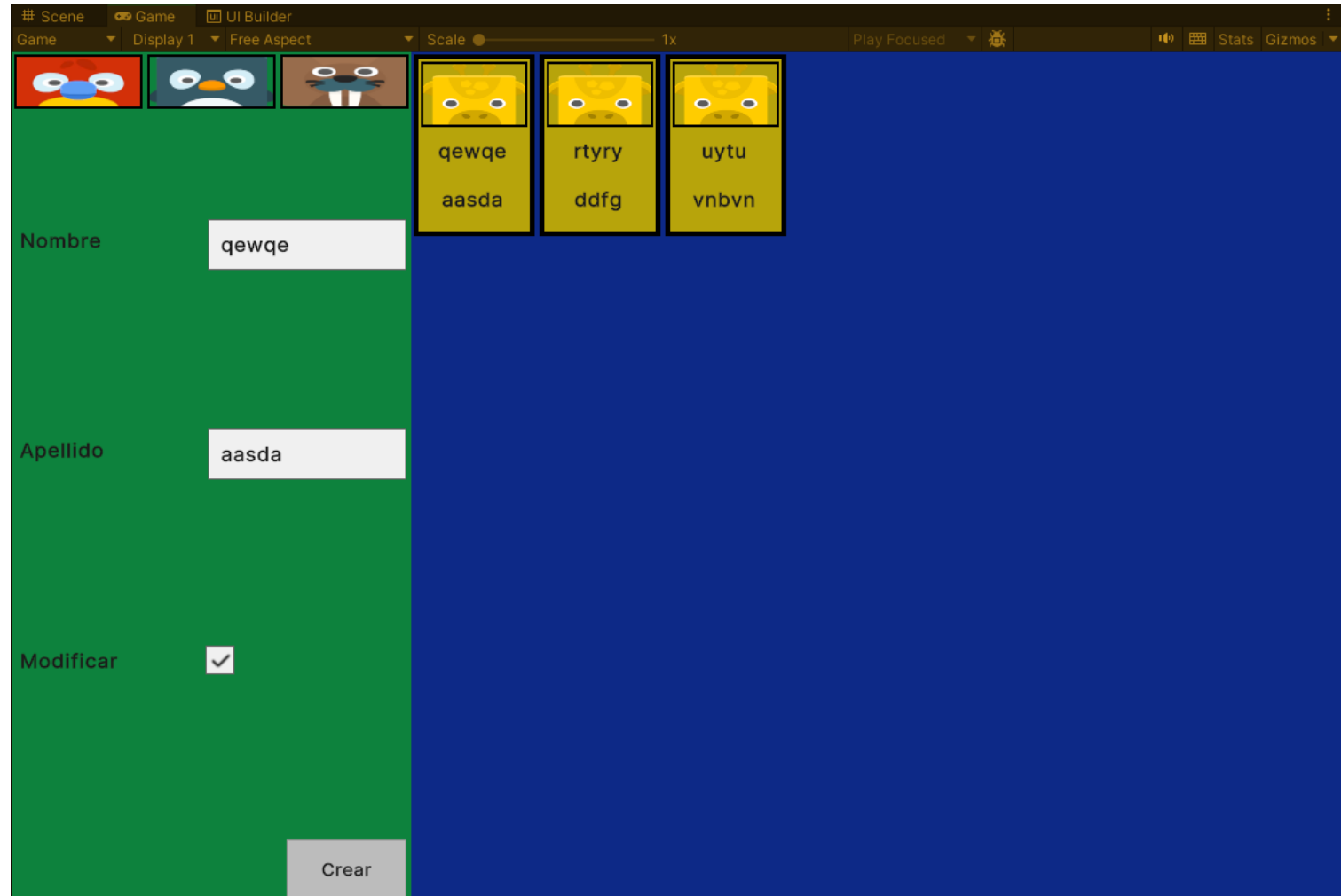
# Funcionamiento



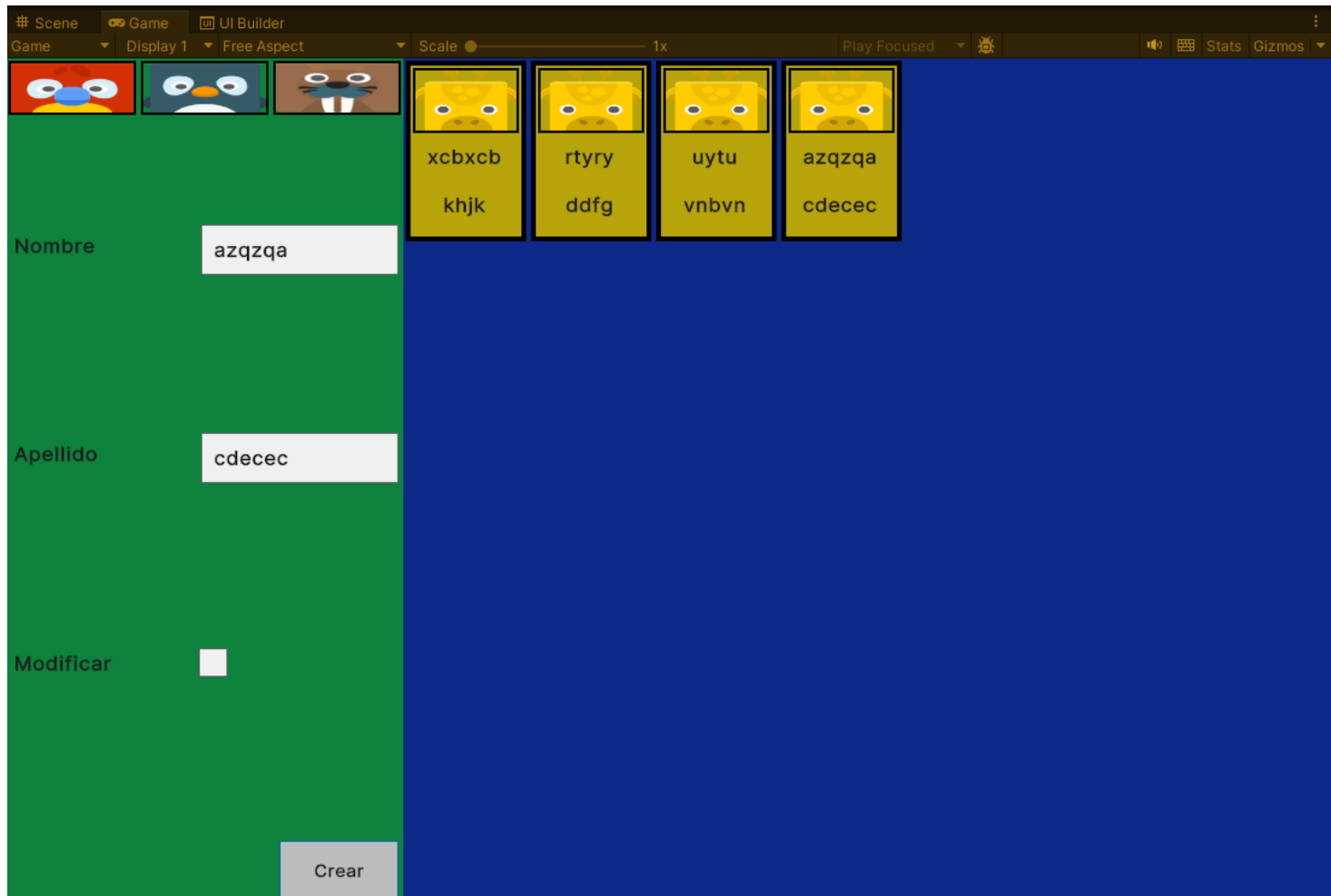
# Creación



# Seleccionamos el primer elemento



# Funcionamiento

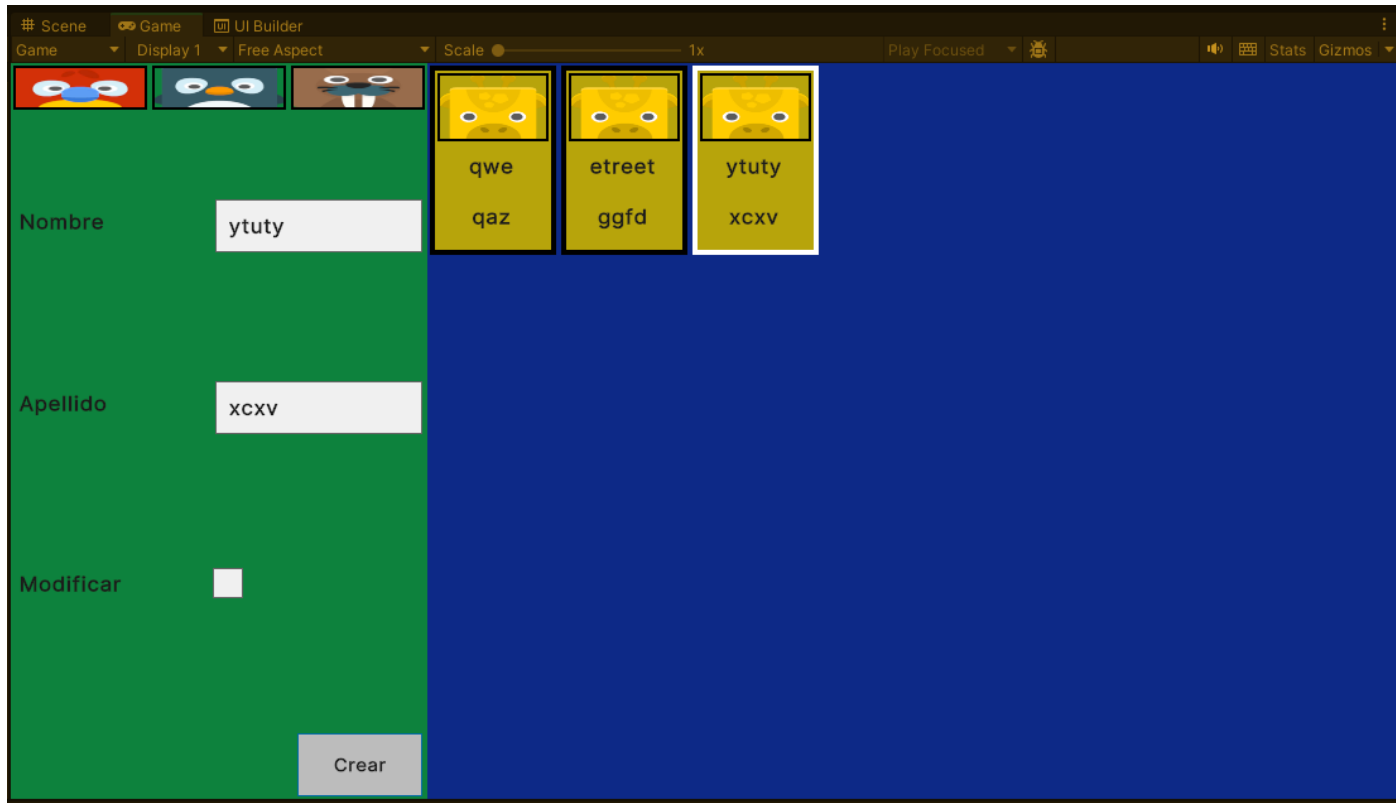


# Índice

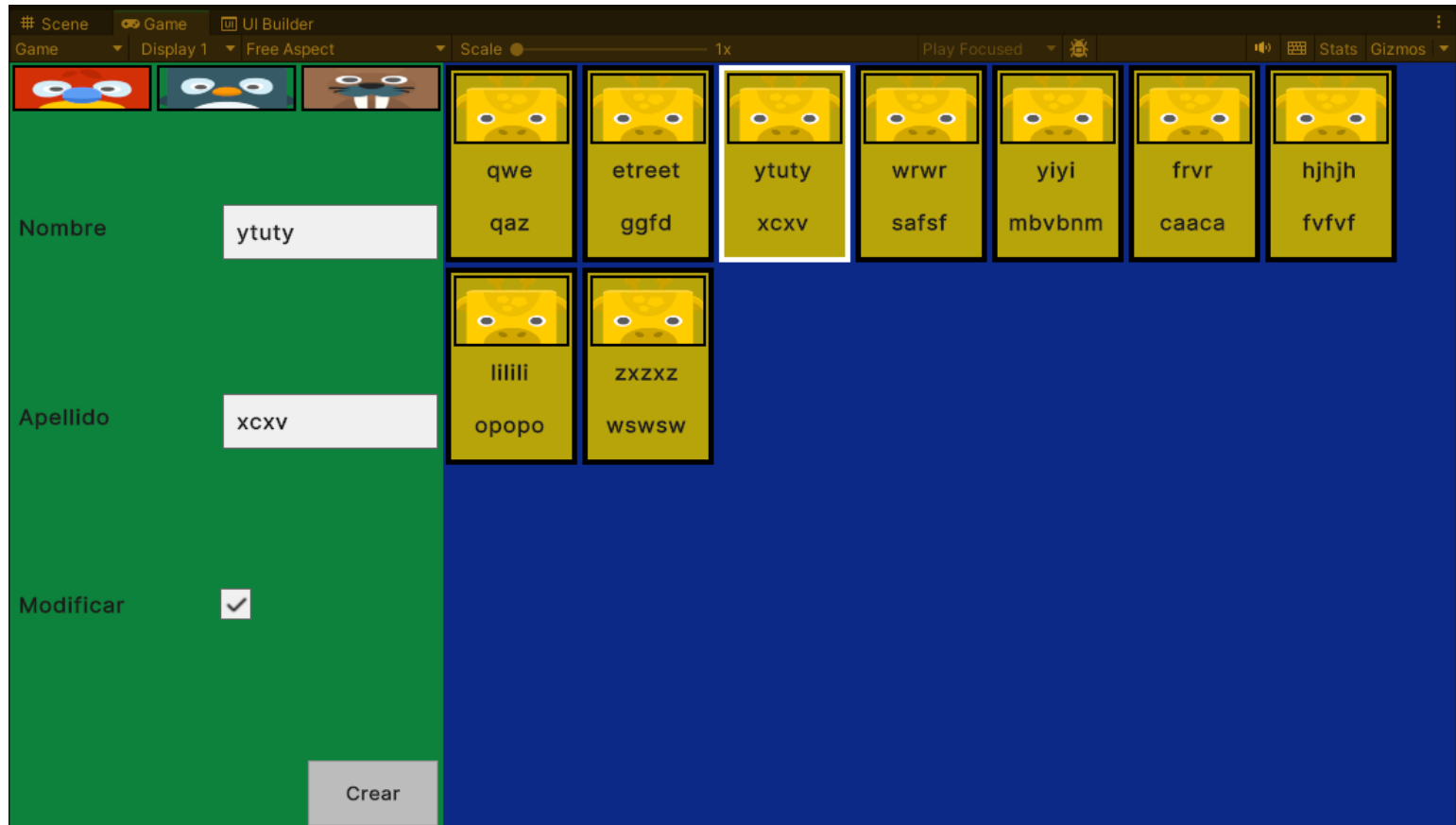
---

- Creación de nuevos elementos
- Modificación de elementos
- **Modificación de atributos en el Template**
- Serialización en formato Json

# Índice



# Índice





# Lab6

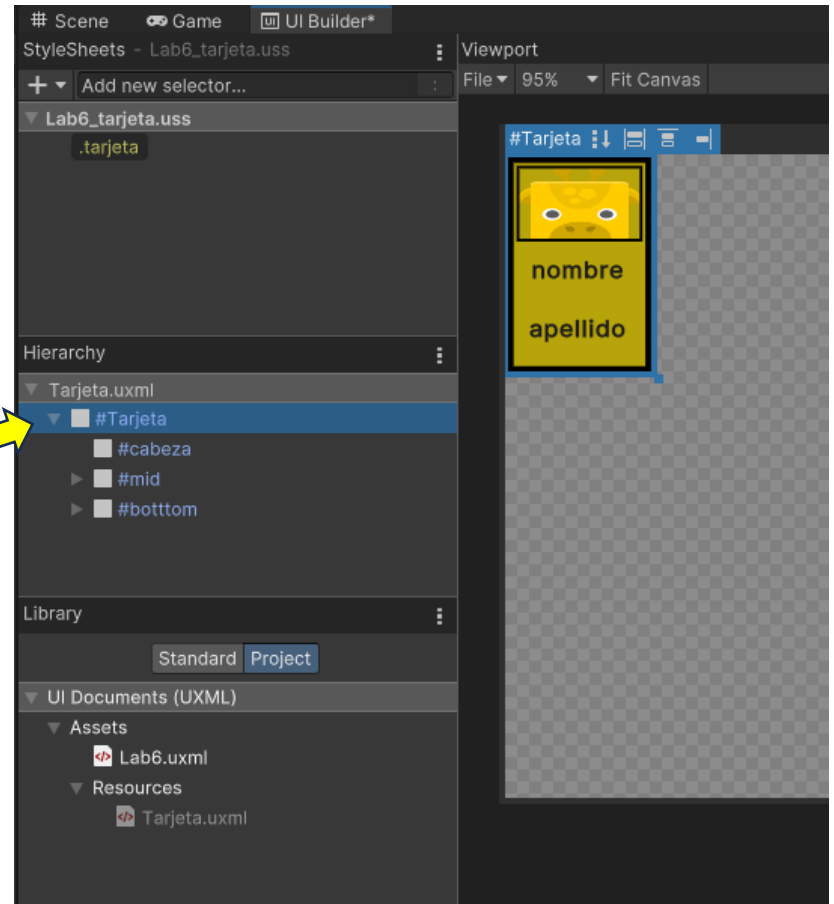
```
Tarjeta.cs  Lab6.cs
Assets > Lab6.cs > Lab6 > NuevaTarjeta
1  using UnityEngine.UIElements;
2  using UnityEngine;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6
7  namespace Lab6_namespace
8  {
9      0 references
10     public class Lab6 : MonoBehaviour
11     {
12         2 references
13         VisualElement botonCrear;
14         5 references
15         Toggle toggleModificar;
16         4 references
17         VisualElement contendor_dcha;
18         4 references
19         TextField input_nombre;
20         4 references
21         TextField input_apellido;
22         6 references
23         Individuo individuoSelec;
24         0 references
25         private void OnEnable()
26         {
27             VisualElement root = GetComponent<UIDocument>().rootVisualElement;
28
29             contendor_dcha = root.Q<VisualElement>("Dcha");
30             input_nombre = root.Q<TextField>("InputNombre");
31             input_apellido = root.Q<TextField>("InputApellido");
32             botonCrear = root.Q<Button>("BotonCrear");
33             toggleModificar = root.Q<Toggle>("ToggleModificar");
34
35             contendor_dcha.RegisterCallback<ClickEvent>(seleccionTarjeta);
36             botonCrear.RegisterCallback<ClickEvent>(NuevaTarjeta);
37             input_nombre.RegisterCallback<ChangeEvent<String>>(CambioNombre);
38             input_apellido.RegisterCallback<ChangeEvent<String>>(CambioApellido);
39         }
40     }
```

# Lab6

```
32
33 1 reference
34 void NuevaTarjeta(ClickEvent evt)
35 {
36     if (!toggleModificar.value)
37     {
38         VisualTreeAsset plantilla = Resources.Load<VisualTreeAsset>("Tarjeta");
39         VisualElement tarjetaPlantilla = plantilla.Instantiate();
40
41         contendor_dcha.Add(tarjetaPlantilla);
42         tarjetas_borde_negro();
43         tarjeta_borde_blanco(tarjetaPlantilla);
44
45         Individuo individuo = new Individuo(input_nombre.value, input_apellido.value);
46         Tarjeta tarjeta = new Tarjeta(tarjetaPlantilla, individuo);
47         individuoSelec = individuo;
48     }
49 }
50
51 1 reference
52 void seleccionTarjeta(ClickEvent e)
53 {
54     VisualElement miTarjeta = e.target as VisualElement;
55     individuoSelec = miTarjeta.userData as Individuo;
56
57     input_nombre.SetValueWithoutNotify(individuoSelec.Nombre);
58     input_apellido.SetValueWithoutNotify(individuoSelec.Apellido);
59     toggleModificar.value = true;
60
61     tarjetas_borde_negro();
62     tarjeta_borde_blanco(miTarjeta);
63 }
```

# Lab6

```
64 void CambioNombre(ChangeEvent<string> evt)
65 {
66     if (toggleModificar.value)
67     {
68         individuoSelec.Nombre = evt.newValue;
69     }
70 }
71
72 1 reference
73 void CambioApellido(ChangeEvent<string> evt)
74 {
75     if (toggleModificar.value)
76     {
77         individuoSelec.Apellido = evt.newValue;
78     }
79 }
80 2 references
81 void tarjetas_borde_negro()
82 {
83     List<VisualElement> lista_tarjetas = contendor_dcha.Children().ToList();
84     lista_tarjetas.ForEach(elem =>
85     {
86         VisualElement tarjeta = elem.Q("Tarjeta");
87
88         tarjeta.style.borderBottomColor = Color.black;
89         tarjeta.style.borderRightColor = Color.black;
90         tarjeta.style.borderTopColor = Color.black;
91         tarjeta.style.borderLeftColor = Color.black;
92     });
93 }
94 2 references
95 void tarjeta_borde_blanco(VisualElement tar)
96 {
97     VisualElement tarjeta = tar.Q("Tarjeta");
98
99     tarjeta.style.borderBottomColor = Color.white;
100     tarjeta.style.borderRightColor = Color.white;
101     tarjeta.style.borderTopColor = Color.white;
102     tarjeta.style.borderLeftColor = Color.white;
103 }
104 }
105 }
```



# Índice

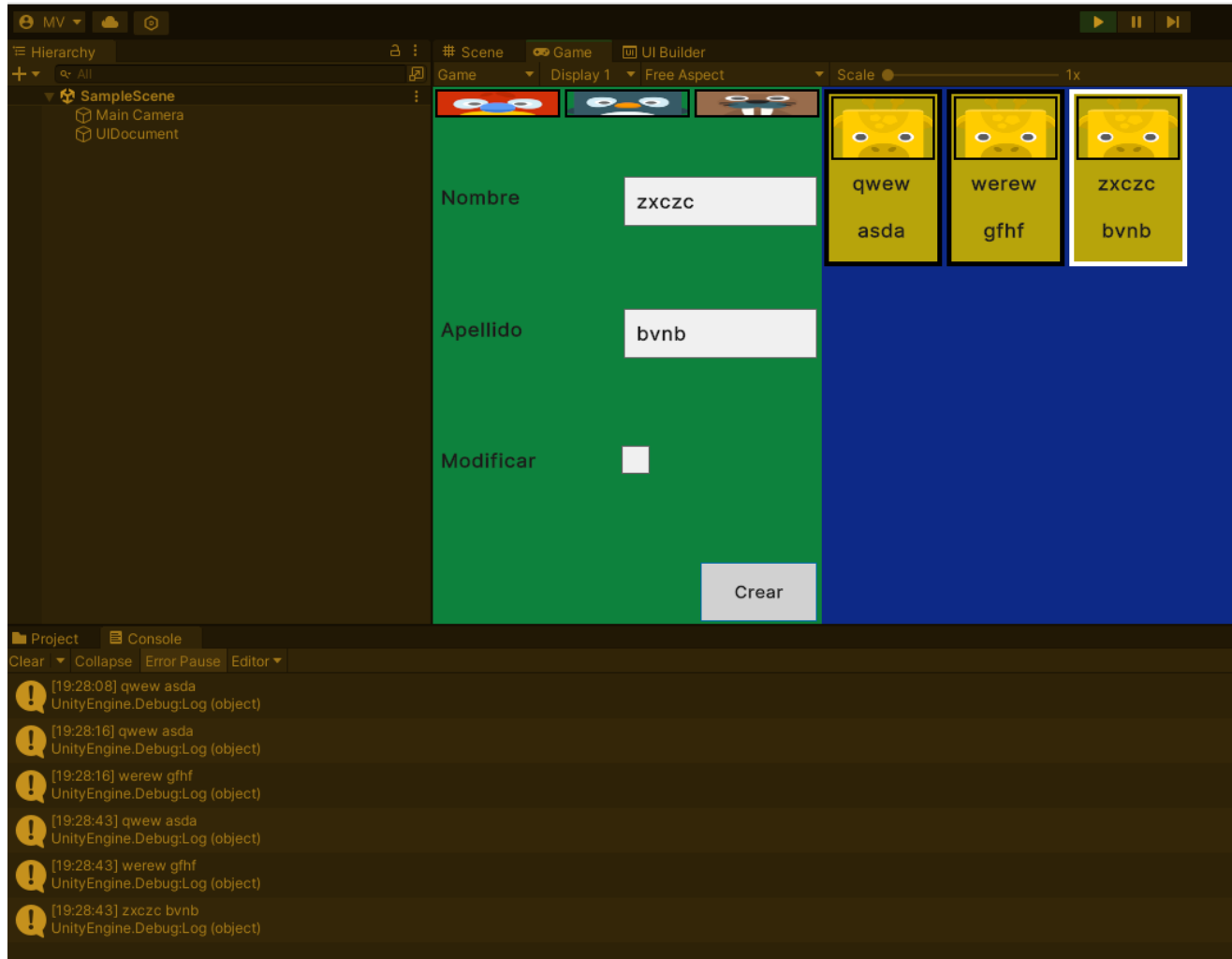
---

- Creación de nuevos elementos
- Modificación de elementos
- Modificación de atributos en el Template
- **Serialización en formato Json**

# Lab6

```
18 2 references
   List<Individuo> list_individuos = new List<Individuo>();
19 0 references
   private void OnEnable()
20 {
21     VisualElement root = GetComponent<UIDocument>().rootVisualElement;
22
23     contendor_dcha = root.Q<VisualElement>("Dcha");
24     input_nombre = root.Q<TextField>("InputNombre");
25     input_apellido = root.Q<TextField>("InputApellido");
26     botonCrear = root.Q<Button>("BotonCrear");
27     toggleModificar = root.Q<Toggle>("ToggleModificar");
28
29     contendor_dcha.RegisterCallback<ClickEvent>(seleccionTarjeta);
30     botonCrear.RegisterCallback<ClickEvent>(NuevaTarjeta);
31     input_nombre.RegisterCallback<ChangeEvent<String>>(CambioNombre);
32     input_apellido.RegisterCallback<ChangeEvent<String>>(CambioApellido);
33 }
34
35 1 reference
   void NuevaTarjeta(ClickEvent evt)
36 {
37     if (!toggleModificar.value)
38     {
39         VisualTreeAsset plantilla = Resources.Load<VisualTreeAsset>("Tarjeta");
40         VisualElement tarjetaPlantilla = plantilla.Instantiate();
41
42         contendor_dcha.Add(tarjetaPlantilla);
43         tarjetas_borde_negro();
44         tarjeta_borde_blanco(tarjetaPlantilla);
45
46         Individuo individuo = new Individuo(input_nombre.value, input_apellido.value);
47         Tarjeta tarjeta = new Tarjeta(tarjetaPlantilla, individuo);
48         individuoSelec = individuo;
49
50         list_individuos.Add(individuo);
51         list_individuos.ForEach(elem => Debug.Log(elem.Nombre + " " + elem.Apellido));
52     }
53 }
54
55 }
```

# Lab6

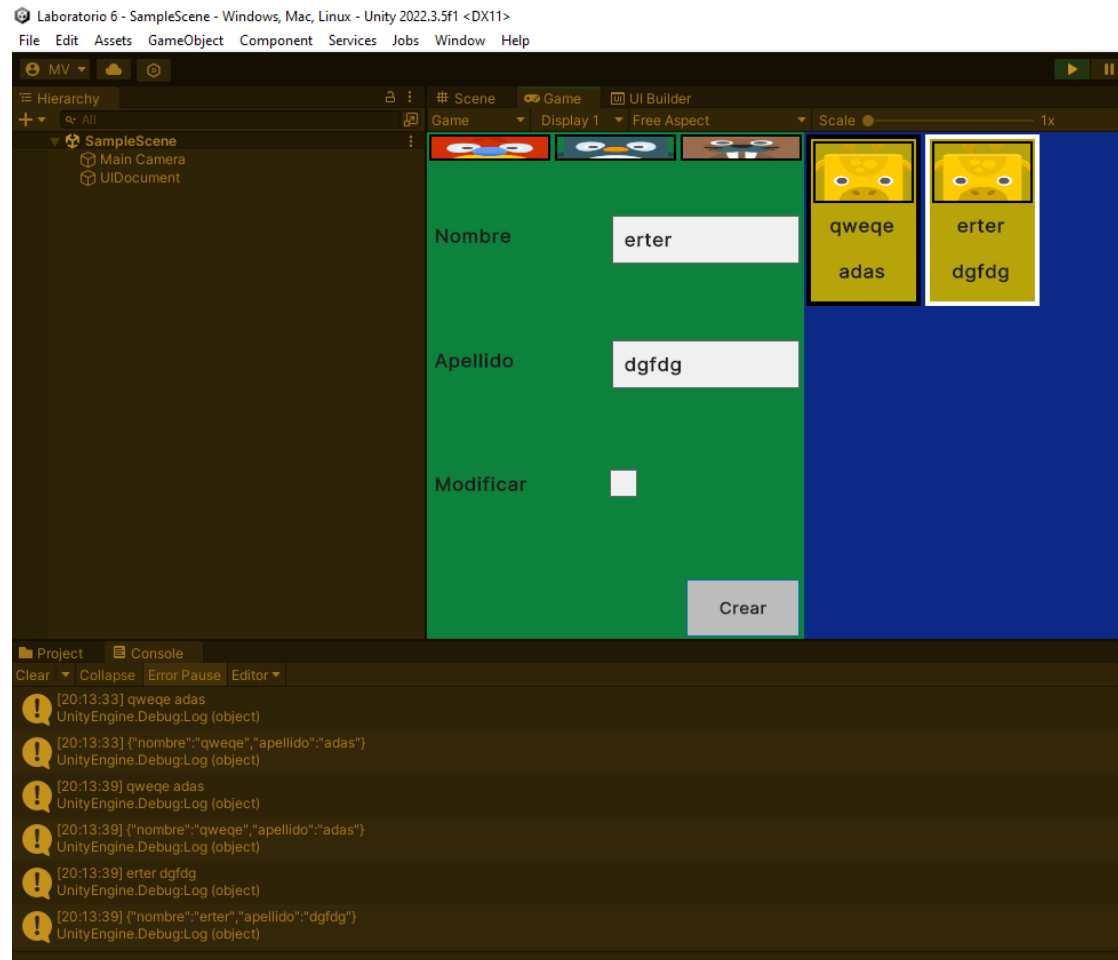


# Json

```
Assets > Lab6.cs > Lab6 > seleccionTarjeta
1  using UnityEngine.UIElements;
2  using UnityEngine;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6
```

```
50  lista_individuos.Add(individuo);
51  lista_individuos.ForEach(elem => {
52      Debug.Log(elem.Nombre + " " + elem.Apellido);
53      string jsonIndividuo = JsonUtility.ToJson(elem);
54      Debug.Log(jsonIndividuo);
55      });
56  }
```

# Json





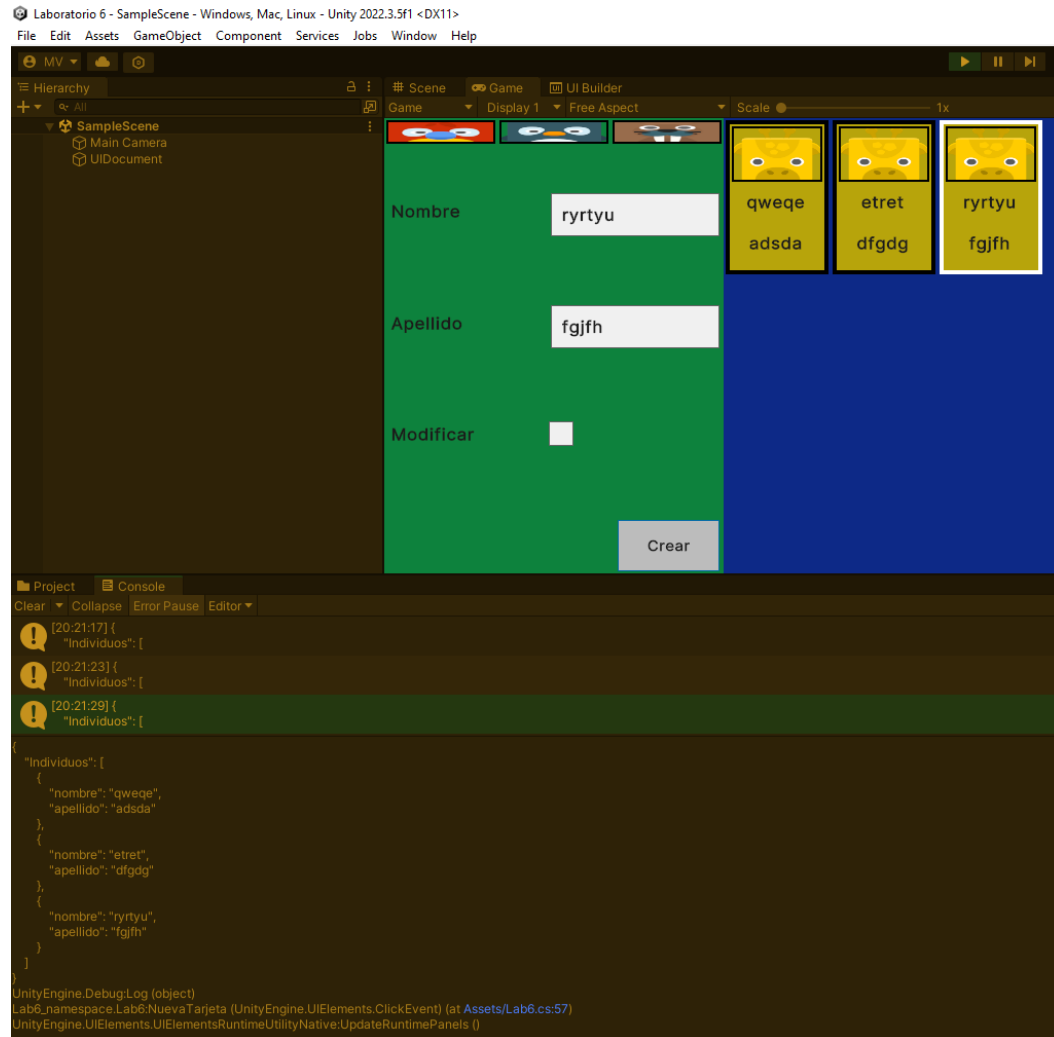
# JsonHelper

```
Tarjeta.cs  Individuo.cs  Lab6.cs  JsonHelper.cs x
Assets > JsonHelper.cs > ...
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using UnityEngine;
5
6  namespace Lab6_namespace
7  {
8      1 reference
      public static class JsonHelperIndividuo
9      {
10         0 references
11         public static List<Individuo> FromJson<Individuo>(string json)
12         {
13             ListaIndividuo<Individuo> listaIndividuo = JsonUtility.FromJson<ListaIndividuo<Individuo>>(json);
14             return listaIndividuo.Individuos;
15         }
16
17         0 references
18         public static string ToJson<Individuo>(List<Individuo> lista)
19         {
20             ListaIndividuo<Individuo> listaIndividuo = new ListaIndividuo<Individuo>();
21             listaIndividuo.Individuos = lista;
22             return JsonUtility.ToJson(listaIndividuo);
23         }
24
25         1 reference
26         public static string ToJson<Individuo>(List<Individuo> lista, bool prettyPrint)
27         {
28             ListaIndividuo<Individuo> listaIndividuo = new ListaIndividuo<Individuo>();
29             listaIndividuo.Individuos = lista;
30             return JsonUtility.ToJson(listaIndividuo, prettyPrint);
31         }
32
33         [Serializable]
34         6 references
35         private class ListaIndividuo<Individuo>
36         {
37             3 references
38             public List<Individuo> Individuos;
39         }
40     }
41 }
```

# Lab6

```
49
50     lista_individuos.Add(individuo);
51     /*
52         lista_individuos.ForEach(elem => {
53             Debug.Log(elem.Nombre + " " + elem.Apellido);
54             string jsonIndividuo = JsonUtility.ToJson(elem);
55             Debug.Log(jsonIndividuo);
56         });
57     */
58     string listaToJson = JsonHelperIndividuo.ToJson(lista_individuos, true);
59     Debug.Log(listaToJson);
60 }
```

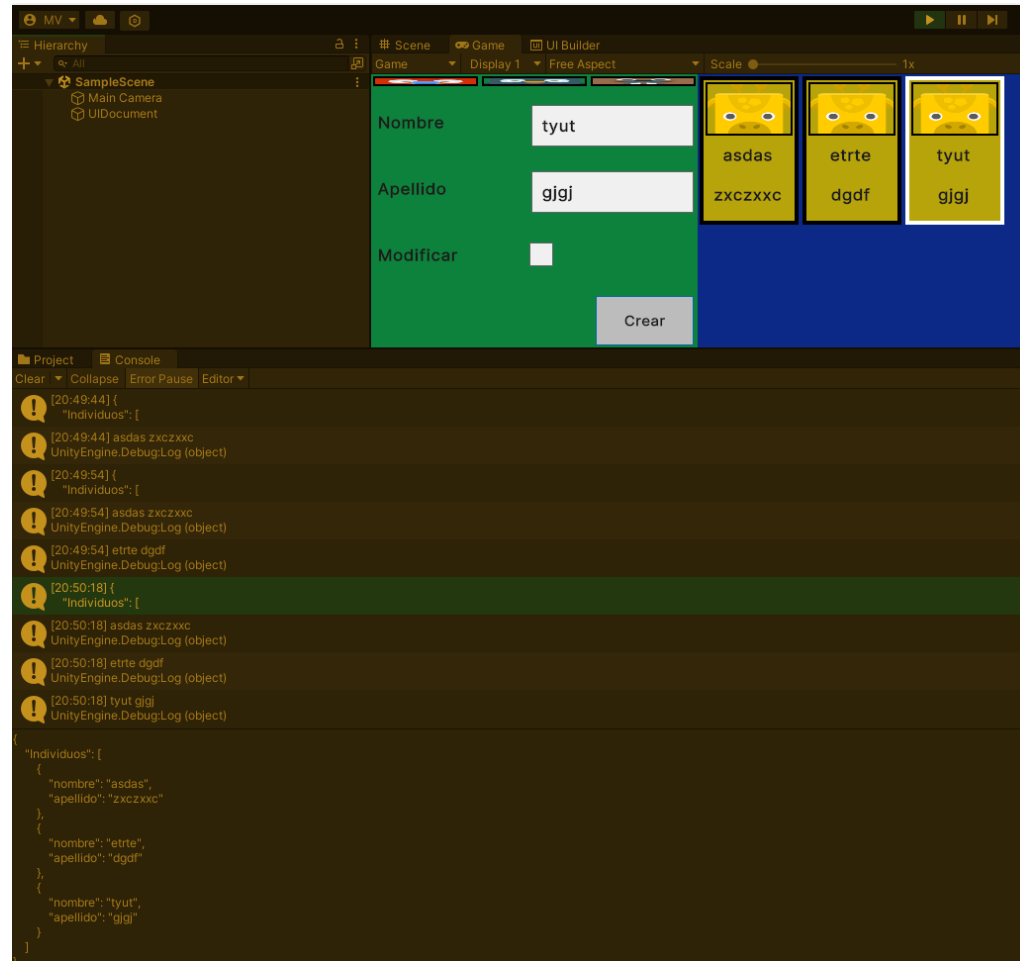
# Json



# Json

```
49
50     lista_individuos.Add(individuo);
51
52     string listaToJson = JsonHelperIndividuo.ToJson(lista_individuos, true);
53     Debug.Log(listaToJson);
54
55     List<Individuo> jsonToLista = JsonHelperIndividuo.FromJson<Individuo>(listaToJson);
56     jsonToLista.ForEach(elem => {
57         Debug.Log(elem.Nombre + " " + elem.Apellido);
58     });
59
60 }
61
```

# Json



# Ejercicios

---

- Añadir la posibilidad de elegir la imagen del top en la tarjeta.
- Añadir un botón de “Guardar”.
  - Este botón guardará en un fichero la “lista\_individuos” en formato Json.
- Modificar el script BaseDatos de la práctica 5, para que lea el archivo creado en el apartado anterior (lista\_individuos en formato Json) y recupere los individuos guardados al iniciarse de nuevo el juego.