

## Pasos a seguir:

- Guardar este documento en nuestro equipo.
- Renombrar según el siguiente ejemplo: ApellidoNombre\_ActividadesUD3
- En el transcurso de las clases, el alumnado irá completando el documento según se indique en el enunciado de las actividades.
- Genera repositorio GIT: UD3\_LMSGI

|                                          |    |
|------------------------------------------|----|
| Actividad 1. XML + namespace.....        | 2  |
| Actividad 2. Inventario XML.....         | 3  |
| Actividad 3. JSON.....                   | 7  |
| Actividad 4. JSON.....                   | 8  |
| Actividad 5. Repositorios en GitHub..... | 11 |
| Actividad 6. JSON- Unity.....            | 12 |

### **Actividad 1. XML + namespace**

XML generado para almacenar entradas de libros (10 entradas)

Captura de pantalla del validador del XML

XML con espacio de nombre ficticio

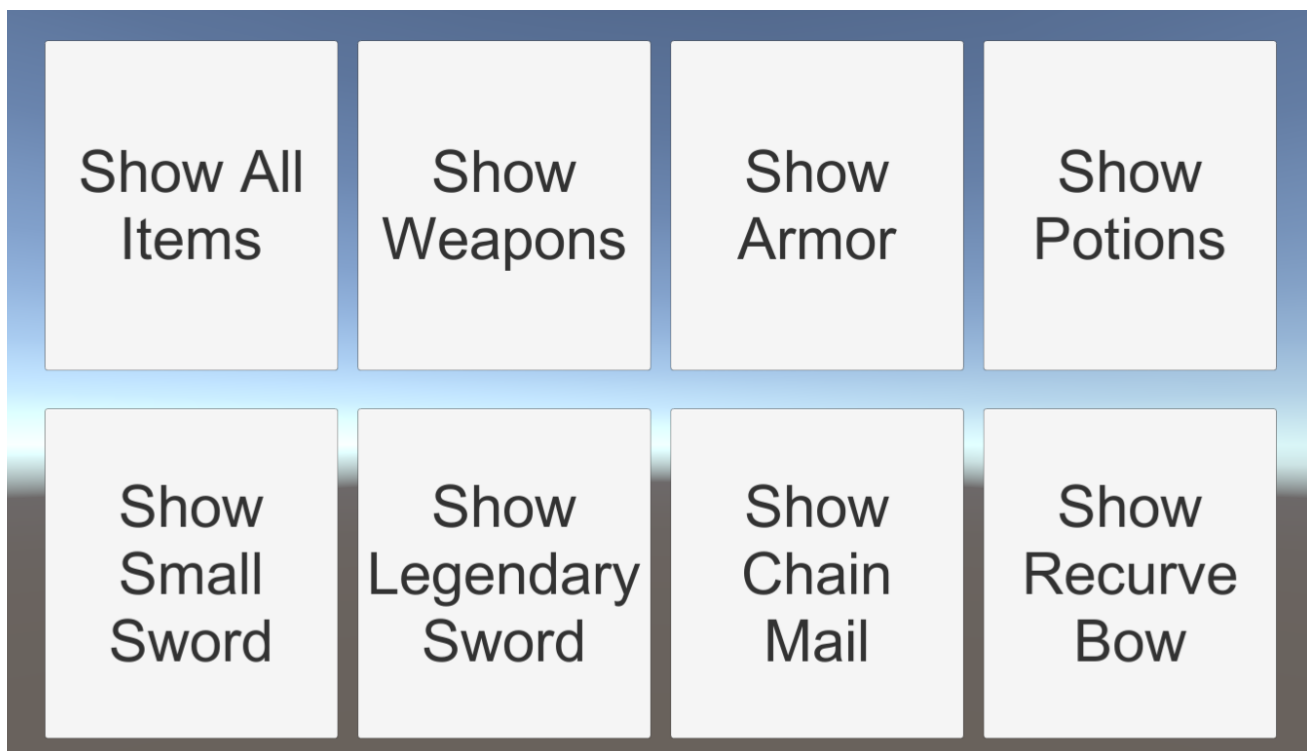
## Actividad 2. Inventario XML

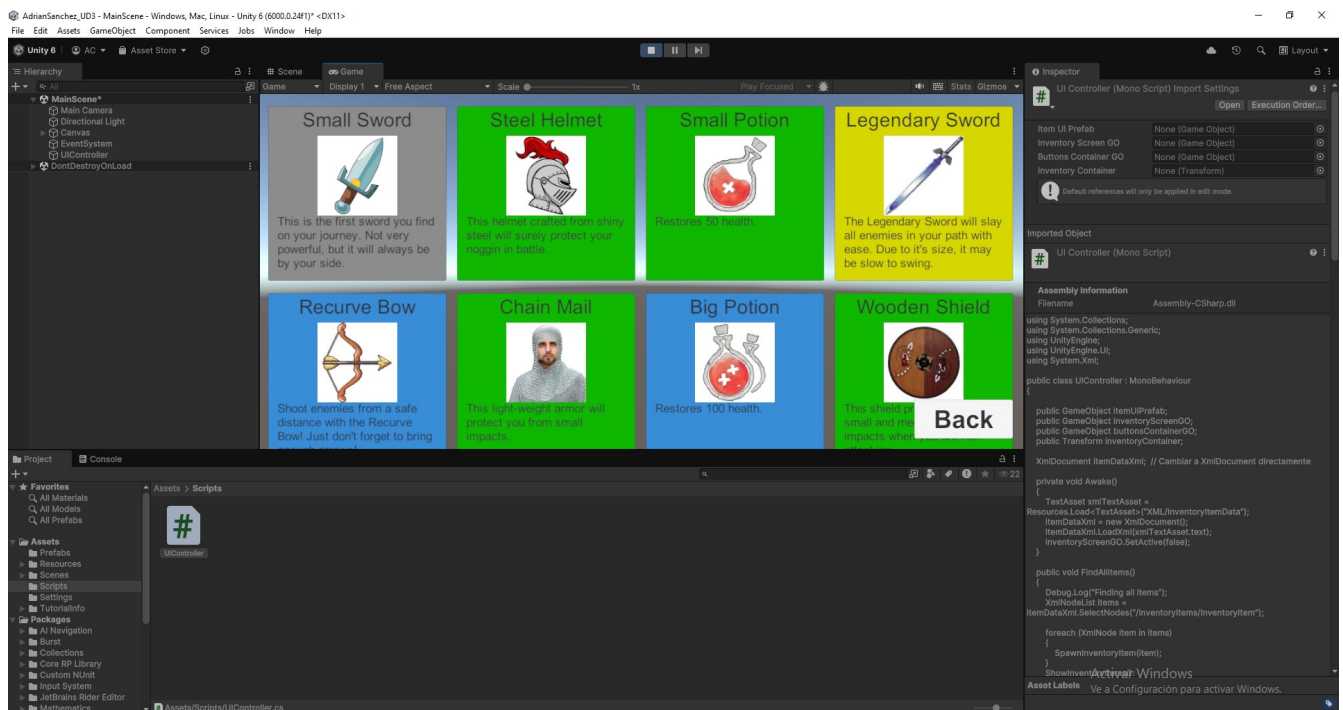
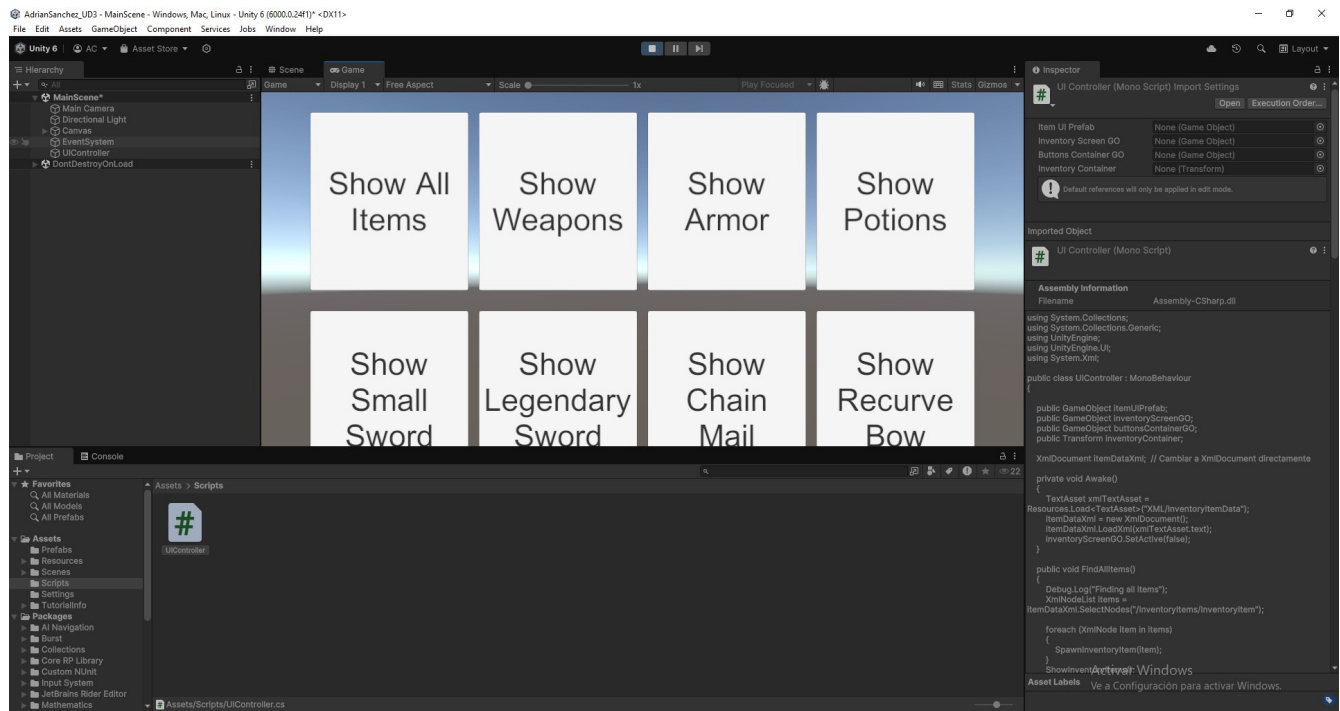
Instrucciones:

1. Crear proyecto(NomApe\_Act2\_UD3), importar el paquete 'Act2-UD3\_LMSGI.unitypackage'. Abrir la escena 'MainScene'

2. Corregir errores. Al ejecutar, debe mostrar la siguiente pantalla:

Cada panel da acceso a un nuevo panel, mostrando los items del inventario del tipo seleccionado.







3. Buscar XML del proyecto y traducir al castellano los datos a cargar de las siguientes etiquetas

XML: `<ItemTitle>` y `<ItemDesc>`

Hacer click en el panel "Show All Items" y adjuntar una captura de pantalla que muestra todos los items con el nombre y descripción en castellano.

4. En el mismo XML, modificar los elementos XML atendiendo a la siguiente traducción:

`<InventoryItems>` -> `<ItemsInventario>`

`</InventoryItems>` -> `</ItemsInventario>`

`<InventoryItem>` -> `<ItemInventario>`

`</InventoryItem>` -> `</ItemInventario>`

`<ItemTitle>` -> `<TituloItem>`

`</ItemTitle>` -> `</TituloItem>`

`<ItemDesc>` -> `<DescItem>`

`</ItemDesc>` -> `</DescItem>`

5. Modificar el script 'UIController.cs' para que cargue correctamente el archivo XML según los nodos modificados.

Adjuntar a continuación el código del script 'UIController.cs'

6. En el mismo XML, modificar los elementos XML atendiendo a la siguiente traducción:

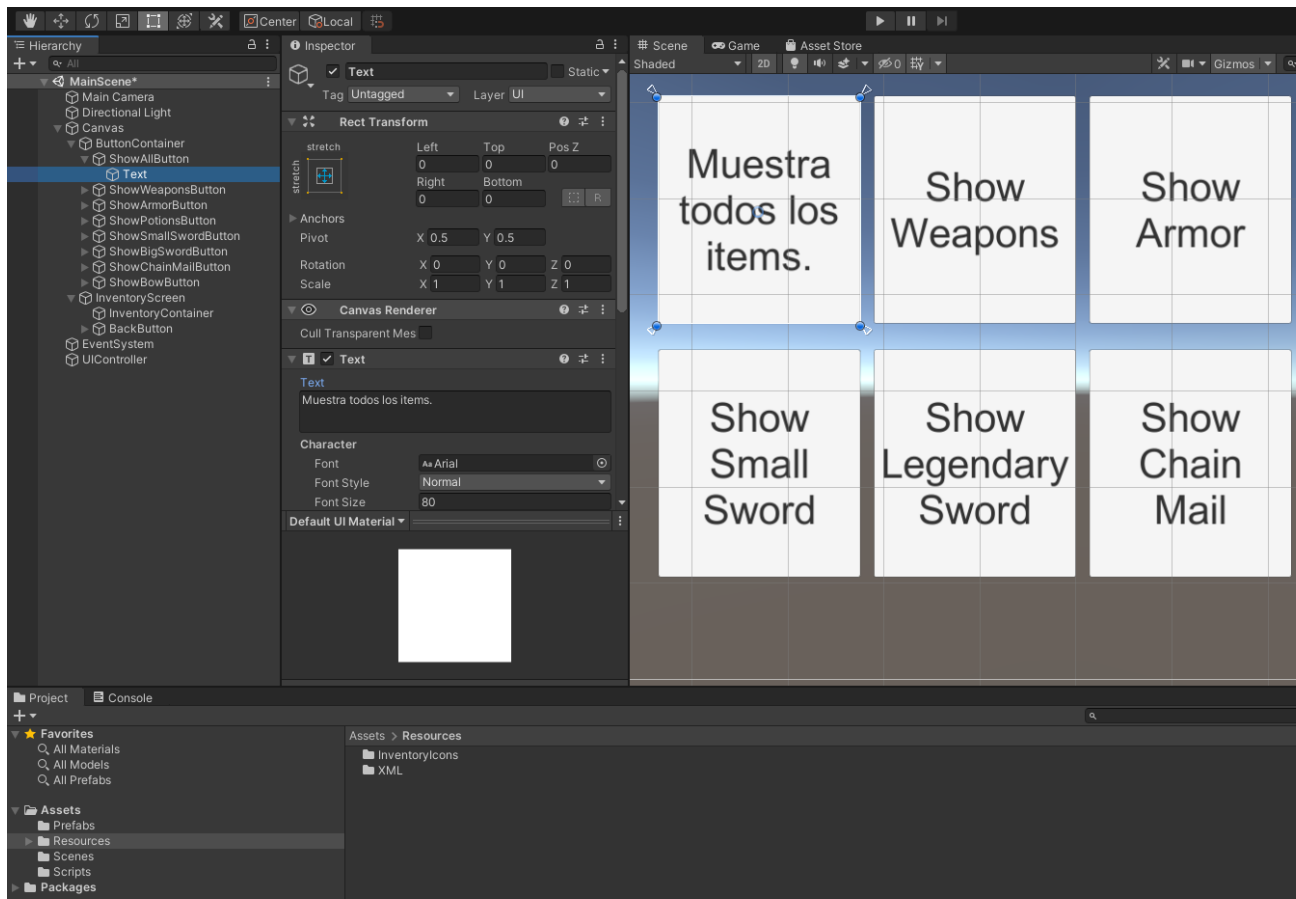
`<Image>` -> `<IMAGEN>`

`</IMAGE>` -> `</IMAGEN>`

7. Modificar el script 'UIController.cs' para que cargue correctamente el archivo XML según los nodos modificados.

Adjuntar a continuación el código MODIFICADO del script 'UIController.cs'

8. En el panel de jerarquía de la escena en Unity, modificar los paneles para que el texto que se muestre sea en castellano.





## Actividad 3. JSON

Realizar el tutorial indicado en el siguiente enlace:

<https://www.eniun.com/tutorial-json/>

## Actividad 4. JSON

### Ejercicio Avanzado 1: Anidación y Referencias en JSON

Considerando la gestión de una escuela, crea un archivo JSON que represente la información de estudiantes, profesores y asignaturas.

Cada estudiante debe tener información personal, como nombre, edad y lista de asignaturas cursadas con las calificaciones.

Cada profesor debe tener información personal y una lista de asignaturas que enseña. Las asignaturas deben incluir un nombre y la cantidad de créditos.

Asegúrate de utilizar anidación y referencias apropiadas.

### Ejercicio Avanzado 2: Manipulación de Datos JSON

Dado el siguiente JSON que representa la información de proyectos en una empresa:

```
{
  "proyectos": [
    {
      "id": 1,
      "nombre": "Sistema de Gestión",
      "equipo": [
        {"id": 101, "nombre": "Juan Pérez"},
        {"id": 102, "nombre": "María Rodríguez"}
      ],
      "tareas": [
        {"id": 201, "descripcion": "Diseño de la interfaz", "completada": true},
        {"id": 202, "descripcion": "Desarrollo del backend", "completada": false}
      ]
    },
    {
      "id": 2,
      "nombre": "Aplicación Móvil",
      "equipo": [
        {"id": 103, "nombre": "Carlos Gómez"},
        {"id": 104, "nombre": "Ana López"}
      ],
      "tareas": [
        {"id": 203, "descripcion": "Desarrollo de funciones principales", "completada": true},
        {"id": 204, "descripcion": "Pruebas de rendimiento", "completada": true}
      ]
    }
  ]
}
```

*Realiza las siguientes operaciones:*

- a) Agrega un nuevo proyecto al registro.
- b) Añade un nuevo miembro al equipo del primer proyecto.
- c) Marca la tarea "Desarrollo del backend" del primer proyecto como completada.

### **Ejercicio Avanzado 3: Validación y Transformación de Datos JSON**

Dado el siguiente JSON que representa la información de empleados de una empresa:

```
{
  "empleados": [
    {
      "id": 1,
      "nombre": "Juan Pérez",
      "cargo": "Desarrollador",
      "salario": 55000
    },
    {
      "id": 2,
      "nombre": "María Rodríguez",
      "cargo": "Analista de Datos",
      "salario": "60000"
    },
    {
      "id": 3,
      "nombre": "Carlos Gómez",
      "cargo": "Diseñador",
      "salario": 75000
    }
  ]
}
```

*Realiza las siguientes operaciones:*

- a) Asegúrate de que todos los empleados tengan un ID único.
- b) Convierte el salario de todos los empleados a números y luego incrementa el salario en un 10%.

## Ejercicio avanzado 4: Sistema de Gestión de Empleados

---

**Contexto:** Imagina que trabajas en una empresa de desarrollo de software y te han asignado la tarea de crear un sistema de gestión de empleados utilizando JSON para almacenar la información de manera estructurada.

---

### Enunciado:

Crea un archivo JSON que represente la información de los empleados de la empresa. Cada empleado debe tener información personal, como nombre, edad, cargo y salario. Además, cada empleado debe tener asignado un conjunto de proyectos en los que trabaja, con detalles sobre cada proyecto, como el nombre, la fecha de inicio y las tareas asignadas.

Utiliza la estructura adecuada en JSON para representar esta información de manera clara y organizada.

## **Actividad 5. Repositorios en GitHub**

1. *Crear un repositorio en el portal de GitHub por cada actividad de la unidad didáctica 3: Act1, Act2, Act3, Act4 .*
2. *En el ordenador del alumno/a, crear un repositorio con el comando git init.*
3. *Establecer link entre el repositorio local y el repositorio remoto en el portal GitHub.*
4. *Realizar push y añadir el directorio principal de cada proyecto al repositorio remoto.*
5. *En cada actividad definida en la plataforma Moodle, añadir capturas de pantalla de los momentos mas relevantes del proceso.*
6. *En cada actividad definida en la plataforma Moodle, añadir el enlace web de cada repositorio para la corrección.*

## Actividad 6. JSON- Unity

Crear un proyecto en Unity y añadir el siguiente script:

```
using UnityEngine;
using System.Collections.Generic;
using System.IO;

public class ManejoJSON : MonoBehaviour
{
    // Ruta del archivo JSON
    private string rutaArchivo = "Assets/datos.json";

    void Start()
    {
        // Leer datos desde el archivo JSON
        string json = File.ReadAllText(rutaArchivo);
        DatosJuego datos = JsonUtility.FromJson<DatosJuego>(json);

        // Mostrar información de cada personaje
        foreach (var personaje in datos.personajes)
        {
            Debug.Log($"Nombre: {personaje.nombre}, Nivel: {personaje.nivel},
Clase: {personaje.clase}");
        }

        // Modificar datos (por ejemplo, incrementar el nivel del aventurero)
        foreach (var personaje in datos.personajes)
        {
            if (personaje.nombre == "Aventurero")
            {
                personaje.nivel++;
            }
        }

        // Escribir de nuevo en el archivo JSON
        string nuevoJson = JsonUtility.ToJson(datos, true);
        File.WriteAllText(rutaArchivo, nuevoJson);
    }
}

[System.Serializable]
public class Personaje
{
    public string nombre;
    public int nivel;
    public string clase;
}

[System.Serializable]
public class DatosJuego
{
    public List<Personaje> personajes;
}
```

Este script utiliza la clase JsonUtility de Unity para leer y escribir en archivos JSON. Asegúrate de que las clases reflejen la estructura de tu archivo JSON. En este caso, se han definido las clases Personaje y DatosJuego. Al ejecutar este script, deberías ver la información de los personajes en la consola de Unity, modificar el nivel del aventurero y guardar los cambios en el archivo JSON.

Para la resolución del ejercicio, es necesario generar el archivo .json que se va a alojar en el directorio /Assets y se llama datos.json.