Documentación de TP I

Aplicación de Motores II

**Alumno:** Slobodianik Esteban Joel

**Profesor:** Miño Santiago Ismael

Tareas

[CustomEditor (CharStats) 3](#_Toc76083821)

[EssentialsSpawnerWindow 6](#_Toc76083822)

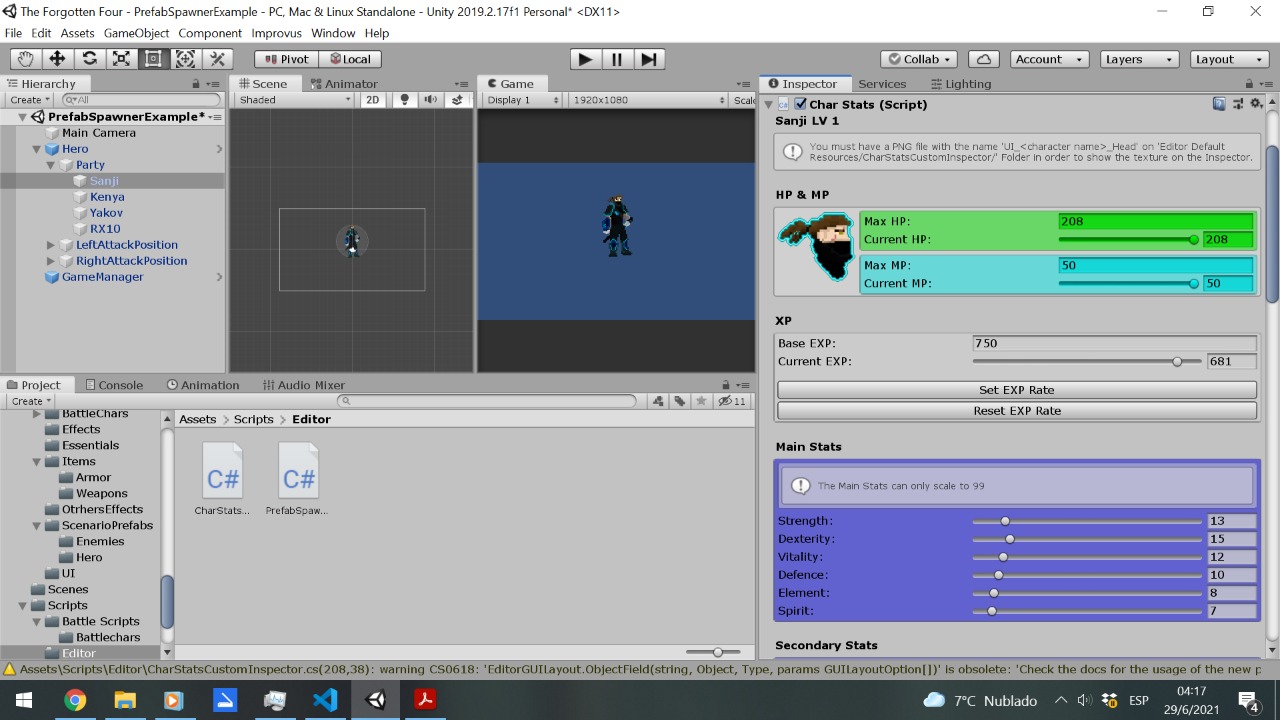
[ScenarioPrefabsManagerWindow 9](#_Toc76083823)

[ExecuteInEditMode (MovingPlatform) 14](#_Toc76083824)

[AssetDataBase 14](#_Toc76083826)

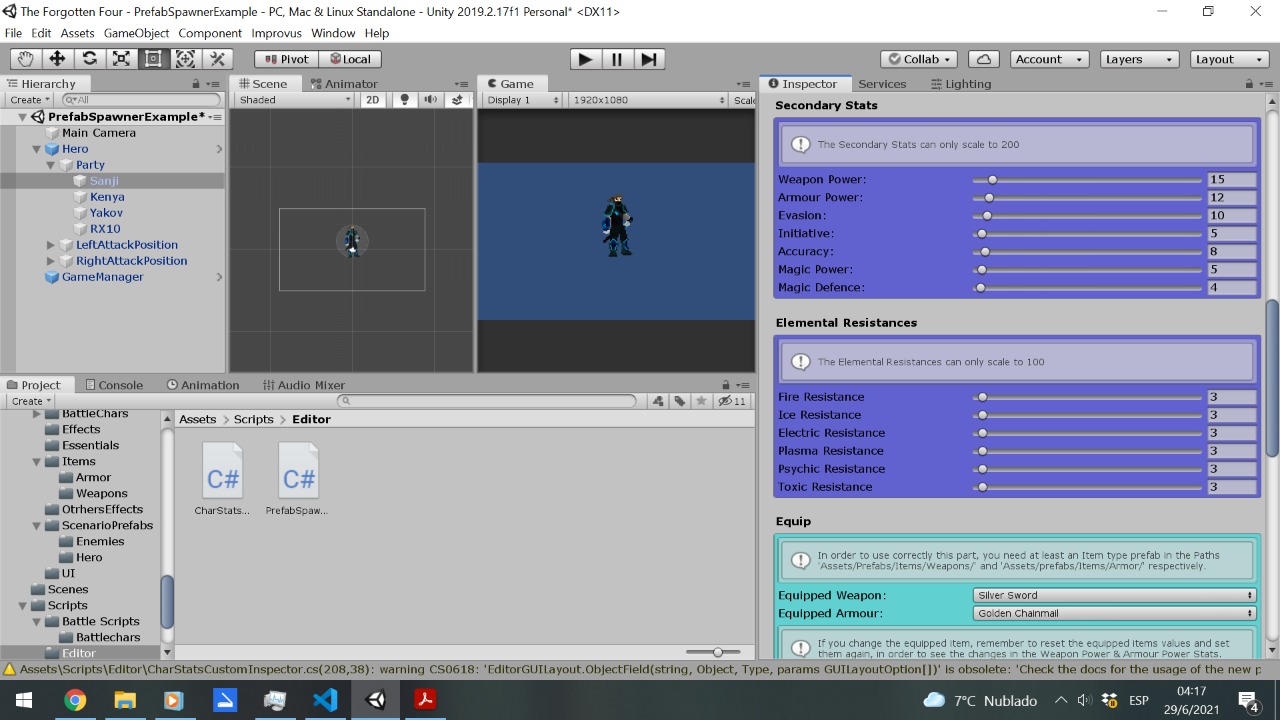
# CustomEditor (CharStats)

Esta herramienta es un CustomInspector del Script *CharStats*, el cuál maneja los stats de los personajes que el jugador maneja en el juego.



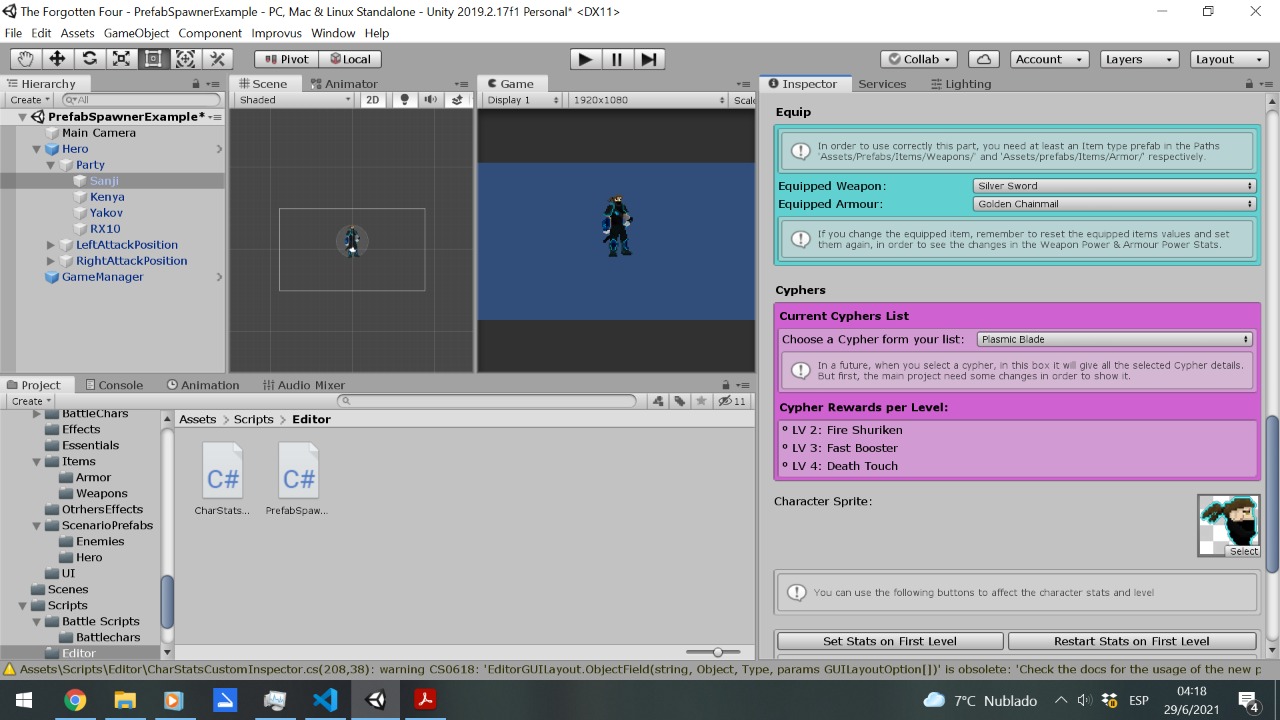
Orden de ítems en la porción del inspector:

1. En el inspector se ve el nombre del personaje y su nivel.
2. Se ve una helpBox que indica que es necesario tener imágenes con el nombre “UI\_<Nombre del Personaje>\_Head.png” en la carpeta *Editor Default Resources/CharStatsCustomInspector/* para que se vea la imagen en la siguiente parte.
3. Se muestra la imagen del personaje solicitada en el helpBox anterior y cuadros donde se puede setear la HP y MP máximos y con un slider se puede setear la HP y MP actuales.
4. Se puede setear la experiencia base y un slider de la experiencia actual. Al hacer click en “Set EXP Rate” se seteará los puntos de experiencia necesarios para subir desde el nivel 1 hasta el nivel 100. Al hacer click en “Reset EXP Rate”
5. Se ven los stats principales del personaje y sus respectivos sliders para setear sus valores (de 1 a 99).



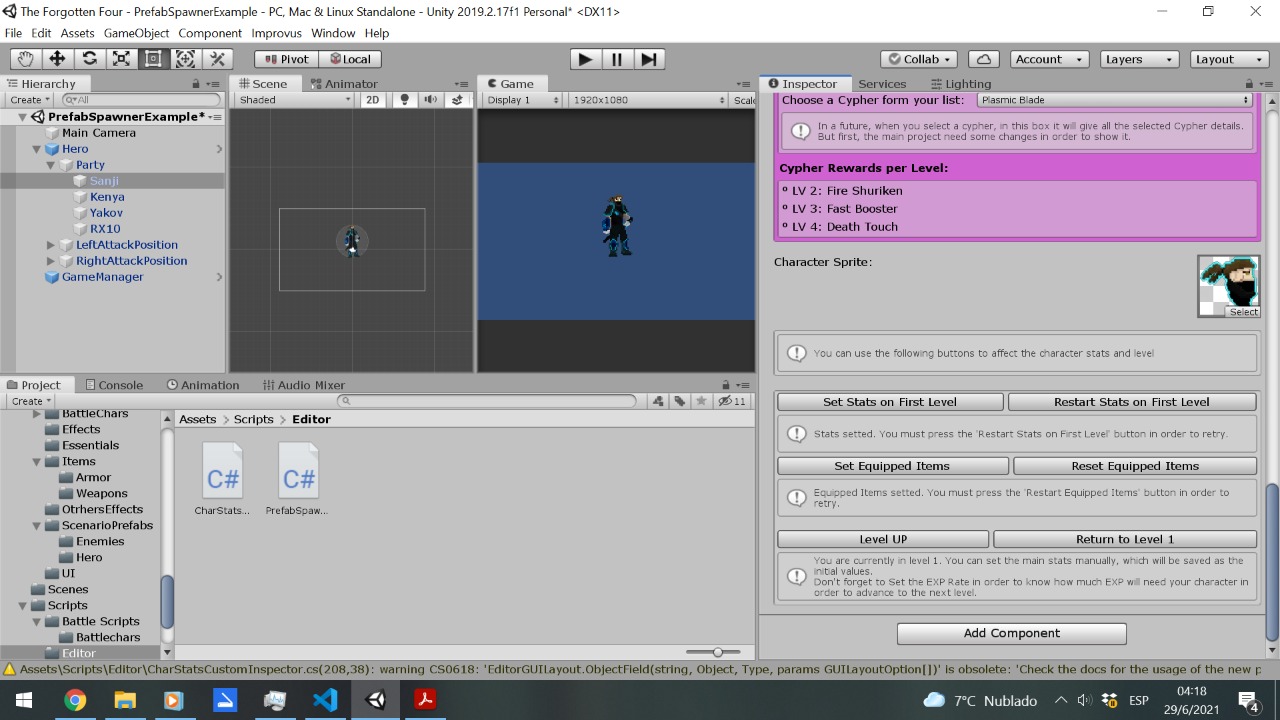
Orden de ítems en la porción del inspector:

1. Stats secundarios que se pueden setear manualmente con sus sliders de 0 a 200.
2. Resistencias elementales que se pueden setear manualmente con sus sliders de 0 a 100.



Orden de ítems en la porción del inspector:

1. Muestra el equipo que tiene el personaje con su respectiva arma y armadura. Estas armas y armaduras deben estar en la carpeta *Prefabs/Items/Weapons* y *Prefabs/Items/Armor* respectivamente. Sólo se mostrarán las armas y armaduras compatibles con el personaje para elegir.
2. Muestra una lista de las magias (Cyphers) actuales del personaje y los Cyphers que se adquieren al subir de nivel.
3. Se muestra un objectField del sprite del personaje que se usará para gran parte del UI.

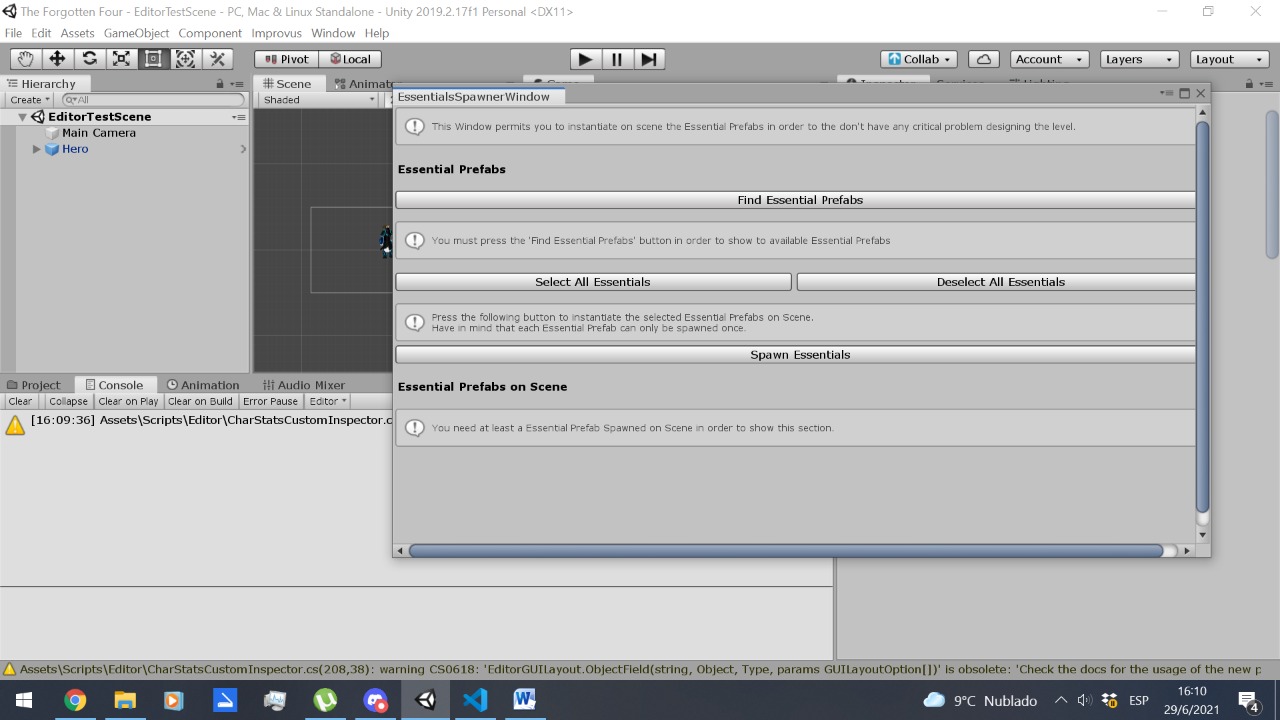


Orden de ítems en la porción del inspector:

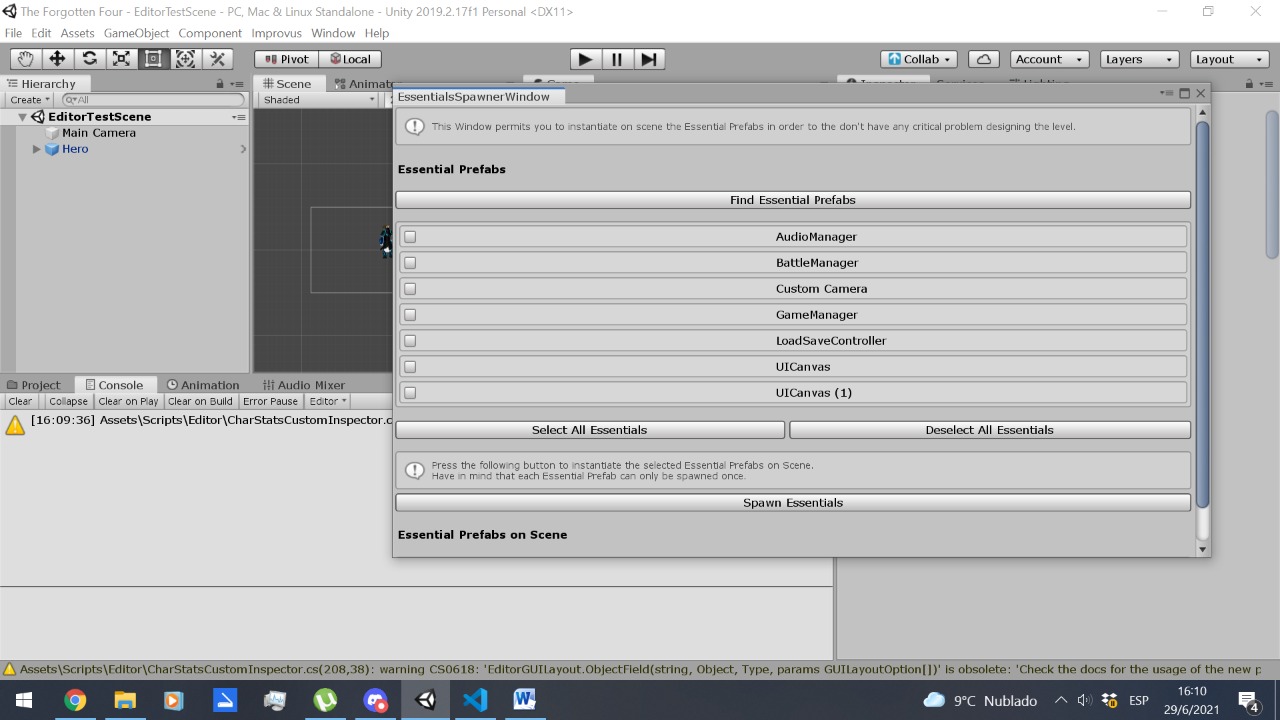
1. “Set Stats on First Level”: por medio de programación se setean los stats secundarios y resistencias elementales, a partir de los valores actuales de los stats principales.
2. “Reset Stats on First Level”: resetea los valores de los stats secundarios y resistencias elementales a 0.
3. “Set Equipped Items”: Le suma el valor del arma y armadura actuales a los stats secundarios *Weapon Power* y *Armour Power* respectivamente.
4. “Reset Equipped Items”: Le resta el valor del arma y armadura actuales stats secundarios *Weapon Power* y *Armour Power* respectivamente.
5. “Level UP”: Sube un nivel del personaje, aumentando los stats secundarios y resistencias elementales. Cada tres niveles también se aumentan los stats principales. Además de que se agrega un Cypher nuevo por cada nivel aumentando.
6. “Return to Level 1”: Devuelve todos los stats y cyphers al estado del nivel 1.

# EssentialsSpawnerWindow

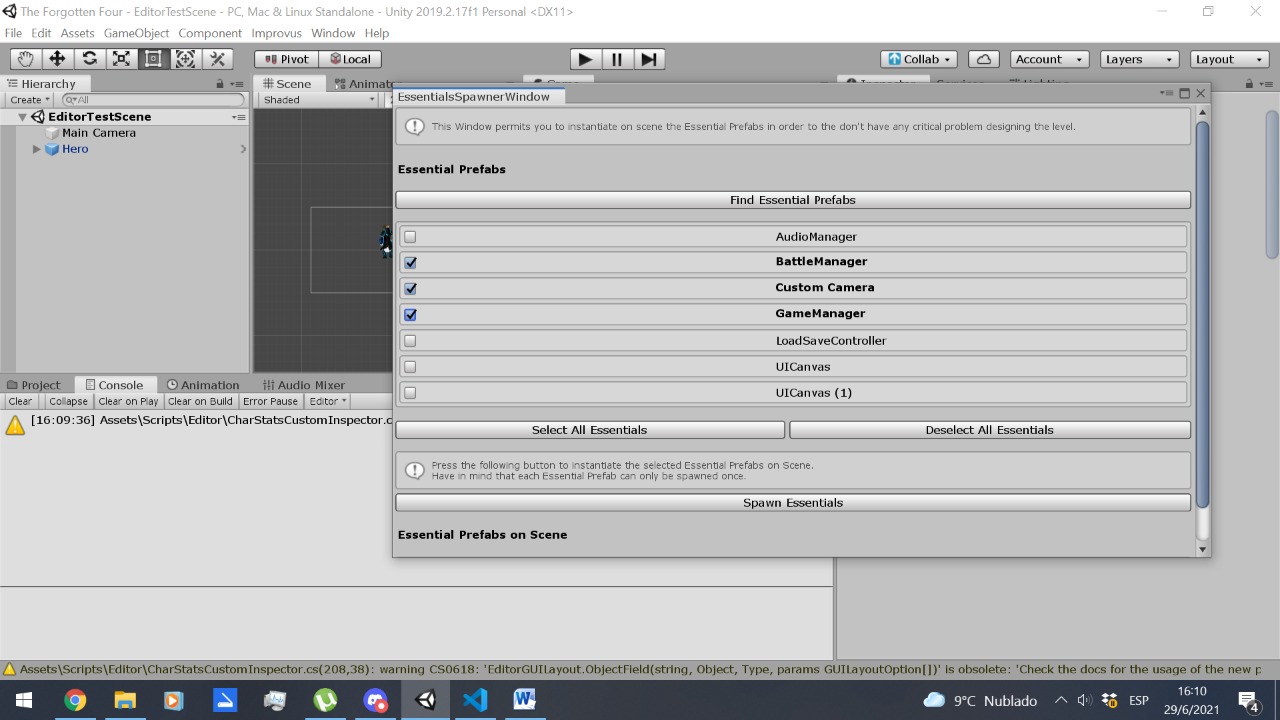
Esta herramienta es una ventana encargada de instanciar en la escena los Prefabs que son considerados “Esenciales” para el juego (tales como el GameManager, AudioManager, etc.).



En primera instancia, tenemos el botón “Find Essential Prefabs”, el cual buscará todos los prefabs ubicados en la carpeta *Prefabs/Essentials*



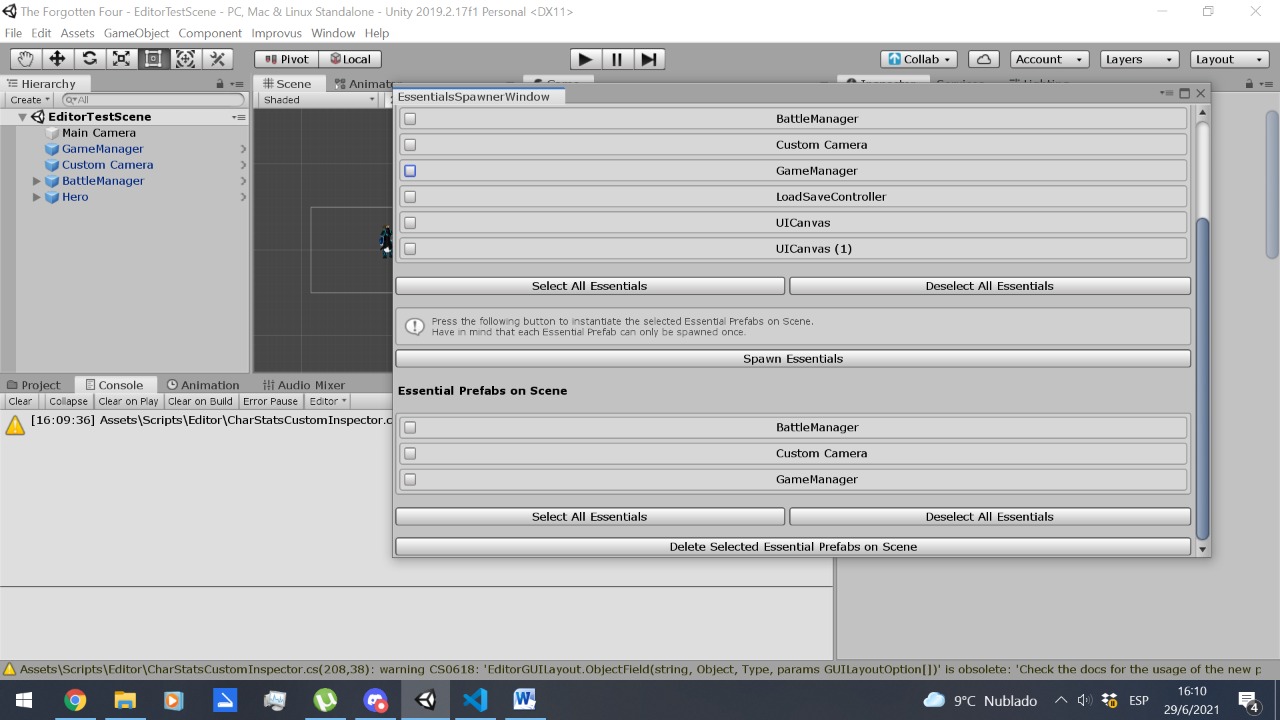
Al hacer click en el botón, se mostrará un listado de todos los prefabs esenciales junto con sus respectivos toggles al seleccionarlos.



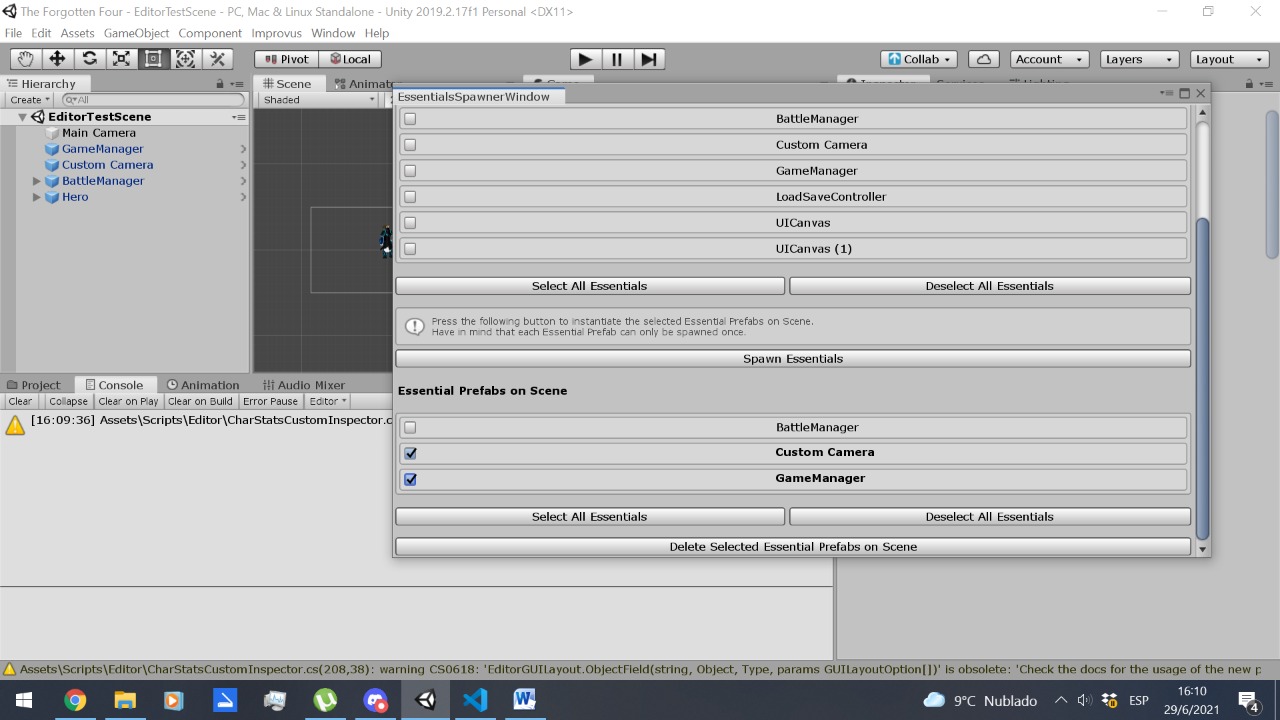
Al seleccionar a uno o más elementos, estos se pondrán en negrita.

Los botones “Select All Essentials” y “Deselect All Essentials” permiten seleccionar y deseleccionar respectivamente todo el listado de prefabs.

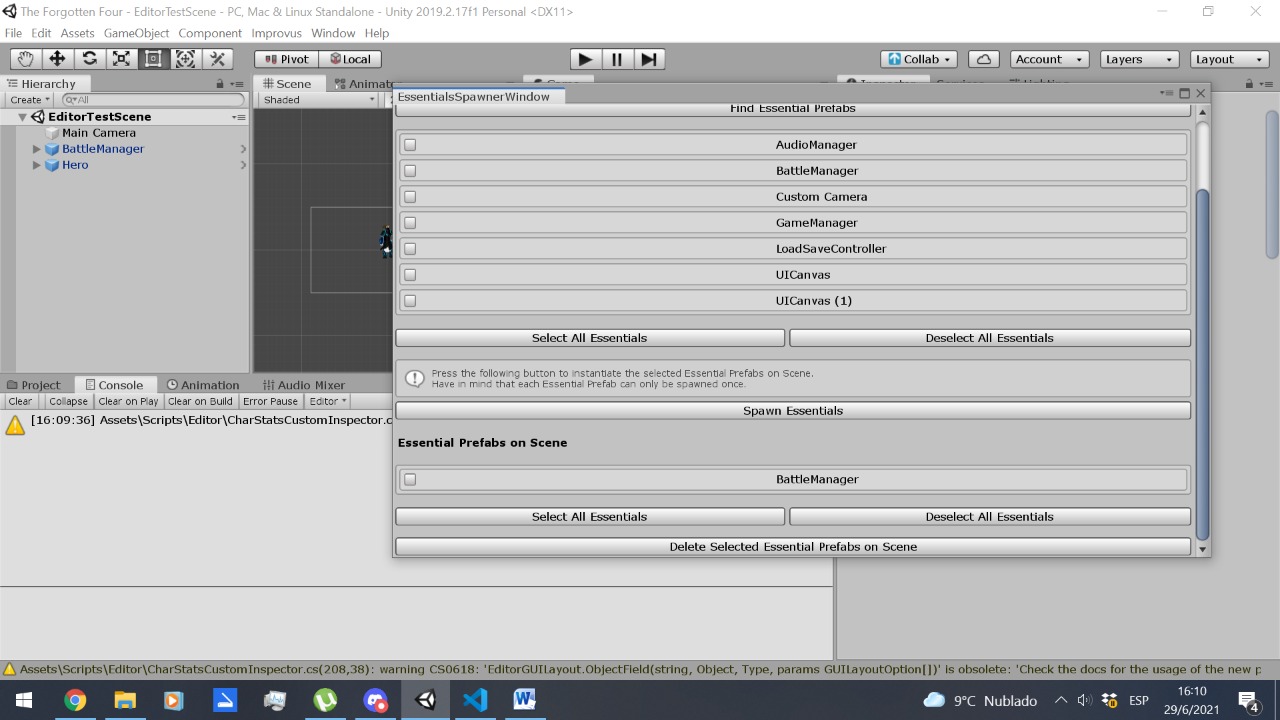
Por último, el botón “Spawn Essentials” instanciará los prefabs seleccionados. Tener en cuenta que la ventana solo instanciará una vez a cada uno de los prefabs esenciales, por lo que si seleccionas varias veces un prefab y le das al botón solo lo instanciará la primera vez.



Cuando se instancien los prefabs, se mostrará en la ventana un listado de los prefabs esenciales en escena con sus respectivos toggles y botones “Select All Essentials” y “Deselect All Essentials”.



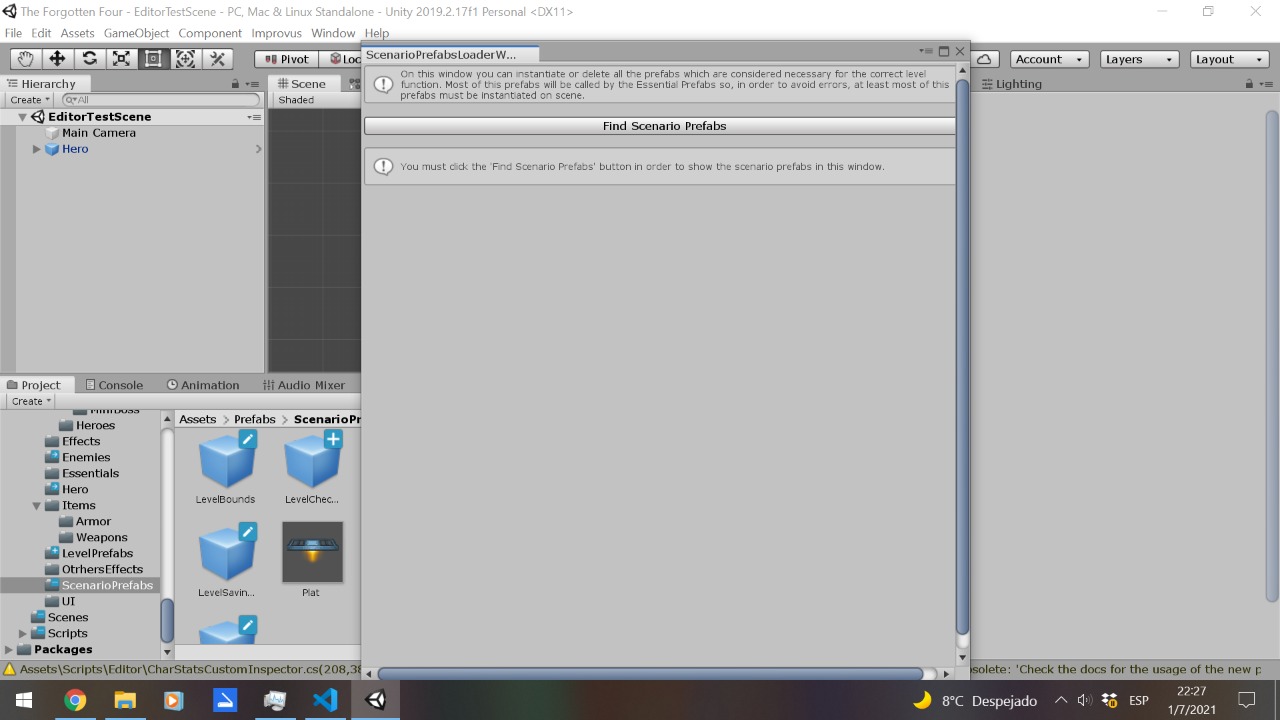
Por último, el botón “Delete Selected Essential Prefabs on Scene”, eliminará de la escena los prefabs seleccionados.



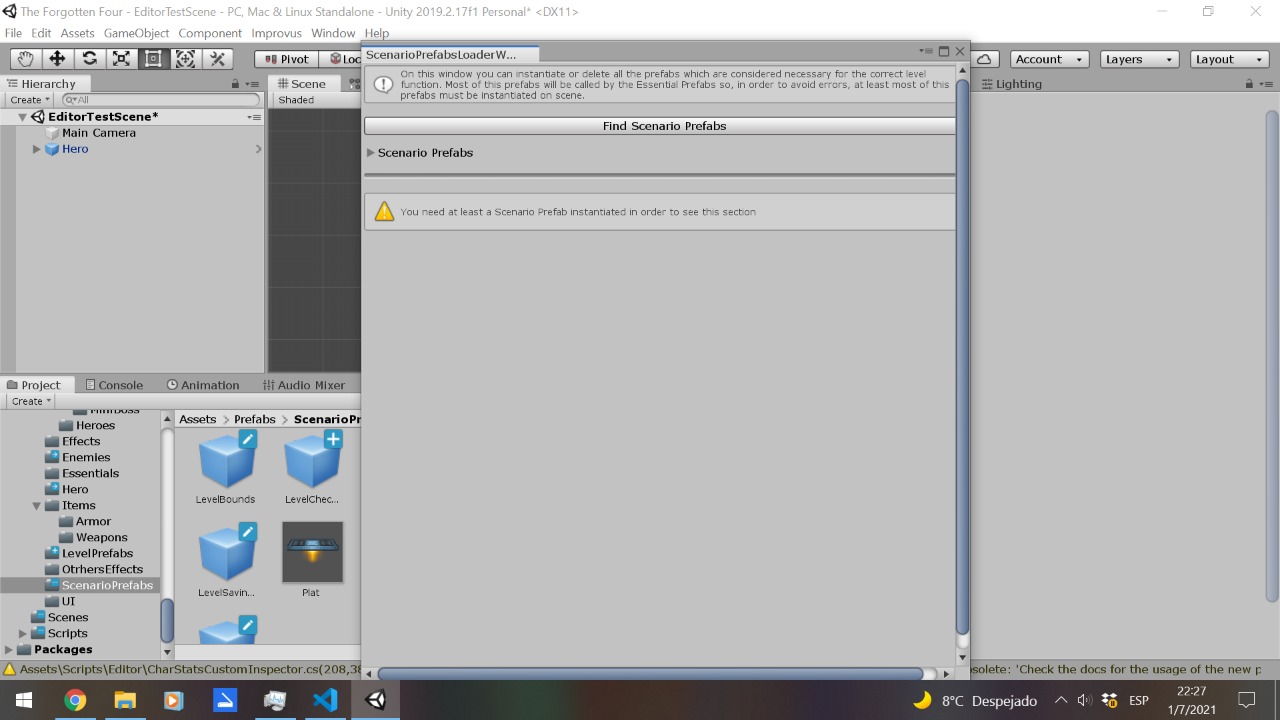
Los prefabs se borrarán tanto de la escena como del listado.

# ScenarioPrefabsManagerWindow

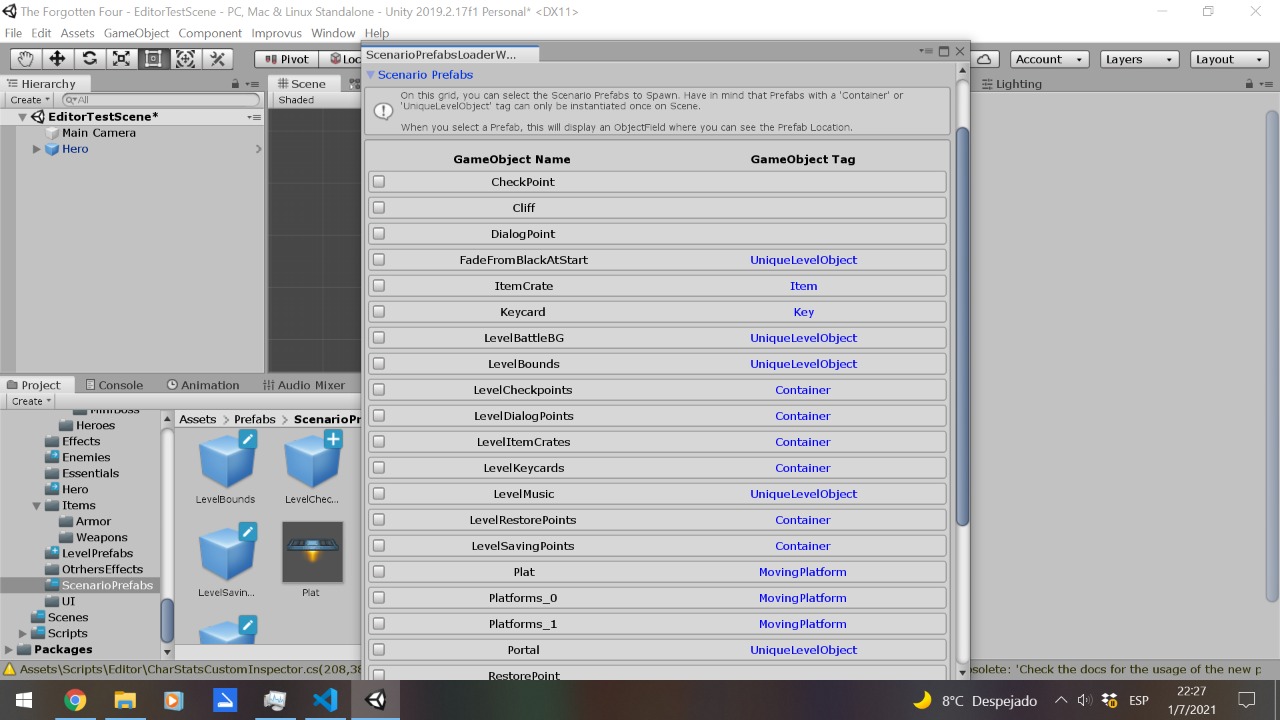
Esta ventana se encarga de instanciar, borrar y organizar los distintos prefabs del escenario que son (de mayor o menor medida) necesarios para el correcto funcionamiento del nivel (principalmente componentes que necesitan los prefabs esenciales).



En primer lugar, se muestra el botón “Find Scenario Prefabs” que al hacer click, mostrará todos los prefabs ubicados en la carpeta *Prefabs/ScenarioPrefabs*.

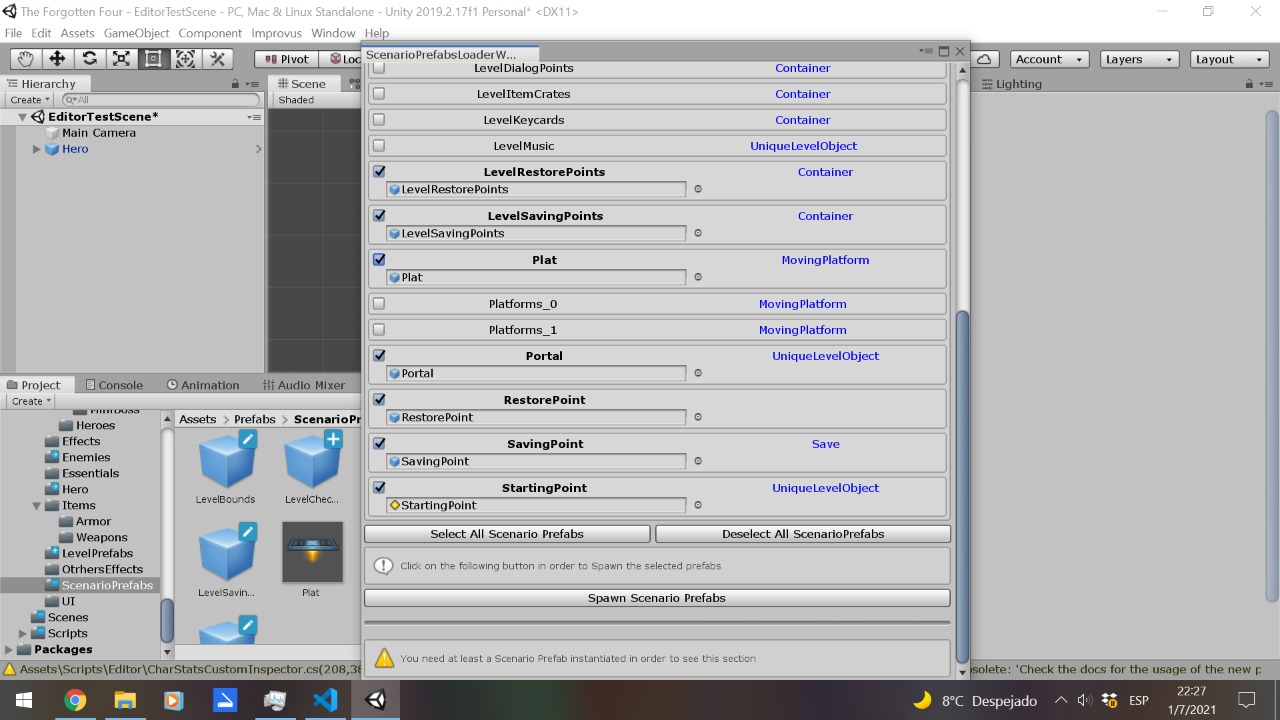


Se mostrará un Foldout que al hacer click se mostrará el listado.

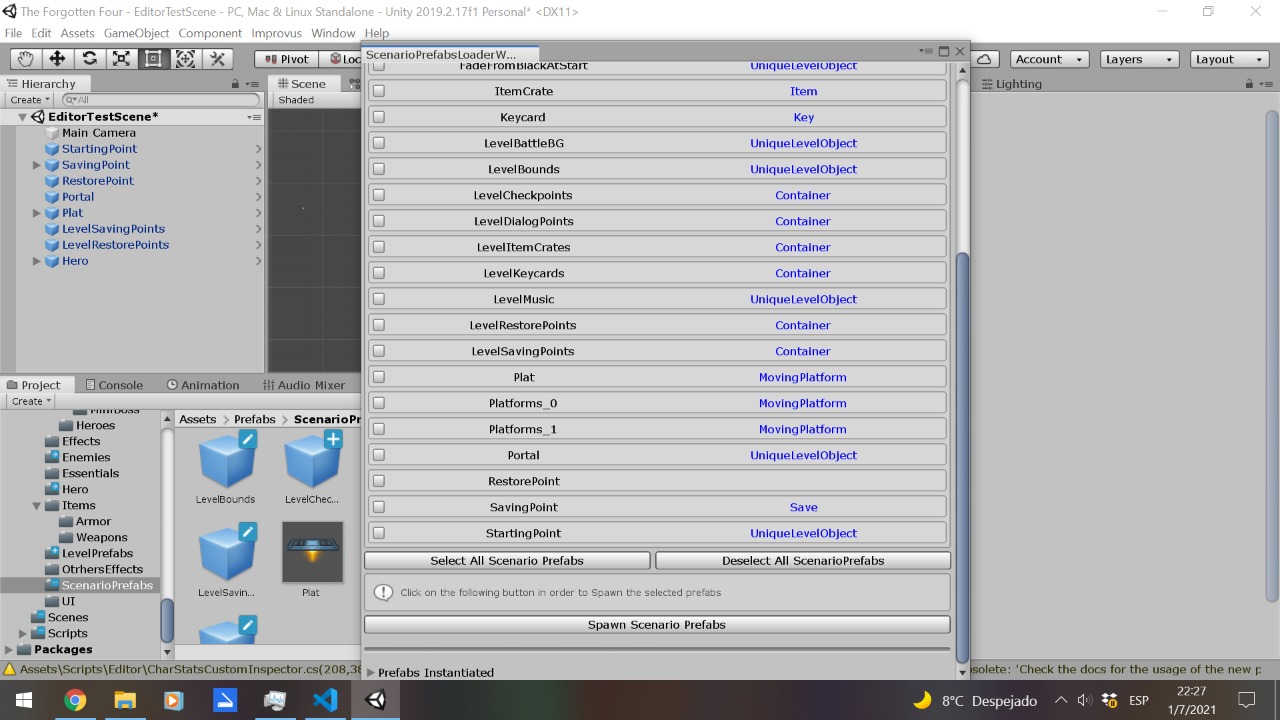


El listado mostrará el Toggle para seleccionar/deseleccionar el elemento, el nombre del prefab y el tag. Este último es necesario para la organización de los Scenario Prefabs. Esta herramienta categoriza los Scenarios Prefabs de la siguiente manera:

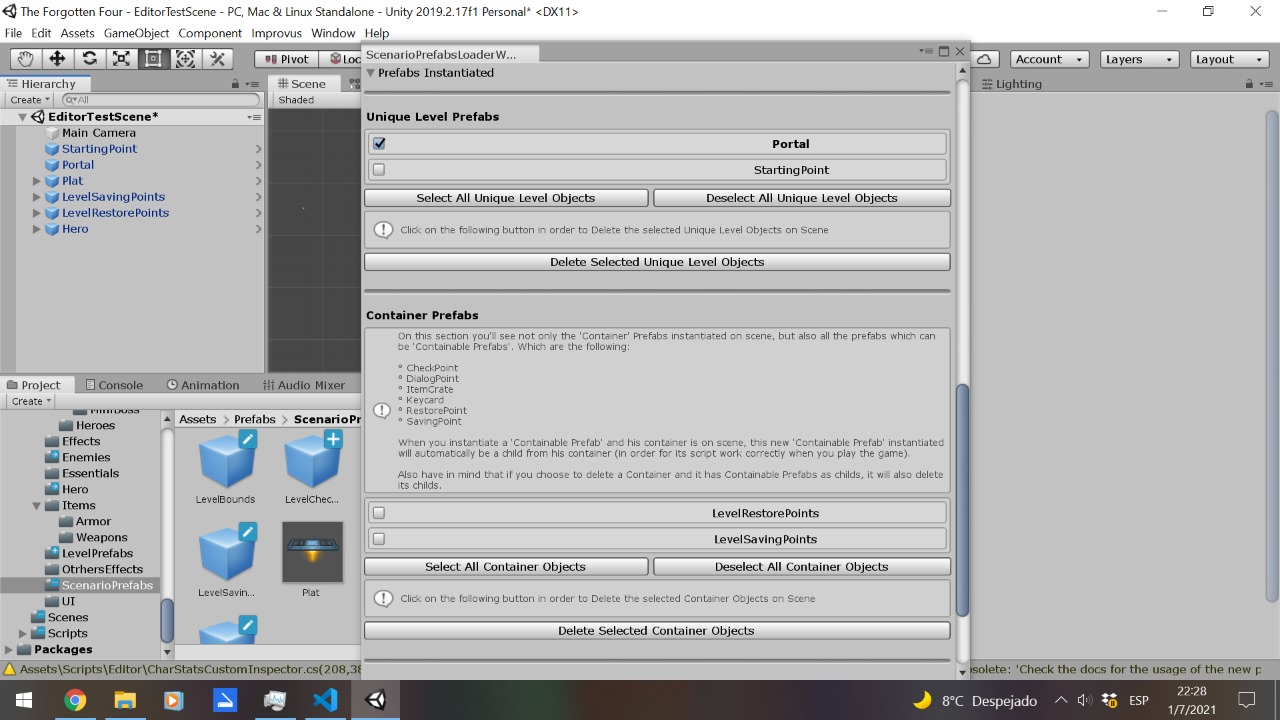
* *Unique Level Objects:* Prefabs que solo pueden instanciarse una vez que son necesarios para el correcto funcionamiento de los prefabs esenciales. Ejemplo: *LevelMusic* contiene la música del nivel, la cual se asignará en el AudioManager a la hora de iniciar un nivel.
* *Containers:* Prefabs que, al igual que los *Unique Level Objects* solo pueden instanciarse una vez y que son necesarios para el correcto funcionamiento de los esenciales, también son considerados contenedores de otros Scenario Prefabs. Ejemplo: *LevelItemCrates* es contenedor de *ItemCrates* el cuál, al iniciar el nivel, asignará todos los *ItemCrates* en un array para que después dicho array sea usado por los prefabs esenciales.
* *Containable Prefabs*: Son los prefabs que son contenidos por los Containers. Pueden instanciarse más de una vez.
* *Other Prefabs*: El resto de prefabs que no son totalmente necesarios pero sí recomendados. Ejemplo: Plataformas movibles y precipicios.



Al igual que la herramienta anterior, al seleccionar uno o varios prefabs los textos se ponen en negrita pero también se mostrará el GameObject para saber qué es lo que se está instanciando (en el caso de que se quiera ver su contenido). También cuenta con los botones “Select All”, “Deselect All” y “Spawn” al igual que la herramienta anterior para seleccionar, deseleccionar e instanciar los prefabs seleccionados.



Al igual que la herramienta anterior, los prefabs serán instanciados en la escena y serán deseleccionados en la ventana.



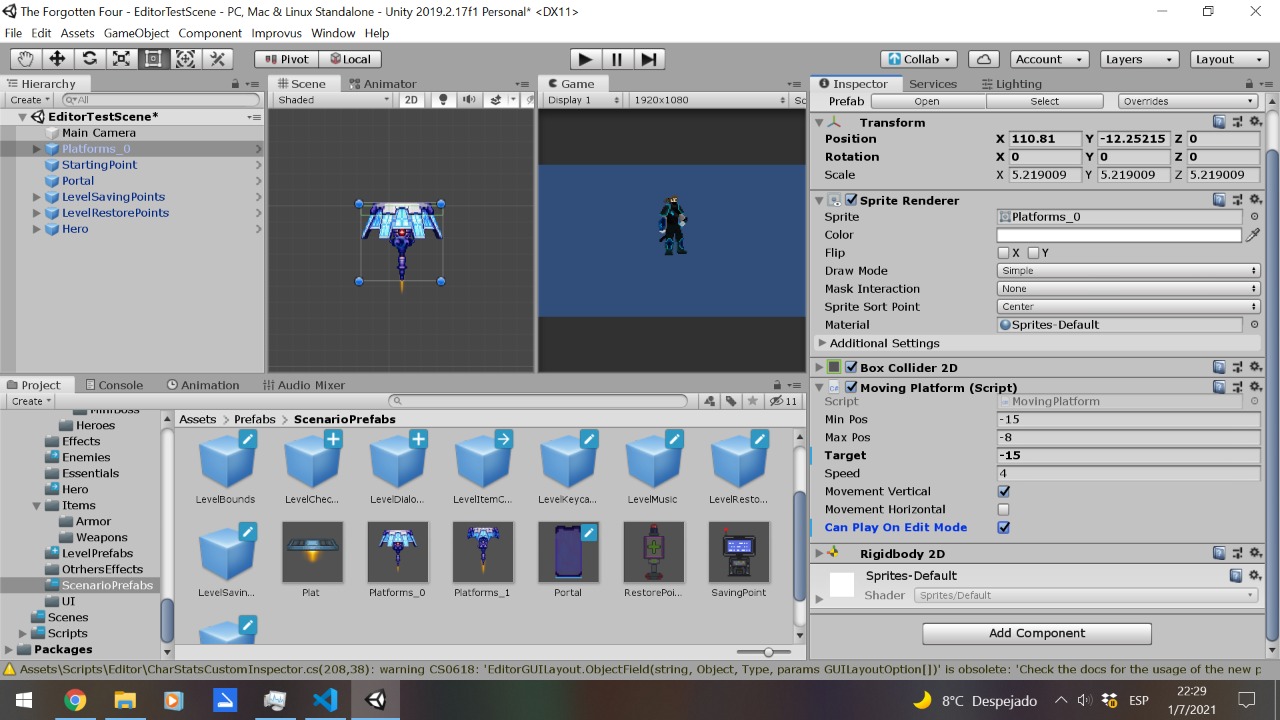
Los prefabs instanciados se mostrarán un Foldout y se dividirán en las categorías anteriormente mencionadas (con excepción de los *Containable Prefabs* que se mostrarán como elementos dentro de los elementos de los *Containers*). Cada sección cuenta con sus botones de “Select All”, “Deselect All” y “Delete”.

Algo para tener en cuenta, cuando se instancia un *Container* buscará si hay prefabs que puede contener en la escena. En el caso de encontrar al menos uno, no solo aparecerá en el listado, sino que también se volverán hijos del *Container* en la escena.

# ExecuteInEditMode (MovingPlatform)

Esta herramienta permite a los GameObjects con el Script *MovingPlatform* poder moverse dentro de la Escena, siempre y cuando la variable “Can Play On Edit Mode” esté en true. Su principal función es testear el movimiento para medir tanto los límites del movimiento como para indicar si el movimiento es vertical, horizontal o ambos (diagonal).

# 



# AssetDataBase

Aclaración: el CustomInspector y los dos EditorWindow utilizan AssetDataBase en sus respectivos Scripts en la carpeta *Scripts/Editor*