**Formato de escenarios y casos de uso Stardew Valley**

**Configuración de los Escenarios**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| createChestSetup | controller | The user performs the action of creating a chest,initializing the chest list, with the condition of having 50 pieces of wood.The expected outcome is the successful creation of the chest. In the first case, the player has enough wood and the chest is created. In the second case, the user does not have enough wood, and the system displays an error message. |
| HarvestSetup | controller | the user performs the action of harvesting the planted seeds, with the condition that 28 days had elapsed after the seed was planted. The expected result is the plant to be added into the users inventory. In the first case the user harvests correctly the seed. In the other case, not enough time has elapsed and the system displays an error message. |
| PlantingSetup |  | The user must be able to plant a specific type of seed in a cropfield, making it so that the seed starts growing |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| PlaceItemSetup | controller | The user performs the action of placing an item inside of a chest, there are no conditions and the expected outcome is for the item to be eliminated from the players inventory and placed inside the chest. First case the item is successfully placed into the chest and eliminated from the player´s inventory. Second case the item is not compatible with the chest´s classification and an error message is displayed by the system. |
| TakeOutItemSetup | controller | The user performs the action of taking out an item from a chest, there are no conditions and the expected result is an item being taken out the chest and saved in the user´s inventory. First case the user successfully takes out the item. |
| ClassifyItemSetup | controller | The user performs the action of classifying a chest by the item type, there are no conditions and the expected result is that the chest starts accepting only items of that type. First case the chest is correctly classified and only a certain type of item is accepted. Second case the user tries to classify a chest with an item inside that does not match the type of item he was trying to classificate the chest with and the system displays an error message. |
| ItemSearchingSetup | controller | The user performs the action of searching an specific item, there are no conditions and the expected result is for the chest to only display an item with a name that matches the search result. First case the user successfully searches and displays the item. Second case the user tries to search an item thats not inside the chest and nothing happens. |
| getSeasonPlantationSetup | plantation | the system receives the plantation´s season and plantation list is initiated |
| getNamePlantationSetup | plantation | the system receives the plantation´s name |
| getGrowthDaysPlantationSetup | plantations | the system receives the plantation´s growth days |
|  |  |  |

…

\* El nombre de los escenarios puede ser setupStage1, setupStage2, etc.

\* La clase es la clase de testing correspondiente al modelo donde acontece el escenario. Por ejemplo si usted está probando User, clase será UserTest.

\* El escenario es la descripción de las condiciones iniciales del escenario.

**Diseño de Casos de Prueba**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| plantation | getName | getNamePlantationSetup |  | se obtiene el nombre de la plantation |
| plantation | getSeason | getSeasonPlantationSetup |  | se obtiene el nombre de la plantation |
| plantation | getGrowthDays | getGrowthDaysPlantationSetup |  | se obtiene el nombre de la plantation |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| controller | moveplantationFromCropfieldToInventory | TakeOutItemSetup | plantationName | the plantation is moved from the cropfield to the inventory |
| controller | createChest | chestSetup | chestName, classification | se crea un coffre exitosamente |
| Chest | addPlantation | placeItemSetup | object plantation | se añade un stack al cofre |
| chest | getClassificationType | ClassifyItemSetup |  | se obtiene el tipo de clasificacion |
| chest | getStacks |  |  | se obtiene un stack |
| Node | getNextNode |  |  | se obtiene el proximo nodo dentro de las listas enlazadas |
| node | getData |  |  | se obtienen los elementos dentro del nodo |
| controller | CheckResourcesForChest | chestSetup |  | se valida que hayan suficientes materiales para construir el cofre |
| controller | movePlantationTochest | PlaceItemSetup | object chest | la plantation es colocada dentro del chest |
| plantation | validateInvalidGrowthDays | plantingSetup | plantation | La semilla es plantada |
| controller | getplantationfrominventory | takeOutItemSetup | plantation | El item es retirado del inventario al cofre |
| Controller | organizechestByGrowDays | organizeChestSetup | plantation | El cofre es organizado por días de crecimiento |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

…

\* Una prueba se compone de un conjunto de casos de prueba.

\* Cada fila representa un ***caso de prueba*** difente

\* En el objetivo de la prueba debe escribir una descripción sobre qué es lo que específicamente está probando del modelo del programa.

\* La clase es la clase del modelo que está siendo puesto a prueba.

\* El método es específicamente el método de la clase que está siendo puesto a prueba.

\* El escenario se refiere al nombre del escenario que usted definió. Todos los casos de prueba corresponden a escenarios.

\* Los valores de entrada son valores que entran al método puesto a prueba.

\* El resultado esperado es lo que se espera que suceda luego de ejecutar el método.