**Formato de escenarios y casos de uso**

**Configuración de los Escenarios**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| stage1 | testUser | Se crea un usuario { "Jacobo", "1006107372" } |
| Stage2 | testShow | Se crea un usuario { "Jacobo", "1006107372" } y se crea un show el { 2022, 12, 2} con la película { "Harry potter"} en una sala {MINI} |
| Stage3 | testShowReserved | Se crea todo lo anterior y adicional se agrega un cliente { "Juan", "1003131"} en la fila [1] y columna [1], que vendría siendo el asiento reservado |

\* 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:** Lo que se esta probando del modelo del programa es la clase CineController es verificar el usuario por su id, verificar que al crear una función que no se pueda crear otra a esa misma hora y también se busca verificar que si hay una función y un cliente ya reservo una silla no se pueda meter otro cliente ahí porque ya esta reservada. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| CineController | userData | testUser | -name  .password  -userId | Verificar correctamente el usuario. |
| CineController | Shows | testShow | -date  -time  -[][] customers  -movie  -durationMovie  -viewersCount  -timeString  -type | Verificar la creación de una función y que no se pueda crear una igual a la misma hora |
| CineController | Catalogue | testShowReserved | -name  -duration | Verificar que haya una función y que haya un cliente que ya reservo una silla para que no se pueda meter otro cliente ahí porque ya está reservada. |

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