**Stage settings**

|  |  |  |
| --- | --- | --- |
| **Name** | **Class** | **Stage** |
| setUp1 | CompanyTest | An object of the Company class in its initial state |
| setUp2 | CompanyTest | An object of the Company class with lastTurnGiven = “D99” |
| setUp3 | CompanyTest | An object of the Company class with lastTurnGiven = “Z99” |
| setUp4 | CompanyTest | An object of the Company class and an object of the User class is added with documentType = CC, documentNumber = “1144105003”, firstName = “Sebastián” and lastName = “Barrera” which is assigned the first "A00" turn. |
| setUp5 | CompanyTest | An object of the Company class and adds two objects of the User class:   * The first one with documentType = CC, documentNumber = “1144105003”, firstName = “Sebastián” and lastName = “Barrera” * The second one withdocumentType = CC, documentNumber = “19493420”, firstName = “Edilberto” and lastName = “Barrera”   Then the first one will take the first Turn “A00” |
| setUp1 | TurnTest | An object of the Turn class with the turn “A00” |
| setUp1 | UserTest | An object of the User Class with documentType = CC, documentNumber = “1144105003”, firstName = “Sebastián” and lastName = “Barrera” without turn |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Target:**  Verify that the alreadyRegist method of the Company class assigns a turn correctly, as well as when the user already has an active turn and then returns the same turn, as when the user doesn't have any active turn and it will generate a new one | | | | |
| **Class** | **Method** | **Stage** | **Input values** | **Result** |
| Company | alreadyRegist | setUp4 | - | The user turn will not change and show A00 |
| Company | areadyRegist | setUp4 | - | The new user turn will be A01 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Target:** Verify that the advanceGivenTurn method of the Company class works correctly, advancing the turn properly, either in the first position, in one in the middle and in the last before turning around. | | | | |
| **Class** | **Method** | **Stage** | **Input values** | **Result** |
| Company | advanceGivenTurn | setUp1 | - | The last turn given will be A00 |
| Company | advanceGivenTurn | setUp2 | - | The last turn given will be E00 |
| Company | advanceGivenTurn | setUp3 | - | The last turn given will be A00 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Target:**  Verify that the searchUser method in Company class works correctly, returning the user searching for when it is needed. In case the user already exists to search and there is more than one user in the program. And in case it doesn't exist and there are no users in the program, it will be throwed the exception WithOutRegisterException. | | | | |
| **Class** | **Method** | **Stage** | **Input values** | **Result** |
| Company | searchUser | setUp1 | documentNumber=”0”, dt="CC” | The user is not found |
| Company | searchUser | setUp5 | - | Return the user searched |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Target:**  Verify that the isPresent method of the Company class works correctly, serving each user correctly, it also confirms that regardless of whether different users have been assigned turns, the turn taken does not progress. It also confirms that regardless of whether different users have been assigned turns, the turn taken does not progress. | | | | |
| **Class** | **Method** | **Stage** | **Input values** | **Result** |
| Company | isPresent | setUp5 | - | Since no turn is yet served, the next turn to attend will be A00 |
| Company | isPresent | setUp5 | - | Having properly served the first user, the next turn will be A02 |
| Company | isPresent | setUp5 | - | Having properly served the next user, the next turn will be A02 |
| Company | isPresent | setUp5 | - | Having properly served the next user, the next turn will be A02 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Target:**  Verify that the addUser method of the Company class works correctly, serving each user correctly, it also confirms that regardless of whether different users have been assigned turns the turn taken does not advance. It also verifies that the same user can be served twice. It also rectifies that the next turn to be attended is consulted, there are even more turns to attend. At the same time verify all the exceptions that can throw. | | | | |
| **Class** | **Method** | **Stage** | **Input values** | **Result** |
| Company | addUser | setUp1 | dt=”CC”, documentNumber=”1144105003”,  firstName=”Sebastián”  lastName=”Barrera” | The controller now have one user with first name Sebastian, last name Barrera, document type CC and document number 1144105003 |
| Company | addUser | setUp1 | dt=”CC”, documentNumber=”1144105003”,  firstName=”Sebastián”  lastName=”Barrera” | The program don’t add the user because he is already added |
| Company | addUser | setUp1 | dt=”CC”, documentNumber=”19493420”,  firstName=”Edilberto”  lastName=”Barrera” | The controller now have two users, the first one with first name Sebastian, last name Barrera, document type CC and document number 1144105003, and the second one with first name Edilberto, last name Barrera, document type CC and document number 19493420 |
| Company | addUser | setUp1 | dt=””, documentNumber=”1144105003”,  firstName=”Sebastián”  lastName=”Barrera” | The exception RequieredFieldsException wil be throwed |
| Company | addUser | setUp1 | dt=”cc”, documentNumber=”1144105003”,  firstName=”Sebastián”  lastName=”Barrera” | The exception InvalidDocumentType wil be throwed |
| Company | addUser | setUp1 | dt=”CC”, documentNumber=””,  firstName=”Sebastián”  lastName=”Barrera” | The exception RequieredFieldsException wil be throwed |
| Company | addUser | setUp1 | dt=”CC”, documentNumber=”1144105003”,  firstName=””  lastName=”Barrera” | The exception RequieredFieldsException wil be throwed |
| Company | addUser | setUp1 | dt=”CC”, documentNumber=”1144105003”,  firstName=”Sebastián”  lastName=”” | The exception RequieredFieldsException wil be throwed |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Target:**  Verify that the Constructor method of the Company class create correctly the object in his initial state | | | | |
| **Class** | **Method** | **Stage** | **Input values** | **Result** |
| Company | Company | - | - | The Company object is create correctly |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Target:**  Verify that the Constructor method of the User class create correctly the object in his initial state | | | | |
| **Class** | **Method** | **Stage** | **Input values** | **Result** |
| User | User | - | dt= CC  dn = “1144105003” fn = “Sebastián”. sn = “Barrera” | The User object is create in a correctly way |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Target:**  Verify that the Constructor method of the Company class create correctly the object in his initial state | | | | |
| **Class** | **Method** | **Stage** | **Input values** | **Result** |
| Turn | Turn | - | - | The Company object is create correctly |