**PRINCIPAL:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Principal | addNewUser(); | Inicializa una nueva instancia de la clase Principal junto con un nuevo usuario con sus respectivos valores. |  | Se obtiene un valor verdadero de la posición en la cual se agregó al nuevo usuario. |

**Objetivo:** Probar que el método de añadir usuarios funciona de manera correcta.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Principal | signIn(); | Inicializa una nueva instancia de la clase Principal junto con un nuevo usuario con sus respectivos valores. |  | Se obtiene un valor verdadero al intentar ingresar con el usuario previamente creado. |
|  |  |  |  |  |

**Objetivo:** Probar que el método de ingresar un usuario funciona de manera correcta.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Principal | searchHotelByName(); | Inicializa una nueva instancia de la clase Principal junto a un ArrayList de hoteles con sus respectivos valores. |  | Se obtiene que, al buscar un hotel por su nombre este sí se encuentra entre los hoteles existentes. |
|  |  |  |  |  |

**Objetivo:** Probar que el método de búsqueda de un hotel por su nombre funciona de manera correcta.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Principal | searchHotelById(); | Inicializa una nueva instancia de la clase Principal junto a un ArrayList de hoteles con sus respectivos valores. |  | Se obtiene que, al buscar un hotel por su id este sí se encuentra entre los hoteles existentes |
|  |  |  |  |  |

**Objetivo:** Probar que el método de búsqueda de un hotel por su id funciona de manera correcta.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Principal | addFavoriteRoom(); | Inicializa una nueva instancia de la clase Principal junto a un nuevo usuario y un hotel, cada uno con sus respectivos valores. |  | Se obtiene un valor verdadero al evaluar si el nuevo hotel favorito del usuario coincide con el asignado. |
|  |  |  |  |  |

**Objetivo:** Probar que el método de añadir un nuevo hotel favorito para el usuario funciona correctamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Principal | createCustomList(); | Inicializa una nueva instancia de la clase Principal junto con un nuevo usuario con sus respectivos valores. |  | Se obtiene que, al buscar la nueva lista creada en el usuario esta es la misma que la asignada. |
|  |  |  |  |  |

**Objetivo:** Probar que el método de crear una nueva lista personalizada para el usuario funciona de manera correcta.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Principal | addRecord(); | Inicializa una nueva instancia de la clase Principal junto con un nuevo usuario con sus respectivos valores. |  | Se obtiene que, al buscar el nuevo historial creado en el usuario es el mismo que el asignado. |
|  |  |  |  |  |

**Objetivo:** Probar que el método de añadir un historial al historial de búsqueda del usuario funciona correctamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Principal | deleteRecord(); | Inicializa una nueva instancia de la clase Principal junto con un nuevo usuario con sus respectivos valores. |  | Se obtiene que, al buscar entre el historial del usuario, efectivamente este ha quedado vacío. |
|  |  |  |  |  |

**Objetivo:** Probar que el método de eliminar el historial de búsqueda del usuario funciona de manera correcta.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Principal | addHotelToCustomList(); | Inicializa una nueva instancia de la clase Principal junto a un nuevo usuario y un hotel, cada uno con sus respectivos valores. |  | Se obtiene que, al buscar entre los hoteles añadidos a la lista personalizada creada por el usuario, se encuentra el hotel asignado previamente. |
|  |  |  |  |  |

**Objetivo:** Probar que el método de añadir un hotel a la lista personalizada elegida por el usuario funciona de manera correcta.

**USER:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| User | searchFavoriteHotel(); | Se inicializa un nuevo usuario con respectivos valores, junto a el se inicializa una serie de hoteles favoritos que se le asignan, un ArrayList de CustomList y una CustomList asignada al usuario y un hotelListed añadido a la CustomList. |  | Se obtiene un resultado verdadero al buscar el hotel instanciado dentro de la lista de hoteles favoritos del usuario. |
| User | searchFavoriteHotel(); | “ ” |  | Se obtiene un valor falso al buscar el hotel instanciado dentro de la lista de hoteles favoritos del usuario. |

**Objetivo:** Probar que el método de buscar un hotel favorito dentro de la lista de hoteles favoritos del usuario funciona correctamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| User | addReservedRoom | Se inicializa un nuevo usuario con respectivos valores, junto a el se inicializa una serie de hoteles favoritos que se le asignan, un ArrayList de CustomList y una CustomList asignada al usuario y un hotelListed añadido a la CustomList. |  | Se obtiene un valor verdadero al comprobar que si la habitación reservada instanciada se encuentra en la lista de habitaciones reservadas del usuario. |
|  |  |  |  |  |

**Objetivo:** Probar que el método de añadir una habitación a las reservas del usuario funciona correctamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| User | createNewCustomList(); | Se inicializa un nuevo usuario con respectivos valores, junto a el se inicializa una serie de hoteles favoritos que se le asignan, un ArrayList de CustomList y una CustomList asignada al usuario y un hotelListed añadido a la CustomList. |  | Se obtiene un valor verdadero al comprobar si la nueva lista personalizada se encuentra entre las listas del usuario. |
|  |  |  |  |  |

**Objetivo:** Probar que el método de crear una nueva lista personalizada del usuario funciona correctamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| User | addHotelToCustomList(); | Se inicializa un nuevo usuario con respectivos valores, junto a el se inicializa una serie de hoteles favoritos que se le asignan, un ArrayList de CustomList y una CustomList asignada al usuario y un hotelListed añadido a la CustomList. |  | Se obtiene un valor verdadero al comprobar si el hotel se a añadido a la lista personalizada seleccionada por el usuario. |
|  |  |  |  |  |

**Objetivo:** Probar que el método de añadir hoteles a una lista personalizada del usuario funciona correctamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| User | addRecord(); | Se inicializa un nuevo usuario con respectivos valores, junto a el se inicializa una serie de hoteles favoritos que se le asignan, un ArrayList de CustomList y una CustomList asignada al usuario y un hotelListed añadido a la CustomList. |  | Se obtiene que, al comparar el historial añadido al historial de búsqueda del usuario es el mismo que el indicado. |
|  |  |  |  |  |

**Objetivo:** Probar que el método de añadir un nuevo registro al historial de búsqueda del usuario funciona de manera correcta.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| User | deleteRecord(); | Se inicializa un nuevo usuario con respectivos valores, junto a el se inicializa una serie de hoteles favoritos que se le asignan, un ArrayList de CustomList y una CustomList asignada al usuario y un hotelListed añadido a la CustomList. |  | Se obtiene que, al buscar en el historial del usuario, este efectivamente se encuentra vacío y eliminado. |
|  |  |  |  |  |

**Objetivo:** Probar que el método de eliminar los registros del historial de búsqueda del usuario funciona de manera correcta.

**CUSTOMLIST:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| CustomList | searchHoteListed(); | Se inicializa una CustomList con unos HotelListed asignados a esa lista. |  | Se obtiene un valor verdadero al buscar el hotel dentro de la lista personalizada mediante el id. |
| CustomList | searchHotelListed(); | “ ” |  | Se obtiene un valor falso al el buscar el hotel dentro de la lista personalizada mediante el id. |

**Objetivo:** Probar que el método de buscar el hotel listado en la lista personalizada del usuario funciona de manera correcta.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| CustomList | addHotelToList(); | Se inicializa una CustomList con unos HotelListed asignados a esa lista. |  | Se obtiene un valor verdadero al comprobar que el hotel añadido previamente se encuentra en la lista personalizada. |

**Objetivo:** Probar que el método de añadir un hotel a la lista personalizada funciona correctamente.