

Formato de escenarios y casos de uso

Names::

Leidy Daniela Londoño - A00392917

Isabella Huila Cerón - A00394751

Nombre	Clase	Escenario
setupStage1	PassengersTest	Un objeto de la clase Passengers con un objeto de la clase Passenger <ul style="list-style-type: none"> - PassengersList - PassengerUno: <ul style="list-style-type: none"> - name: "Daniel" , chair: "1A" , id: "2" , miles: 0
setupStage2	PassengersTest	Un objeto de la clase Passengers con un objeto de la clase Passenger <ul style="list-style-type: none"> - PassengersList - PassengerUno: <ul style="list-style-type: none"> - name: "Luisa" , chair: "5B" , id: "11" , miles: 200
setUpStage3	PassengersTest	Un objeto de la clase Passengers con un objeto de la clase Passenger <ul style="list-style-type: none"> - PassengersList - PassengerUno: <ul style="list-style-type: none"> - name: "Isabella" , chair: "3A" , id: "4" , miles: 1200

Objetivo de la Prueba: Verificar que el método Insert de la clase HashTable agregue correctamente un nuevo pasajero, además manejando las colisiones y en caso de que haya repetido, lanzar una excepción

Clase	Método	Escenario	Valores de Entrada	Resultado
HashTable	insert	setupStage1	Passanger: name: "Johan" ,chair: "2B" , id: "6" , miles: 100	La tabla hash ahora queda con dos objetos, el que tenía que era el pasajero Daniel y este nuevo que ingresó que el id sería igual a 6.
HashTable	insert	setupStage2	name: "Andrea" ,chair: "2B" , id:: "20" , miles: 1200	La tabla hash queda con dos objetos, el que tenía que era la pasajera daniela y el nuevo que ingresó que tendría el id 20. En este caso existe una colisión, pero la función la maneja y dejaría ingresar
HashTable	insert	setupStage3	name: "Isabella" , chair: "3A" , id: "4" , miles: 1200	La tabla queda solo con un pasajero y no se agrega esta nueva entrada, en vez de esto, lanza una excepción indicando que el elemento está duplicado

Nombre	Clase	Escenario
setupStage4	PassengersTest	Un objeto de la clase Passengers con tres objetos de la clase Passenger <ul style="list-style-type: none"> - PassengersList - PassengerUno: <ul style="list-style-type: none"> - name: "Isabella" , chair: "3A" , id: "4" , miles: 1200 - name: "Paula" , chair: "4A" , id: "11" , miles: 0 - name: "Camilo" , chair: "2C" , id: "12" , miles: 10000

Objetivo de la Prueba: Verificar que el método search busque correctamente un pasajero. Teniendo en cuenta si no encuentra nada y manejando las colisiones				
Clase	Método	Escenario	Valores de Entrada	Resultado
HashTable	search	setupStage4	name: "Daniela" , chair: "3C" , id: "20" , miles: 10000 key: 20	Devuelve el pasajero buscado, en este caso, sería el pasajero con nombre Daniela
HashTable	search	setupStage4	key: 20	No devuelve nada, ya que no existe un pasajero con esta key
HashTable	search	setupStage4	name: "Juan" , chair: "5E" , id: "20" , miles: 1000 Key: 20	Existe una colisión, pero el método buscar debe manejarlo correctamente. Así que el método devuelve correctamente el valor buscado, que sería el pasajero con el nombre Juan