# Casos de Prueba

## Configuración de Escenarios

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setupStage1Cola | Cola | Tres personas esperando el ascensor para montarse en el segundo piso (Ingresaron en el orden puesto).   * Juan * Daniel * Pablo |
| setupStage1Stack | Stack | Tres personas montadas en el ascensor (Están en el siguiente orden).   * Juan * Daniel * Pablo |
| setupStage1HasMap | HasMap | Tres personas ubicadas en distintos puntos finales (Nombre, piso en el que quedaron)   * Juan 3 * Daniel 2 * Pablo 5 |

## Test de Queue

**Objetivo de la prueba­:** Verificar que el método Add y remove estén cumpliendo bien su función. Se verifica que al añadir una persona esperando el ascensor el numero aumente, y al momento de ingresar a este la primera persona que salga si es la primera persona que estaba esperando.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Queue | Add | setupStage1Cola | Name = “Carlos”  Piso = 2 | 4 personas esperando el ascensor. (Se verifica que “Carlos” sea la última persona en la cola) |
| Queue | Remove | setupStage1Cola | Se elimina a la primera persona que tiene que salir de la cola | La primera persona que se elimina de la Queue es “Juan” |

## Test de adición de Stack

**Objetivo de la prueba­:** Verificar que el funcionamiento dentro del ascensor funciona correctamente. Haciendo que la ultima persona en ingresar al ascensor sea la primera en salir de este.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Stack | Add | setupStage1Stack | Name = “Carlos” | Se verifica que la nueva stack contenga 4 personas y que “Carlos” sea la primera persona en salir |
| Stack | Remove | setupStage1Stack | Se elimina a la primera persona que tiene que salir de la cola | Se verifica que “Pablo” ya no esta en la Stack |

## Test de HashMap

**Objetivo de la prueba:** Verificar que se puedan agregar personas con su respectiva oficina al hashMap. En caso de haber colisión la persona que intento ingresar de ultima se queda sin oficina.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| HashMap | put | setupStage1HashMap | * Name = “Carlos” * Piso = 1 | Ahora el edificio x tiene 4 personas dentro de el |
| HashMap | put | setupStage1HashMap | * Name = “Carlos” * Piso = 3 | Al haber colisión Carlos queda sin oficina |