## **Datos del estudiante**

Nombre: Duvan Andres Diaz Montañez Id: 000401676 Fecha: 4 septiembre 2020

## **PreguntaS obligatorias**

1. Describir objetos o entidades y sus relaciones necesarias para cumplir con los requisitos.

R/: Se implementaron datos primitivos que almacenan los datos primordiales del pasajero temporalmente para después ser añadidos uno por uno a su respectiva lista, teniendo en cuenta el tipo de información.

1. Presente el tipo de estructuras de datos utilizadas en su solución y porqué.

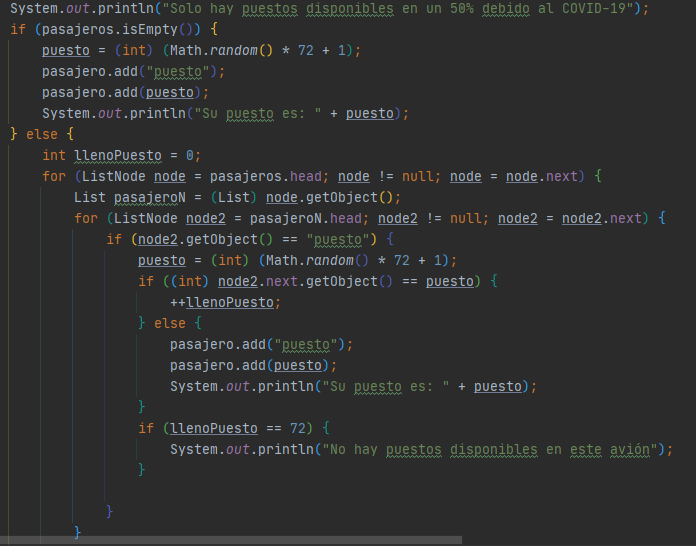
R/: En el funcionamiento del programa se usó una lista simplemente enlazada debido a que cuando lo plantee deduje que no era necesario devolverse entre nodos y demás, solo se necesitan recorridos que busquen cierta información y sea extraída.

Se crearon 4 de estas listas primordiales las cuales son:

* Pasajeros, la cual almacena una lista por cada pasajero registrado.
* Pasajero, esta lista almacena toda la información primordial y requerida del pasajero, como nombre, apellido, identificación, etc. para hacer la compra del vuelo.
* Vuelos, esta lista almacena listas de todos los vuelos dados por el aeropuerto.
* Vuelo, esta lista almacena toda la información del vuelo, como hora de salida, ciudad de salida, etc.

1. Especifique el proceso de reasignación de pasajeros al vuelo siguiente.

R/:

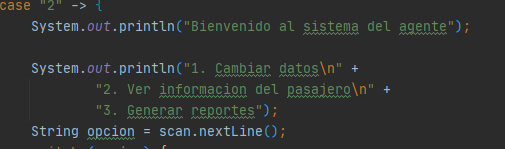


Primero se verifica que la lista de pasajeros no este vacía para no hacer validación de si los asientos del vuelo elegido por el pasajero ya están ocupados. En caso de que si haya pasajeros hace la validación. Consiste en recorrer cada pasajero y validar si el vuelo asignado al nuevo ya ha sido asignado antes con otro pasajero. Cabe destacar que los puestos se le asignan aleatoriamente al pasajero, si el pasajero se le es asignado un puesto se le dice cual es, en caso contrario, se le dice que debe buscar otro vuelo porque en el que intenta ingresar ya esta lleno en su totalidad del 50% debido a los problemas del COVID-19.

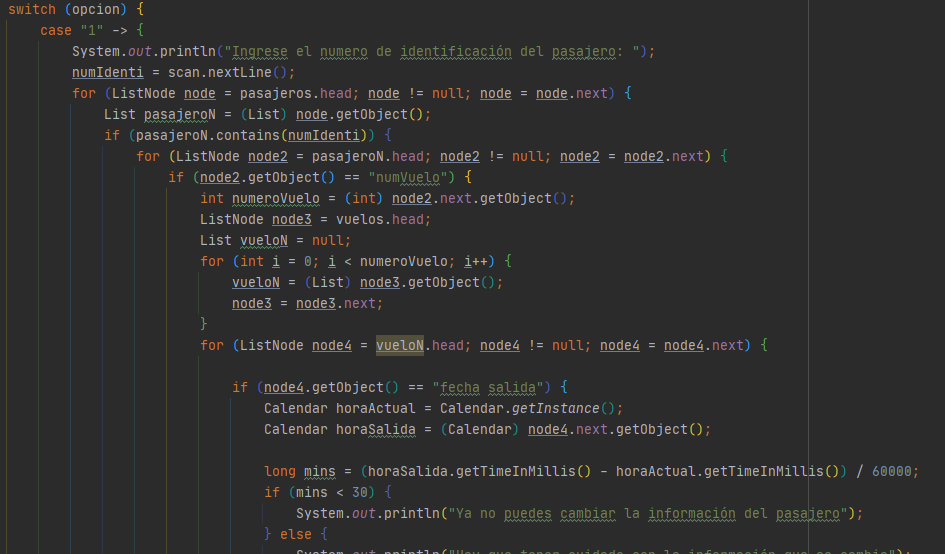
## **Describir una validación o un procedimiento, Use los fragmentos de código necesarios (uno de los PUNTOs)**

1. Documente las validaciones necesarias para la asignación asientos de los pasajeros.
2. Documente las validaciones necesarias para la asignación de carga de los pasajeros.
3. Documente las validaciones de restricción de tiempo en los procedimientos de abordaje.

R/: En primer lugar, para validar esto en el check-in, en mi caso esta validación la hace el agente de vuelos, por tanto el agente tendrá la posibilidad de hacer cambios en el sistema siempre y cuando no sea 30 minutos antes de abordar.



Si se elige la opción 1 lo mandara a una serie de validaciones las cuales son: Pedir el número de identificación del pasajero, para poder saber de que pasajero se necesita cambiar la información, luego se valida la hora del vuelo de este y la actual para poder saber si aun puede ser modificada la información



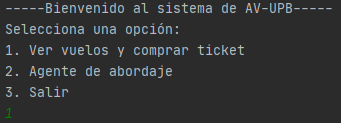
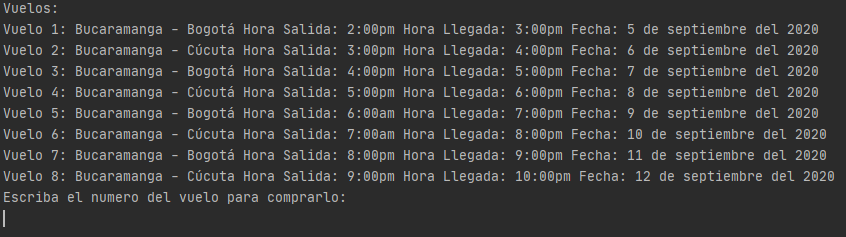
Entonces cuando ya es verificado el numero de vuelo de esta persona con su respectiva fecha y hora, se resta la fecha y hora del vuelo con la fecha y hora actual, la cual se obtiene con el método getTimeInMillis de la clase Calendar, luego se divide por 60.000 para poder obtener el tiempo en minutos debido a que esta clase de Java no tiene un get especial para los minutos. Una vez que hace la validación se valida si la hora restante es menor a 30, si es así, no le permitirá hacer ningún tipo de cambio, en caso de que si fuera mayor a 30 le permite hacer cualquier cambio, como nombre, apellido, peso de la maleta y demás.

1. Documente las validaciones de cupo de pasajeros y los horarios en la venta de pasajes.
2. Documente el procedimiento el registro de pasajeros (venta).
3. Documente el procedimiento de abordaje de pasajeros (check-in).
4. Documente el procedimiento de actualización de los datos de pasajeros.
5. Documente el procedimiento de navegación de los registros de pasajeros en su sistema.

## **Describir con evidencias (uno de los PUNTOs)**

1. Mediante una prueba unitaria demuestre el funcionamiento de la navegación de pasajeros en su sistema.

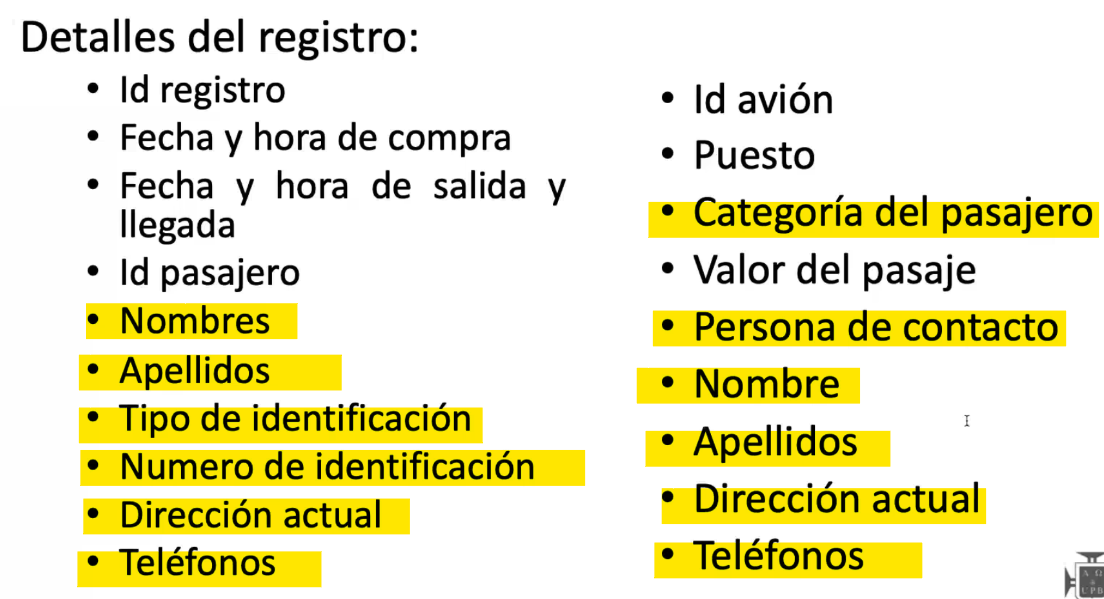
R/:

* Como menú principal el pasajero podrá ver los vuelos y poder comprar sus tickets 1
* Ahora el sistema arrojara la siguiente información:
* Ahora el pasajero elige el vuelo que desee:

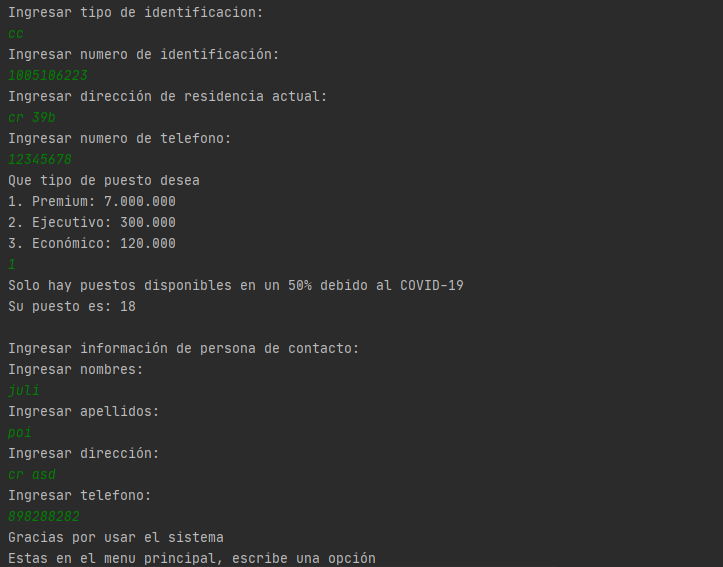


* Ahora el pasajero debe ingresar todos sus datos personales para poder comprar el vuelo: 

Debe digitar estos datos personales, los no seleccionados se llenarán automáticamente:



Aquí se puede observar algunas opciones rellenadas con su respectiva información y con la respuesta final arrojada por el sistema.



Esto es todo lo que podría realizar el pasajero, el resto de las opciones solo las podrá ver y hacer el agente.

1. Mediante una prueba unitaria demuestre el funcionamiento de un procedimiento que involucre el uso de estructuras de datos (Listas).
2. Mediante una prueba unitaria demuestre el funcionamiento de la venta de pasajes.
3. Mediante una prueba unitaria demuestre el funcionamiento de búsquedas de pasajeros.
4. Mediante una prueba unitaria demuestre el funcionamiento de búsquedas de carga que pertenece a un pasajero.

## **Especifique con diagramas o seudocódigo (uno de los PUNTOs)**

1. Especifique crear reportes de los pasajeros ordenados por nombres o apellidos o id tiquete de cada avión.
2. Especifique el proceso de reporte de los costos detallados y totales del vuelo.

costosTotales = 0

nombre = nulo

valorPasaje = 0

1. Para nodo = 0 hasta nodo no sea en pasajeros
   1. Dentro del paso 2. Para nodo2 = 0 hasta nodo2 no sea nulo en pasajero
2. Si nodo2 = “nombre” entonces

nombre = Al nodo siguiente del nodo2

1. Si nodo2 = “valorPasaje” entonces

valorPasaje = Al nodo siguiente del nodo2

costosTotales se le suma el nodo siguiente del nodo2

1. Al terminar el paso a. Imprimir ("Nombre: " + nombre + "Valor de su pasaje: " + valorPasaje)
2. Al terminar el paso 1. Imprimir ("El costo total de los vuelos es: " + costoTotales)