

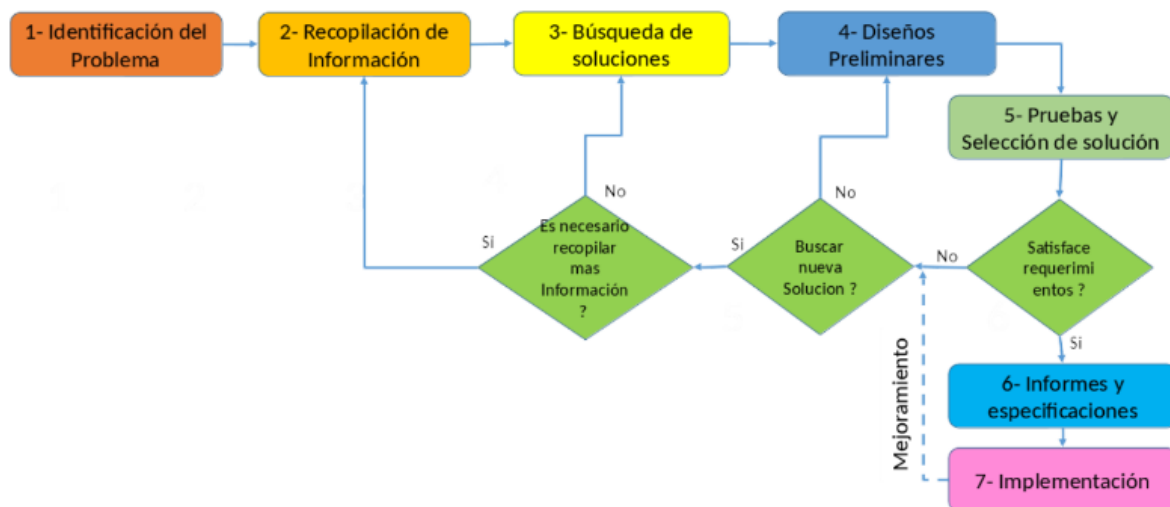
Contexto problemático

Una reconocida aerolínea necesita un sistema cuyo objetivo principal es mejorar el orden en el proceso de ingreso y salida del avión a través de la base de datos con la cual el sistema debe contar una vez finalizado.

Desarrollo de la Solución

Para resolver la situación anterior se eligió el Método de la Ingeniería para desarrollar la solución siguiendo un enfoque sistemático y acorde con la situación problemática planteada.

Con base en la descripción del Método de la Ingeniería del libro “Introduction to Engineering” de Paul Wright, se definió el siguiente diagrama de flujo, cuyos pasos seguiremos en el desarrollo de la solución.



Paso 1. Identificación del Problema

Se reconocen de manera concreta las necesidades propias de la situación problemática así como sus síntomas y condiciones bajo las cuales debe ser resuelta.

Identificación de necesidades y síntomas

- Los usuarios del sistema de la aerolínea necesitan mejorar el orden de ingreso y salida de los aviones
- El sistema debe permitir cargar la información de los pasajeros correspondientes a ese vuelo por medio de algún archivo de texto plano
- Se requiere buscar la manera más eficiente posible la información completa de un pasajero en la sala de abordaje y registrar su llegada a la sala.
- se espera premiar la puntualidad de los pasajeros ingresando al avión en el orden de llegada (aunque se sigue manteniendo el llamado por secciones del avión, empezando por las más alejadas a la puerta de ingreso)
- el sistema debe mostrar el orden en el que deben ingresar los pasajeros al miembro de la tripulación encargado
- La primera clase no solo será llamada por su orden de llegada sino también por otros datos como: millas acumuladas, atención especial requerida, tercera edad u otros datos relevantes que se puedan ocurrir.
- El sistema debe mostrar el orden en que deben salir los pasajeros, en este caso, quienes primero salen son aquellos que se encuentran en las primeras filas -
- para cada fila el orden está establecido por cercanía al pasillo u orden de llegada como última instancia.

Definición del problema

La aerolínea requiere un sistema que le permita a la persona encargada de la tripulación del vuelo visualizar el orden de ingreso y de salida de los pasajeros, dependiendo de diferentes tipos de características y restricciones tales como su orden de llegada o si es primera clase.

Paso 2. Recopilación de información

Con el objetivo de tener total claridad en los conceptos involucrados se hace una búsqueda de las definiciones de los términos más estrechamente relacionados con el problema planteado. Es importante realizar esta búsqueda en fuentes reconocidas y confiables para conocer cuáles elementos hacen parte del problema y cuáles no.

DEFINICIONES

FUENTES

<https://es.wikipedia.org>

Miembro de la tripulación

El término personal de a bordo puede incluir a pilotos, oficiales de vuelo, auxiliar de vuelo y comisarios de abordaje. En las aeronaves de carga, generalmente basta con un piloto y un oficial de vuelo.

Paso 3. Búsqueda de Soluciones Creativas

- a. Podemos hacer que los clientes se carguen directamente en la controladora usando una tabla hash como estructura de datos.
- b. Podemos hacer una clase extra a parte del controller que almacene y cargue todos los usuarios en una tabla hash.
- c. Podemos hacer que cuando los usuarios estén en “la sala de espera” a punto de abordar el avión, estén guardados todos en una cola de prioridad. De esta forma, se tendrá en cuenta a los usuarios premium para poder salir antes y se espera que el la estructura de datos vaya sacando el resto de usuarios según su orden de llegada.
- d. Podemos hacer una clase “Sala de espera” que tenga una cola sencilla y una cola de prioridad. De esta forma, los usuarios normales serán ingresados al avión según su orden de llegada siempre y cuando la cola de prioridad esté vacía. Si hay usuarios en la cola de prioridad, estos deberán ser ingresados antes al avión y según su prioridad en la cola.
- e. Para el modelo del avión podemos crear una pila que en cada posición contenga una pareja de pilas. Las pilas de la pareja, deberán tener la capacidad máxima de almacenar a 3 usuarios cada una. De esta forma, las filas de cada pareja serán una hilera de asientos en el avión y podrán ser sacados uno de cada lado.
- f. Para el modelo del avión, podemos usar solo dos pilas de forma que cada una guarde una sección del avión, aclarando que sería para aviones con dos secciones. Esto permite que el programa pueda trabajar más óptimamente y como son pilas. Si se llenan en el orden adecuado, a la hora de sacarlos del avión deberán salir también de forma correcta.
- g. El sistema puede implementar persistencia para almacenar grupos de clientes y simular vuelos con grupos de ellos en cuando el usuario operador lo indique.

- h. El sistema puede solicitar simplemente un archivo de texto plano que contiene la información de todos los pasajeros que van a un vuelo. Donde el orden en el que estén los pasajeros en el archivo de texto, será tomado como el orden de llegada.
- i. El sistema podría preguntar al usuario por el número de secciones total que tendrá el avión en el que volarán los pasajeros, el número de sillas por cada sección, cuántas de estas secciones son de primera clase y la ubicación de las secciones de primera clase (Adelante, en el medio o atrás).

Paso 4. transición de la formulación de ideas a los diseños preliminares

Tras el análisis de las ideas planteadas podemos hablar de dos modelos bastante factibles que podrían dar una solución efectiva al problema planteado.

- La primera solución es un sistema de software que tendrá su interfaz de usuario a través de la consola. Dicho sistema, al inicio de su ejecución, tendrá una base de datos con 36 pasajeros los cuales podrá ingresar en repetidas ocasiones y con diferentes órdenes a un vuelo. En este sistema, los pasajeros serán guardados directamente en la clase controladora usando una tabla hash. Al momento de iniciar un vuelo, el sistema guardará a todos los pasajeros que vayan llegando en una cola de prioridad, ubicada también en la clase controladora, que se encargará de que los usuarios entren inicialmente en el orden que su prioridad indique o si son usuarios estándar, entrarán simplemente organizados por orden de llegada. Al momento de simular el vuelo, los pasajeros estarán almacenados en un stack, desde la clase controladora, que en cada posición contendrá stacks con tres pasajeros. Donde, cada uno de los stacks internos va a representar una sección de alguno de los grupos de sillas en el avión. Usando el Stack externo, se garantiza que al momento de vaciar el avión, los pasajeros irán saliendo en orden de su sección y en orden de su cercanía al pasillo. Porque, el primero en entrar a un stack interno será el que está al lado de la ventana y el último en entrar al stack interno será el que está más cerca del pasillo.
- La segunda solución es un sistema de software que tendrá su interfaz de usuario a través de la consola. Dicho sistema, al inicio de su ejecución, tendrá una base de datos con 36 pasajeros los cuales podrá ingresar en repetidas ocasiones y con diferentes órdenes a un vuelo. En este sistema, habrá una clase encargada de la base de datos de los pasajeros externa a la clase controladora que guardará a todos los usuarios que se esperan cuando se inicia un vuelo. En este sistema de software, también habrá una clase externa a la controladora que simulará la sala de espera de la aerolínea, dicha clase tendrá una cola y una cola de prioridad. En la cola, se almacenarán solamente a los usuarios estándar, que serán ingresados al avión en orden de llegada, siempre y cuando la cola de prioridad esté vacía. En la cola de prioridad se guardarán los usuarios premium, que no solo entrarán al avión en función de su orden de llegada, sino también, en función de su prioridad. El sistema, tendrá una clase capaz de simular el avión que sabrá en qué orden deben estar sentados los pasajeros y también podrá retornar el orden en el que los pasajeros deben salir. Para esto, dicha clase usa dos stacks, uno para cada grupo de secciones del avión, en los cuales si se usan de forma adecuada (Sacando por

grupos o de a un individuo por vez de forma que se seguirá respetando el orden de salida correcto.)

Nota:

Cabe resaltar, que para ambos sistemas, el usuario podrá iniciar vuelos, con nuevos pasajeros en cada uno, hasta que seleccione la opción de cerrar el programa. Y que solamente los usuarios de primera clase tendrán prioridad a la hora de entrar al sistema. La prioridad de dichos pasajeros será calculada en función de las millas de cada usuario y de atributos booleanos como si el usuario tiene atención especial requerida o si es de la tercera edad.

Paso 5. Evaluación Y selección de la mejor solución

Criterios

- Criterio A. Exactitud. La solución responde de forma correcta a todos los eventos a los que puede ser sometido.
[2] La solución responde de forma precisa (Se prefiere una solución precisa)
[1] La solución responde de forma imprecisa

- Criterio B. Espacio en memoria. La solución debe usar la menor cantidad posible de espacio en memoria.
[3] La solución usa la menor cantidad posible de atributos y de estructuras (Se prefiere que la solución use la menor cantidad de estructuras y atributos posibles)

[2] La solución usa más atributos y estructuras que los mínimamente necesarios para dar solución a los eventos planteados.

[3] La solución usa muchos más atributos y estructuras de los estrictamente necesarios.

- Criterio C. Facilidad en implementación algorítmica

[3] Su implementación se torna intuitiva y se puede emular de “forma visual” fácilmente. (Se prefiere que su implementación algorítmica sea intuitiva para que el equipo de desarrollo lleve el proyecto a calidad de forma mas rapida)

[2] Su implementación se torna manejable pero no es fácil de emular de “forma visual”.

[1] Su implementación se torna muy compleja de implementar y es muy difícil de emular de “forma visual”.

- Criterio D. Emulación correcta del modelo empresarial de la aerolínea.

[2] El sistema simula correctamente el modelo empresarial de la aerolínea.
(Es preferible que sea de esta forma para garantizar el funcionamiento correcto del sistema se software)

[1] El sistema simula no simula de forma adecuada el modelo empresarial de la aerolínea.

Evaluación

	Criterio A	Criterio B	Criterio C	Criterio D	Puntos totales
Primera solución	2	2	2	1	7
Segunda solución	2	3	3	2	10

Selección

Siguiendo los resultados obtenidos en el análisis anterior, se debe llevar a producción la segunda solución. Puesto que, tiene más puntos y esto significa que atiende mejor a los criterios de calidad planteados.

Paso 6. Preparación de informes y especificaciones

Hemos decidido plantear en este paso las tablas de análisis de requerimientos para cada una de las funcionalidades que debe tener el software y el diseño de la solución será presentado como un diagrama de clases. Para ver estos elementos, ver los archivos **ClassDiagram** y **RequirementAnalysis** en la carpeta doc del proyecto donde también se encuentra este documento.

Paso 7. Implementación del diseño

Lista de tareas a implementar:

- Construcción de todas las estructuras de datos necesarias para solucionar los eventos planteados.
- Inserción de todas las clases planteadas en el modelo del sistema.
- Implementación de todos los menues en la interfaz de usuario
- Implementación de las funcionalidades del modelo
- Conexión del modelo del sistema y la interfaz de usuario mediante la controladora