# **PontiAir - Portal de Reservas**

La ***PontiAir*** es una reconocida aerolínea establecida en todo el país, esta ha expandido constantemente sus servicios y ahora requiere ofrecer un exclusivo portal de reservas de tiquetes aéreos. ***PontiAir*** tiene una trayectoria destacada en la industria de la aviación, con la cual se ha consolidado como una de las principales aerolíneas a nivel mundial con una flota moderna, rutas globales y con un servicio de alta calidad, ***PontiAir*** se ha ganado la confianza de millones de viajeros. Dado su fuerte popularidad le ha pedido a un grupo de ingenieros a establecer un nuevo portal de reserva para que sus clientes puedan acceder a su servicio de venta de tiquetes aéreos.

El objetivo del presente proyecto es construir un programa que simule un **portal** **de reservas de tiquetes aéreos**, para esto se deberá consolidar un programa en **C++** el cual permita almacenar los registros de las reservas de tiquetes aéreos y permitir búsquedas por diferentes parámetros.

## Características principales:

De las **Reservas** de tiquetes se debe almacenar los siguientes datos:



1. Nombre del Cliente.
2. Número de Documento.
3. Sexo del Cliente (“M”, “F”, “N”). (N: representa no aplica)
4. Cantidad de Pasajeros para la reserva.
5. Numero de reserva. (en 000 inicialmente, si no se ha reservado)
6. Seguro de vuelo. (SI – NO)

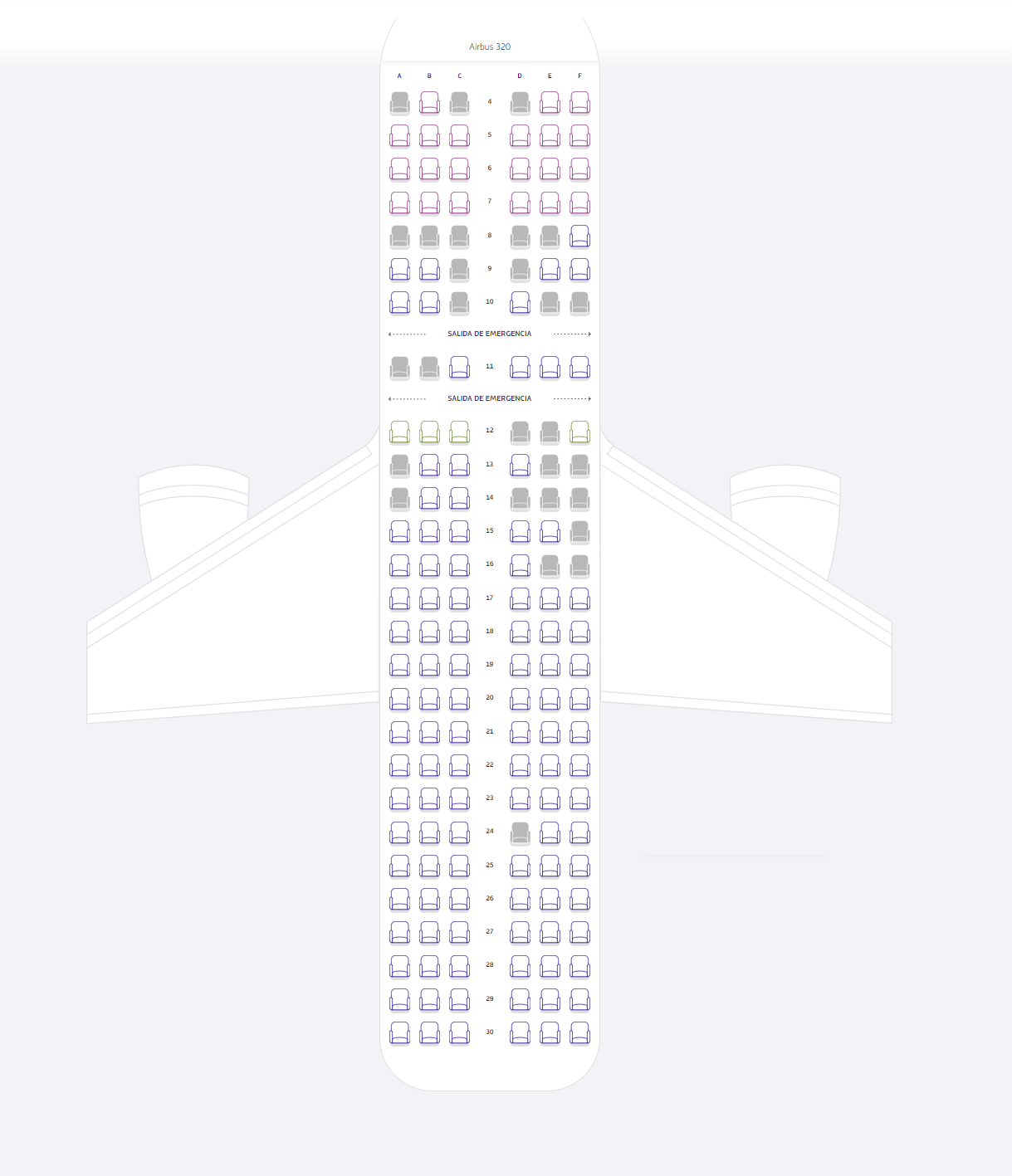
Cada uno de los **Asientos** en el ***portal*** **de la aerolínea** debe tener los siguientes datos:

1. Número del vuelo.
2. Destino.
3. Fila.
4. Numero de asiento.
5. Estado. (Libre – Ocupado)
6. Categoría.
7. Numero de reserva. (en 000 inicialmente, si no se ha reservado)

En **Estado** se debe indicar si el asiento se encuentra **disponible** o si está en estado **ocupado**. (Los asientos por defecto deben estar en **estado disponible**)

Para cada avión se mantienen unas **categorías** de asientos. A continuación, se puede ver la disposición de asientos de 6 filas (A-F) y de 27 filas (4 - 30).

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente 

La siguiente tabla muestra los precios en miles de pesos por **categoría**, según la categoría del asiento

|  |  |  |  |
| --- | --- | --- | --- |
| **Bussines (4-10)** | **Emergencia (11)** | **Estándar (12 - 20)** | **Económica (21-30)** |
| 500 | 250 | 200 | 100 |
|  |  |  |  |

Los pasajeros de una reserva solicitada deberán ubicarse en **sillas continuas por fila**. Antes de asignar una **reserva** se deberá conocer si hay asientos disponibles para la cantidad de pasajeros solicitadas por el cliente. Si no hay asientos disponibles para esta cantidad de personas en la fila, se deberá informarle al cliente que no se puede completar la reserva.

Si se elige la fila entre las salidas de emergencia, se deberá notificar al cliente si acepta los términos y condiciones de esta ubicación.

La información se representará en una matriz, en donde las filas representarán con un número (4 - 30) y las columnas se representarán con una letra. (A - F)

Si el cliente solicita un seguro de vuelo, deberá cobrarse 100 miles de pesos adicionales.

## Desarrollo del proyecto

El portal de reservas está representado con **registros** con la información del cliente**,** el programa esperado debe generar un menú principal que debe admitir las siguientes opciones:

## Menú principal

1. Ingresar reserva. (Se deberá cambiar el estado de los asientos al generarse).
2. Finalizar reserva. (Eliminar un registro, se debe liberar los asientos ocupados)
3. Consultar los registros por fila. (Ver todos los asientos por fila)
4. Buscar reserva. (Por nombre o por número de documento del cliente).
5. Mostrar todo el portal. (todas las reservas que han sido vendidas por cada categoría).
6. Consultar asientos ocupados. (Debe mostrar todos los asientos que actualmente no están disponibles)
7. Calcular el total de reservas vendidas por categoría.
8. Calcular el total de ganancias por reservas vendidas de todo el avión.
9. Salir.

## Observaciones

* Para efectos prácticos, las corresponden a solo un avión con un único trayecto.
* Para términos prácticos, se deberá tener almacenado inicialmente asientos reservados, como **mínimo** en **5 filas y 3 columnas**, según las instrucciones definidas anteriormente.
* Se conoce que máximo hay un total de **6 asientos por fila**.
* Un mismo cliente puede tener varias reservas compradas. (Puede tener varias reservas, con asientos reservados para toda su familia y/o amigos)
* Si un cliente quiere adquirir nuevas reservas (caso de más de 6 sillas), lo hará de nuevo en la ejecución de ingresar reserva. (No se realizará reservas en diferentes filas en la misma transacción)
* El numero de la reserva se le asigna un numero de 3 cifras automáticamente, el cual deberá ir aumentando a medida que se generen nuevas reservas. (100, 101, 102,103, etc.)
* Si al realizar una reserva, la cantidad de personas para la reserva supera al número de pasajeros máximo por fila, el sistema alertará que se supera el límite y no le dejará realizar la reserva.
* No se debe usar la librería <**vector**> ni <**algorithm**>.
* Si un cliente adquirió un seguro de vuelo se debe cobrar 100 miles de pesos más.

## Entregable

1. Diseño del proyecto **(Primera entrega: 5%):** hasta el **11 de noviembre 11:59pm** vía actividad habilitada para tal fin en **Brightspace** – (después de las 11:59pm se cerrará y después no se aceptará ninguna entrega, no se aceptará ninguna excusa), un documento que contenga:

* Descripción de las estructuras de datos que van a utilizar para el desarrollo del proyecto:
  + Descripción y uso detallado de la estructura.
  + Nombre descriptivo del atributo.
  + Tipo de dato del atributo.
  + Descripción de rangos o características que deban ser validadas por el programa con respecto al atributo (i.e. obligatoriedad, rangos de números, formatos de cadenas, etc.)
* La documentación de cada posible función que piensan utilizar con:
  + Nombre de la función.
  + Sinopsis: Descripción textual de lo que hace la función.
  + Argumentos: descripción completa de todos y cada uno de los argumentos (parámetros) de la función (tipo, nombre, descripción detallada y uso).
  + Descripción: Descripción textual del algoritmo realizado por la función, separado en pasos claros y detallados.
  + Retorno: Descripción de lo que retorna la función (tipo de dato y significado).
  + Precondición: Qué condiciones deben cumplir los argumentos (parámetros) para que la función se ejecute exitosamente.
  + Postcondición: como quedan los parámetros pasados por referencia y el retorno en contexto del algoritmo descrito.
* Diseños de posibles pantallas del programa describiendo como sería la interacción entre el usuario y el programa.

1. **Entrega Final 14 de noviembre** es necesario subir el **archivo.cpp** y **archivo.h** en la actividad habilitada para tal fin en **Brightspace** – (después de las 11:59pm se cerrará y después no se aceptará ninguna entrega, no se aceptará ninguna excusa)
2. **Sustentación del Proyecto (10%)** **(semana 17 - Salón de programación)** (Presencial),deben estar **todos los integrantes para sustentar.**

## Condiciones de Entrega

* El proyecto se podrá realizar en grupos de **máximo 3 personas**.
* El código **se verificará** para comprobar que no haya plagio. Si se encuentra alguna inconsistencia, quedará con 0 la definitiva del proyecto.
* No se debe usar Chat gpt, ni compartir código con otros grupos. (Se comprobará la originalidad del código)
* La documentación del proyecto hace referencia a las funciones y estructuras que se manejaron en el proyecto y su propósito. Se debe describir en forma de manual técnico, cómo el sistema cumple con los requerimientos planteados anteriormente. **En ningún caso** el código fuente se considera documentación y **NO** se debe incluir en la misma.
* La sustentación del proyecto que puede ser (Modificación al proyecto y/o Adición de funcionalidad) de manera **individual o grupal**.
* Es necesario **sustentar correctamente** el proyecto, si no se cumple con la sustentación se reconocerá el **25%** de la nota final del proyecto.