

## **DISEÑO DE SOLUCIONES COMPUTACIONALES BASADAS EN PROGRAMACIÓN ORIENTADA A OBJETOS**

### **Objetivos**

- Diseñar soluciones computacionales empleando programación orientada a objetos a partir de la descripción de requerimientos
- Identificar las clases, atributos y métodos requeridos en una solución computacional y presentar estos elementos a través de diagramas de clase
- Aplicar herencia, polimorfismo, clases abstractas para la solución de problemas
- Emplear archivos y excepciones para guardar y cargar información
- Hacer uso de interfaces gráficas para el diseño de soluciones
- Implementar la solución en Python y diseñar pruebas que demuestren la solución del problema

### **Recursos requeridos**

- PC con Visual Studio Code instalado

### **Actividades a desarrollar por los estudiantes:**

#### **Parte 1. Conformación de equipo de trabajo**

El proyecto se podrá presentar en grupos de hasta 3 estudiantes, se recomienda desarrollar esta actividad en grupo para mejorar el uso del tiempo.

#### **Parte 2. Diseño de la solución computacional**

- Realice la lectura detallada de los requerimientos del sistema.
- Identifique las clases, métodos y atributos que permitirán solucionar el problema. En este proyecto se espera que se haga uso de herencia y de una clase abstracta.
- Realice los diagramas de clase, pueden utilizar software como Lucidchart o Draw.io para diagramar.
- Exporte los diagramas en un archivo pdf el cual deberá enviar a través del Moodle y presentar el día de la sustentación del proyecto.
- Diseñe una interfaz gráfica que permita la interacción con el programa. En el caso de Python se debe hacer uso de Tkinter.

#### **Parte 3. Implementación de la solución computacional**

Realice la implementación de su diseño usando Python u otro lenguaje de programación diferente a JAVA que use el paradigma de orientado a objetos. La implementación debe ser coherente con el diagrama de clases presentado. Para este proyecto se requiere el uso de interfaces gráficas; ¡¡¡puede usar toda su imaginación para crear una interfaz llamativa y funcional!!!.

*Nota: solo se recibe el proyecto 1 en Python u otro lenguaje de programación diferente a JAVA que desee explorar.*

#### **Parte 4. Diseño de pruebas de funcionamiento**

Determine que pruebas se requieren para demostrar que su implementación cumple con los requisitos del problema, es decir que responda a los requerimientos descritos en el documento adjunto. Estas pruebas deberán ser presentadas durante la sustentación.

### Parte 5. Instrucciones de entrega y sustentación

La entrega se realizará en dos etapas:

- El diagrama de clases y diagramación de la interfaz en formato PDF debe subirse a la plataforma a más tardar el **17 de julio**
- La implementación en Python debe ser presentada y sustentada el **24 de julio** en el horario de clase

Para la sustentación:

- Prepare con su equipo de trabajos una sustentación de máximo 7 minutos, en los cuales deberá presentar el diagrama de clases, mostrar el código de las clases principales, y además realizar las pruebas que considere necesario para mostrar el funcionamiento. La sustentación no puede exceder los 7 minutos.

Para tener en cuenta:

- El trabajo se puede (y se recomienda) presentar en grupos de hasta 3 estudiantes
- Solo se otorga la nota a los integrantes del grupo que se encuentren el día de la sustentación
- Este es un evento evaluativo, sino presenta el proyecto el 24 de julio debe presentar una excusa válida para reprogramar la entrega (teniendo en cuenta que el semestre finaliza el sábado 26 de julio).
- El proyecto incluye la presentación de los diagramas de clases y el diseño de la interfaz gráfica que debe ser enviado en pdf por el Moodle
- No se recibe código por correo electrónico o a través del Moodle, el código debe ser presentado y sustentado el 24 de julio
- La sustentación debe durar máximo 7 minutos, se controlará el tiempo. Grupos o estudiantes que se extiendan de este tiempo serán interrumpidos, y se asignará la nota en base a lo que mostraron.

## RUBRICA DE EVALUACIÓN

En la siguiente tabla se presenta la rúbrica con la cual se realizará la calificación del proyecto. Leerla con atención para tener en cuenta los diferentes aspectos en la elaboración de los diagramas, implementación y sustentación del proyecto.

Rubrica	Descripción	Ponderación	
<b>Diseño</b>	Presentación de diagramas de clase	5%	10%
	Los diagramas de clase incluyen la información suficiente para identificar las clases, atributos y métodos más importantes del sistema	5%	
<b>Implementación</b>	El programa corre sin ningún error de compilación	5%	20%
	Se evidencia que las clases implementadas corresponden con los diagramas de clase presentados	5%	
	Se presenta una implementación empleando interfaz grafica	10%	
<b>Archivos</b>	Realiza la lectura y escritura de archivos	5%	10%
	Demuestra la persistencia de los datos empleando archivos de texto	5%	
<b>Interfaz gráfica</b>	La interfaz grafica que se presenta es llamativa y funcional, coherente con las especificaciones del problema	10%	20%
	Es fácil y claro utilizar la interfaz	10%	
<b>Pruebas</b>	Se presentan pruebas que evidencian la funcionalidad del sistema	10%	20%
	Las pruebas evidencias que el programa es robusto y permite trabajar con posibles entradas erróneas	10%	
<b>Sustentación</b>	Presenta(n) con claridad el diseño, la implementación y las pruebas del sistema	10%	20%
	Hacen uso adecuado del tiempo	10%	

## FECHAS IMPORTANTES

Actividad	Fechas
Envío de diagramas de clase a través del Moodle	17 de julio
Sustentación	24 de julio

## ESPECIFICACIONES DEL SISTEMA – SOFTWARE PARA LA RESERVA DE VUELOS

Una aerolínea nos ha contratado para crear un software que permita realizar las reservas de sus vuelos. La aerolínea nos proporciona información sobre los vuelos diarios en un archivo de texto, que incluye el código del vuelo, la ciudad de origen, la ciudad de destino, el horario, y el número de sillas preferencial y económica para cada vuelo. Esta aerolínea maneja precios estándar, por cada silla preferencial se debe pagar \$850.000 y por cada silla de un vuelo económico \$235.000. El sistema debe permitir:

1. Los usuarios deben registrarse usando su nombre, correo electrónico, número de documento y creando una contraseña. El sistema permite iniciar la sesión empleando el número de documento y contraseña correcta. Así mismo, debe permitir cambiar la contraseña.
2. Se debe tener un administrador del sistema, el cual puede modificar la información de los vuelos, tales como agregar nuevos vuelos (ingresado el código del vuelo, la ciudad de origen, destino, horario, y número de sillas preferencial y económica). El administrador podrá consultar cuáles de los vuelos cuenta con sillas vendidas y los datos de los pasajeros.
3. Para los usuarios, una vez ingresaron debe permitir buscar vuelos según su origen o destino, el sistema debe indicar los horarios disponibles y el número de sillas disponibles.
4. El usuario, una vez ingrese en la sesión podrá reservar vuelos, ingresando los datos de los pasajeros (un usuario registrado podrá comprar hasta 3 sillas en el mismo vuelo ingresando la información de los otros pasajeros). El sistema deberá calcular el precio total, mostrar el resumen de la compra, y una vez confirmada, se genera un número de reserva.
5. Un usuario podrá gestionar todas las reservas que ha realizado, permitiendo cancelar o modificar una reserva, así como realizar check in.
6. Cuando el usuario realiza el check in, se muestra toda la información del vuelo y el usuario acumula millas, por cada reserva acumula 500 millas. Al sumar 2000 millas podrá reservar un vuelo en silla preferencial pagando la silla económica.
7. Al realizar el check in el usuario debe seleccionar el tipo y cantidad de maletas: equipaje de mano (incluido en sillas preferenciales y económicas), maleta de cabina (maleta de mano de máximo 10 kg, incluido para sillas preferenciales y con un costo de \$40.000 para sillas económicas), y maleta de bodega (1 maleta incluida para sillas preferenciales – las maletas adicionales para sillas preferenciales y sillas económicas tienen un costo proporcional al peso y volumen).
8. El sistema debe permitir almacenar la información en archivos txt de tal forma que una vez se cargue nuevamente el sistema, los usuarios registrados y reservas realizadas se mantengan.