

## *Proyecto 2 - Patrones de Diseño*

*Yuen Law*

*Semestre I, 2021*



Figura 1: Foto por Sara Haj-Hassan de freeimages.com

### *Objetivos*

1. Aplicar conceptos de ingeniería de software.
2. Aplicar el patrón de diseño MVC.
3. Aprender sobre el uso de sockets en Java para comunicación de aplicaciones.
4. Aprender sobre el uso de hilos en Java para la ejecución de tareas concurrentes.

### *Descripción*

El proyecto se trata de diseñar y programar una serie de aplicaciones que simule un sistema de control aéreo de un aeropuerto. Específicamente, el usuario debe controlar la llegada y aterrizaje de aviones que van llegando. Cada aplicación controla o muestra un aspecto diferente del sistema.

Estas son:

- **Vuelos:** Es la aplicación que simula la llegada de los aviones al aeropuerto. Se encarga de notificar al controlador cuando un avión se aproxima y requiere aterrizar. Debe generar diferentes tipos de avión: carga, pasajeros, privados. Y cada uno de éstos tendrá

requerimientos diferentes de pista por su tamaño. Los aviones además pueden venir a tiempo o con retraso.

- **Ventana controlador:** Esta es la ventana principal para el usuario. Conforme llegan los vuelos, el usuario debe asignarle a cada avión una pista de aterrizaje adecuada que esté libre; y una vez que haya aterrizado, una puerta de embarque.
- **Controlador:** Es la aplicación que simula el movimiento de los aviones dentro del aeropuerto. Una vez que se le ha asignado una pista de aterrizaje, esta aplicación inicia un contador de aterrizaje para ese avión y le notifica a la ventana del controlador para que se le asigne una puerta. Después, debe iniciar otro contador que simule el tiempo de traslado de la pista a la puerta otro más para simular el tiempo de desembarque. Finalmente cuando el avión desocupa la puerta y sale de la simulación.
- **Ventana de información:** Es la ventana que usualmente se ve en la sala de espera. Debe mostrar los vuelos esperados y el estado actual: atrasado/a tiempo, aterrizando, taxi, llegado (# la puerta).

### *Comunicación*

La comunicación entre las 4 aplicaciones es por medio de sockets.

*Vuelos-Controlador* La aplicación de vuelos debe enviarle la información de cada avión cuando éste se aproxima al aeropuerto.

*Controlador-Ventana* El controlador debe enviarle la información necesaria al usuario para que pueda tomar decisiones acertadas: tipo de avión, pistas y puertas libres, etc.

*Controlador-Información* La ventana de información debe recibir información de los aviones y sus estados para mostrarlos en la sala de espera.

### *Hilos*

Como en esta ocasión, se trata de una simulación en tiempo real, el uso de hilos es necesario. Se deben implementar hilos para la comunicación y para la simulación en la aplicación Controlador.

### *Requerimientos*

- Se debe programar usando el paradigma orientado a objetos.

- Para el control de desarrollo se debe usar la pizarra de proyecto en Github y el sistema kanban para registrar el progreso de las actividades realizadas.

### *Entregables*

1. Documentación PDF, en el formato IEEE para conferencias (máximo 8 páginas) con las siguientes secciones: Introducción (planteamiento del problema), Diseño (Diagrama de clases, métodos (descripción detallada de la solución), Conclusiones y Referencias.
2. Código en un repositorio de Git. El último commit realizado debe ser antes de la fecha y hora límite de entrega.

### *Aspectos administrativos*

1. Fecha de entrega: martes 8 de junio, hora: 12 media noche
2. El pdf debe ser enviado por medio de Teams
3. El proyecto debe ser desarrollado en Java.

### *Evaluación*

Rubro	Valor
Documento PDF	(20)
- Introducción	2.5
- Diseño	10
- Conclusión	2.5
- Referencias	5
Código	(60)
- Conexión y mensajería	15
- Controlador	15
- Vuelos	10
- Ventana de información	10
- Ventana de controlador	10
Avance	(10)
Workflow	(10)
- Uso de git	5
- uso del board de github	5
Total	100