

Instituto tecnológico de Costa Rica

Escuela de Ingeniería en Computadores

Documento de acreditación

Algoritmos y estructuras de datos 1

Grupo 02

Proyecto #3 "AirWar"

Profesor: Leonardo Araya Martínez

Estudiantes:

Fiorela Sofía González Rubí

Steven Josué Pérez Aguilar

II semestre, 2024

Tabla de contenido

Introducción.....	2
Diseño	3
Requerimientos (historias de usuario).....	3
5 problemas con más de una posible solución (2 alternativas para cada solución, ventajas y desventajas de cada alternativa, justificación de alternativa seleccionada).....	6
Diagrama de clases UML.....	9
Diagrama de arquitectura	10
Checklist de historias de usuario implementadas pendientes	10

Introducción

Este proyecto tiene como objetivo principal implementar una solución a un problema haciendo uso de grafos y algoritmos de ordenamiento como agente secundario, en el desarrollo del juego AirWar. En este juego, el jugador podrá visualizar rutas entre aeropuertos y portaaviones (representado con un grafo), por las cuales transitarán aviones, el objetivo del juego será derribar la mayor cantidad de aviones posibles. La solución propuesta se centra en la implementación de grafos para representar las rutas y destinos, algoritmos creados para su uso con grafos para encontrar las mejores rutas posibles y algoritmos de ordenamiento para gestionar características adicionales del juego.

En este documento se busca ilustrar mediante la documentación del desarrollo de este proyecto la adquisición del atributo esperado, diseño en su etapa inicial, demostrando las capacidades desde identificar los requerimientos dados para el proyecto, identificar y analizar alternativas de solución para conseguir la más eficiente posible y por último, diseñar de forma creativa la solución seleccionada asegurando el cumplimiento de las especificaciones solicitadas para cumplir con el proyecto.

Diseño

Requerimientos (historias de usuario)

Historia de Usuario 001	
Nombre	Destrucción de aviones
Descripción	Yo, como jugador, quiero destruir la mayor cantidad de aviones posible en un periodo de tiempo definido, para lograr una alta puntuación y mejorar mi habilidad en el juego.
Criterios de aceptación	<p>Se puede establecer un límite de tiempo al iniciar el juego.</p> <p>Se cuenta el número de aviones destruidos al finalizar el periodo.</p> <p>Se muestra la puntuación final al jugador al terminar el tiempo.</p>

Historia de Usuario 002	
Nombre	Generación de aeropuertos, portaaviones y rutas
Descripción	Yo, como usuario, quiero que se generen aleatoriamente aeropuertos, portaaviones y rutas para hacer el mapa de juego dinámico.
Criterios de aceptación	<p>Los aeropuertos, portaaviones y rutas se generan de forma aleatoria al iniciar el juego.</p> <p>El mapa se modela utilizando un grafo representado por listas de adyacencia.</p> <p>Las rutas generadas conectan adecuadamente los puntos del grafo.</p>

Historia de Usuario 003	
Nombre	Batería antiaérea
Descripción	Yo, como jugador, quiero controlar una batería antiaérea que se mueva constantemente de izquierda a derecha, para disparar y derribar a los aviones.
Criterios de aceptación	<p>La batería se mueve de forma constante entre la izquierda y derecha de la pantalla.</p> <p>Al hacer clic, se dispara una bala con trayectoria recta.</p>

Historia de Usuario 004	
Nombre	Generación de rutas con pesos
Descripción	Yo, como usuario, quiero que las rutas entre aeropuertos y portaaviones tengan distintos pesos, basados en la distancia y el tipo de destino, para que los aviones optimicen sus rutas y haya variabilidad en los costos.
Criterios de aceptación	<p>Cada ruta generada tiene un peso determinado por la distancia.</p> <p>Las rutas a portaaviones tienen un costo adicional.</p> <p>Las rutas interoceánicas tienen un peso mayor que las rutas continentales.</p>

Historia de Usuario 005	
Nombre	Selección de destinos y recarga de aviones
Descripción	Yo, como usuario, quiero que los aviones elijan destinos y calculen la mejor ruta considerando los pesos de las rutas, para que puedan realizar vuelos eficientes y optimizados.
Criterios de aceptación	<p>Los aviones eligen su destino de manera aleatoria al despegar.</p> <p>La mejor ruta se calcula usando los pesos de las rutas.</p> <p>Al aterrizar, los aviones esperan un tiempo aleatorio y recargan combustible antes de retomar vuelo.</p>

Historia de Usuario 006	
Nombre	Racionamiento de combustible en aeropuertos
Descripción	Yo, como usuario, quiero que los aeropuertos racionen el combustible a los aviones, para que se pueda gestionar eficientemente el combustible.
Criterios de aceptación	<p>Los aeropuertos racionan el combustible de acuerdo a una regla definida.</p> <p>Si un avión se queda sin combustible antes de aterrizar, se cae.</p>

Historia de Usuario 007	
Nombre	Construcción de nuevos aviones
Descripción	Yo, como usuario, quiero que los aeropuertos construyan nuevos aviones, limitados por la capacidad de sus hangares, para mantener un flujo continuo de aviones en el juego.
Criterios de aceptación	<p>Cada aeropuerto construye nuevos aviones de forma periódica.</p> <p>La cantidad de aviones construidos no excede la capacidad del hangar.</p> <p>Cada avión recibe un ID único generado aleatoriamente mediante GUID.</p>

Historia de Usuario 008	
Nombre	Implementación de módulos AI en aviones
Descripción	Yo, como usuario, quiero que cada avión tenga módulos de inteligencia artificial que gestionen su funcionamiento, para simular el comportamiento autónomo y mejorar la jugabilidad.
Criterios de aceptación	<p>Cada avión tiene cuatro módulos AI: Pilot, Copilot, Maintenance y Space Awareness.</p> <p>Cada módulo tiene un ID único de tres letras, un rol específico y un contador de horas de vuelo.</p> <p>Se puede consultar la lista de aviones derribados, ordenados por ID usando Merge Sort.</p>

Historia de Usuario 009	
Nombre	Ordenación de tripulación de aviones derribados
Descripción	Yo, como usuario, quiero poder ver la tripulación de los aviones derribados y ordenarlos por ID, rol o horas de vuelo, para analizar el desempeño y organización de cada tripulación.
Criterios de aceptación	<p>Es posible obtener la lista de módulos de AI para cada avión derribado.</p> <p>Los módulos se pueden ordenar por ID, rol o horas de vuelo usando Selection Sort.</p> <p>La información de la tripulación ordenada se muestra en pantalla.</p>

Historia de Usuario 010	
Nombre	Visualización de datos relevantes del juego
Descripción	Yo, como usuario, quiero ver en pantalla los datos importantes del juego, como rutas de los aviones, pesos de las rutas y atributos de los aviones, para entender mejor la dinámica del juego y tomar decisiones.
Criterios de aceptación	<p>Se muestran en pantalla las rutas calculadas por los aviones.</p> <p>Se indican los pesos de cada ruta y atributos relevantes de los aviones.</p> <p>La información se actualiza en tiempo real conforme avanza el juego.</p>

5 problemas con más de una posible solución (2 alternativas para cada solución, ventajas y desventajas de cada alternativa, justificación de alternativa seleccionada)

Problema 1: Generación de rutas con pesos.

2 alternativas para la solución:

Alternativa 1: Implementar pesos considerando distancia, destino y tipo de ruta.

Alternativa 2: Asignar pesos estáticos basados únicamente en la distancia entre puntos.

Ventajas y desventajas de cada alternativa:

Alternativa 1:

Ventajas: Considera todos los factores relevantes y hace el juego más dinámico (distancia, tipo de ruta y destino).

Desventajas: Mayor complejidad en la implementación.

Alternativa 2:

Ventajas: Más fácil de implementar y rápida de ejecutar.

Desventajas: No refleja completamente los requerimientos, ya que ignora la relevancia del destino y tipo de ruta.

Justificación de alternativa seleccionada:

Se selecciona la Alternativa 1 porque cumple con los requerimientos al considerar todos los factores relevantes y mejora la calidad del sistema de rutas dinámicas, mejorando la calidad general del producto final.

Problema 2: Racionamiento de combustible en aeropuertos.

2 alternativas para la solución:

Alternativa 1: Distribuir el combustible proporcionalmente según la cantidad total disponible.

Alternativa 2: Asignar combustible de forma aleatoria a los aviones que lo soliciten.

Ventajas y desventajas de cada alternativa:

Alternativa 1:

Ventajas: Justa y sigue el criterio solicitado según los requerimientos.

Desventajas: Podría no ser suficiente para vuelos largos o en casos de escasez en los aeropuertos.

Alternativa 2:

Ventajas: Simplicidad y rapidez en la asignación.

Desventajas: Puede resultar en una distribución ineficiente o injusta, dejando vuelos sin combustible.

Justificación de alternativa seleccionada:

Se selecciona la Alternativa 1 porque asegura una distribución más justa del combustible y cumple con los requerimientos, maximizando la equidad en las asignaciones de combustible.

Problema 3: Ordenar aviones derribados.

2 alternativas para la solución:

Alternativa 1: Ordenar la lista de aviones derribados por ID usando Merge Sort.

Alternativa 2: Ordenar la lista de aviones derribados usando un algoritmo más simple como Bubble Sort.

Ventajas y desventajas de cada alternativa:

Alternativa 1:

Ventajas: Merge Sort es más eficiente y adecuado para listas grandes.

Desventajas: Más difícil de implementar en comparación con otros algoritmos más básicos.

Alternativa 2:

Ventajas: Bubble Sort es más fácil de implementar.

Desventajas: Muy ineficiente para listas grandes y no garantiza un rendimiento adecuado para el juego.

Justificación de alternativa seleccionada:

Se selecciona la Alternativa 1 porque Merge Sort es más eficiente para grandes cantidades de datos, lo cual empieza a ser importante conforme las cantidades de aviones derribados incrementan y es el algoritmo solicitado en los requerimientos.

Problema 4: Generación de IDs únicos para aviones y módulos AI**2 alternativas para la solución:**

Alternativa 1: Generar IDs usando GUIDs.

Alternativa 2: Crear un sistema de IDs secuenciales únicos basado en un contador.

Ventajas y desventajas de cada alternativa:

Alternativa 1:

Ventajas: Garantiza existencia única del IDs.

Desventajas: Los GUIDs son largos y menos legibles para los usuarios.

Alternativa 2:

Ventajas: IDs más simples y fáciles de interpretar.

Desventajas: Riesgo de duplicación si no se gestiona correctamente.

Justificación de alternativa seleccionada:

Se selecciona la Alternativa 1 porque los GUIDs garantizan existencia única, cumple con lo solicitado en los requerimientos y elimina posibles problemas de duplicación.

Problema 5: Visualización de datos relevantes en pantalla**2 alternativas para la solución:**

Alternativa 1: Mostrar todos los datos relevantes del juego en una sola pantalla (la única), como lo solicitan los requerimientos.

Alternativa 2: Dividir los datos en pantallas separadas para facilitar la lectura.

Ventajas y desventajas de cada alternativa:

Alternativa 1:

Ventajas: Muestra toda la información en un solo lugar, como lo solicitan los requerimientos.

Desventajas: Puede sobrecargar visualmente al jugador, dificultando la interpretación rápida de los datos e interfiriendo en rasgos visuales fundamentales del juego.

Alternativa 2:

Ventajas: Mejora la claridad al dividir la información, permitiendo una visualización más ordenada y legible.

Desventajas: Requiere navegación adicional por parte del usuario y no mantiene los datos actualizados siempre a la vista del jugador.

Justificación de alternativa seleccionada:

Se selecciona la Alternativa 1 porque los requerimientos solicitan mostrar todos los datos relevantes en una sola pantalla y aunque eso podría afectar la claridad visual, cumple con las especificaciones del proyecto.

Diagrama de clases UML

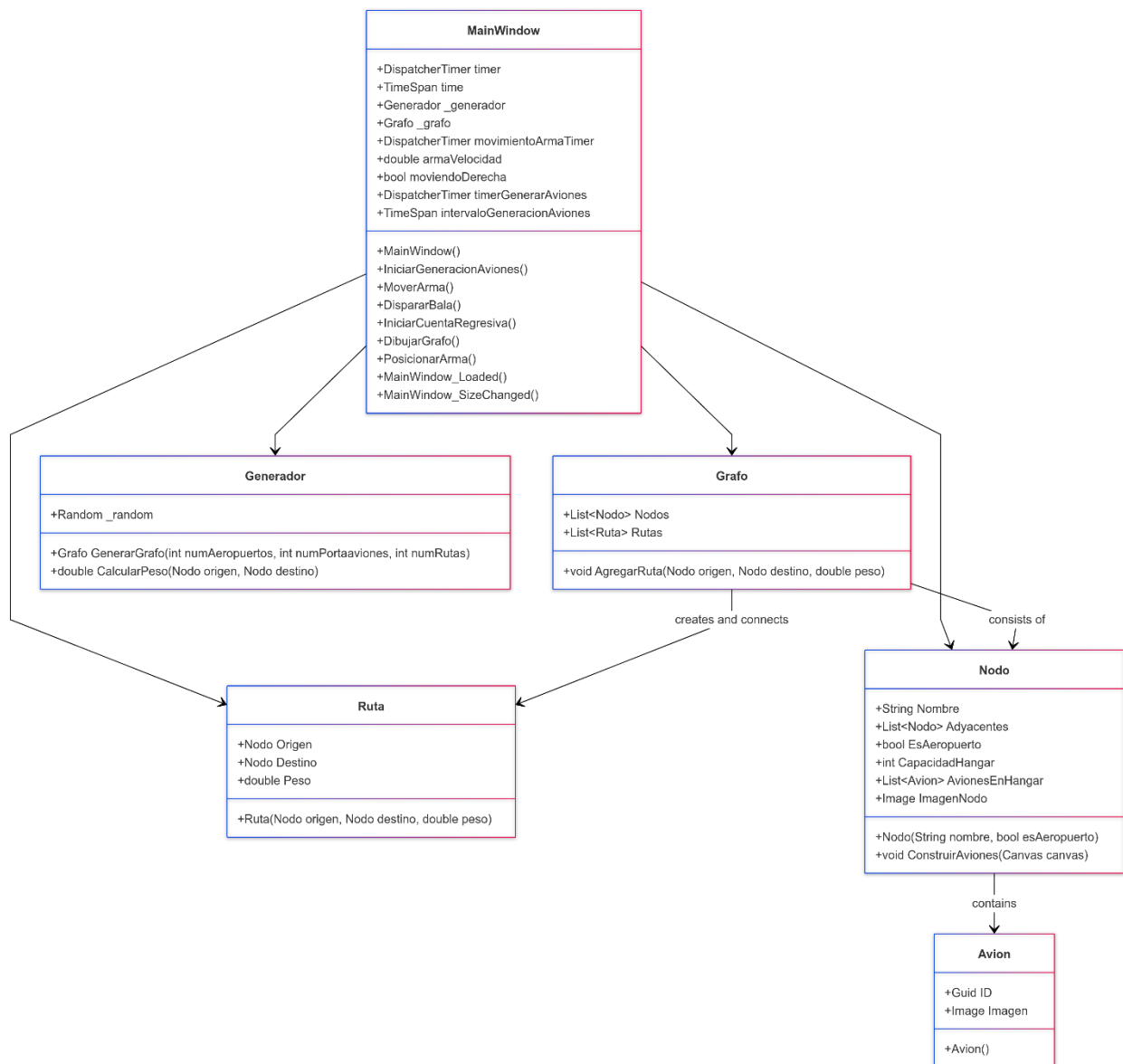
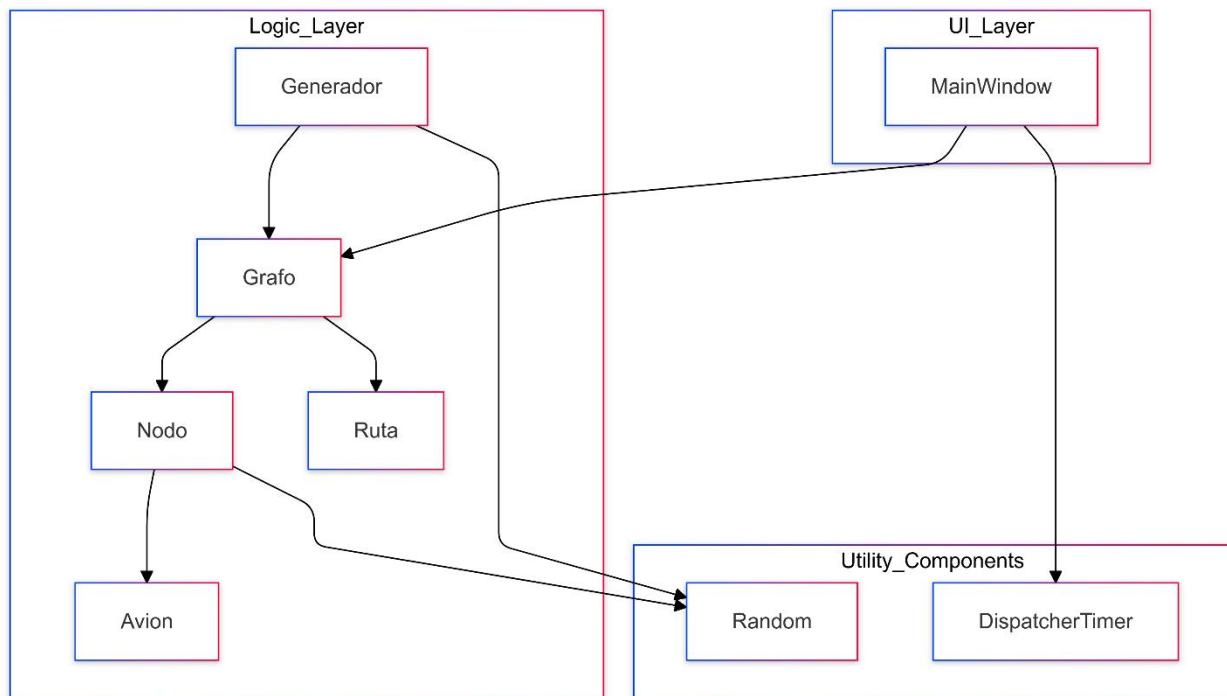


Diagrama de arquitectura

Architecture Diagram for AirWar



Checklist de historias de usuario implementadas pendientes

- ☐ HDU 001 Destrucción de aviones
- ☒ HDU 002 Generación de aeropuertos, portaaviones y rutas
- ☒ HDU 003 Batería antiaérea
- ☒ HDU 004 Generación de rutas con pesos
- ☐ HDU 005 Selección de destinos y recarga de aviones
- ☐ HDU 006 Racionamiento de combustible en aeropuertos
- ☒ HDU 007 Construcción de nuevos aviones
- ☐ HDU 008 Implementación de módulos AI en aviones
- ☐ HDU 009 Ordenación de tripulación de aviones derribados
- ☐ HDU 010 Visualización de datos relevantes del juego