

# AlgoritmoQuest: La Odisea UML

Gamificación Estructural | Ingeniería | Ingeniería de sistemas | Tema: algoritmos y UML

## Contexto Narrativo

### Contexto Narrativo: La odisea de los Ingenieros del Futuro

En un futuro próximo, la humanidad enfrenta una crisis tecnológica sin precedentes. Un vasto sistema global de información, llamado “La Red Central”, se ha vuelto inestable debido a errores en sus algoritmos y diseños deficientes que amenazan con colapsar infraestructuras críticas en todo el planeta. La única esperanza reside en un grupo selecto de ingenieros del sistema, ustedes, estudiantes de Ingeniería de Sistemas, quienes han sido convocados para formar parte de la élite llamada “Los Arquitectos del Código”. Su misión es restaurar el orden en La Red Central dominando el arte de los algoritmos y el modelado UML para diseñar soluciones robustas, claras y eficientes.

Ambientación:

- El aula se transforma en la “Base de Operaciones de Los Arquitectos”, equipada con estaciones de trabajo donde se desarrollarán misiones y retos.
- Los estudiantes asumen roles de especialistas: Analistas de Requerimientos, Diseñadores UML, Programadores de Algoritmos y Líderes de Equipo.
- Cada equipo representa una escuadra encargada de resolver un módulo específico dentro del sistema global.

Rol de los estudiantes dentro de la narrativa:

- **Analista de Requerimientos:** Responsable de interpretar las necesidades del sistema y definir parámetros claros para los algoritmos.
- **Diseñador UML:** Encargado de modelar las soluciones a través de diagramas UML: de casos de uso, de clases, de secuencia y de actividad.
- **Programador de Algoritmos:** Diseña y optimiza los algoritmos que implementan las funcionalidades definidas y modeladas.
- **Líder de Equipo:** Coordina, negocia y asegura la comunicación efectiva para que el equipo cumpla objetivos y respete tiempos.

Misión principal:

Restaurar la estabilidad de La Red Central diseñando, analizando y optimizando algoritmos complejos y modelando sistemas con UML para garantizar un funcionamiento eficiente y seguro. Esto implica completar una serie de misiones (actividades) que abordan desde la comprensión básica de algoritmos hasta la aplicación avanzada de diagramas UML, fortaleciendo así las competencias técnicas y habilidades blandas.

Conexión con el tema de aprendizaje:

El uso de algoritmos y UML no solo es fundamental para resolver problemas complejos de ingeniería sino que, dentro de la narrativa, representa el “lenguaje” y “mapa” que permitirá a los estudiantes entender, comunicar y solucionar las

fallas en el sistema global. Así, cada misión gamificada se alinea con un objetivo pedagógico concreto que se traduce en habilidades prácticas para la ingeniería de sistemas.

Además, el marco narrativo fomenta el trabajo colaborativo, la creatividad para diseñar soluciones, el pensamiento crítico para optimizar algoritmos y diagramas, la negociación para la toma de decisiones en equipo, y la responsabilidad para cumplir con roles y plazos, integrando las competencias del siglo XXI de manera natural y motivadora.

## Mecánicas de Juego

### Mecánicas de Juego Detalladas

#### Sistema de Puntos:

- Los estudiantes acumulan puntos (llamados “Creditos de Código”) por completar actividades, entregar diagramas UML correctos, diseñar algoritmos eficientes y colaborar efectivamente.
- Los puntos varían según la complejidad y calidad, por ejemplo: 50 puntos por análisis correcto, 70 por diagramas UML completos, 100 por algoritmos eficientes que cumplen requisitos, y 30 por participación activa en equipo.
- Los puntos se registran semanalmente y se actualizan en una “Tabla de Clasificación” visible para todos para fomentar competencia sana.

#### Niveles:

- Los niveles representan el progreso y dominio: “Aprendiz de Arquitecto” (0-199 puntos), “Ingeniero en Formación” (200-399 puntos), “Arquitecto Certificado” (400-599 puntos), “Maestro Arquitecto” (600+ puntos).
- Al subir de nivel, los estudiantes desbloquean “Misiones Avanzadas” y tienen acceso a recursos exclusivos (videos, plantillas UML, ejemplos de algoritmos).

#### Insignias:

- Se otorgan insignias digitales por logros específicos, tales como “Maestro del Diagrama de Clases”, “Genio del Algoritmo”, “Líder Colaborativo”, “Negociador Efectivo” y “Responsable del Código”.
- Las insignias se muestran en la plataforma virtual o en el aula (carteles o stickers) y se pueden coleccionar para reconocimiento final.

#### Retos:

- Se plantean retos individuales y grupales que deben resolverse en tiempo limitado, por ejemplo: diseñar un algoritmo para ordenar datos, modelar un caso de uso complejo o negociar roles y tareas en el equipo.
- Los retos tienen niveles de dificultad creciente y premian con puntos extra y “boosters” (ayudas especiales en futuras actividades).

#### Recompensas:

- Además de puntos e insignias, los estudiantes ganan “Boosters” que pueden usar para extender tiempo, pedir pistas o recibir retroalimentación adicional.

- Se entregan reconocimientos simbólicos (certificados digitales, menciones en boletines) para motivar la participación continua.

#### **Progresión:**

- La experiencia está diseñada en módulos secuenciales donde la complejidad y autonomía aumentan progresivamente.
- La tabla de clasificación y niveles permiten visualizar el avance colectivo e individual, promoviendo un sentido de competencia y colaboración simultáneamente.

#### **Retroalimentación Inmediata:**

- Después de cada actividad, los docentes y compañeros dan feedback inmediato mediante rúbricas simplificadas, comentarios orales y digitales para corregir y mejorar.
- Se utilizan herramientas digitales para validación automática en algunos ejercicios (plataformas de UML y pseudocódigo).

En conjunto, estas mecánicas estructuran una experiencia dinámica, desafiante y motivadora que integra la gamificación estructural con objetivos pedagógicos claros, fomentando las competencias del siglo XXI en un entorno realista y atractivo.

## **Actividades Gamificadas**

### **Actividades Gamificadas Paso a Paso**

#### **Actividad 1: "Misión Reconocimiento - Análisis de Requerimientos"**

**Descripción:** Los estudiantes asumen el rol de Analistas para identificar y comprender los requerimientos funcionales y no funcionales del sistema que deben mejorar.

#### **Instrucciones:**

- Se entrega un caso de estudio con una problemática concreta de La Red Central (por ejemplo, un fallo en la gestión de usuarios).
- En equipos de 4, discuten y listan requerimientos claros y medibles.
- Cada equipo presenta un documento con al menos 5 requerimientos funcionales y 3 no funcionales.

**Tiempo estimado:** 90 minutos

**Materiales:** Documento de caso de estudio, hojas, computadora con procesador de texto.

**Integración con mecánicas:** Entrega correcta otorga 50 puntos por equipo, más 10 puntos individuales por participación activa. El equipo gana la insignia "Analista Preciso" si todos los requerimientos son claros, completos y bien justificados.

#### **Actividad 2: "Construyendo el Mapa - Diagramas UML de Casos de Uso"**

**Descripción:** Como Diseñadores UML, los estudiantes modelan los casos de uso basados en los requerimientos analizados.

**Instrucciones:**

- Usando herramientas digitales gratuitas (como draw.io o Lucidchart), crean diagramas de casos de uso que reflejen las funcionalidades principales.
- Identifican actores, casos, relaciones y describen brevemente cada caso de uso.
- Presentan su diagrama para revisión y retroalimentación.

**Tiempo estimado:** 120 minutos

**Materiales:** Computadoras con acceso a internet y software de diagramación, plantilla guía.

**Integración con mecánicas:** Diagramas completos y correctos suman 70 puntos. Se otorga insignia “Maestro de Casos de Uso” si el equipo entrega un diagrama profesional y bien explicado. Posibilidad de usar un booster para extender tiempo si es necesario.

### **Actividad 3: "El Arte del Algoritmo - Diseño y Optimización"**

**Descripción:** Los Programadores diseñan algoritmos para resolver problemas específicos detectados en La Red Central, optimizando desempeño y recursos.

**Instrucciones:**

- Se presenta un problema concreto (por ejemplo, ordenar registros de usuarios para facilitar búsquedas rápidas).
- Los estudiantes diseñan pseudocódigo o diagramas de flujo para el algoritmo que resuelve el problema.
- Se evalúa eficiencia en complejidad, claridad y corrección.

**Tiempo estimado:** 150 minutos

**Materiales:** Hojas para diagramas de flujo, software de pseudocódigo (como Visual Paradigm Online o herramientas offline).

**Integración con mecánicas:** Algoritmos bien diseñados otorgan hasta 100 puntos. El equipo puede negociar con el líder para usar un booster que permita recibir pistas sobre optimización. Se entrega insignia “Genio del Algoritmo” si el diseño es óptimo y documentado.

### **Actividad 4: "La Sinfonía UML - Diagramas de Clases y Secuencia"**

**Descripción:** Los diseñadores UML avanzan para construir diagramas detallados que reflejen la estructura y comportamiento del sistema.

**Instrucciones:**

- Diseñan diagramas de clases identificando atributos, métodos y relaciones entre clases.
- Modelan diagramas de secuencia para representar la interacción entre objetos en casos específicos.
- Integran ambos diagramas en una presentación conjunta para explicar la solución.

**Tiempo estimado:** 180 minutos

**Materiales:** Software de diagramación, plantillas UML, ejemplos previos.

**Integración con mecánicas:** Diagramas completos suman 80 puntos. Insignia “Arquitecto UML” se otorga por claridad, precisión y creatividad. El equipo puede usar boosters para pedir revisión anticipada y mejorar calidad.

### **Actividad 5: "Negociación y Liderazgo - Coordinación de Equipo"**

**Descripción:** El líder de equipo debe gestionar roles, tiempos y negociaciones para cumplir las misiones, fomentando habilidades blandas clave.

**Instrucciones:**

- El líder organiza las tareas en un tablero Kanban físico o digital.
- Facilita sesiones de negociación para resolver conflictos o reasignar tareas en caso de imprevistos.
- Presenta un informe final de gestión con lecciones aprendidas.

**Tiempo estimado:** Durante todo el desarrollo de las actividades (aprox. 2 semanas de clase)

**Materiales:** Tablero Kanban (pizarra o Trello), formatos de informe, cronogramas.

**Integración con mecánicas:** Evaluación continua con 30 puntos por gestión efectiva, insignia “Líder Colaborativo”. El docente puede otorgar puntos extra por iniciativa y resolución de conflictos.

### **Actividad 6: "Reto Final - Integración y Presentación de la Solución"**

**Descripción:** Los equipos presentan una solución integral que incluye análisis de requerimientos, diagramas UML completos y algoritmos diseñados, defendiendo sus decisiones ante un jurado académico.

**Instrucciones:**

- Preparan una presentación multimedia que integre todos los elementos desarrollados.
- Responden preguntas técnicas y de gestión planteadas por docentes y compañeros.
- Reciben retroalimentación y evaluación final.

**Tiempo estimado:** 3 horas (preparación y presentación)

**Materiales:** Computadora, proyector, materiales impresos, rúbricas de evaluación.

**Integración con mecánicas:** Puntos hasta 150 según calidad, defensa y trabajo en equipo. Insignia “Maestro Arquitecto” para el equipo con mejor desempeño global. Se actualizan niveles y tabla de clasificación final.

Estas actividades combinan colaboración, creatividad, pensamiento crítico y resolución de problemas, asegurando que el aprendizaje de algoritmos y UML sea profundo, práctico y motivador.

## **Reglas y Condiciones**

### **Reglas Claras del Juego**

- **Condiciones de victoria:** El equipo que alcance el nivel “Maestro Arquitecto” con la mayor cantidad de puntos al finalizar el módulo obtiene un reconocimiento especial y puede participar en una competencia interuniversitaria.

- **Roles obligatorios:** Cada equipo debe mantener los roles asignados (Analista, Diseñador UML, Programador, Líder) durante todas las actividades. Cambios solo con aprobación docente.
- **Turnos y tiempos:** Las actividades tienen tiempos definidos; el uso de boosters permite extender tiempos o recibir ayudas limitadas.
- **Penalizaciones:** Entregar trabajos incompletos o plagiar resultados implica pérdida de puntos (hasta 30 por actividad) y suspensión temporal de boosters.
- **Tabla de puntos:** Se actualiza semanalmente y es pública para fomentar transparencia y motivación.
- **Sistema de logros:** Las insignias se otorgan automáticamente al cumplir criterios específicos. Acumular tres insignias otorga un bonus de 50 puntos.
- **Colaboración y respeto:** Se espera que todos los miembros participen activamente y respeten opiniones. Incumplimientos pueden conllevar reducción de puntos individuales.
- **Entrega y revisión:** Los trabajos deben entregarse en las fechas establecidas; el docente dará retroalimentación inmediata para mejorar antes de la entrega final.
- **Uso de boosters:** Cada equipo tiene 3 boosters por módulo, que deben usarse estratégicamente para apoyo o extensión.

## Evaluación Gamificada

### Evaluación dentro del Sistema Gamificado

#### Criterios de Evaluación:

- *Calidad Técnica:* Precisión y corrección de algoritmos y diagramas UML.
- *Creatividad:* Originalidad en soluciones propuestas y diseños innovadores.
- *Colaboración:* Participación activa y efectiva en equipo, liderazgo y negociación.
- *Responsabilidad:* Cumplimiento de roles, tiempos y entrega de trabajos.
- *Comunicación:* Claridad en exposiciones y documentación.

#### Rúbricas Integradas:

- Para algoritmos: evaluación por claridad (0-20), eficiencia (0-30), documentación (0-20), corrección (0-30).
- Para UML: evaluación por completitud (0-25), precisión (0-25), legibilidad y presentación (0-25), innovación (0-25).
- Para colaboración: participación (0-40), liderazgo y negociación (0-30), responsabilidad (0-30).

#### Evidencias de Aprendizaje:

- Documentos de análisis de requerimientos.
- Diagramas UML entregados y presentados.
- Algoritmos diseñados en pseudocódigo y diagramas de flujo.
- Informes de gestión y negociación.
- Presentaciones finales defendidas ante jurado.

### Reflexión Final:

Al concluir la odisea, cada estudiante escribe una reflexión personal sobre su aprendizaje, desafíos enfrentados, habilidades desarrolladas y cómo aplicarán estos conocimientos en su formación profesional.

### Cierre de la Narrativa:

La Red Central vuelve a estabilizarse gracias al trabajo conjunto de Los Arquitectos del Código. Se celebra la restauración del sistema con un reconocimiento simbólico y la entrega de certificados que avalan sus competencias técnicas y habilidades blandas, preparando a los futuros ingenieros para retos reales del mundo tecnológico.

## Recomendaciones Logísticas

### Recomendaciones para la Implementación

- **Tiempo necesario:** Aproximadamente 4 a 6 semanas de clase (2-3 sesiones semanales de 90-120 minutos) para cubrir todas las actividades con profundidad.
- **Espacio físico:** Aula equipada con mesas para trabajo en equipo, pizarras para tableros Kanban, proyector para presentaciones y espacio para exposiciones orales.
- **Materiales y herramientas TIC:**
  - Computadoras con acceso a internet.
  - Software gratuito para diagramas UML (draw.io, Lucidchart, Visual Paradigm Online).
  - Herramientas para pseudocódigo y diagramas de flujo (Visual Paradigm, Pencil Project).
  - Plataformas para gestión colaborativa (Trello, Google Drive).
  - Plantillas y guías impresas para análisis y documentación.
- **Tamaño del grupo:** Ideal entre 16 y 24 estudiantes para formar 4-6 equipos de trabajo, permitiendo roles definidos y colaboración efectiva.
- **Preparación previa del docente:**
  - Familiarizarse con herramientas digitales y preparar casos de estudio contextualizados.
  - Diseñar rúbricas claras y guías de evaluación.
  - Planificar cronogramas y mecanismos de retroalimentación inmediata.
  - Capacitar a estudiantes en roles y expectativas.
- **Posibles dificultades y cómo superarlas:**
  - *Desconocimiento técnico:* Realizar sesiones introductorias previas para nivelar conocimientos básicos de UML y algoritmos.
  - *Desigualdad en participación:* Vigilar que cada miembro cumpla rol; usar evaluaciones individuales y grupales.
  - *Problemas técnicos con software:* Tener alternativas offline y materiales impresos para respaldo.
  - *Gestión del tiempo:* Usar boosters estratégicamente y ajustar tiempos si es necesario.

- *Falta de motivación:* Mantener la narrativa atractiva, ofrecer reconocimientos y promover competencia sana.