



Universidad  
de Alcalá

# DOCUMENTACIÓN

## Práctica Laboratorio - “*Number Tycoon*”

Universidad de Alcalá  
Grado en Ingeniería en Sistemas de Información (GISI)  
Patrones Software  
Tutor: Salvador Otón Tortosa

David Serrano Díaz  
Alejandro Torres de Diego  
Rodrigo Palomo Cuenca

Universidad de Alcalá  
Madrid, Spain

[david.serranod@edu.uah.es](mailto:david.serranod@edu.uah.es) 03226056F  
[alejandro.torresd@edu.uah.es](mailto:alejandro.torresd@edu.uah.es) 03202122Q  
[rodrigo.palomo@edu.uah.es](mailto:rodrigo.palomo@edu.uah.es) 03220316V



## Índice

---

<b>1. Enunciado</b>	<b>2</b>
1.1 Usuarios de la Aplicación	2
1.2 Elementos del Sistema	2
1.2.1 Minas	2
1.2.2 Pozos	2
1.2.3 Módulos de Operación	3
1.2.4 Dividers y Mergers	3
1.2.5 Cintas Transportadoras	3
1.2.6 Sistema de Puntos y Mejoras	3
1.3 Funcionamiento de la Aplicación	3
1.4 Almacenamiento de Datos	3
1.5 Excepciones y Validaciones	4
1.6 Consideraciones Técnicas	4
<b>2. Requisitos</b>	<b>5</b>
2.1 Funcionales	5
2.2 No funcionales	6
<b>3. Manual de Usuario</b>	<b>7</b>
<b>4. Diseño completo de la aplicación (UML)</b>	<b>7</b>
4.1 Diagramas de clases	7
4.2 Casos de uso	7
<b>5. Patrones utilizados (enunciación y justificación)</b>	<b>7</b>
<b>X. Roadmap de desarrollo</b>	<b>8</b>
<b>Y. Falta por hacer</b>	<b>8</b>



## 1. Enunciado

Se contempla el diseño y realización de una aplicación informática denominada “Number Tycoon”, un videojuego de simulación y gestión en el que el jugador construye y administra una red de minas, pozos y máquinas matemáticas.

El objetivo es producir, transformar y consumir números de forma automatizada para obtener puntos, mejorar la infraestructura y optimizar la red de producción.

El jugador deberá planificar la disposición de sus máquinas, controlar los flujos numéricos, y conseguir una red eficiente que maximice la puntuación obtenida. La aplicación combinará elementos de automatización, estrategia y gestión en tiempo real.

### 1.1 Usuarios de la Aplicación

La aplicación está diseñada para un único tipo de usuario:

- **Jugador:**

Es el usuario principal de la aplicación. Puede iniciar una nueva partida o continuar una existente. Durante la sesión, el jugador podrá construir, mejorar o eliminar estructuras, activar minas, colocar módulos de operación y optimizar el flujo numérico.

El jugador obtiene puntos al consumir números en los pozos, y con ellos puede adquirir nuevas minas, módulos y mejoras.

### 1.2 Elementos del Sistema

El sistema está compuesto por los siguientes elementos de juego, que conforman la base del simulador:

#### 1.2.1 Minas

Generan un flujo continuo de números enteros

- Cada mina produce un tipo de número base (por ejemplo, 1, 2, 3...).
- Las minas están precolocadas en el mapa y se activan al comprarlas.
- Pueden mejorarse para aumentar su eficiencia (+1 unidad de producción).

#### 1.2.2 Pozos

Consumen un número específico.

- Cada pozo tiene una dificultad y una puntuación fija.
- Ejemplo: el Pozo 3 consume números “3” y otorga 2 puntos por cada número procesado.
- Nuevos pozos aparecen cuando se alcanza un umbral de 100 números consumidos.



### 1.2.3 Módulos de Operación

Transforman los números mediante operaciones matemáticas:

- Suma, Multiplicación, Resta, División (entera), etc.
- Disponen de dos entradas y una salida.
- Pueden colocarse libremente en el mapa y conectarse a otros elementos mediante cintas.

### 1.2.4 Dividers y Mergers

- Divider: divide un flujo numérico en dos salidas.
  - Merger: combina dos flujos del mismo tipo en una salida.
- Estos elementos permiten organizar y optimizar los flujos numéricos del sistema.

### 1.2.5 Cintas Transportadoras

Conectan minas, pozos y módulos.

- No tienen coste de colocación ni de eliminación.
- Los flujos numéricos se representan visualmente como números en movimiento.

### 1.2.6 Sistema de Puntos y Mejoras

- Los puntos obtenidos permiten comprar nuevas estructuras.
- Se incluyen mejoras de velocidad de cintas (hasta  $\times 3$ ) y eficiencia de minas (+1 unidad).

## 1.3 Funcionamiento de la Aplicación

El videojuego se ejecutará en una interfaz gráfica desarrollada con la librería pygame, donde se representará un mapa que contiene minas, pozos, cintas y módulos conectados.

El sistema debe permitir las siguientes operaciones:

- Construir y eliminar estructuras libremente.
- Comprar nuevas minas, módulos o mejoras mediante puntos.
- Gestionar el flujo numérico entre elementos conectados.
- Visualizar en tiempo real los números moviéndose por las cintas.
- Guardar y cargar partidas, conservando el progreso del jugador.
- Mostrar avisos y efectos visuales al alcanzar hitos, como la aparición de un nuevo pozo.

## 1.4 Almacenamiento de Datos

El sistema debe guardar y recuperar el progreso del jugador, incluyendo:

- Estructuras activas (minas, pozos, módulos, cintas).
- Puntos acumulados.
- Mejoras adquiridas.
- Configuración del mapa.

Los datos se almacenarán en archivos de texto o binarios (.json o .dat), y se cargarán automáticamente al iniciar la aplicación.



## 1.5 Excepciones y Validaciones

Además de las excepciones generales del lenguaje, el sistema controlará las siguientes situaciones:

- Colocar un módulo o pozo en una posición ocupada.
- Intentar comprar sin puntos suficientes.
- Eliminar un elemento inexistente.
- Conectar módulos incompatibles.
- Activar una mina o pozo ya activo.
- Superar los límites máximos de mejora de velocidad o eficiencia.

## 1.6 Consideraciones Técnicas

- Lenguaje: Python 3.x
- Librerías: pygame, dataclasses, abc, json
- Arquitectura: Orientada a objetos, con aplicación de al menos 10 patrones de diseño (2 creacionales, 2 estructurales, 2 de comportamiento y 4 adicionales sin contar los fundamentales).
- Plataforma: Aplicación de escritorio (Windows, Linux, macOS).



## 2. Requisitos

### 2.1 Funcionales

Ref	Descripción	Prioridad
RF001	Existen minas y pozos: de las minas sale un flujo de números; los pozos los consumen y otorgan puntos.	Alta
RF002	Cada pozo tiene una dificultad fija y una asignación de puntos predefinida. Ejemplo: el pozo 3 consume “treses” y da 2 puntos por número.	Media
RF003	Los pozos aparecen al alcanzar un umbral de 100 números consumidos en el anterior; su ubicación será aleatoria dentro del mapa.	Alta
RF004	El jugador puede comprar minas, módulos de operación (suma, multiplicación), divisores y mergers.	Alta
RF005	El juego es continuo y sin estancias, permitiendo la reutilización de estructuras existentes.	Media
RF006	El jugador puede crear y eliminar construcciones libremente.	Alta
RF007	Al eliminar una máquina, el jugador recupera los puntos invertidos (total o parcialmente).	Media
RF008	Las cintas transportadoras no tienen coste de colocación ni eliminación.	Alta
RF009	Las minas están preestablecidas en el mapa desde el inicio; al comprarlas se activan.	Alta
RF010	El escenario inicial incluye una mina 1 y un pozo 1 conectados por una cinta gratuita, con saldo inicial 0.	Alta
RF011	Los módulos de suma y multiplicación poseen dos entradas numéricas.	Alta
RF012	Sistema de upgrades: mejora de velocidad de cintas (hasta x3) y de eficiencia de minas (+1 unidad).	Media
RF013	El <i>merger</i> combina dos flujos numéricos de la misma base en una única salida.	Alta



## 2.2 No funcionales

Ref	Descripción	Prioridad
<b>RNF001</b>	Animación visual cuando el jugador obtiene puntos.	<b>Media</b>
<b>RNF002</b>	Los flujos numéricos se representan en movimiento sobre las cintas transportadoras.	<b>Alta</b>
<b>RNF003</b>	Efecto visual destacado cuando aparece un nuevo pozo.	<b>Media</b>
<b>RNF004</b>	El juego incluirá música y animaciones variadas para ambientación.	<b>Media</b>
<b>RNF005</b>	Existirá un menú principal con opciones de inicio, continuar y salir.	<b>Alta</b>
<b>RNF006</b>	El sistema incluirá ajustes de resolución y control de audio.	<b>Alta</b>
<b>RNF007</b>	El mapeo de controles estará preestablecido, pendiente de definición detallada.	<b>Baja</b>



### **3. Manual de Usuario**

## **4. Diseño completo de la aplicación (UML)**

### **4.1 Diagramas de clases**

### **4.2 Casos de uso**

## **5. Patrones utilizados (enunciación y justificación)**





## X. Roadmap de desarrollo

Fase	Objetivo principal	Requisito	Resultado esperado
<b>Build 0.1 Núcleo Lógico (MVP)</b>	Implementar el flujo base de minas y pozos, con sistema de puntos y umbrales.	RF001, RF002, RF003, RF010, RF011	Juego funcional con generación de números, pozos progresivos y puntuación.
<b>Build 0.2 Sistema de Construcción y Costes</b>	Permitir al jugador construir, eliminar y comprar máquinas y cintas.	RF004, RF007, RF008, RF009	Sistema económico funcional y estructuras dinámicas.
<b>Build 0.3 Operadores Matemáticos</b>	Introducir módulos de suma, multiplicación, divisor y merger.	RF013, RF015	Red matemática interactiva y configurable.
<b>Build 0.4 Sistema de Upgrades</b>	Añadir mejoras de minas y cintas.	RF014	Jugabilidad progresiva con incentivos de optimización.
<b>Build 0.5 Experiencia de Usuario</b>	Añadir animaciones, visualización de flujos y feedback visual.	RNF001, RNF002, RNF003	Experiencia fluida y visualmente atractiva.
<b>Build 0.6 Interfaz y Configuración</b>	Crear menú principal y ajustes básicos.	RNF004, RNF005, RNF006	Interfaz funcional, audio configurable y presentación sólida.
<b>Build 0.7 Estabilización Final</b>	Integración de controles definidos y balance general.	RNF007 + revisión global	Versión estable y lista para ampliaciones futuras.

## Y. Falta por hacer

1. Concreción total de requisitos de interfaz y controles
2. Requisitos/apartado específico de concreción de patrones software a usar y donde