Manual Técnico – Totito Chino (Timbiriche)

1. Descripción general

Este programa implementa el juego **Totito Chino (Timbiriche)** como parte de la práctica de **Estructura de Datos** en C++.
El código hace uso de listas enlazadas, pilas y colas para manejar tablero, turnos y powerups.

2. Lenguaje y entorno

- **Lenguaje**: C++ (estándar C++17 o superior).
- **Entorno de desarrollo recomendado**:
  - **Dev-C++** (para Windows, sencillo de instalar).
  - **Code::Blocks** (multiplataforma).
  - **g++** (compilador GNU para Linux/macOS).

3. Requisitos del sistema

- **Windows** 7/10/11 (32 o 64 bits) o
- **Linux** (cualquier distribución con g++ instalado).
- **Memoria**: mínimo 2 GB RAM.
- **Almacenamiento**: mínimo 200 MB libres.
- **Compilador C++**:
  - Windows: Dev-C++ ya incluye Mingw/g++.
  - Linux/macOS: instalar con el gestor de paquetes:
  - sudo apt install g++

4. Instalación de Dev-C++

1. Descargar **Dev-C++** desde https://sourceforge.net/projects/orwelldevcpp/.
2. Ejecutar el instalador.
3. Seguir el asistente → seleccionar carpeta de instalación → finalizar.
4. Abrir Dev-C++.

5. Creación del proyecto en Dev-C++

1. Abrir Dev-C++.
2. Ir a **Archivo → Nuevo → Proyecto → Proyecto Consola C++**.
3. Asignar nombre (ej: TotitoChino) y carpeta de guardado.
4. Copiar los archivos .cpp y .h del proyecto en esa carpeta.
5. En Dev-C++, añadir los archivos al proyecto (menú **Proyecto → Añadir a proyecto**).

6. Compilación y ejecución en Dev-C++

1. Presionar **F9** o ir a **Ejecutar → Compilar y ejecutar**.
2. Si la compilación fue correcta, se abre una ventana de consola mostrando el menú del juego.
3. En caso de errores:
    - Revisar que el proyecto esté en modo **C++**.
    - Revisar que todas las librerías estándar (iostream, vector, etc.) estén bien escritas.

7. Compilación desde terminal (opcional)

Si se usa g++ directamente:

g++ -std=c++17 -Wall -Wextra -o totito main.cpp game.cpp board.cpp player.cpp powerups.cpp tads.cpp
./totito

8. Estructura del proyecto

- main.cpp: punto de entrada del programa.
- game.cpp / game.h: control de flujo del juego y reglas.
- board.cpp / board.h: representación del tablero.
- player.cpp / player.h: información y lógica de los jugadores.
- powerups.cpp / powerups.h: implementación de los powerups.
- tads.cpp / tads.h: implementación de estructuras de datos.
- Makefile: (opcional) archivo para compilar automáticamente.

9. Funcionamiento interno

- Se muestra un menú en consola.
- El usuario ingresa parámetros de tablero y jugadores.
- El tablero se gestiona mediante listas y matrices.
- Los turnos de jugadores se almacenan en una **cola**.
- Los powerups de cada jugador se almacenan en una **pila**.
- Cuando se llena una casilla, se asigna al jugador, se incrementa puntaje y se aplican reglas especiales.

10. Resolución de problemas

- **Error "undefined reference"**: revisar que todos los archivos .cpp estén añadidos al proyecto.
- **La ventana se cierra al terminar**: en Dev-C++, añadir system("pause"); antes de return 0;.
- **Errores de acentos/ñ en consola**: cambiar la codificación a UTF-8.

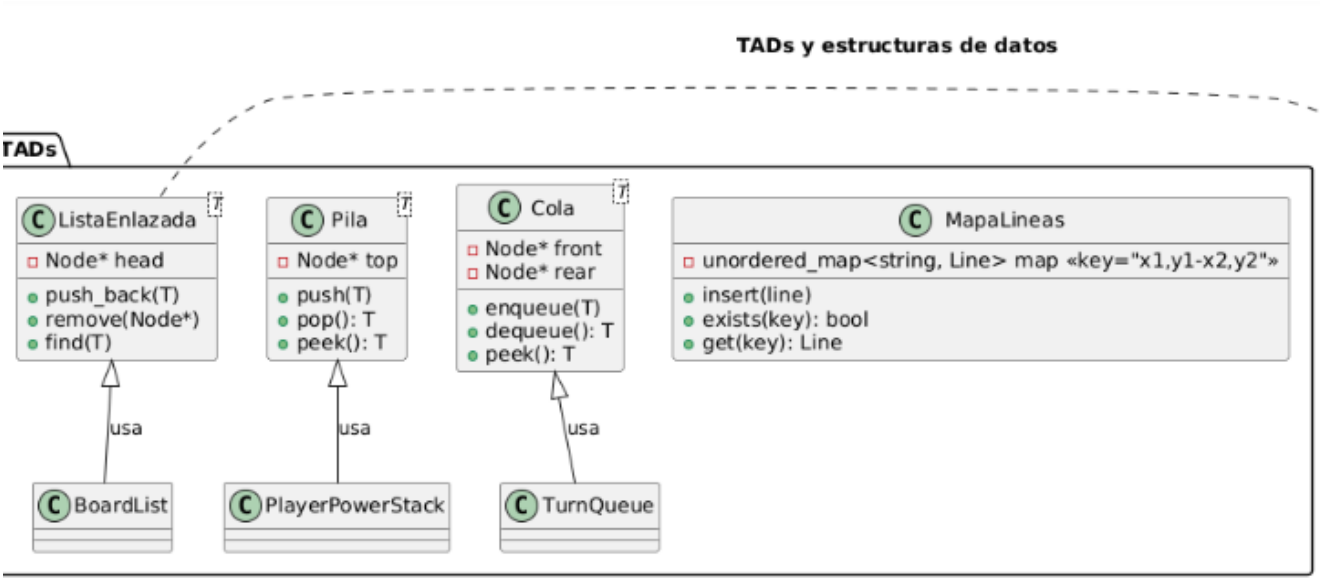

Diagrama de Clases - Totito Chino