



UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA (UTEC)

Curso: Programación II
Proyecto Final (4 semanas)

Rival Frontiers

13 de octubre de 2025

Resumen

Diseña un juego de estrategia por turnos en consola donde la civilización principal se enfrenta a un guardián controlado por un segundo rol de jugador; ambos compiten en un tablero matricial que alberga enclaves neutrales y estructuras clave cuyas posesiones cambian según los eventos de cada turno. La partida se define al administrar recursos, cumplir misiones de campaña y superar umbrales de dominio territorial, mientras la derrota acecha si el guardián conquista objetivos críticos o agota las reservas del rival. El proyecto debe fusionar POO (herencia, polimorfismo, sobrecarga de operadores) con lógica procedural (ciclo de turnos, validaciones, reportes), utilizando exclusivamente librerías estándar de C++ como `<iostream>`, `<vector>`, `<string>` y `<fstream>`.

Objetivos de aprendizaje

- Diseñar arquitecturas modulares que separen dominio, interfaz, sistemas de eventos y persistencia.
- Modelar mundos discretos con contenedores estándar y patrones flexibles para estados, recursos, disparadores y efectos.
- Orquestrar ciclos de juego por turnos combinando decisiones del jugador, simulaciones del mundo y mantenimiento del estado global.
- Documentar decisiones técnicas, definir criterios de prueba y planificar extensiones futuras del sistema.

Alcance jugable

- Campaña en solitario de 8–12 turnos base, ampliable a escenarios personalizados.
- Civilización jugable con **progresión de recursos**, **habilidades desbloqueables** y **misiones narrativas**.
- Retadores sistémicos: guardianes automáticos, trampas activables y objetivos que reaccionan mediante reglas predefinidas.
- Eventos dinámicos: refuerzos programados, climas adversos, rutas comerciales o sabotajes que influyen en el tablero.

Requisitos funcionales mínimos

1. **Mapa 2D** (p.ej., 12x12) con terrenos como Llanura (**LL**), Bosque (**BO**), Montaña (**MO**), Agua (**AG**) y variantes opcionales.
2. **Ciclo de turno** dividido en fase del jugador, fase del mundo (eventos/rutinas) y registro de sucesos.
3. **Comportamientos polimórficos** en Terreno (costos, bonos), Unidad (movimiento, ataque) y Edificio (efecto por turno).
4. **Condiciones de victoria y derrota**: dominio territorial, puntaje objetivo y chequeos de recursos u objetivos narrativos.
5. **Interfaz de consola** con menús, reportes en tiempo real y notificaciones de cambios en el mundo.

Requisitos técnicos

POO

- **Terreno** → **Llanura, Bosque, Montaña, Agua, Pantano.**
 - costo_movimiento(const Unidad&)
 - bono_defensa(const Unidad&)
- **Controlador** → **Jugador, Sistema, Neutral.**
 - resolver_fase(Contexto&)
 - aplicar_rutina(const Evento&)
- **Unidad** → **Soldado, Arquero, Caballero, Mago, Ingeniero.**
 - mover(const Coordenada&)
 - atacar(Unidad&, Contexto&)
 - habilidad_especial(Contexto&)
- **Edificio** → **Granja, Cuartel, Torre, Forja.**
 - efecto_turno(Faccion&)
 - reaccion_evento(const Evento&)
- **Sobrecargas.**
 - << impresión del estado (mapa, bitácora, reportes).
 - += anexar celdas al dominio del jugador o marcar zonas activadas por eventos.
 - > comparar poder militar, dominio o prioridades de objetivos.

Sistemas del mundo

- Motor de eventos que combine cronogramas, disparadores y rutinas predefinidas.
- Tablas de estado que alteren el tablero (refuerzos programados, clima, bonus temporales).
- Bitácora de acciones y replay básico de turnos para depurar.

Procedural

- Asignar un **presupuesto de 2 puntos de acción por turno** al jugador y registrar su consumo (movimientos, construcciones, habilidades). Rutinas del mundo se ejecutan fuera de este presupuesto.
- Definir el ciclo de cada turno: decisiones del jugador, fase del mundo (rutinas/eventos), resolución de sucesos y actualización de indicadores.
- Establecer validaciones transversales para reglas de movimiento, combate, construcción y consumo de recursos.
- Ofrecer utilidades reutilizables (persistencia, generación de mapas, diagnóstico, exportación de reportes).

Mecánicas clave

- **Economía:** producción de comida, metal y energía para reclutar unidades o activar habilidades.
- **Tecnología:** *tech tree* sencillo con mejoras permanentes o temporales.
- **Moral:** modificador que depende de victorias, clima y eventos narrativos.
- **Economía de acciones:** cada turno el jugador dispone de 2 puntos de acción; las rutinas del mundo se resuelven con scripts predefinidos.
- **Guardianes automáticos:** scripts de patrullaje, defensa de objetivos y contra-ofensivas predefinidas.
- **Misiones opcionales:** escoltar caravanas, defender aldeas, recuperar artefactos.

Sistema de recursos

- **Pool inicial:** Comida 18, Metal 12 y Energía 7 (como se observa en el panel superior de ejemplo).
- **Producción por turno:** cada J1Gr aporta +2 comida, J1Cu aporta +1 metal y habilita reclutamientos, J1To aporta +1 energía y extiende la visión; edificios neutrales capturados pasan a producir para el jugador.
- **Costos de mantenimiento:** cada unidad consumida al cierre del turno reduce 1 comida; si no hay comida suficiente, la moral cae y la unidad queda inactiva.
- **Costos de acciones:** los puntos de acción determinan el número de decisiones por turno, pero además algunas acciones consumen recursos según la tabla siguiente.

Acción	Comida	Metal	Energía
Mover unidad	0	0	0
Reclutar J1S/J1A/J1C	1	1	0
Construir J1Gr	0	1	1
Construir J1To	0	2	1
Activar J1I / desplegar ingeniero	1	0	1
Habilidad especial	0	0	1

Los recursos se descuentan en el momento de la acción; si el jugador no dispone del costo exacto, la acción queda bloqueada hasta el siguiente turno.

Interfaz y experiencia de usuario

- Panel superior con información del turno, recursos, progreso del objetivo y alertas del mundo (moral o clima se añaden solo si el escenario lo necesita).
- Menú de acciones contextual: mover, atacar, construir, habilidades, reportes, guardar/cargar.
- Reportes detallados por turno (acciones propias, reacciones del mundo, eventos, cambios de dominio).
- Sistema de ayuda en línea (atajos, glosario de iconos, explicación de terrenos).
- Registro opcional de telemetría para análisis posterior (CSV o texto).

Layout de consola

Panel superior (ejemplo)

```
1 Turno #8
2 Objetivo: Alcanzar 60% de dominio antes del turno 20
3 Recursos: Comida=18 | Metal=12 | Energía=7
4 Jugador: Dominio=25% | Sistema: Dominio=22%
5 Alerta: Guardianes activos en cuadrante noreste
```

Mapa 2D (vista compacta)

```
Turno #8 | Dominio jugador: 25% | Dominio sistema: 22% |
  Neutrales: 53%
Objetivo activo: Recuperar los centros logísticos del norte

      0      1      2      3      4      5
0 [ J1S/LL][ .. /B0][ J2S/LL][ .. /AG][ J1Cu/LL][ .. /..]
1 [ .. /..][ J1A/B0][ J1M/LL][ J2To/LL][ J1Gr/LL][ .. /..]
2 [ .. /M0][ J2S/LL][ J2M/LL][ J2Gr/LL][ .. /B0][ .. /AG]
3 [ .. /..][ .. /..][ NeCu/LL][ .. /..][ J1S/B0][ .. /..]
4 [ .. /..][ .. /..][ J2A/LL][ .. /..][ .. /..][ J2Cu/LL]
5 [ .. /AG][ .. /..][ J1To/LL][ J1S/LL][ J2S/B0][ .. /..]
```

Evolución del tablero (paso a paso)

1. **Estado 0 — Preludio.** Ocupación inicial: J1Cu en (0,0), patrulla J1S en (1,1), guardianes J2S/J2M/J2A en (0,2), (2,2) y (4,2), con NeCu sin capturar en (3,2). Recursos disponibles: Comida 18, Metal 12, Energía 7.

```
1      0      1      2      3      4      5
2 0 [ J1Cu/LL][ .. /B0][ J2S/LL][ .. /AG][ .. /LL][ .. /LL]
3 1 [ .. /LL][ J1S/LL][ .. /LL][ .. /LL][ .. /LL][ .. /LL]
4 2 [ .. /M0][ .. /LL][ J2M/LL][ J2Gr/LL][ .. /B0][ .. /AG]
5 3 [ .. /LL][ .. /LL][ NeCu/LL][ .. /LL][ .. /LL][ .. /LL]
6 4 [ .. /LL][ .. /LL][ J2A/LL][ .. /LL][ .. /LL][ J2Cu/LL]
7 5 [ .. /AG][ .. /LL][ .. /LL][ .. /LL][ .. /B0][ .. /LL]
```

2. **Estado 1 — Turno 1 (jugador).** Acciones (2/2): J1S avanza de (1,1) a (1,3) y, como segunda acción, se construye J1Gr en (1,3) para sostener la ofensiva. Recursos tras las acciones: Comida 18, Metal 11, Energía 6 (coste de la granja, que producirá +2 comida desde el siguiente turno).

	0	1	2	3	4	5
1	0	[J1Cu/LL][../B0][J2S/LL][../AG][../LL][
2	0	[J1Cu/LL][../B0][J2S/LL][../AG][../LL][
3	1	[../LL][../LL][J1Gr/LL][../LL][
4	2	[../M0][../LL][J2M/LL][J2Gr/LL][
5	3	[../LL][../LL][NeCu/LL][../LL][
6	4	[../LL][../LL][J2A/LL][../LL][
7	5	[../AG][../LL][../LL][../LL][

3. **Estado 2 — Fase 1 del mundo.** Rutina defensiva: J2S rota de (0,2) a (1,2) y J2M se acerca al centro para cubrir (2,2).

	0	1	2	3	4	5
1	0	[J1Cu/LL][../B0][../LL][../AG][../LL][
2	0	[J1Cu/LL][../B0][../LL][../AG][../LL][
3	1	[../LL][../LL][J2S/LL][J1Gr/LL][
4	2	[../M0][J2M/LL][../LL][J2Gr/LL][
5	3	[../LL][../LL][NeCu/LL][../LL][
6	4	[../LL][../LL][J2A/LL][../LL][
7	5	[../AG][../LL][../LL][../LL][

4. **Estado 3 — Turno 2 (jugador).** Acciones (2/2): J1S avanza desde (3,4) hasta (2,4) capturando NeCu al pasar por (3,2); la segunda acción erige J1To en (3,4) para vigilar el corredor. Recursos tras las acciones: Comida 18, Metal 9, Energía 5 (coste de la torre); NeCu pasa a producir +1 metal a partir del siguiente turno.

	0	1	2	3	4	5
1	0	[J1Cu/LL][../B0][../LL][../AG][../LL][
2	0	[J1Cu/LL][../B0][../LL][../AG][../LL][
3	1	[../LL][../LL][J2S/LL][J1Gr/LL][
4	2	[../M0][J2M/LL][../LL][J2Gr/LL][
5	3	[../LL][../LL][J1Cu/LL][../LL][
6	4	[../LL][../LL][J2A/LL][../LL][
7	5	[../AG][../LL][J1To/LL][../LL][

5. **Estado 4 — Fase 2 del mundo.** Rutina del sistema (sin costo para el jugador): J2S se desplaza de (1,2) a (2,2), J2M baja a (3,1), J2A ocupa (3,3) y surge un nuevo J2S en (4,4). Recursos del jugador permanecen en Comida 18, Metal 9, Energía 5.

	0	1	2	3	4	5
1	0	[J1Cu/LL][../B0][../LL][../AG][../LL][
2	0	[J1Cu/LL][../B0][../LL][../AG][../LL][
3	1	[../LL][../LL][J1Gr/LL][J1S/LL][
4	2	[../M0][../LL][J2S/LL][J2Gr/LL][
5	3	[../LL][J2M/LL][J1Cu/LL][J2A/LL][
6	4	[../LL][../LL][../LL][J2S/LL][
7	5	[../AG][../LL][J1To/LL][../LL][

6. **Estado 5 — Turno 3 (jugador).** Acciones (2/2): se gasta un punto para que J1S avance de (1,4) a (2,4) y elimine al guardián de ese corredor (sin coste adicional); el segundo punto despliega J1I en (1,1) y, usando su habilidad, funda J1Gr en (4,3). Recursos tras las acciones: Comida 17, Metal 8, Energía 3.

	0	1	2	3	4	5
1	0	[J1Cu/LL][.. /B0][.. /LL][.. /AG][.. /LL][.. /LL]				
2	1	[.. /LL][J1I/LL][.. /LL][J1Gr/LL][.. /LL][.. /LL]				
3	2	[.. /M0][.. /LL][.. /LL][J2Gr/LL][J1S/B0][.. /AG]				
4	3	[.. /LL][J2M/LL][J1Cu/LL][J2A/LL][J1To/LL][.. /LL]				
5	4	[.. /LL][.. /LL][.. /LL][J1Gr/LL][J2S/LL][J2Cu/LL]				
6	5	[.. /AG][.. /LL][J1To/LL][.. /LL][.. /B0][.. /LL]				
7						

Nota sobre prefijos neutrales. El código Ne resalta estructuras sin propietario inicial; si prefieres mostrarlas como casillas libres, puedes reemplazarlas por .. siempre que ajustes la leyenda.

Leyenda de códigos

Convención general — cada celda se representa como PrefijoTipo/Terreno.

Prefijo	Control
J1	Jugador 1 - Civilización controlada por la persona
J2	Jugador 2 - Guardianes automáticos gestionados por rutinas del sistema
Ne	Enclave neutral o estructura sin propietario

Tipo	Jugador 1	Jugador 2 (guardian)
Soldado	J1S	J2S
Arquero	J1A	J2A
Caballero	J1C	J2C
Mago	J1M	J2M
Ingeniero	J1I	J2I

Código	Descripción
J1Cu, J2Cu, NeCu	Cuarteles activos: generan unidades o sirven de punto de refuerzo.
J1To, J2To, NeTo	Torres defensivas: otorgan bono de defensa y visión.
J1Gr, J2Gr, NeGr	Granjas o centros logísticos: producen recursos por turno.
..	Casilla libre sin unidad ni edificio.

Restricciones y reglas

- Una sola unidad activa por casilla; construir requiere desocuparla previamente.
- Cada turno del jugador dispone de **2 puntos de acción**; mover, construir o activar habilidades consume 1 punto. Los efectos del mundo (guardianes/eventos) no consumen ese presupuesto.
- El desplazamiento es ortogonal (arriba, abajo, izquierda, derecha) y respeta el costo del terreno y la fase de control.
- Terrenos de agua (AG) exigen unidades náuticas; montañas (MO) incrementan el costo de movimiento.
- El jugador y el sistema solo pueden construir o activar estructuras en casillas bajo su control y libres de enemigos.
- La victoria por dominio se verifica al cierre del turno del jugador; la derrota se evalúa si el sistema activa objetivos críticos o deja al jugador sin recursos vitales.

Flujo de partida (ejemplo)

1. **Turno 1 (Jugador):** recluta J1S en J1Cu, asegura J1To y captura (3,4) para elevar el dominio.
2. **Fase 1 del mundo:** los guardianes desplazan J2S hacia (1,2), refuerzan J2To y activan alarmas locales.
3. **Turno 2 (Jugador):** activa J1Gr para producir recursos, lanza un ataque sobre J2Gr y estabiliza el frente sur.
4. **Fase 2 del mundo:** el sistema intenta recuperar NeCu, redistribuye fuerzas y reclama el 22 % del dominio.
5. **Evaluación:** tras cada ronda se verifican los objetivos; si ninguno se cumple, la campaña continúa.

Cronograma sugerido (4 semanas)

Semana	Actividad principal
1	Definir arquitectura, modelos de datos, generación de mapas y visualización en consola.
2	Implementar ciclo de turnos, unidades, terrenos y combate básico.
3	Integrar eventos programados, guardianes automáticos y condiciones de victoria/derrota.
4	Ajustes de balance, reportes, persistencia opcional, pruebas y demo final.

Criterios de evaluación

Criterio	Peso	Puntaje máx.
POO (herencia/polimorfismo/encapsulación)	40 %	8.0
Sistemas del mundo (eventos y guardianes)	20 %	4.0
Arquitectura de datos y flujo de turnos	15 %	3.0
Sobrecarga de operadores y utilidades	10 %	2.0
Experiencia de usuario y reportes	10 %	2.0
Calidad, organización y compilación	5 %	1.0
Total	100 %	20.0

Entregables

- Código fuente organizado en **src/** con archivos **.h/.cpp** y scripts de construcción.
- Documento técnico (máx. 3 páginas) que describa arquitectura, sistemas de eventos, balance y casos de prueba.
- Registro de una partida completa (log o replay) con comentarios clave.
- Presentación (5–7 minutos) con demo, decisiones de diseño y oportunidades de extensión.