

UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA



Curso:

Programación 2

PROYECTO: Rival Frontiers

Integrantes:

Barona Rodriguez, Eduardo Francisco

Apellido, Nombre

Roman Rodriguez, Jianpierre Patrick

Apellido, Nombre

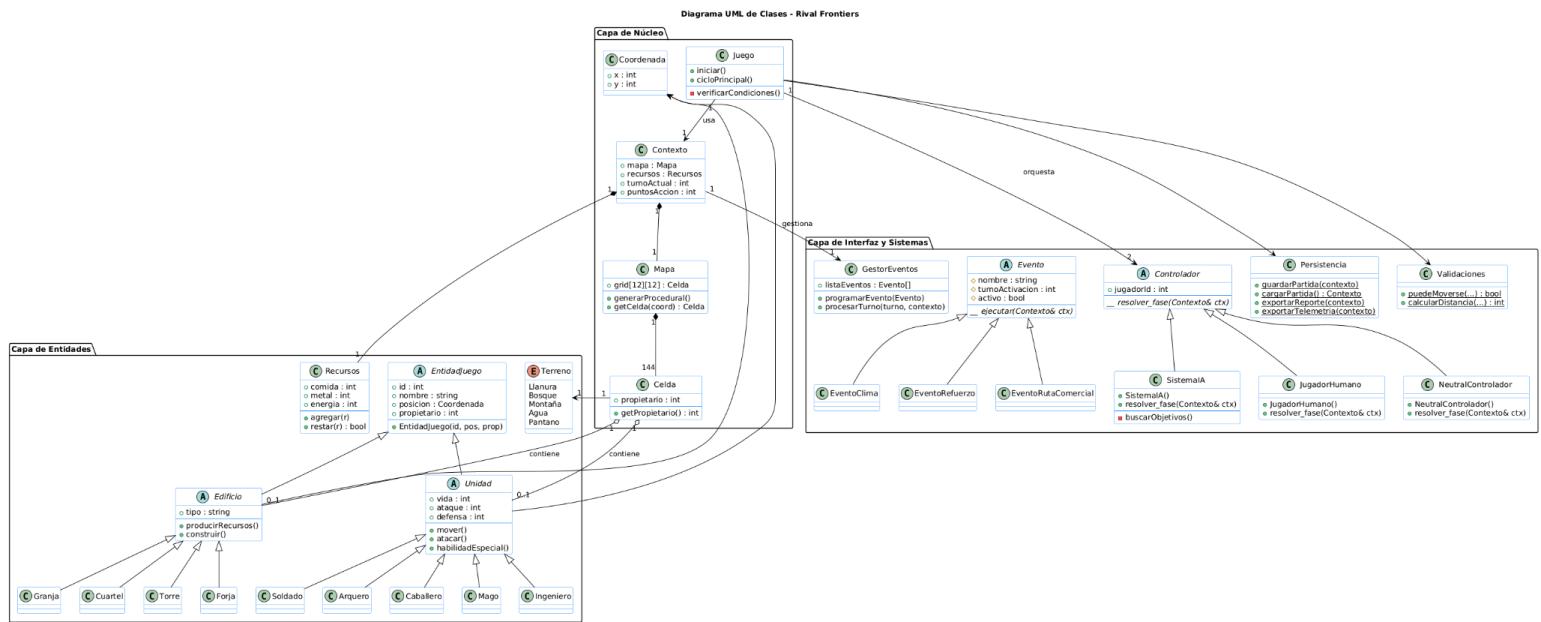
Apellido, Nombre

Profesor:

Rivas Medina, Rubén

Lima, Barranco

Arquitectura



1. Núcleo del Juego

- **Juego:** Controla el ciclo de turnos, fases y victoria.
 - **Contexto:** Guarda el estado global (recursos, mapa, turno, eventos).
 - **Mapa:** Cuadrícula 12×12 con terreno generado y entidades colocadas.
 - **Celda:** Elemento del mapa que puede tener terreno, unidad, edificio y propietario.

2. Entidades del Juego

<p>Unidades (5 tipos) Soldado, Arquero, Caballero, Mago, Ingeniero</p> <ul style="list-style-type: none"> • Tienen atributos propios (vida, ataque, defensa). • Poseen habilidades especiales con efectos temporales o en área. 	<p>Edificios (4 tipos) Granja, Cuartel, Torre, Forja</p> <ul style="list-style-type: none"> • Producen recursos cada turno. • Pueden ser neutrales y capturados.
<p>Sistemas de Soporte</p> <ul style="list-style-type: none"> • Recursos: Economía con Comida, Metal y Energía. • Combate: Ataques influenciados por terreno. • Eventos: Acciones automáticas por turno. • Persistencia: Guardado y carga de partidas, más reportes. 	<p>Terrenos (5 tipos) Llanura, Bosque, Montaña, Agua, Pantano</p> <ul style="list-style-type: none"> • Modifican movimiento y defensa.

Sistemas de Eventos

3 Arquitectura de Eventos

El sistema implementa un **gestor de eventos temporales** que permite programar y ejecutar eventos en turnos específicos:

```
C/C++
class Evento {
protected:
    string nombre;
    int turnoActivacion;
    bool activo;
public:
    virtual void ejecutar(Contexto& ctx) = 0;
};
```

Evento	Descripción	Efecto
EventoRefuerzo	Llegada de unidades aliadas	Spawning de N unidades en posiciones específicas
EventoClima	Condiciones meteorológicas	Modificadores temporales de movimiento/combate
EventoRutaComercial	Bonificación económica	+N recursos para jugador especificado

2.3 Flujo de Procesamiento

1. Programación: `gestorEventos.programarEvento(new EventoRefuerzo(5, 1, 3))`
2. Cada Turno: `gestorEventos.procesarTurno(turnoActual, contexto)`
3. Activación: `evento->ejecutar(contexto) si turnoActivacion <= turnoActual`
4. Limpieza: `gestorEventos.limpiarEventosCompletados()`

3. BALANCE DEL JUEGO

Sistema Económico:

Recursos iniciales: 18 comida, 12 metal, 8 energía.

Unidades y edificios tienen costos variados según rol; magos e ingenieros requieren energía, construcciones avanzadas requieren más metal.

Balance de Combate:

Cada unidad tiene vida, ataque y defensa diferenciados y habilidades especiales (ej.: arquero daño x1.5, mago AoE, caballero daño x2).

El terreno afecta movimiento y defensa: bosque (+2), montaña (+3), agua intransitable.

Condiciones de Victoria:

Ganar con 60% de dominio antes del turno 20; perder por tiempo si no se alcanza; desempate por mayor territorio.

4. CASOS DE PRUEBA

ID	Objetivo	Descripción Resumida	Resultado Esperado
CP-001	Inicialización del juego	Verificar generación del mapa 12x12, bases en (0,0)/(11,11), 3 unidades por jugador y 3–5 edificios neutrales.	Mapa balanceado y correctamente configurado.
CP-002	Movimiento de unidades	Validar movimientos permitidos, bloqueo por agua/unidades enemigas y captura de edificios neutrales.	Solo movimientos válidos aceptados.
CP-003	Sistema de combate	Probar daño: Soldado (ATK 3) vs Arquero (DEF 1). Cálculo con fórmula $\max(\text{ATK} - \text{DEF}, 1)$.	Daño = 2; arquero queda con 5 de vida.
CP-004	Ciclo completo del turno	Verificar transición: recursos → acciones jugador → IA → incremento de turno.	Flujo de turnos correcto.
CP-005	Captura de edificios	Unidad entra en edificio neutral; cambio de propietario y actualización del código. Unidad permanece fuera.	Edificio capturado correctamente.
CP-006	Exportación de reportes	Generación de reporte_partida.txt con secciones completas.	Archivo creado con formato legible.
CP-008	IA básica	IA detecta unidades, expande territorio, se mueve hacia objetivos y recluta si tiene recursos.	IA toma decisiones.