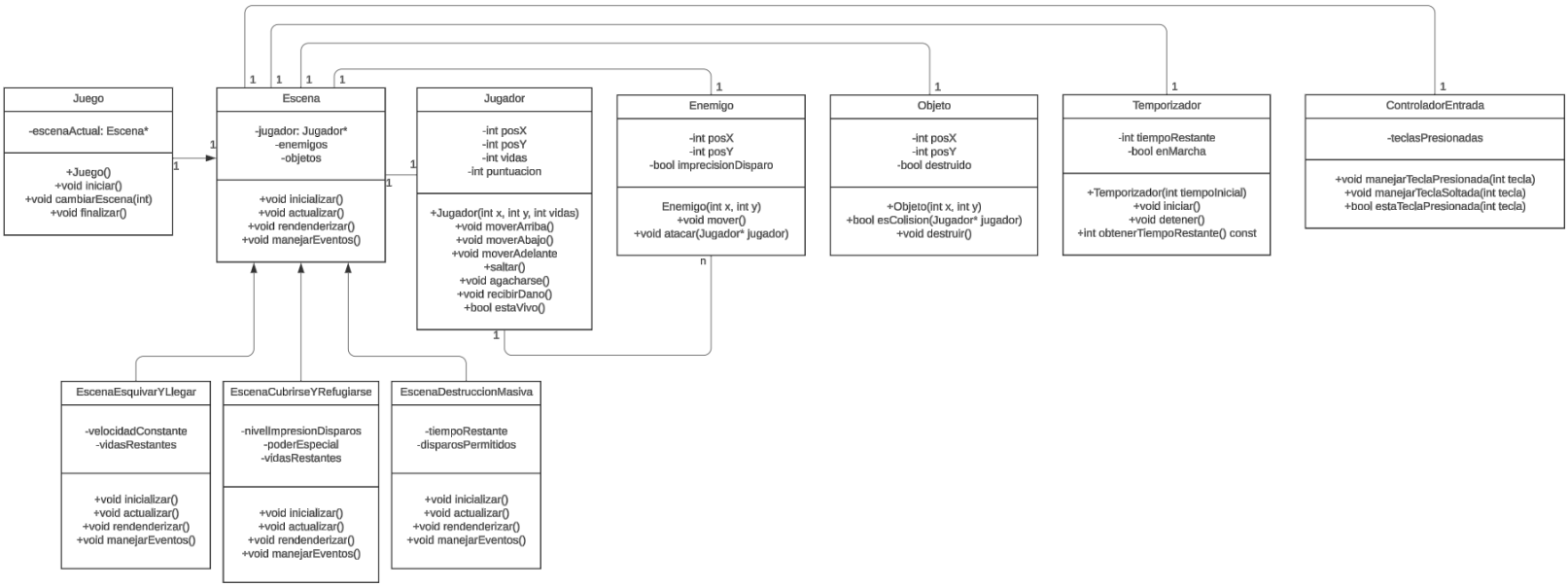
INFORME DE PROYECTO FINAL

Sergio Andrés Berbesí Builes – Salome Bermúdez Macias

a. Análisis del problema y consideraciones para la alternativa de solución propuesta:

- Juego: La clase principal que gestiona el flujo del juego y cambia entre escenas.
- Escena: Una clase abstracta de la cual derivan las diferentes escenas del juego.
- Jugador: Representa al jugador con sus atributos y métodos para moverse y recibir daño.
- Enemigo: Representa a los enemigos con métodos para moverse y atacar al jugador.
- Objeto: Representa los objetos interactivos en las escenas, con métodos para manejar colisiones y destrucción.
- Temporizador: Gestiona el tiempo en las escenas.
- ControladorEntrada: Maneja las entradas del usuario (teclado).
- EscenaEsquivarYLlegar, EscenaCubrirseYRefugiarse, EscenaDestruccionMasiva: Clases derivadas de Escena, cada una con atributos y métodos específicos para su respectiva mecánica de juego.

b. Diagrama de clases:



c. Algoritmos implementados:

```
class Juego {
public:
  Juego();
  void iniciar();
  void cambiarEscena(int escena);
  void finalizar();
private:
  Escena* escenaActual;
};

class Escena {
public:
  virtual void inicializar() = 0;
  virtual void actualizar() = 0;
  virtual void renderizar() = 0;
  virtual void manejarEventos() = 0;
private:
  Jugador* jugador;
  Enemigo* enemigos;
  Objeto* objetos;
};

class Jugador {
public:
  Jugador(int x, int y, int vidas);
  void moverArriba();
  void moverAbajo();
  void moverAdelante();
  void saltar();
  void agacharse();
  void recibirDano();
  bool estaVivo() const;
private:
  int posX, posY;
  int vidas;
  int puntuacion;
};

class Enemigo {
public:
  Enemigo(int x, int y);
  void mover();
  void atacar(Jugador* jugador);
private:
  int posX, posY;
```

```
class EscenaEsquivarYLlegar : public Escena {
public:
  void inicializar() override;
  void actualizar() override;
  void renderizar() override;
  void manejarEventos() override;
private:
  int velocidadConstante;
  int vidasRestantes;
};

class EscenaCubrirseYRefugiarse : public Escena {
public:
  void inicializar() override;
  void actualizar() override;
  void renderizar() override;
  void manejarEventos() override;
private:
  int nivelImprecisionDisparos;
  int poderEspecial;
  int vidasRestantes;
};

class EscenaDestruccionMasiva : public Escena {
public:
  void inicializar() override;
  void actualizar() override;
  void renderizar() override;
  void manejarEventos() override;
private:
  int tiempoRestante;
  int disparosPermitidos;
};

class Temporizador {
public:
  Temporizador(int tiempoInicial);
  void iniciar();
  void detener();
  int obtenerTiempoRestante() const;
private:
  int tiempoRestante;
  bool enMarcha;
};
```

```
    bool imprecisionDisparo;
};

class Objeto {
public:
    Objeto(int x, int y);
    bool esColision(Jugador* jugador) const;
    void destruir();
private:
    int posX, posY;
    bool destruido;
};

class ControladorEntrada {
public:
    static void manejarTeclaPresionada(int tecla);
    static void manejarTeclaSoltada(int tecla);
};
```