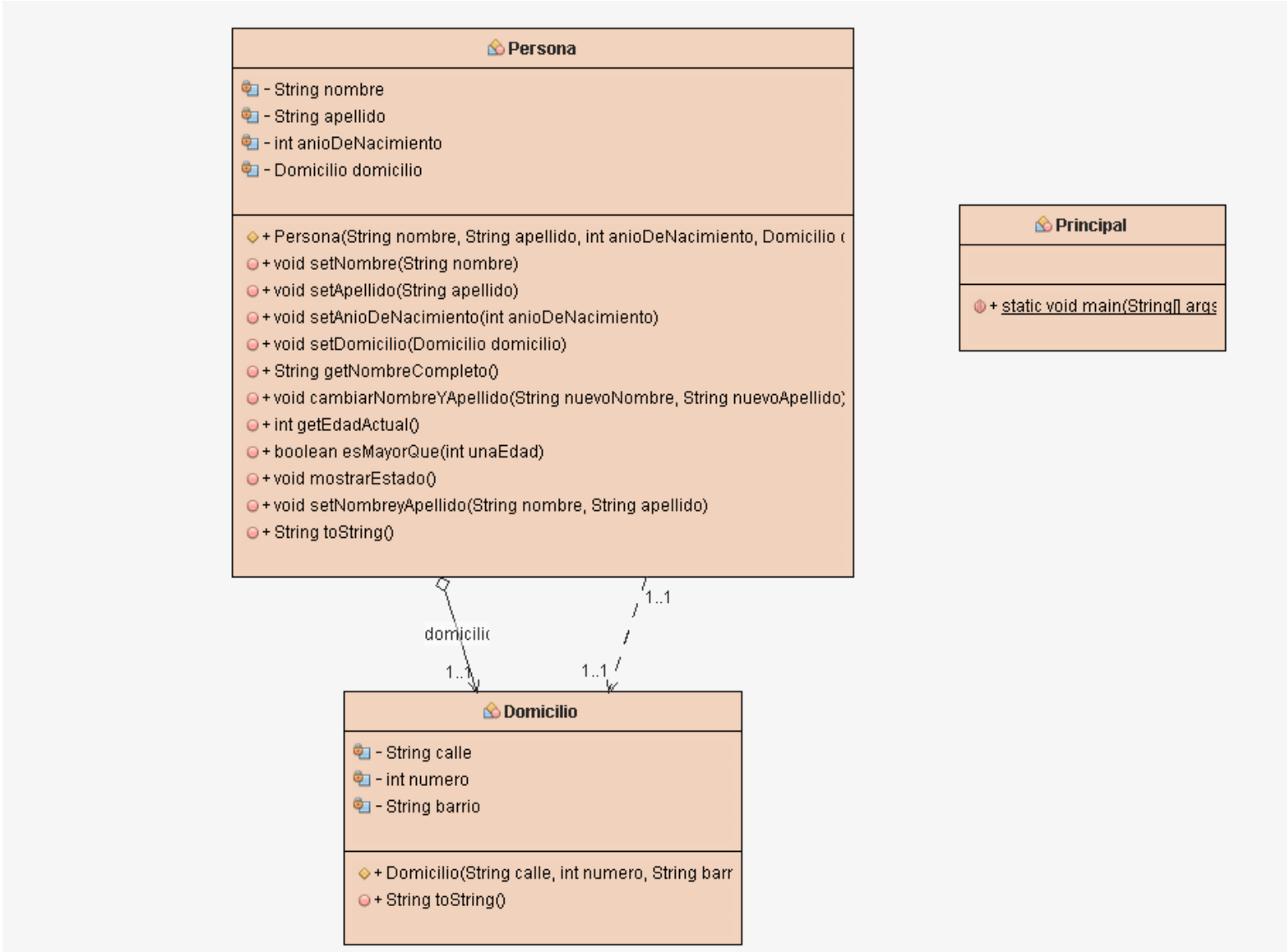
✅

Ejercicio 9

Refactoreá la clase Persona del ejercicio 1), cambiando el año de nacimiento por su fecha de nacimiento y agregando el atributo domicilio, que contenga calle, altura y barrio.

💡 **Refactorizar** es el proceso de modificar el código existente para mejorarlo, sin cambiar su comportamiento externo. En otras palabras, se trata de reorganizar y simplificar el código sin alterar su funcionamiento. El objetivo de la refactorización es mejorar la calidad del código, haciéndolo más legible, mantenible y escalable, y reducir la cantidad de errores y bugs. La refactorización se realiza para mejorar el diseño, la estructura y la eficiencia del código, eliminando duplicidades y reduciendo la complejidad. En resumen, refactorizar es mejorar el código existente para hacerlo más fácil de entender y mantener en el futuro.

Diagrama UML



Proyecto Java:

```
package ejercicio_09;

import java.time.LocalDate;

public class Persona {
    private String nombre;
    private String apellido;
    private int anioDeNacimiento;
    private Domicilio domicilio;


    // CONSTRUCTOR
    public Persona(String nombre, String apellido, int anioDeNacimiento, Domicilio dom) {
        setNombre(nombre);
        setApellido(apellido);
        setAnioDeNacimiento(anioDeNacimiento);
        setDomicilio(dom);
    }

    public void setNombre(String nombre) {
        if(nombre != null && nombre.length() >= 3) {
```

```
            this.nombre = nombre;
        }
    }

    public void setApellido(String apellido) {
        if(apellido != null && apellido.length() >= 2) {
            this.apellido = apellido;
        }
    }

    public void setAnioDeNacimiento(int anioDeNacimiento) {
        if (anioDeNacimiento > 0) {
             this.anioDeNacimiento = anioDeNacimiento;
        }
    }

    public void setDomicilio(Domicilio domicilio) {
        this.domicilio = domicilio;
    }

    public String getNombreCompleto (){
        return nombre + " " + apellido;
    }

    public void cambiarNombreYApellido(String nuevoNombre, String nuevoApellido) {
        setNombre(nuevoNombre);
        setApellido(nuevoApellido);
    }

    public int getEdadActual() {
        // Hubo que importar la clase 'LocalDate'
        int anioActual = LocalDate.now().getYear();
        int edad = anioActual - anioDeNacimiento;
        return edad;
    }

    public boolean esMayorQue(int unaEdad){
        return getEdadActual() >= unaEdad;
    }

    public void mostrarEstado () {
        System.out.println( toString() );
    }

    public void setNombreyApellido(String nombre,String apellido) {
        if(nombre != null && apellido != null){
        this.nombre = nombre;
        this.apellido = apellido;
        }
    }

    @Override
    public String toString() {
        return "Persona{" + "nombre=" + nombre + ", apellido=" + apellido + ", anioDeNacimiento=" + anioDeNacimiento + ", domicilio=" + domicilio + '}';
    }

}
```

```
package ejercicio_09;

public class Domicilio {
    private String calle;
    private int numero;
    private String barrio;

    public Domicilio(String calle, int numero, String barrio) {
        this.calle = calle;
        this.numero = numero;
        this.barrio = barrio;
    }

    @Override
    public String toString() {
        return "Domicilio{" + "calle=" + calle + ", numero=" + numero + ", barrio=" + barrio + '}';
    }

}
```

```
package ejercicio_09;

public class Principal {

    public static void main(String[] args) {
        Domicilio dom = new Domicilio("Arcos",2702,"Belgrano");
        Persona p = new Persona("Francisco","Chiminelli",2001,dom);
        p.mostrarEstado();
    }
}
```

👩‍💻 ***Explicación del programa:***

Esta segunda versión consiste en 3 archivos Java, `Persona.java`, `Domicilio.java` y `Principal.java`.

La clase `Persona` ahora tiene un constructor que recibe parámetros y se encarga de validarlos mediante los métodos `set`, además de tener un nuevo atributo `Domicilio`.

Además, se ha modificado el método `mostrarEstado` para llamar al método `toString` en lugar de imprimir los atributos manualmente.

La clase `Domicilio` es una nueva clase que modela el domicilio de una persona.

En el archivo `Principal.java` se ha modificado la creación de un objeto `Persona` para pasar un objeto `Domicilio` como argumento al constructor.