

# Programación y Diseño Orientado a Objetos. ETSIIT. UGR

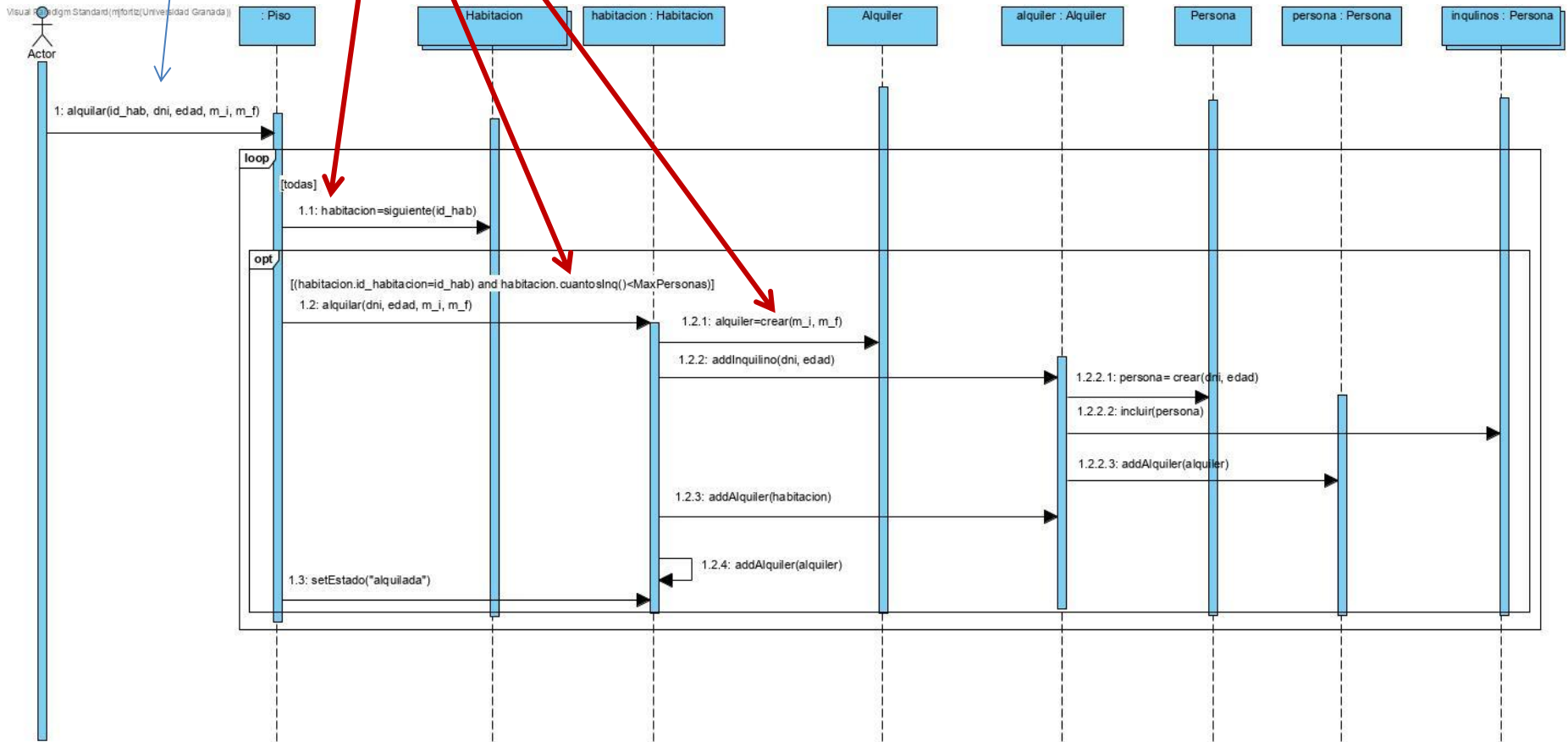
Paso de diagrama de secuencia a  
código Java y Ruby

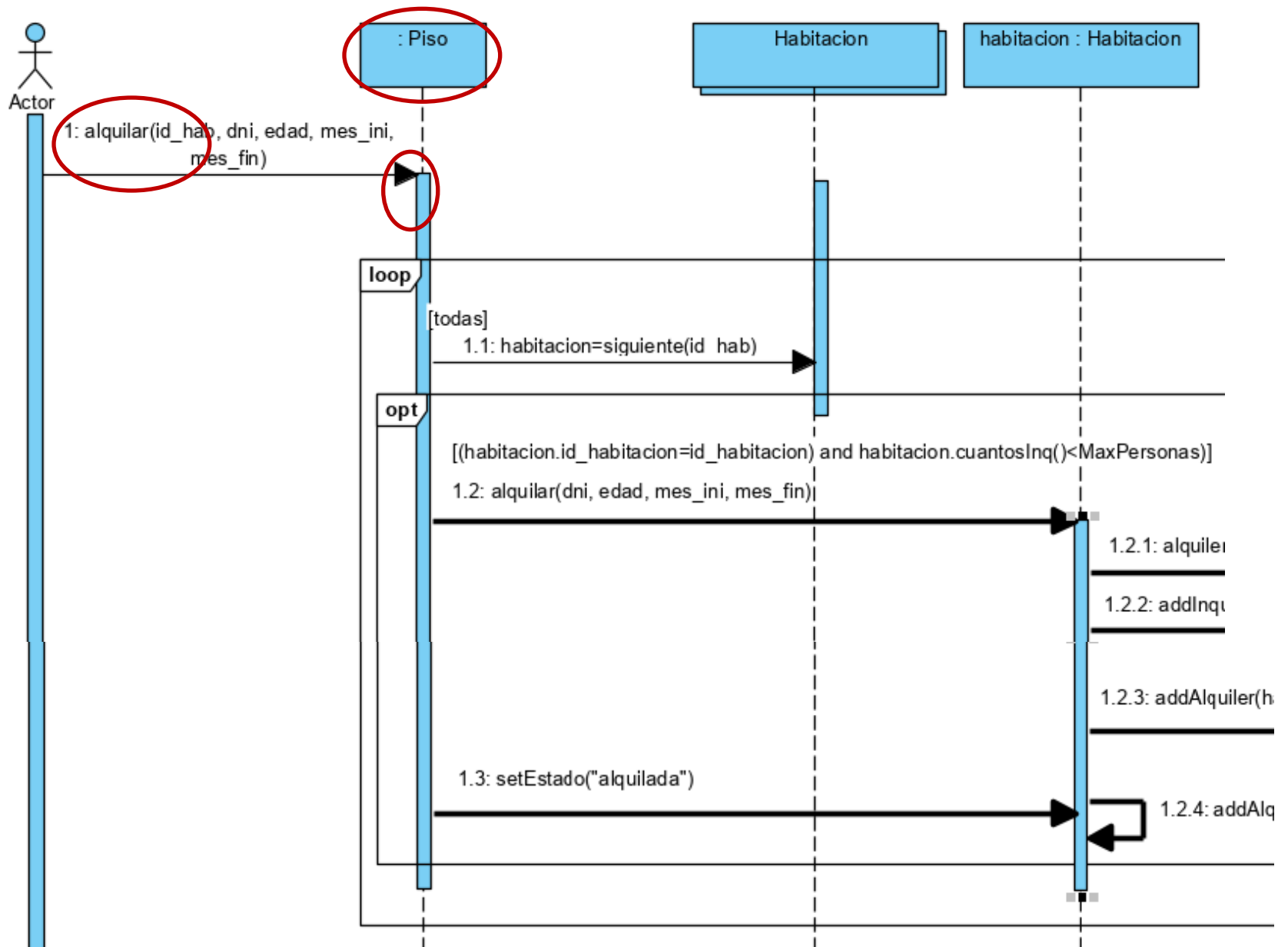
María José Rodríguez Fórtiz. LSI.

Método a implementar

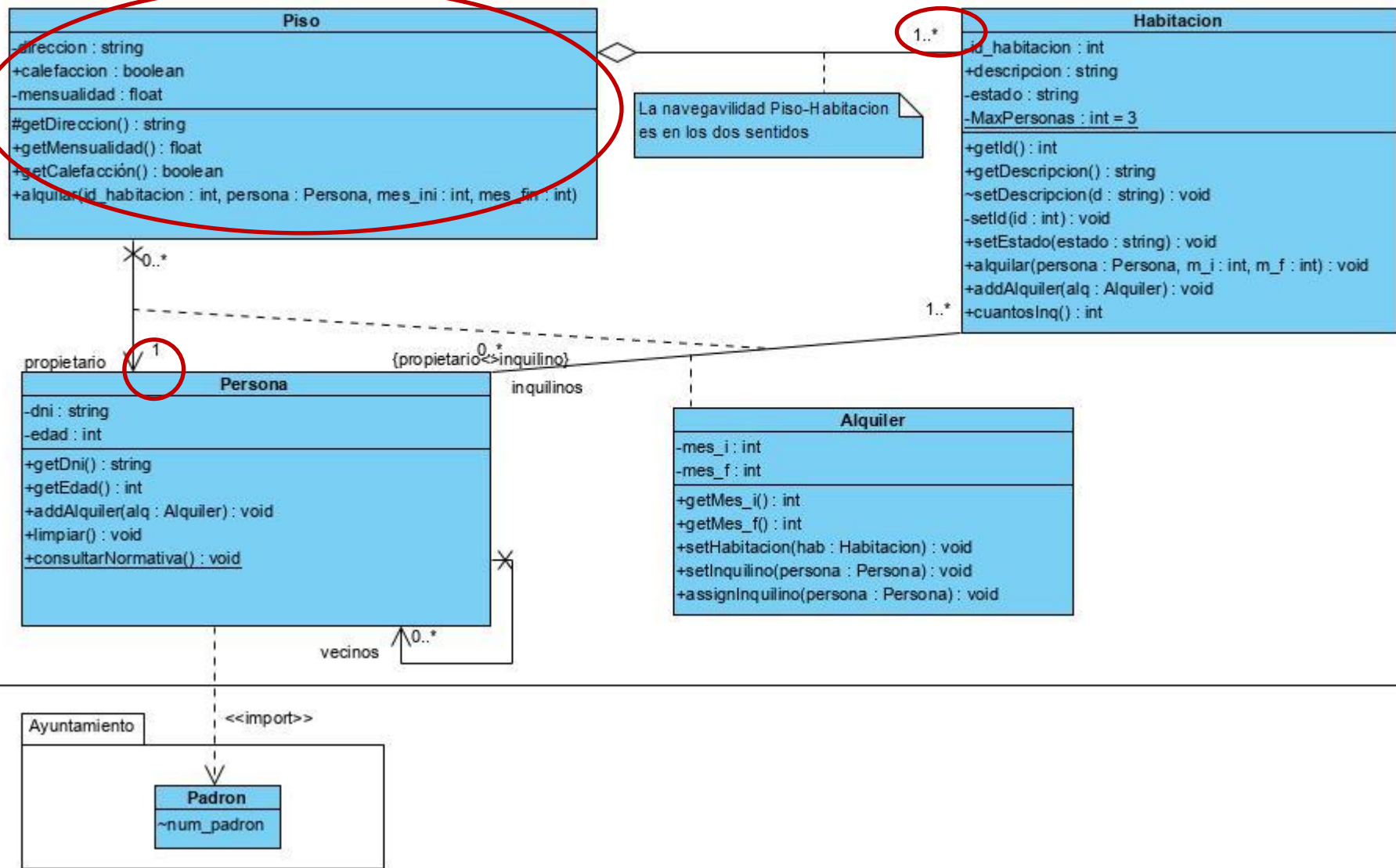
Envío de mensajes

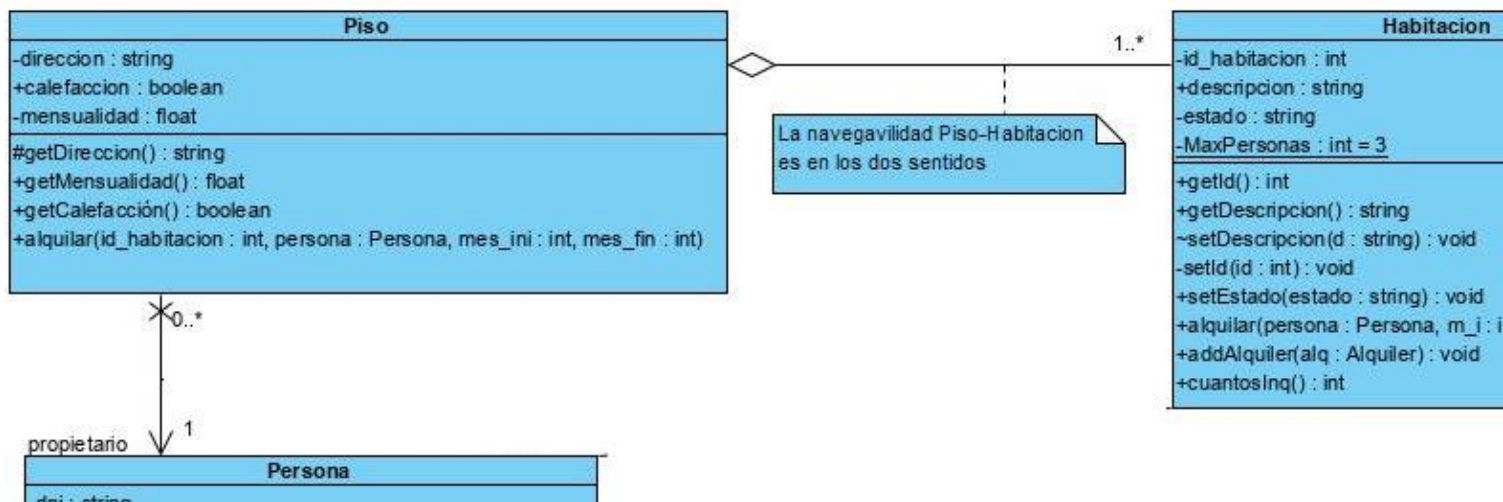
Líneas de vida





## Arrendamiento





## Ruby

```

module Arrendamiento
class Piso{
  #constructor, inicializa atributos de instancia
  def initialize(d, c, m, p)
    #básicos
    @direccion=d
    @calefaccion=c
    @mensualidad=m
    #de referencia
    @habitaciones=Array.new
    @propietario=nil
  end

  #método a implementar con diagrama de secuencia
  def alquilar(id_habitacion, persona, mes_ini, mes_fin)
    ...
  end
end
end

```

## Java

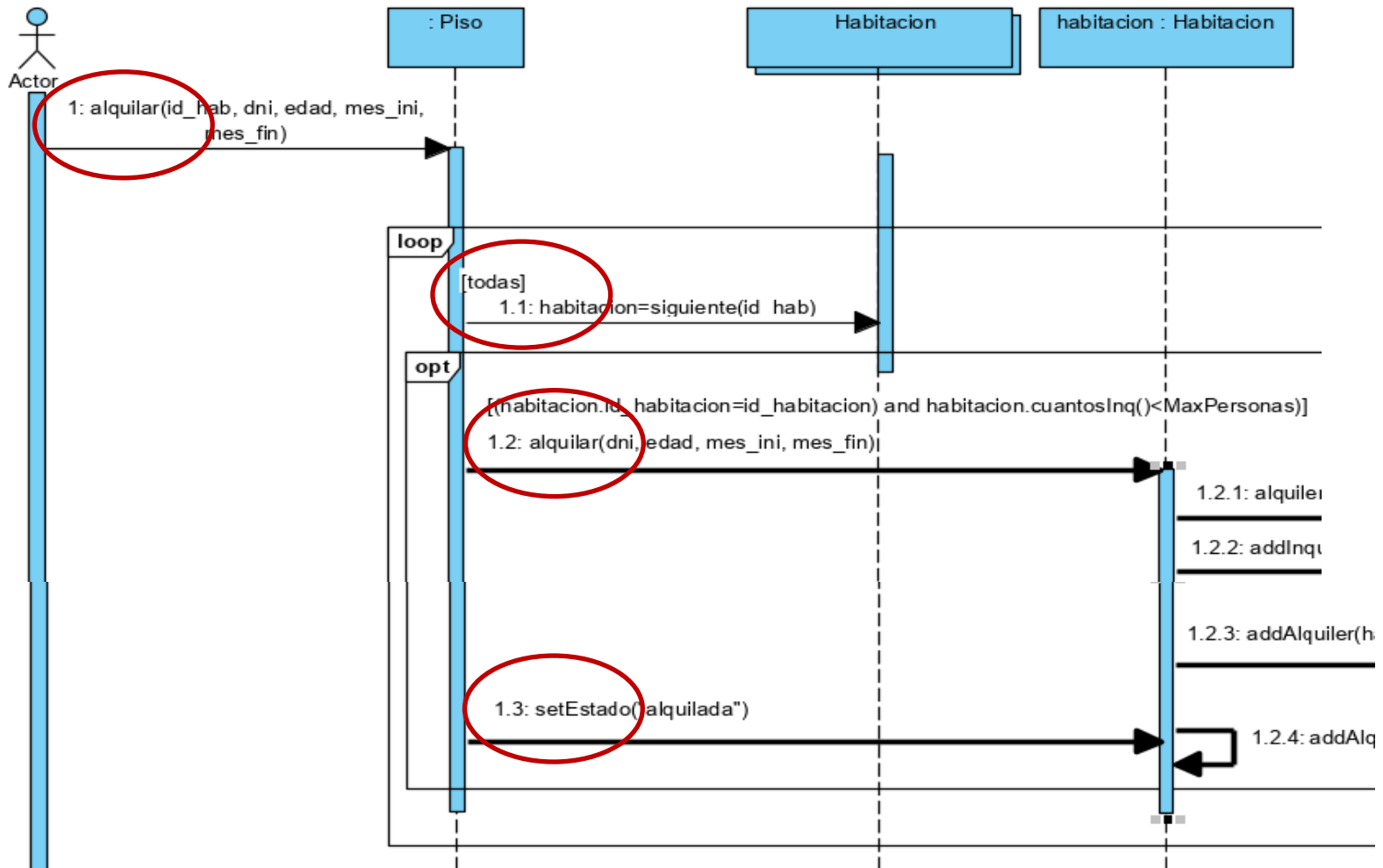
```

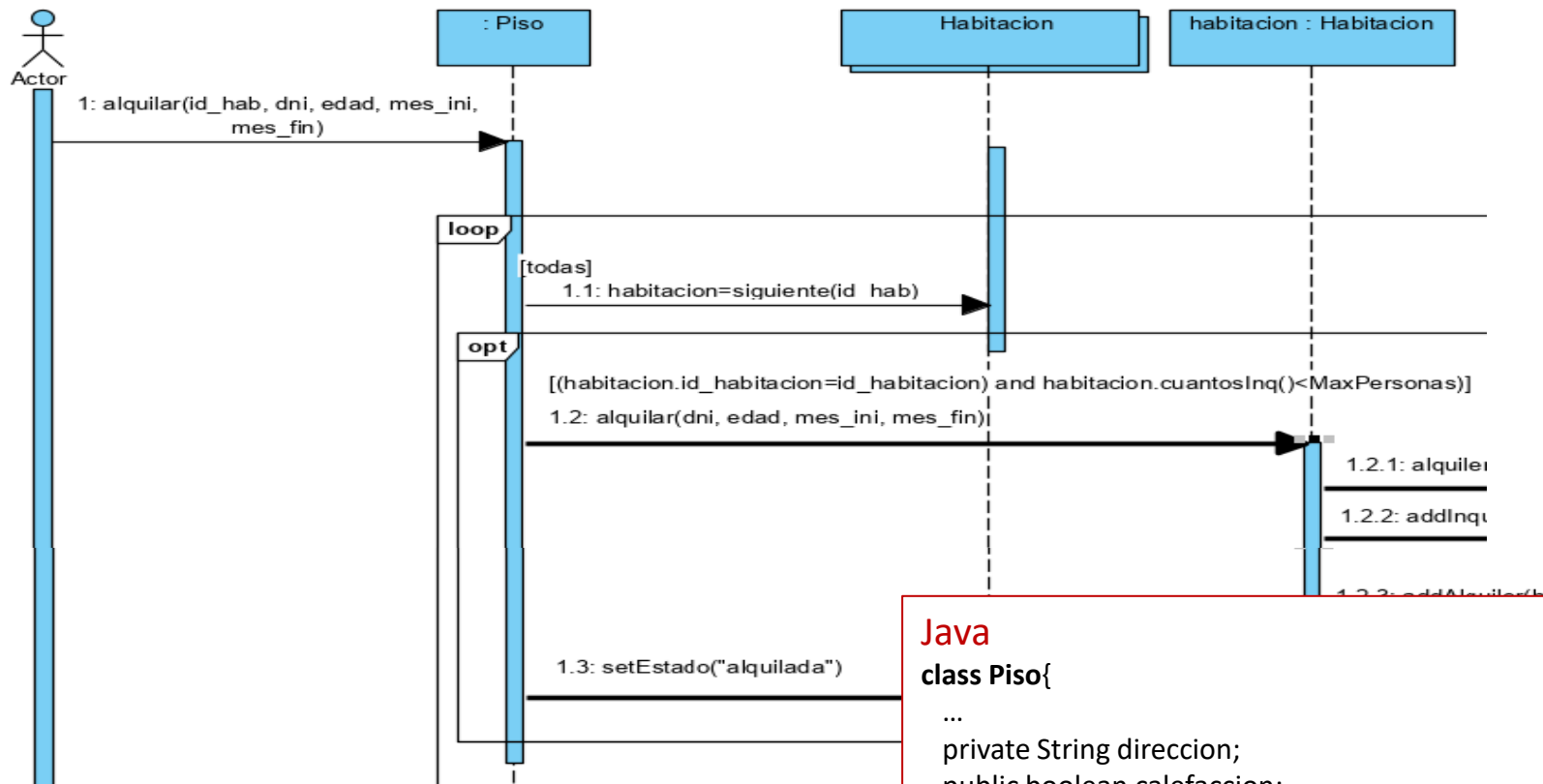
package Arrendamiento;
class Piso{
  //atributos de instancia básicos
  private String direccion;
  public boolean calefaccion;
  private float mensualidad;
  //atributos de instancia de referencia
  private ArrayList<Habitacion> habitaciones;
  private Persona propietario;
  //constructor, inicializa atributos de instancia
  Piso(String d, boolean c, float m, Persona p){
    direccion=d;
    calefaccion=c;
    mensualidad=m;
    habitaciones=new ArrayList<>();
    propietario=null; }

  //método a implementar con diagrama de secuencia
  public void alquilar(int id_habitacion, Persona persona,
    int mes_ini, int mes_in){ ...}

  ... }

```





## Ruby

class Piso

...

**def alquilar(id\_habitacion, persona, mes\_ini, mes\_fin)**

(1.1) for habitacion in @habitaciones

(opt) if ((habitacion.id\_habitacion == id\_habitacion) and  
(habitacion.cuantosInq<Habitacion.get\_MaxPersonas))

(1.2) habitacion.alquilar(dni, edad, mes\_ini, mes\_fin)

(1.3) habitacion.setEstado('alquilada')

end

end

**end**

end

## Java

**class Piso{**

...

private String direccion;

public boolean calefaccion;

private float mensualidad;

private ArrayList<Habitacion> **habitaciones;**

private Persona propietario;

**public void alquilar(int id\_habitacion, Persona persona,  
int mes\_i, int mes\_f){**

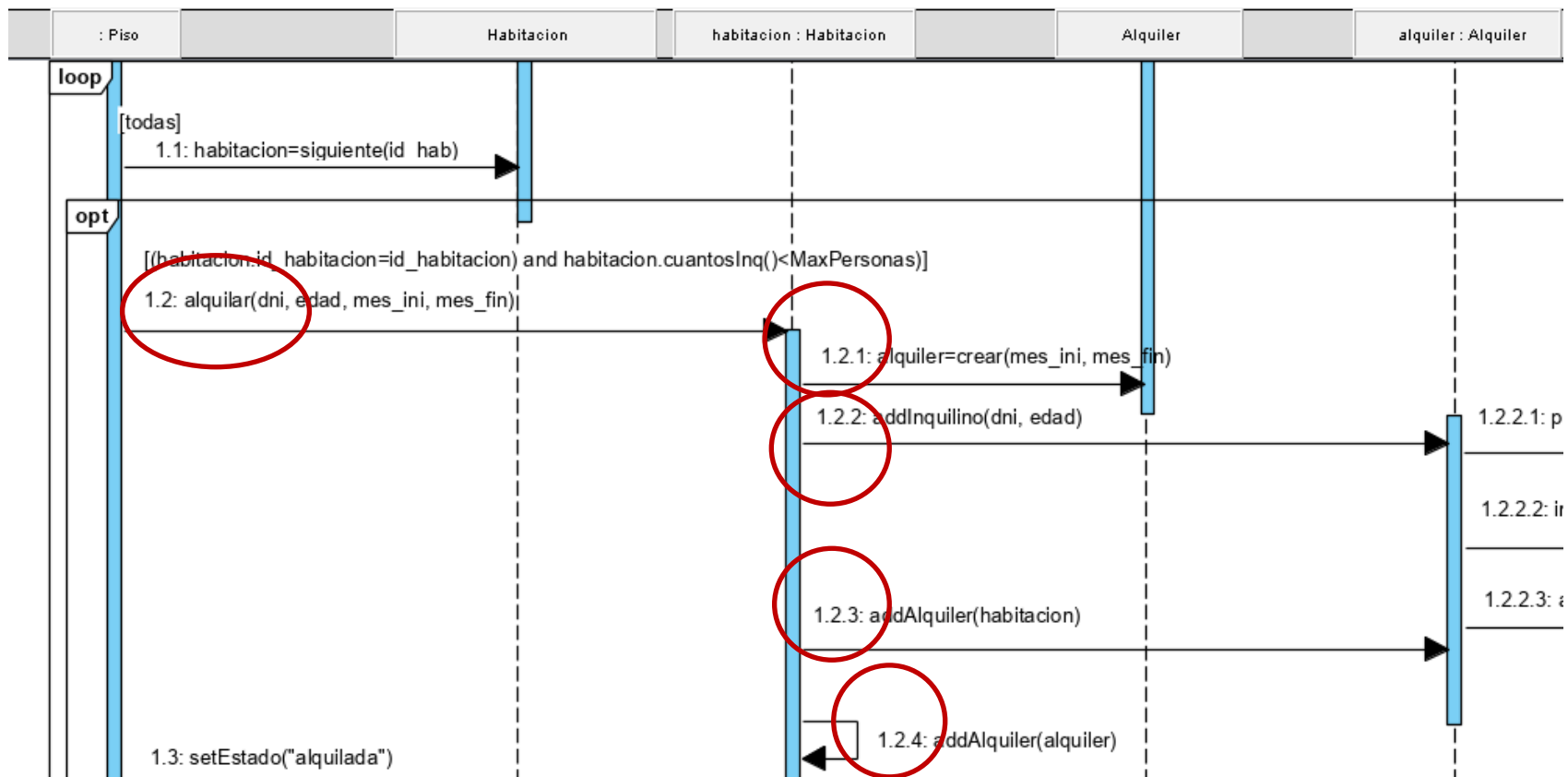
(1.1) for (Habitacion habitacion: habitaciones)

(opt) if ((habitacion.id\_habitacion==id\_habitacion)  
and (habitacion.cuantosInq<Habitacion.MaxPersonas))

(1.2) { habitacion.alquilar(dni, edad, mes\_ini, mes\_fin);

(1.3) habitacion.setEstado("alquilada");}}

//otras formas de 1.1: for (i=0; i++; i<habitaciones.size), while, ...



## Ruby

```

class Habitacion{
  ....
  def alquilar(dni, edad, mes_ini, mes_fin)

    (1.2.1) alquiler= Alquiler.new(mes_ini, mes_fin)
    (1.2.2) alquiler.addInquilino(dni, edad)
    (1.2.3) alquiler.addAlquiler(self)
    (1.2.4) self.addAlquiler(alquiler)

  end
end
  
```

## Java

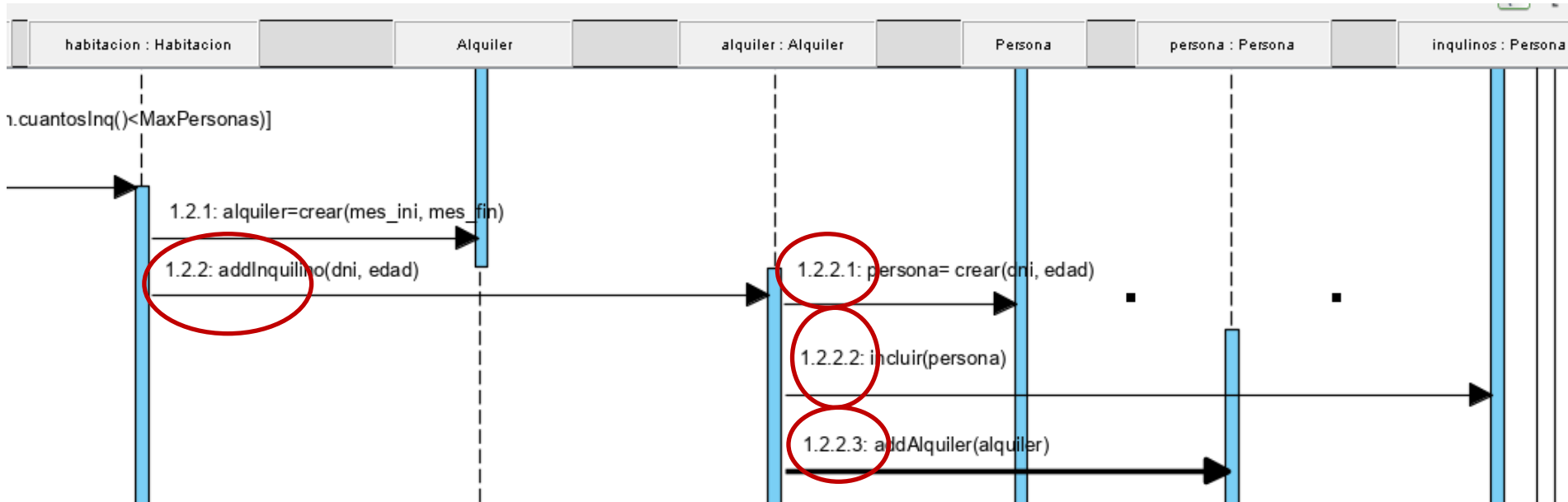
```

class Habitacion{
  public void alquilar(String dni, int edad, int mes_ini, int mes_in){

    (1.2.1) Alquiler alquiler= new Alquiler(mes_ini, mes_fin);
    (1.2.2) alquiler.addInquilino(dni, edad);
    (1.2.3) alquiler.addAlquiler(this);
    (1.2.4) this.addAlquiler(alquiler);

  }
}
  
```





## Ruby

```
class Alquiler{
```

```
  ....
```

```
  def addInquilino(dni, edad)
```

```
    (1.2.2.1) persona= Persona.new(dni,edad)
```

```
    (1.2.2.2) @inquilinos<<persona #atributo de instancia de ref.
```

```
    (1.2.2.3) persona.addAlquiler(self)
```

```
  end
```

```
end
```

## Java

```
class Alquiler{
```

```
  ...
```

```
  public void addInquilino(String dni, int edad){
```

```
    (1.2.2.1) Persona persona= new Persona(dni, edad);
```

```
    (1.2.2.2) inquilinos.add(persona); //atributo de instancia de ref.
```

```
    (1.2.2.3) persona.addAlquiler(this);
```

```
  }
```

```
}
```