UML: Diagramas de Interacción

Dpto. Lenguajes y Sistemas Informáticos
Universidad de Granada

Programación y Diseño Orientado a Objetos

(Curso 2023-2024)

Créditos

- Las siguientes imágenes e ilustraciones son libres y se han obtenido de:
  - Emojis, `https://pixabay.com/images/id-2074153/`
- El resto de imágenes e ilustraciones son de creación propia, al igual que los ejemplos de código

Objetivos

- Saber interpretar los diagramas de secuencia y comunicación
- Saber implementarlos

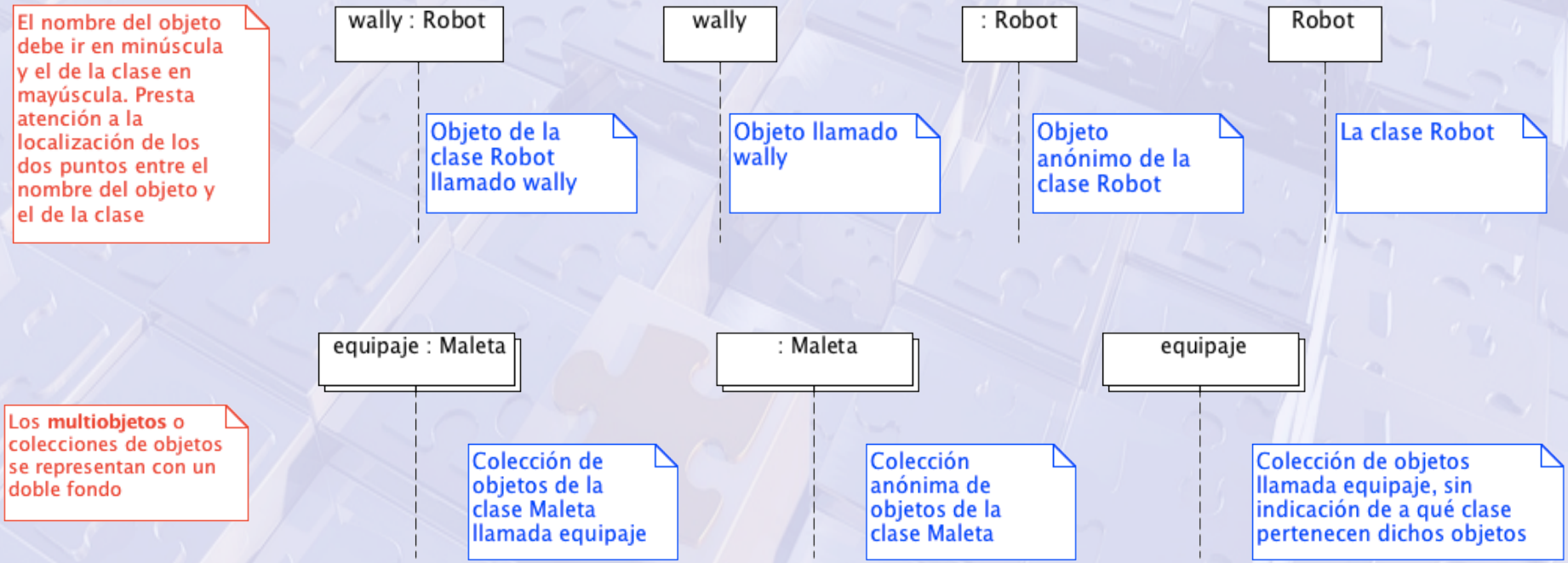
Contenidos

Diagramas de interacción

- Su propósito es mostrar el comportamiento del sistema a través de las interacciones entre los elementos del modelo
- Hay dos tipos básicos:
  - **Diagramas de secuencia**: Enfatizan la secuencia temporal de los mensajes enviados entre objetos
  - **Diagramas de colaboración**: Enfatizan la relación entre los objetos receptores y emisores de los mensajes
- Elementos:
  - Participantes: Objetos y clases que forman parte de la interacción
  - Mensajes: El flujo y su secuencia entre los participantes

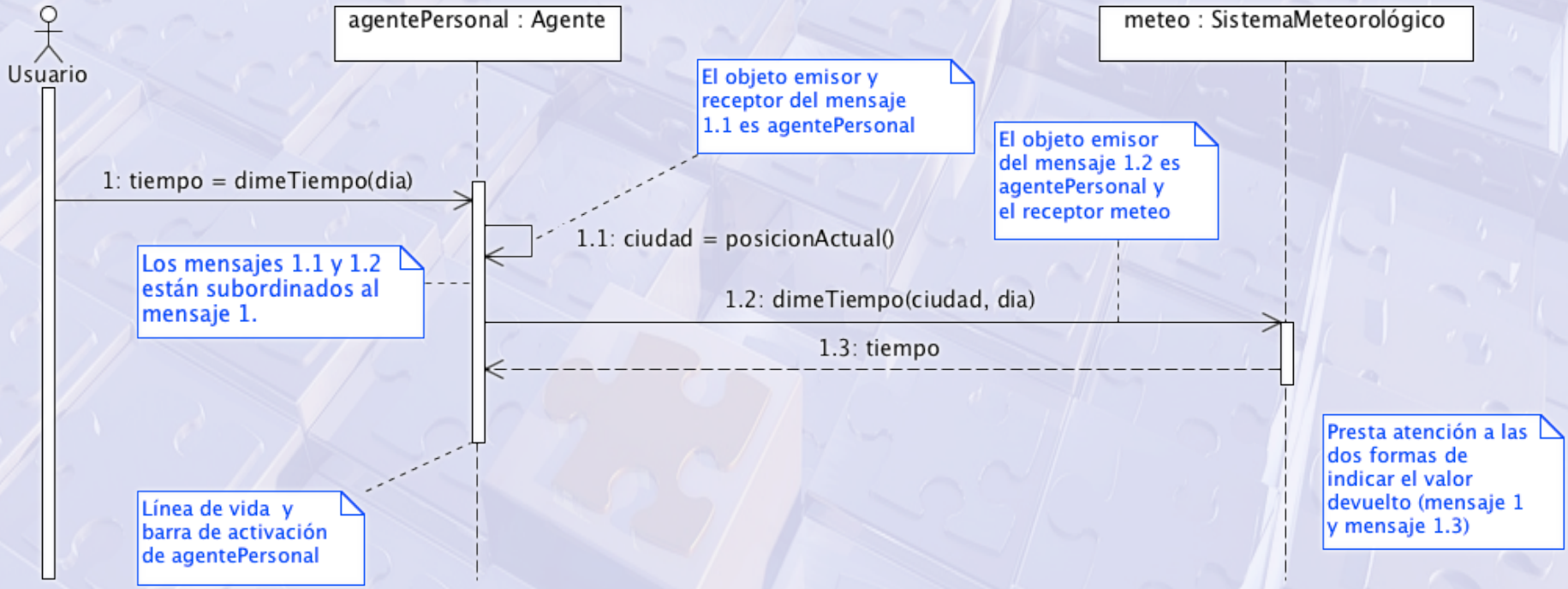
Diagramas de secuencia

- Los **participantes** se muestran en una caja



Diagramas de secuencia

- **Mensajes**: Emisor y Receptor



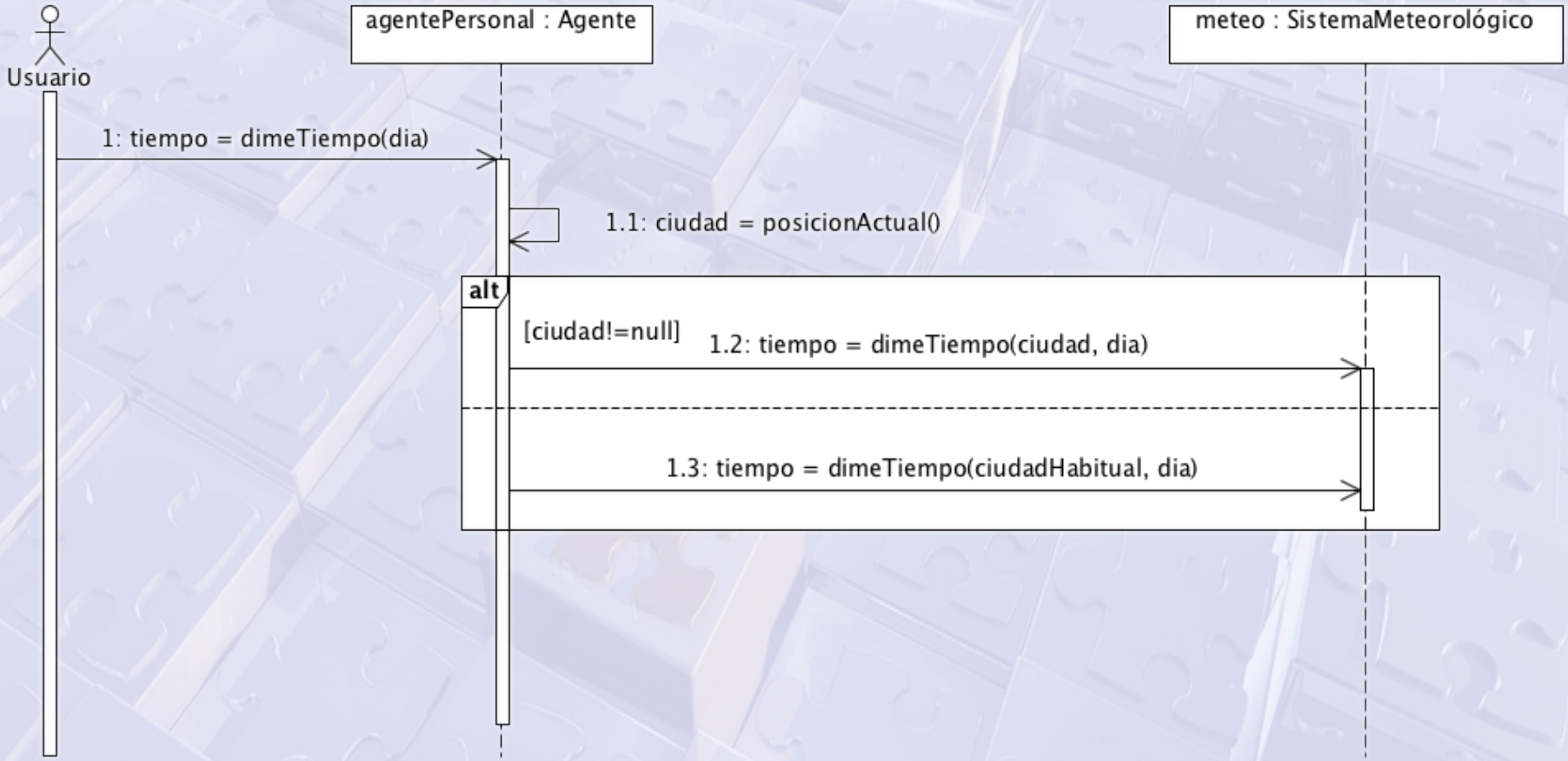
Diagramas de secuencia

Ruby: Implementación del diagrama anterior

```ruby
class Agente

...

  def dimeTiempo (dia)
    # No se indica receptor, es el propio objeto
    ciudad = posicionActual

    # ¿Cómo sabemos que meteo es un atributo?
    @meteo.dimeTiempo (ciudad, dia)

    # Devuelve el resultado del último paso de mensaje
  end

...

end
```

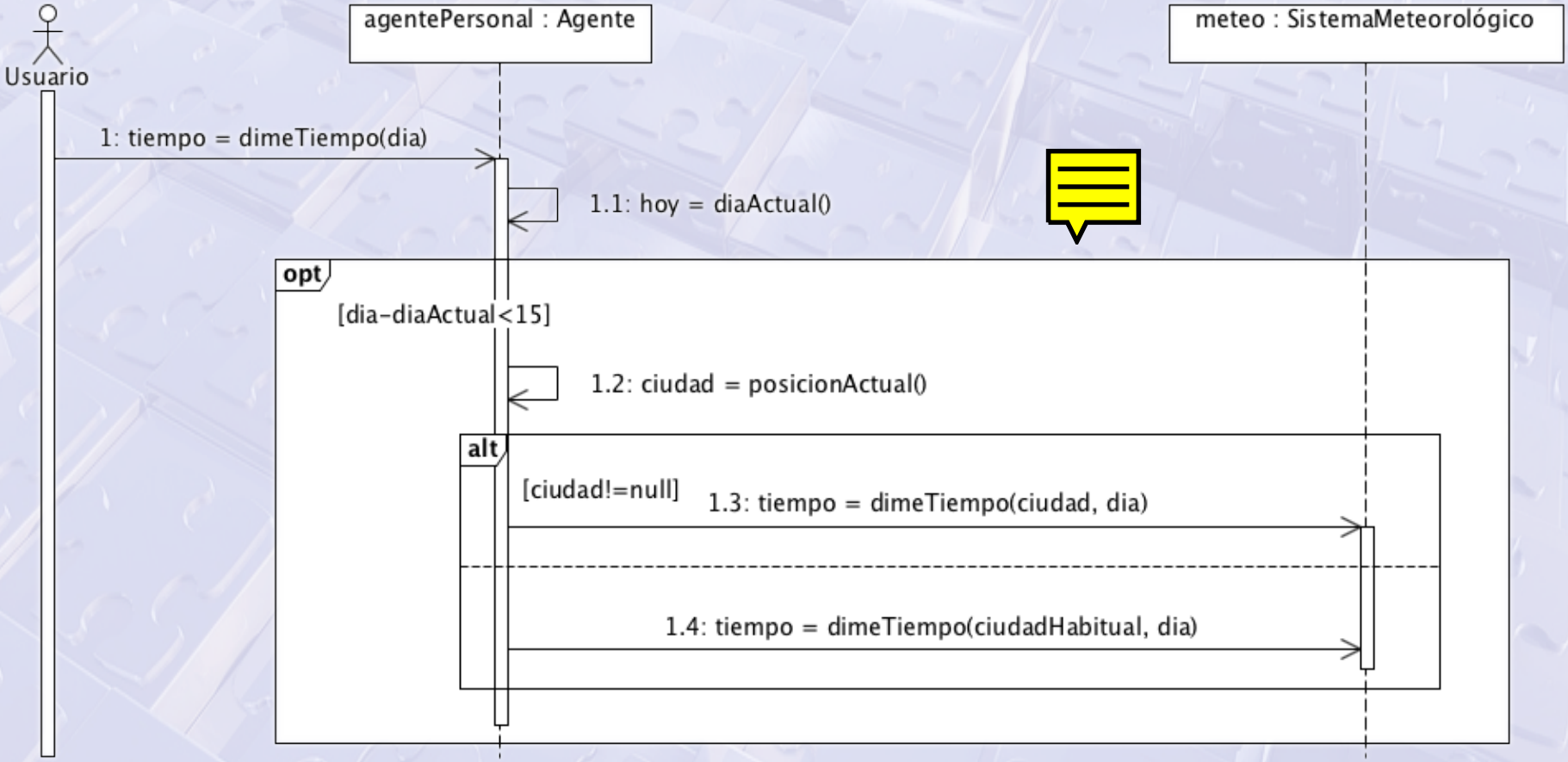
Diagramas de secuencia

- Fragmentos: **Condicionales**



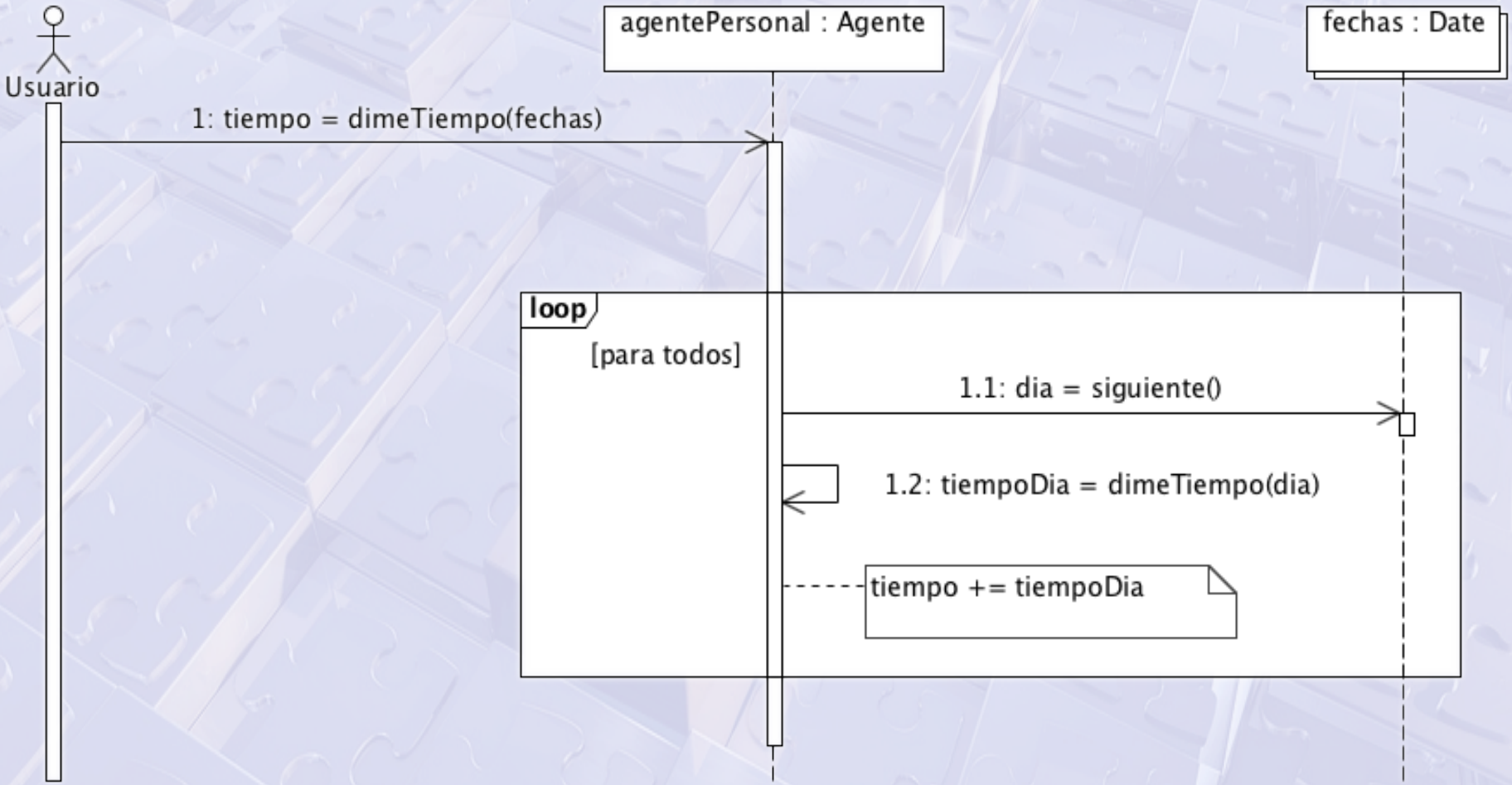
Diagramas de secuencia

- Fragmentos: **Condicionales**



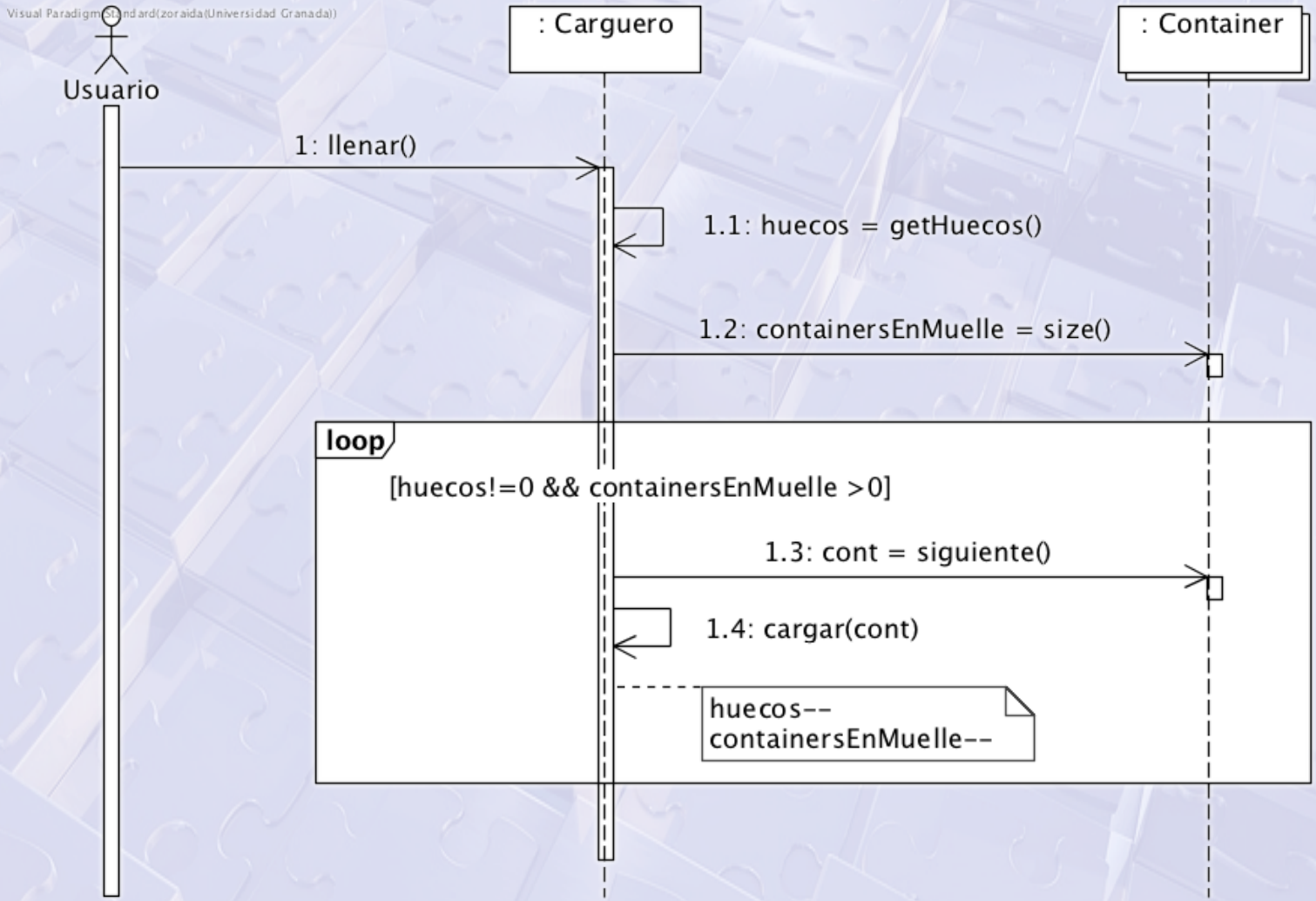
Diagramas de secuencia

- Fragmentos: **Bucles**



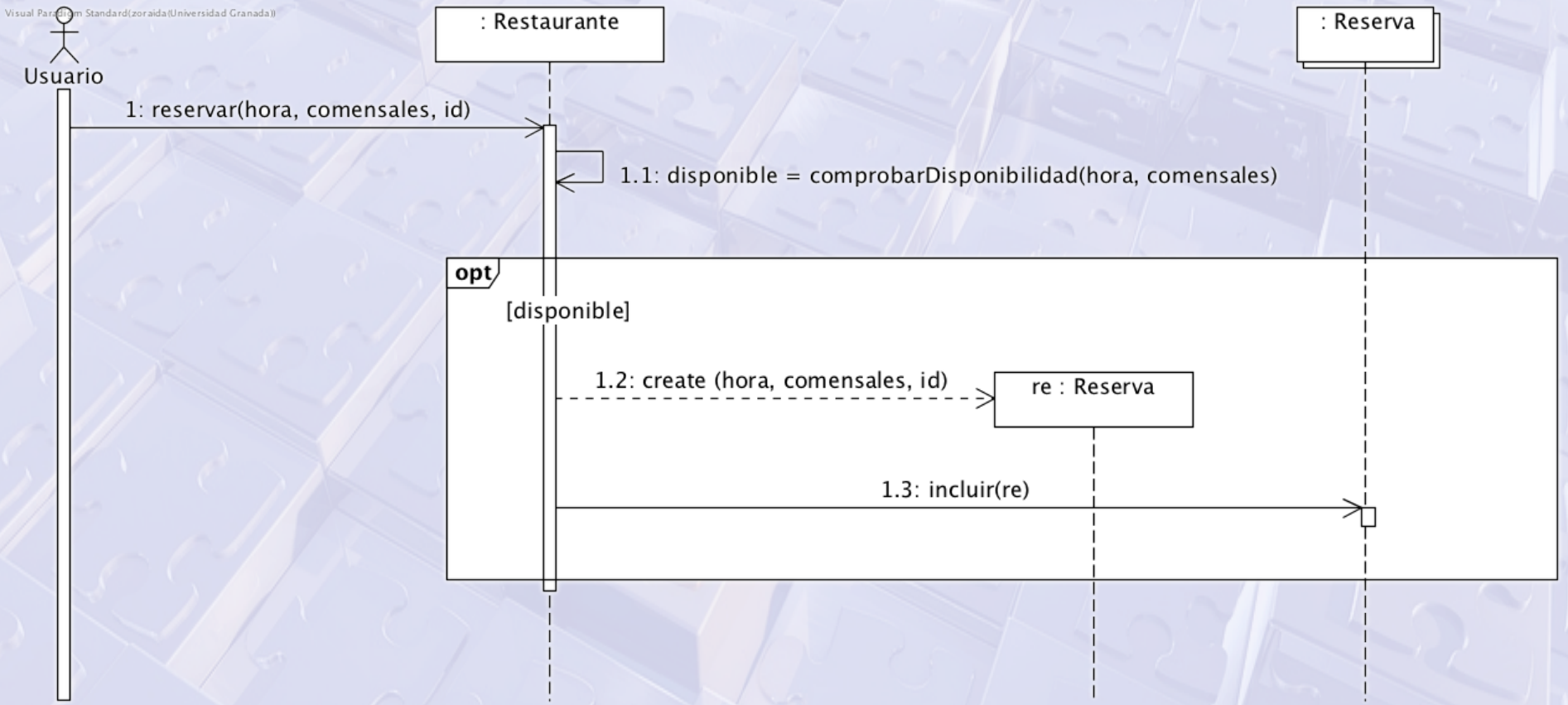
Diagramas de secuencia

- Fragmentos: **Bucles**



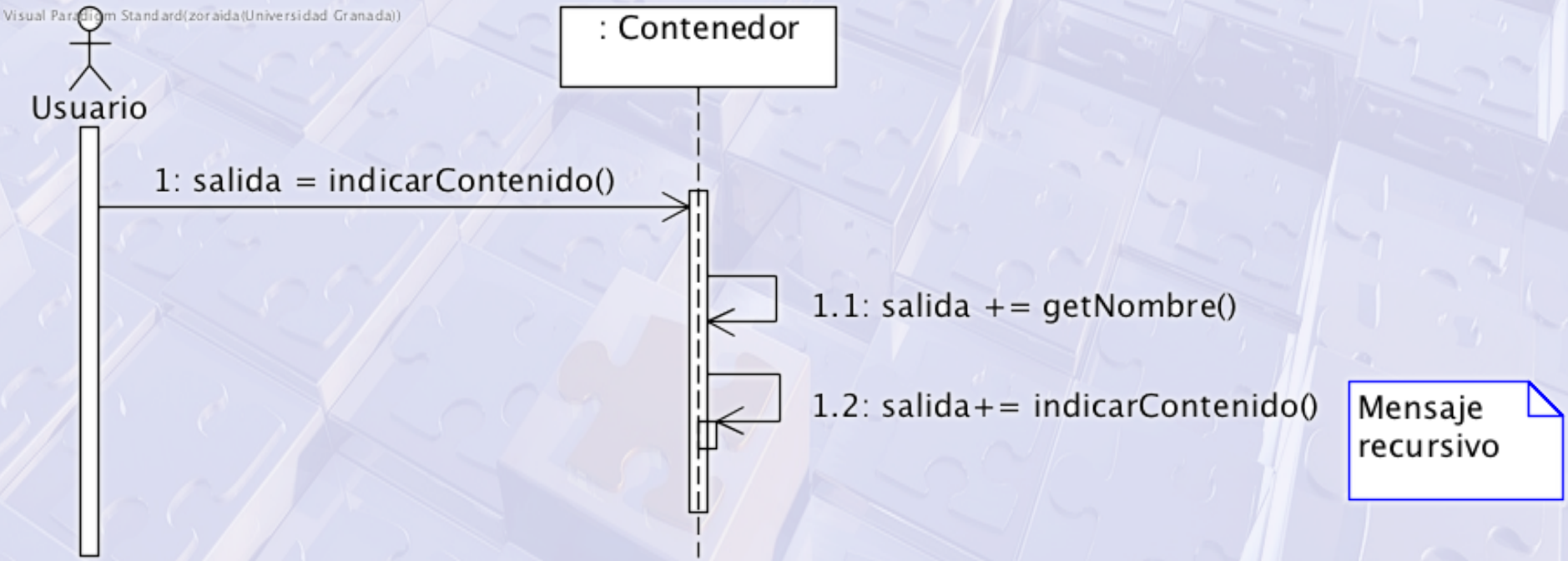
Diagramas de secuencia

- **Creación de instancias**



Diagramas de secuencia

- **Recursividad**



Diagramas de comunicación

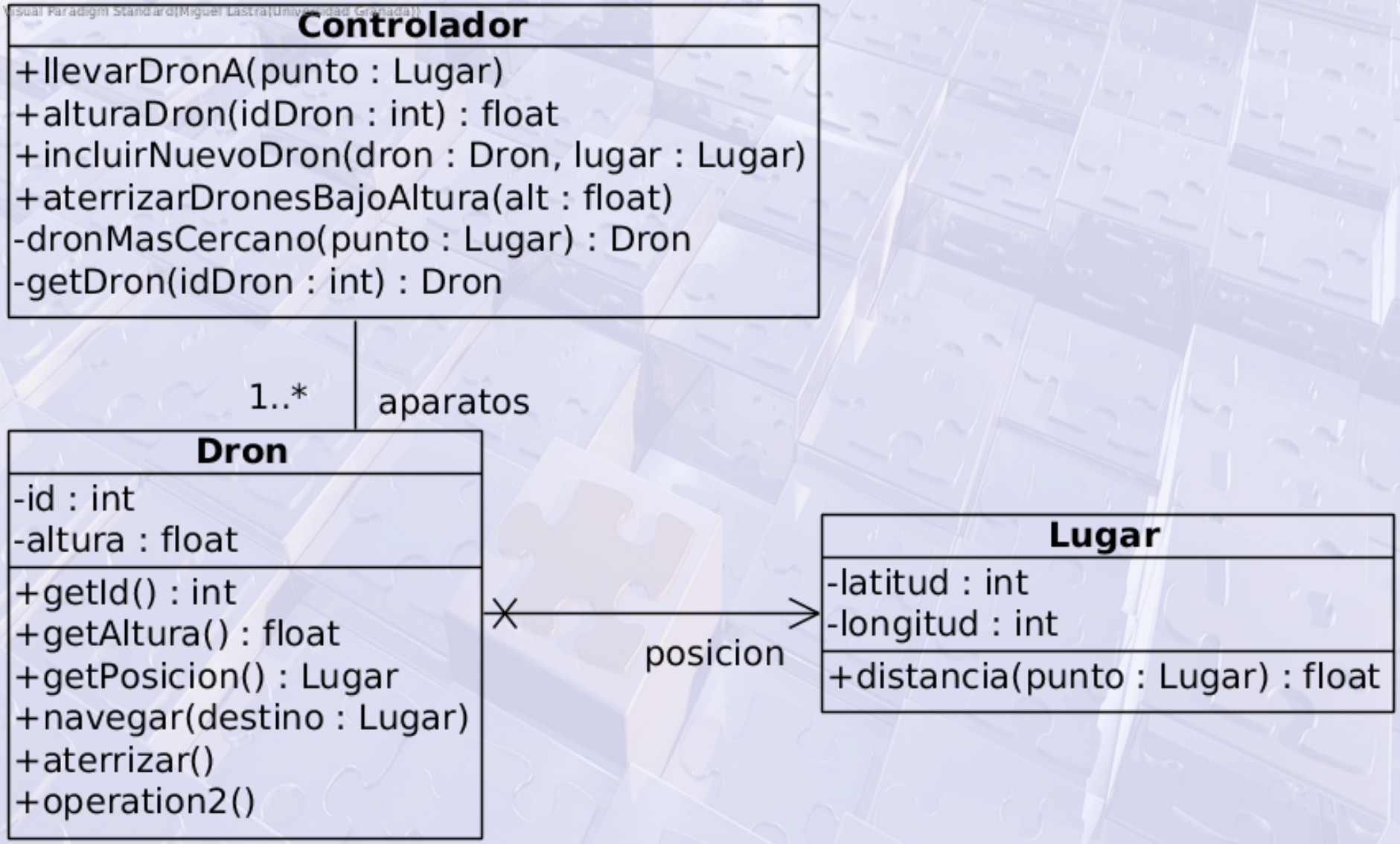
- Muestran de forma visual muy clara las vías de comunicación que deben darse entre los participantes para que pueda llevarse a cabo el envío de mensajes entre ellos
- Las vías de comunicación (enlaces) son el elemento principal y el orden temporal de los mensajes un elemento secundario

Diagramas de comunicación

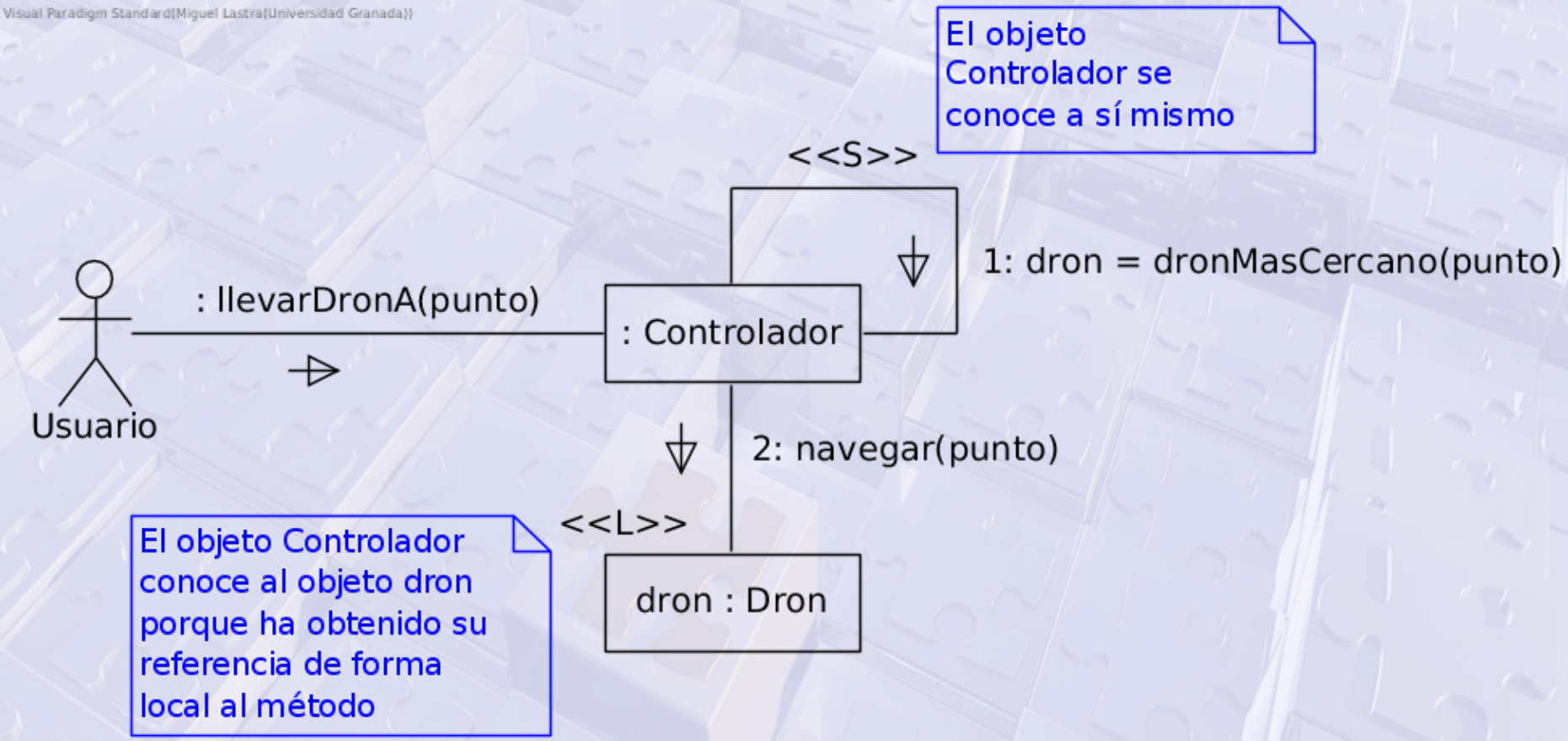
- Las vías de comunicación se representan mediante líneas que unen a los participantes

- Tipos de enlaces:
  - Global (G): Uno de los participantes pertenece a un ámbito superior. Ej: un atributo de clase
  - Asociación (A): Entre los participantes existe una asociación
  - Parámetro (P): Uno de los objetos es pasado como parámetro a un método del otro participante
  - Local (L): Uno de los participantes es un objeto local a un método del otro participante
  - Self (S): Un objeto también puede enviarse mensajes a sí mismo

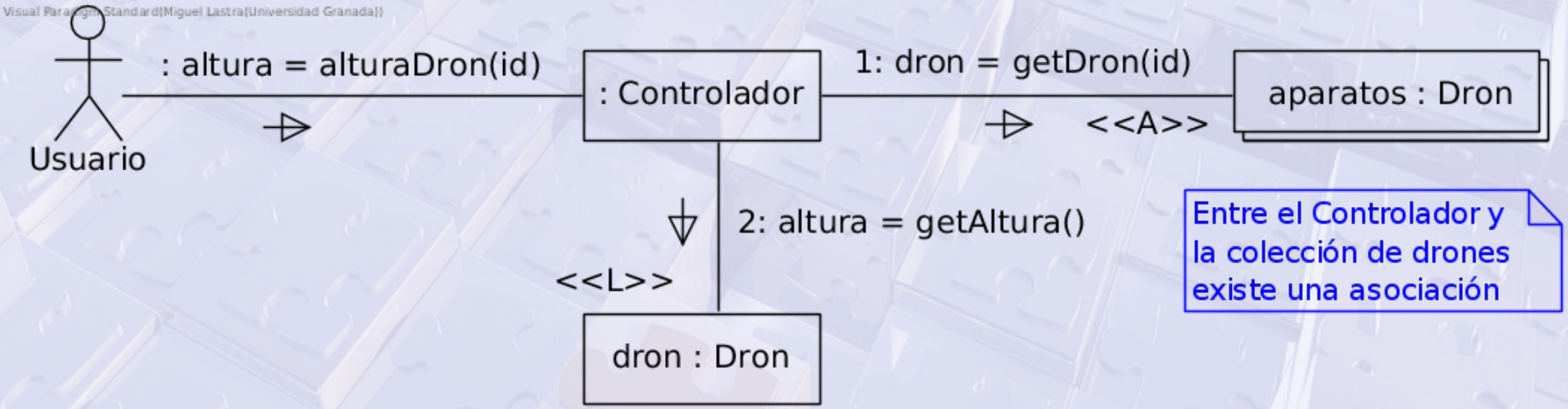
DC para los ejemplos siguientes



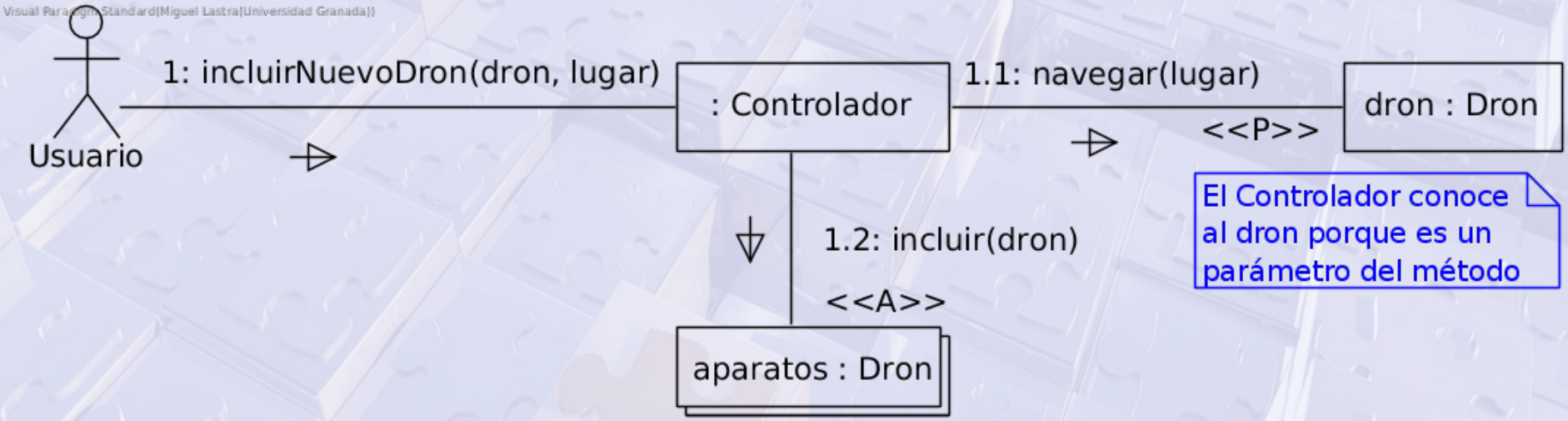
Ejemplo 1


Visual Paradigm Standard(Miguel Lastra(Universidad Granada))
El objeto Controlador se conoce a sí mismo
<<S>>
1: dron = dronMasCercano(punto)
: llevarDronA(punto)
: Controlador
Usuario
2: navegar(punto)
<<L>>
El objeto Controlador conoce al objeto dron porque ha obtenido su referencia de forma local al método
dron : Dron

Ejemplo 2

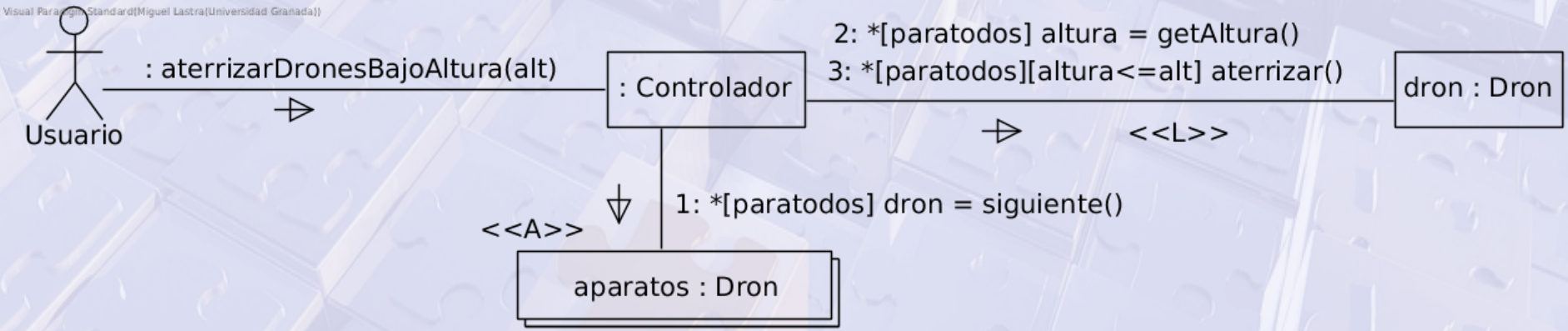

Visual Paradigm Standard(Miguel Lastra(Universidad Granada))
Usuario
: altura = alturaDron(id)
: Controlador
1: dron = getDron(id)
aparatos : Dron
<<A>>
2: altura = getAltura()
<<L>>
dron : Dron
Entre el Controlador y la colección de drones existe una asociación

Ejemplo 3


Visual Paradigm Standard(Miguel Lastra(Universidad Granada))
Usuario
1: incluirNuevoDron(dron, lugar)
: Controlador
1.1: navegar(lugar)
<<P>>
dron : Dron
1.2: incluir(dron)
<<A>>
aparatos : Dron
El Controlador conoce al dron porque es un parámetro del método

Ejemplo 4

- Condicionales y bucles



Diagramas de interacción → *Diseño* ←

- Recordar que el objetivo de los diagramas UML son:
  - Especificar las características de un sistema antes de su construcción
  - Visualizar gráficamente un sistema software de forma que sea entendible
  - Documentar un sistema para facilitar su mantenimiento, revisión y modificación
- En definitiva, facilitar la tarea del equipo de desarrollo
- Si la especificación de un método (sobre todo los de comunicación) es una maraña de flechas donde es más fácil perderse que aclararse:
  1. Tal vez ese tipo de diagrama no sea el más adecuado para esa especificación
  2. Tal vez haya que subdividir un diagrama grande en varios pequeños
  3. Tal vez el método deba subdividirse en diversas tareas más pequeñas y más fáciles de especificar de una manera clara y fácilmente entendible (supondrá un desarrollo y mantenimiento más fácil)

UML: Diagramas de Interacción

Dpto. Lenguajes y Sistemas Informáticos
Universidad de Granada

Programación y Diseño Orientado a Objetos

(Curso 2023-2024)

Temporary page!

LaTeX was unable to guess the total number of pages correctl
was some unprocessed data that should have been added to
page this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus pa
away, because LaTeX now knows how many pages to expect f
document.