

Programación Orientada a Objetos

```
var Carousel = {  
  init : function(options, el){  
    var base = this;
```

```
    base.$elem = $(el);
```

```
    // options passed via js override options passed via data attributes  
    base.options = $.extend({}, $.fn.owlCarousel.options, base.$elem.data(), options);
```

```
    base.userOptions = options;  
    base.loadContent();
```

```
  },
```

```
  loadContent : function(){  
    var base = this;
```

```
    if (typeof base.options.beforeInit === "function") {  
      base.options.beforeInit.apply(this, [base.$elem]);  
    }
```

```
    if (typeof base.options.jsonPath === "string") {  
      var url = base.options.jsonPath;
```

```
      function getData(data) {
```

```
        if (typeof base.options.jsonSuccess === "function") {  
          base.options.jsonSuccess.apply(this, [data]);
```

Diagrama de Clases

Método final

Map	
Language	Hyperlink
Body Condition	Constraint
Precondition	Post Condition
Base	Parameters
Stereotype	
Name	golpear
Return Type	void
Type Modifier	
Visibility	public
Static	false
Abstract	false
Leaf	true
Definition	
metodo que se encarga de que el personaje golpee la pelota @param game, que representa el juego POong @param pelota, representa la pelota actual del juego	

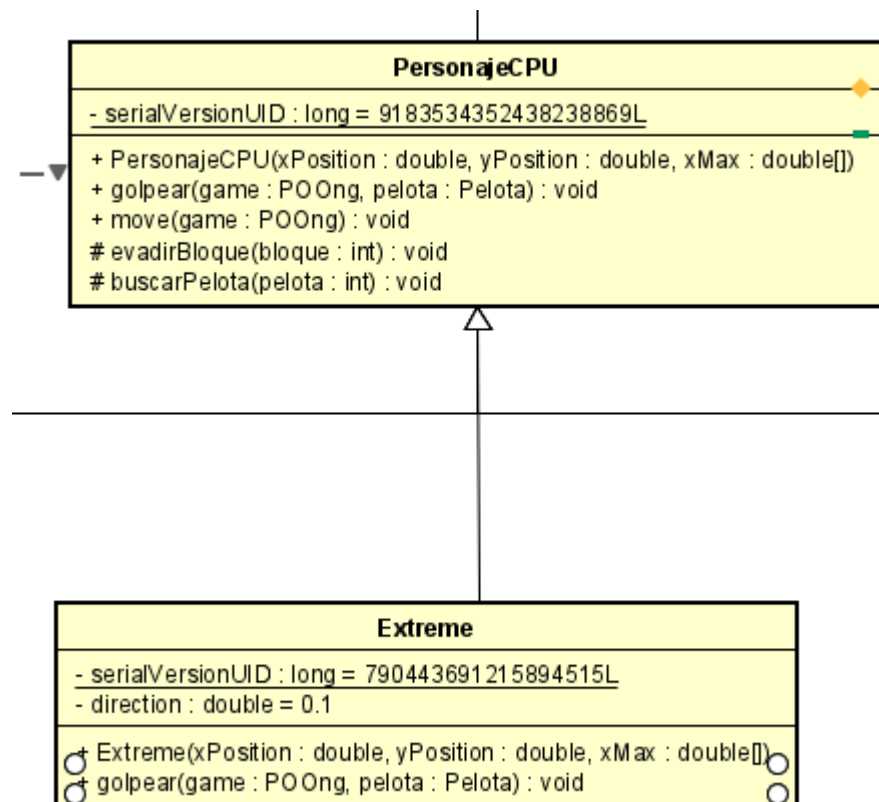


Diagrama de secuencia



- **Permiten diseñar interacciones entre objetos en un método.**
- **Visualiza la responsabilidad de cada objeto en un método.**
- **Permite visualizar complejidad del código en cuanto a acoplamiento.**
- **Muestra la comunicación entre objetos y los mensajes que envían.**
- **Muestra la secuencia (orden) en el que suceden las interacciones.**



Diagrama de secuencia

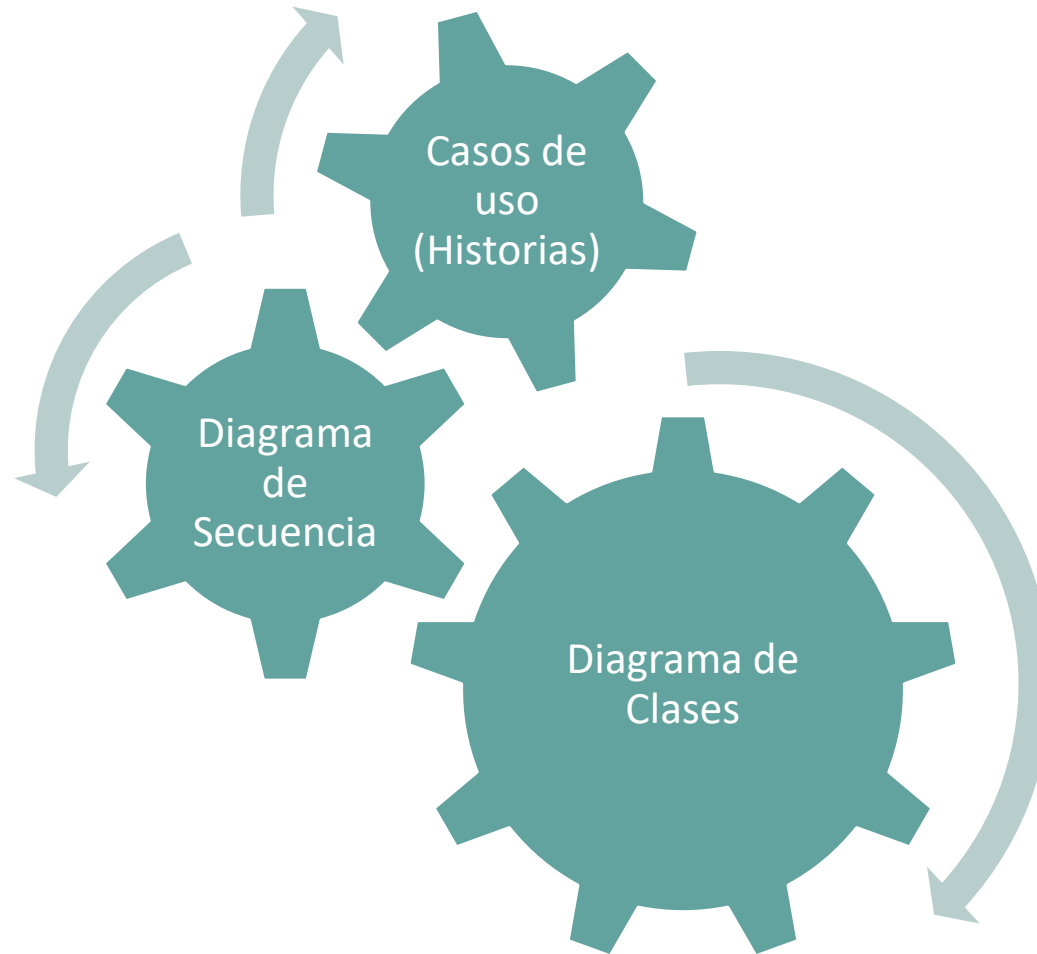


Diagrama de secuencia

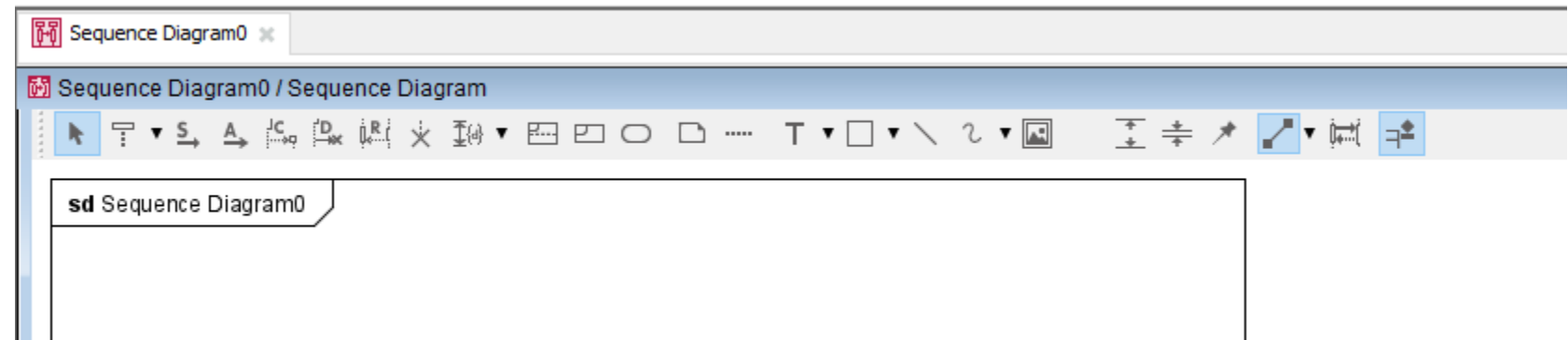
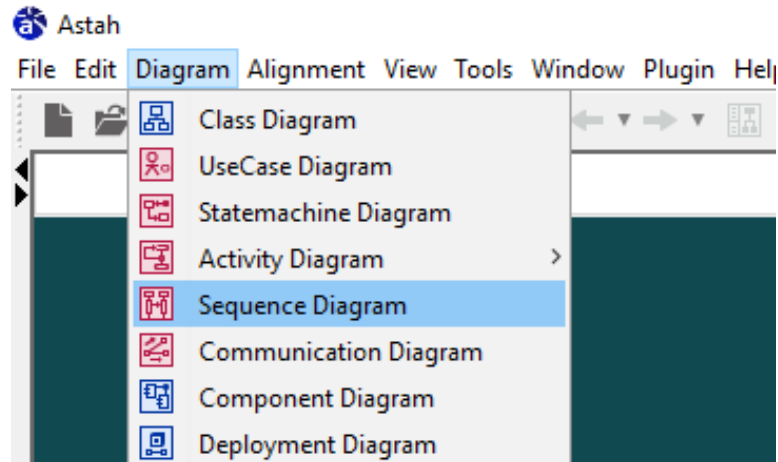
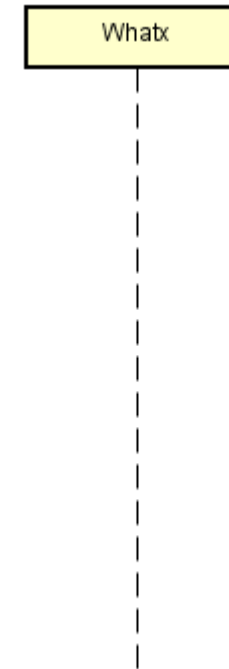
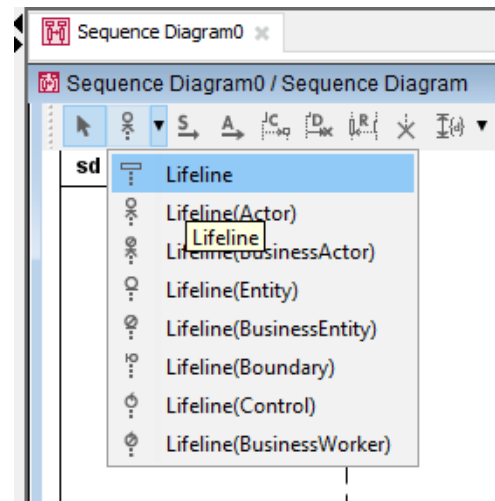
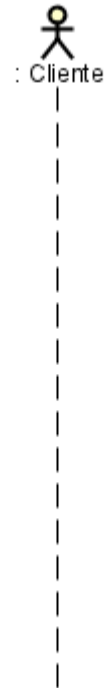
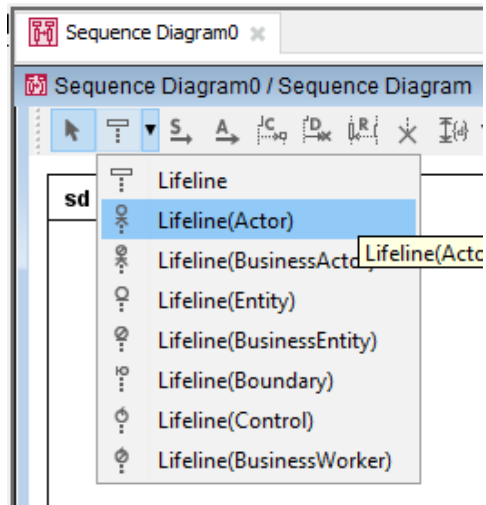
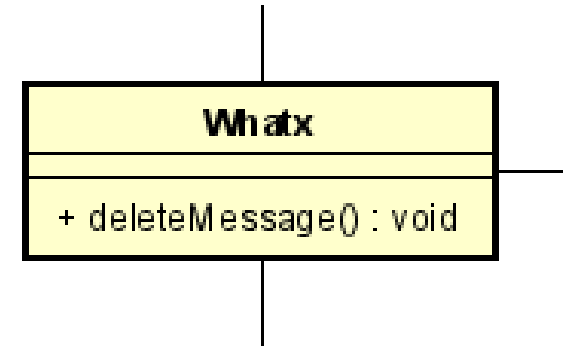
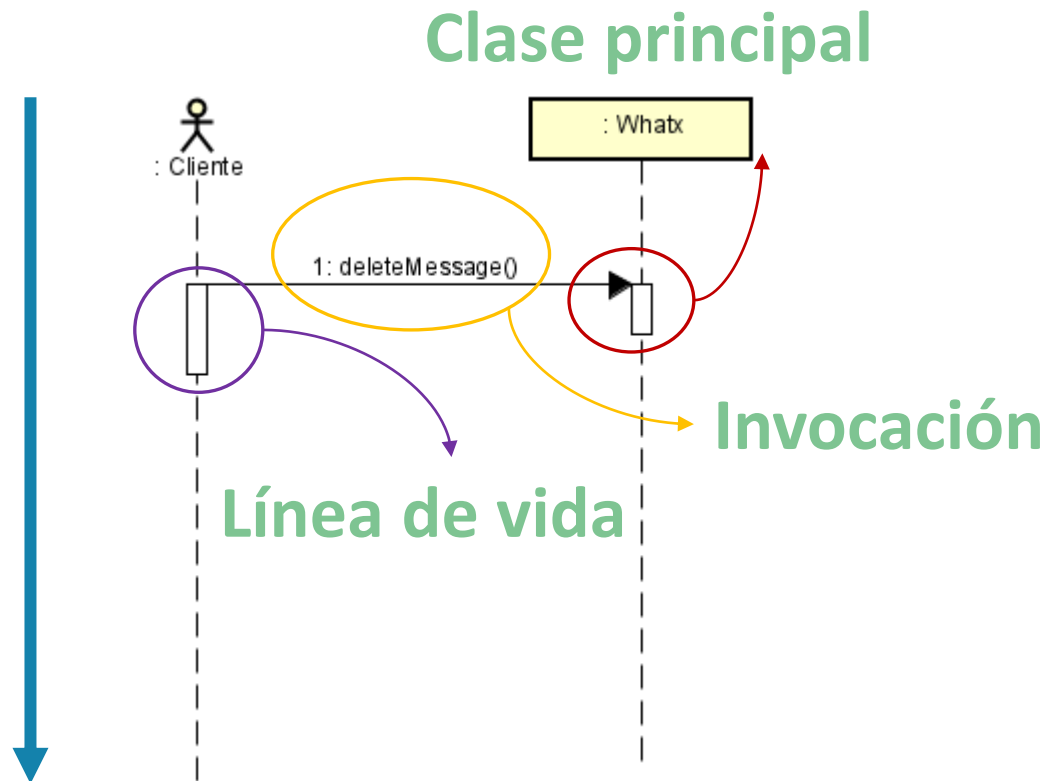


Diagrama de secuencia



Base	Hyperlink
Name	<input type="text"/>
Base Class	Whatx <input type="button" value="v"/>
<input type="button" value="Property"/> <input type="button" value="New"/>	

Diagrama de secuencia



Map

Base Stereotype Constraint Hyperlink

Name deleteMessage

Argument

Guard

Return Variable

Return Type

Operation <<Unspecified>>

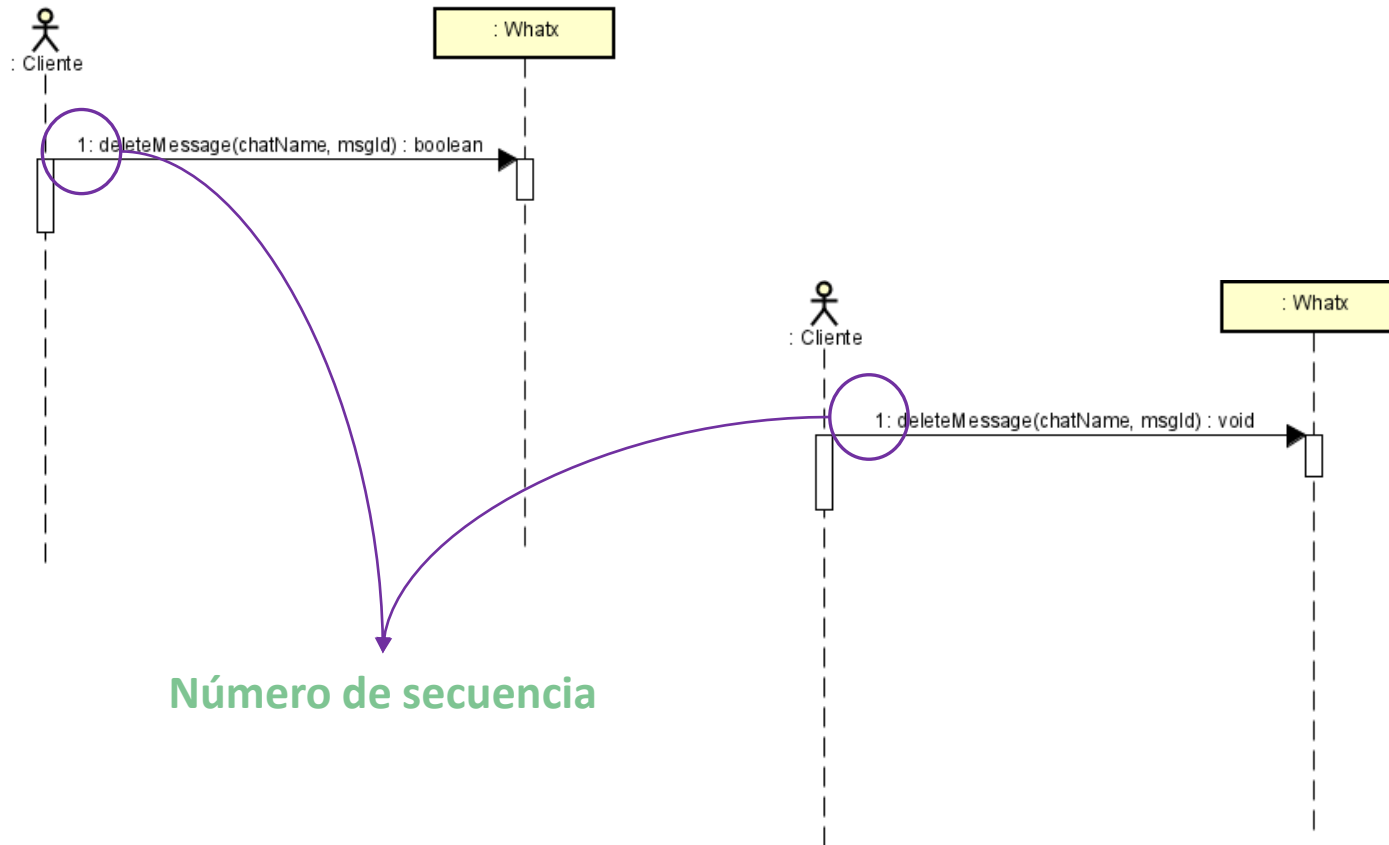
Property New

Source

Target

☐ Asynchronous

Diagrama de secuencia



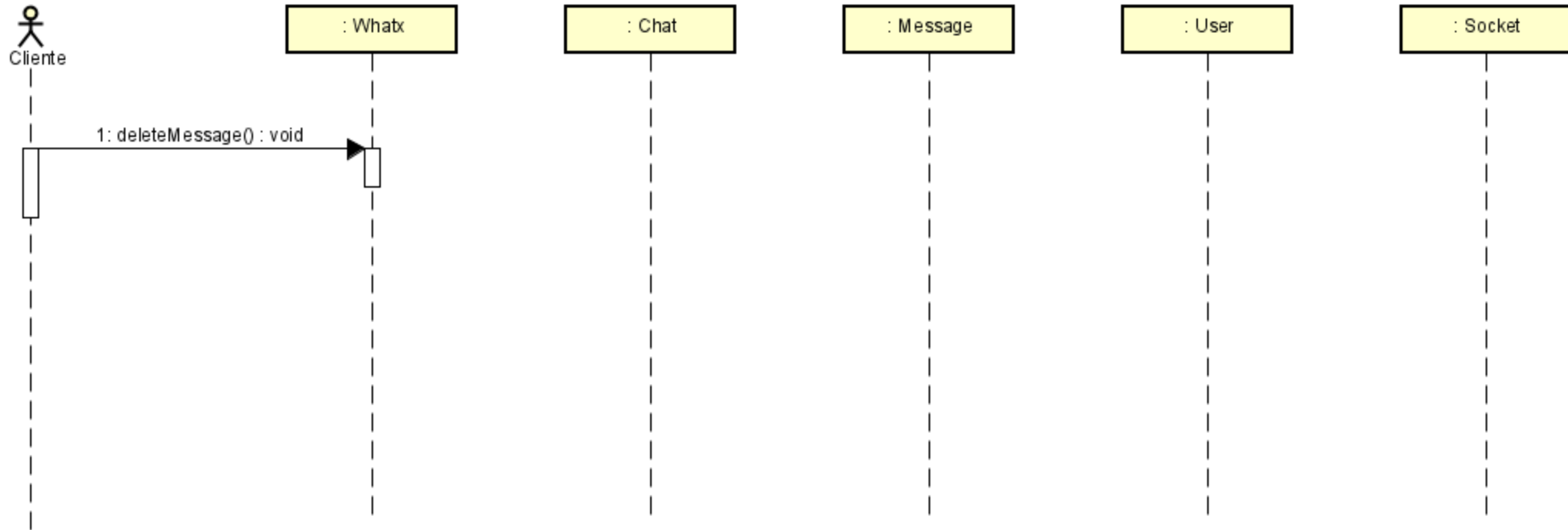
Número de secuencia

Identificar parámetros y retorno

Base	Stereotype	Constraint	Hyperlink
Name	deleteMessage		
Argument	chatName, msgId		
Guard			
Return Variable			
Return Type	boolean		
Operation	deleteMessage():void - ...		
		Property	New
Source			

Base	Stereotype	Constraint	Hyperlink
Name	deleteMessage		
Argument	chatName, msgId		
Guard			
Return Variable			
Return Type			
Operation	deleteMessage():void - ...		
		Property	New
Source			

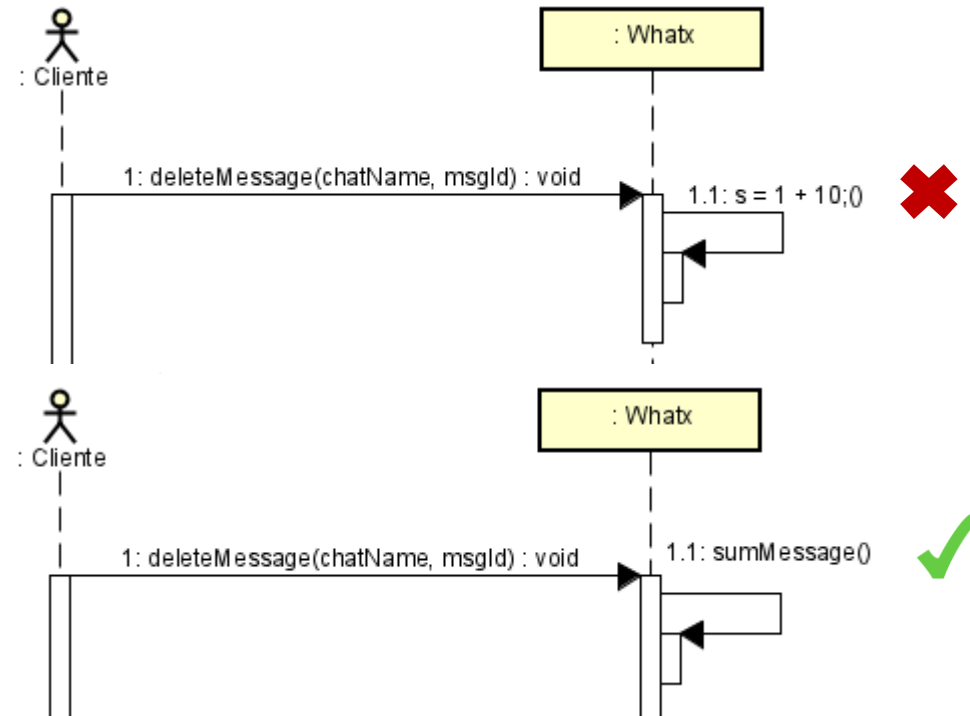
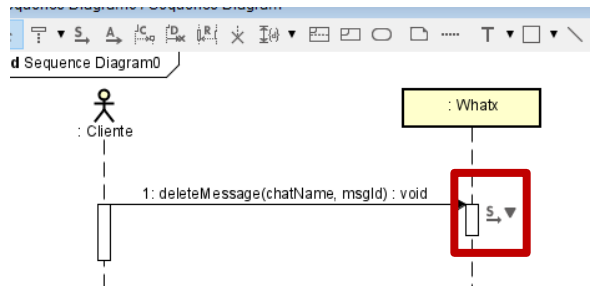
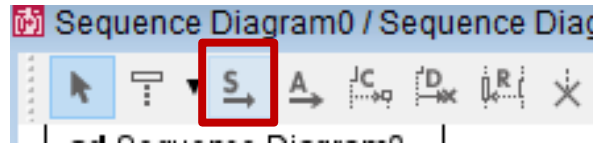
Diagrama de secuencia



Identificar los objetos (u otros sistemas) que interactúan

Si no nos dicen, diseñar hasta el nivel 3 (máximo). Ej: 1.1.1

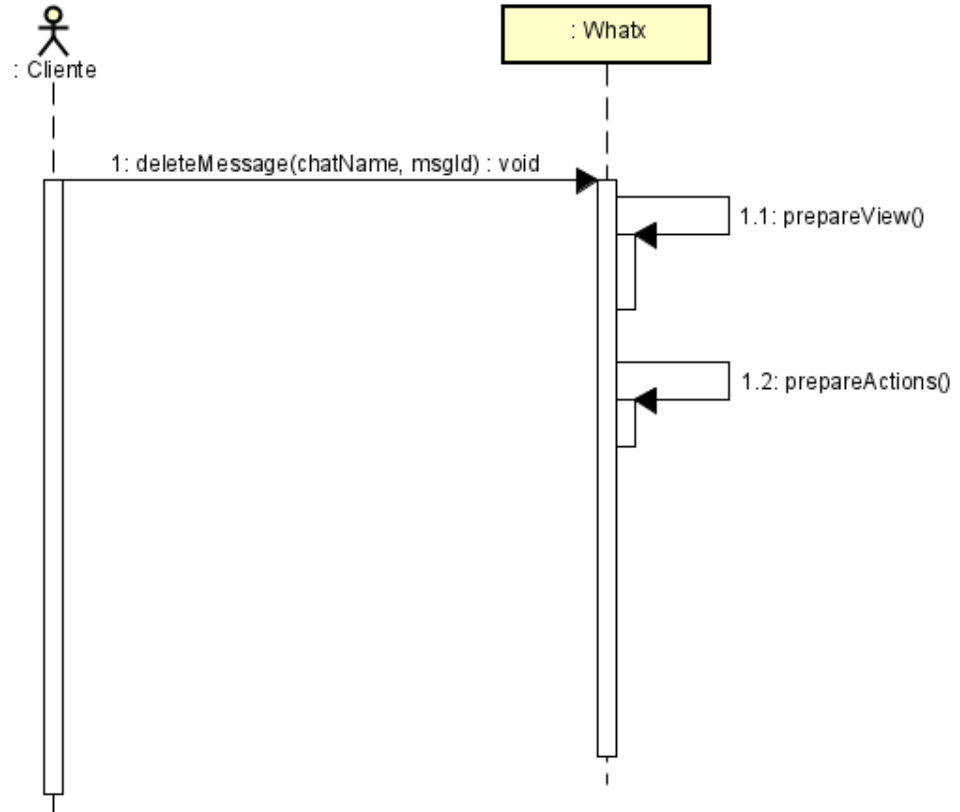
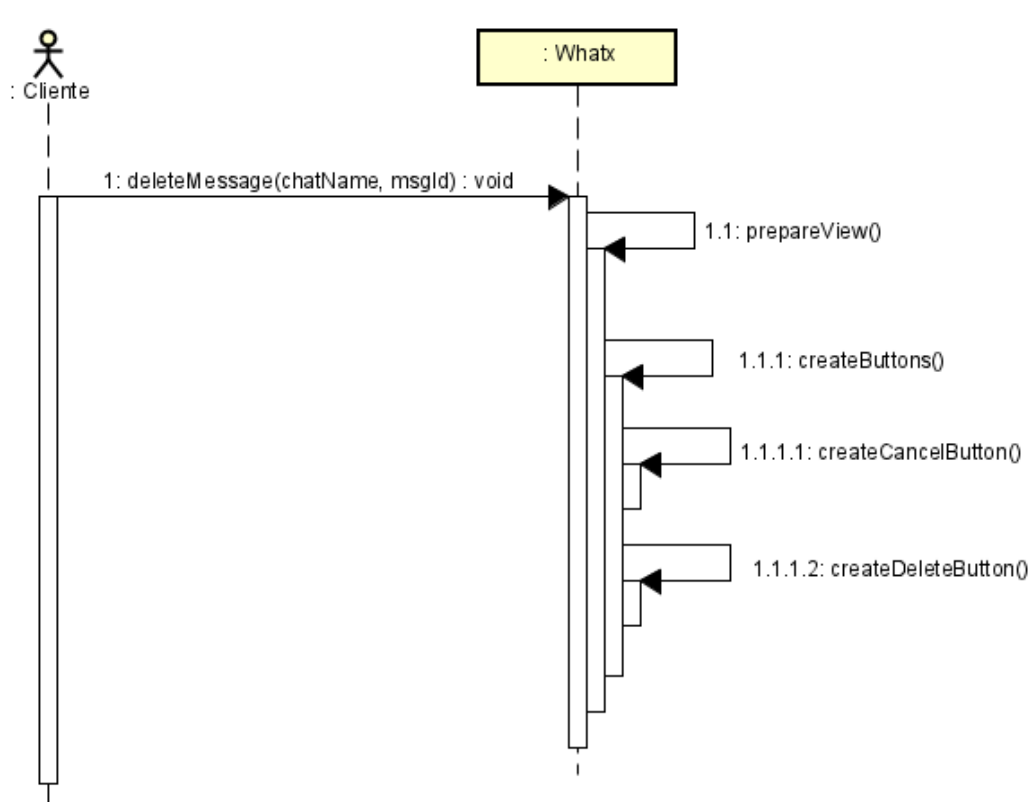
Diagrama de secuencia



Identificar las invocaciones requeridas

No se diseñan líneas de código, si INVOCACIONES

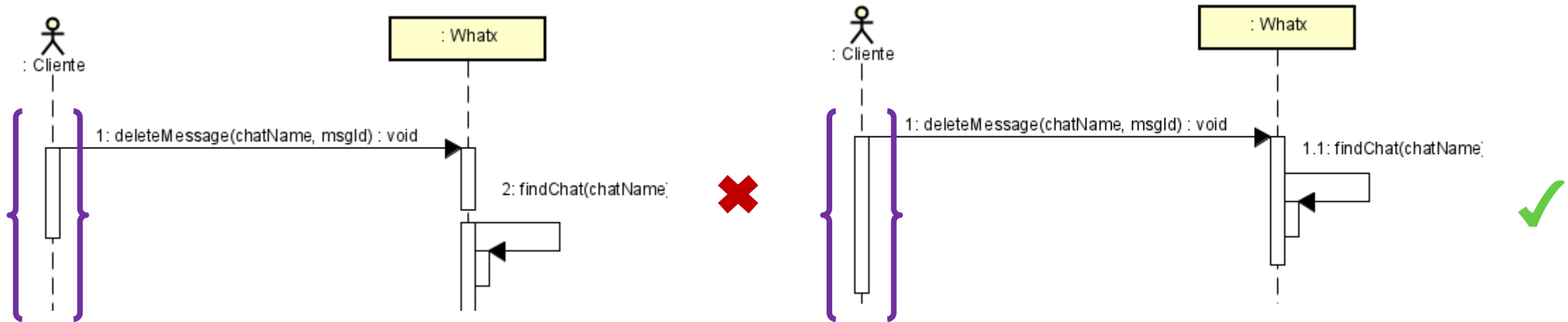
Diagrama de secuencia



Identificar las invocaciones requeridas

Adicionar los métodos que encapsulen comportamientos clave para el método principal (1) o máximo 3 niveles de especificación

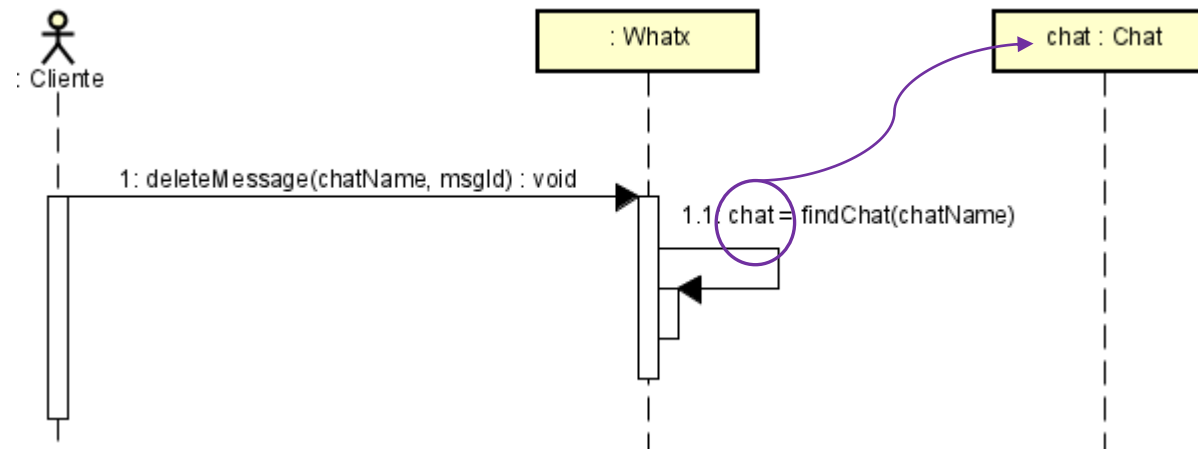
Diagrama de secuencia



Identificar las invocaciones requeridas

Mantenga la secuencia 1, 1.1, 1.1.1 no 1,2,3

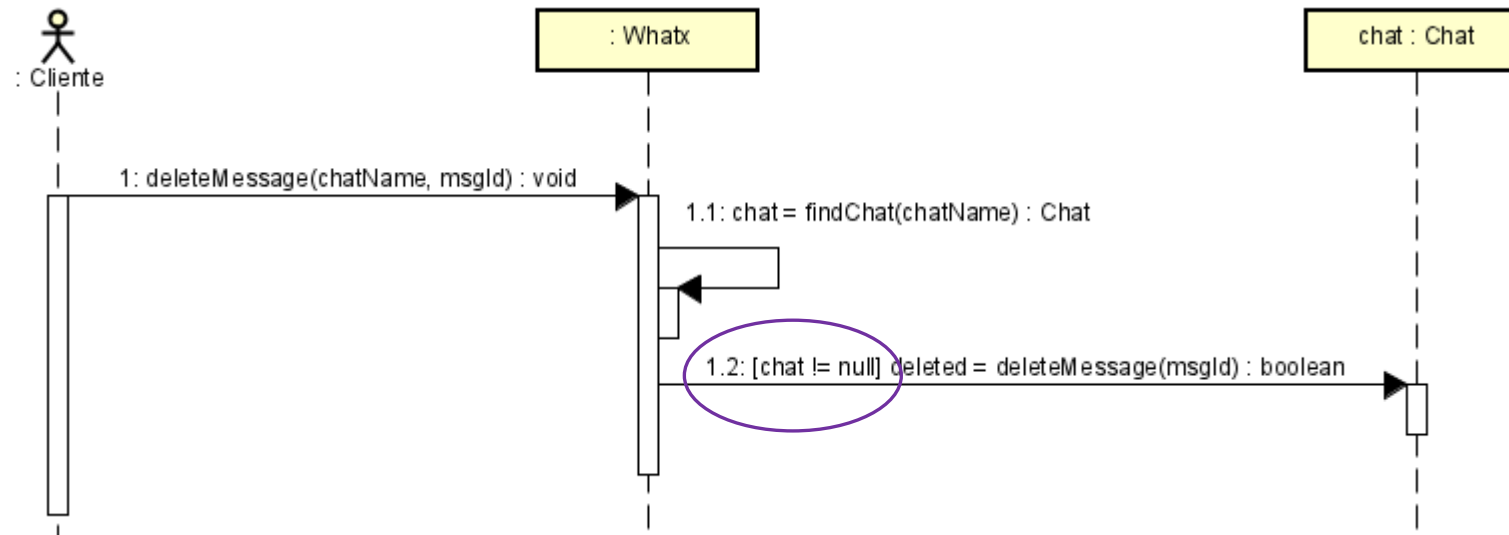
Diagrama de secuencia



Base	Stereotype	Constraint	Hyperlink
Name	findChat		
Argument	chatName		
Guard			
Return Variable	chat		
Return Type			
Operation	<<Unspecified>>		
<input type="button" value="Property"/> <input type="button" value="New"/>			

Variables de retorno
Definir solo si se utilizan después

Diagrama de secuencia

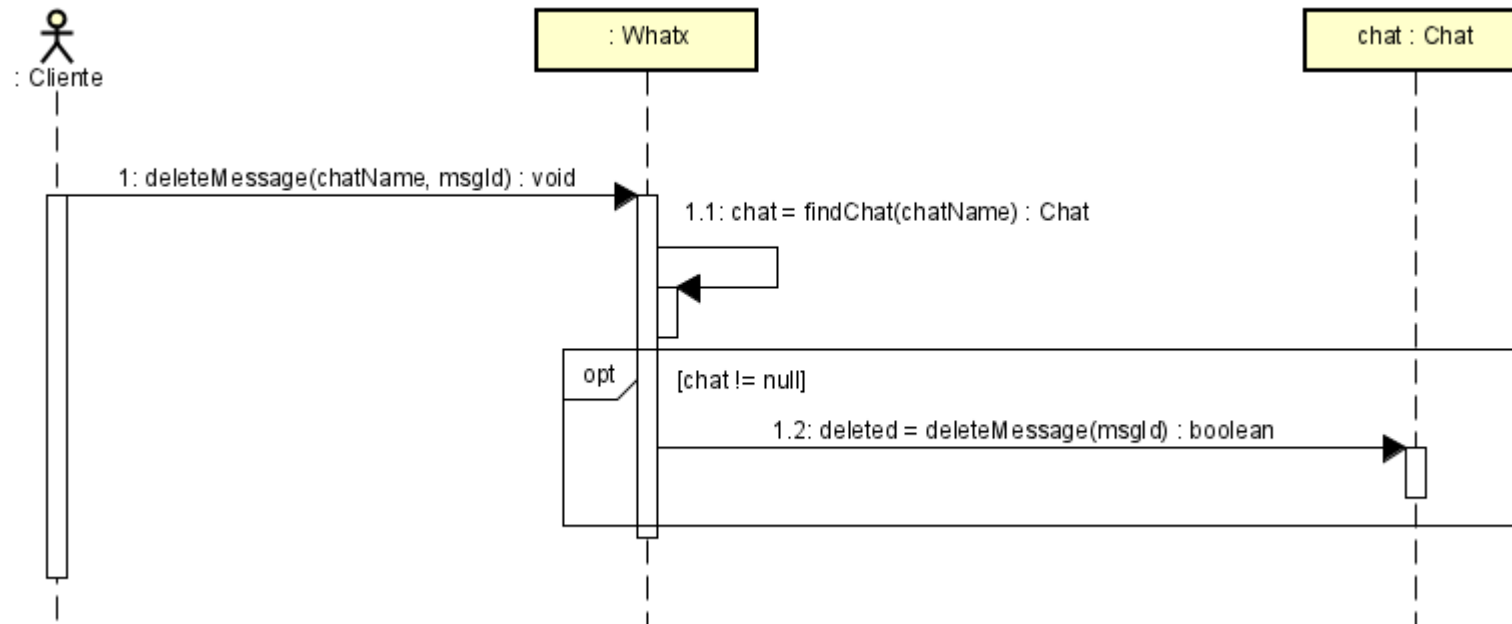
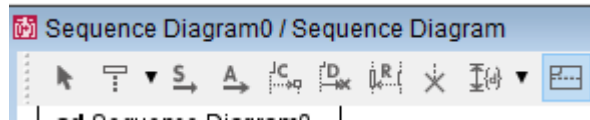


Base	Stereotype	Constraint	Hyperlink
Name	deleteMessage		
Argument	msgId		
Guard	chat != null		
Return Variable	deleted		
Return Type	boolean		

Condicionales: Opción 1

[IF] Si solo una invocación se afecta por el condicional

Diagrama de secuencia



Base	Stereotype	Constraint	Hyperlink
Name	deleteMessage		
Argument	msgId		
Guard			
Return Variable	deleted		
Return Type	boolean		
Operation	<<Unspecified>>		

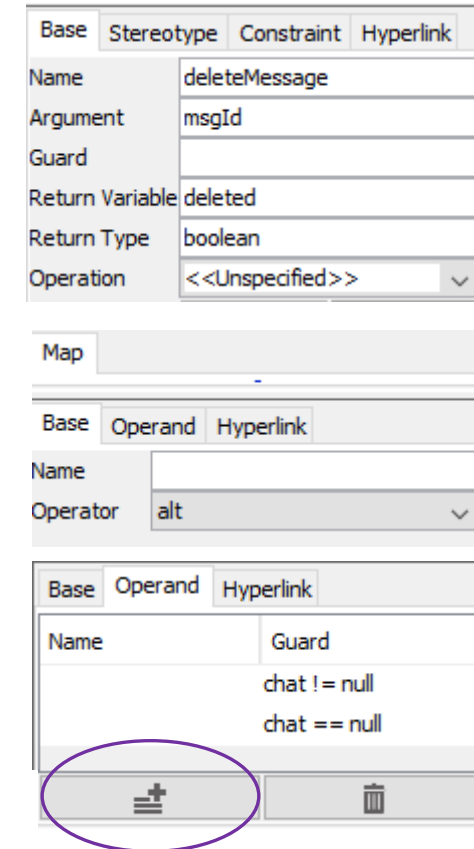
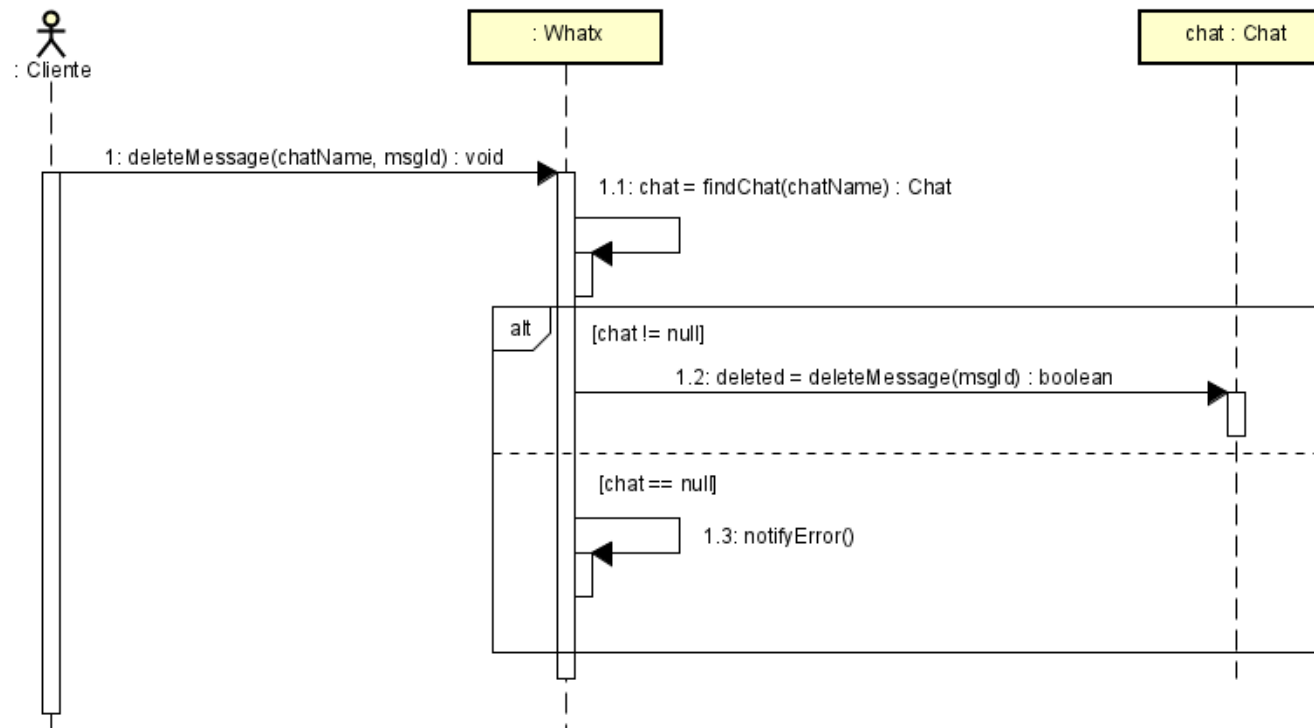
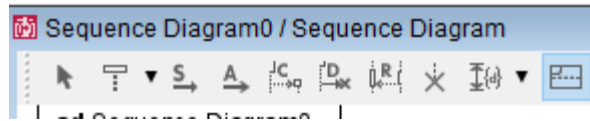
Base	Operand	Hyperlink
Name		
Operator	opt	

Base	Operand	Hyperlink
Name	Guard	
	chat != null	

Condicionales: Opción 2

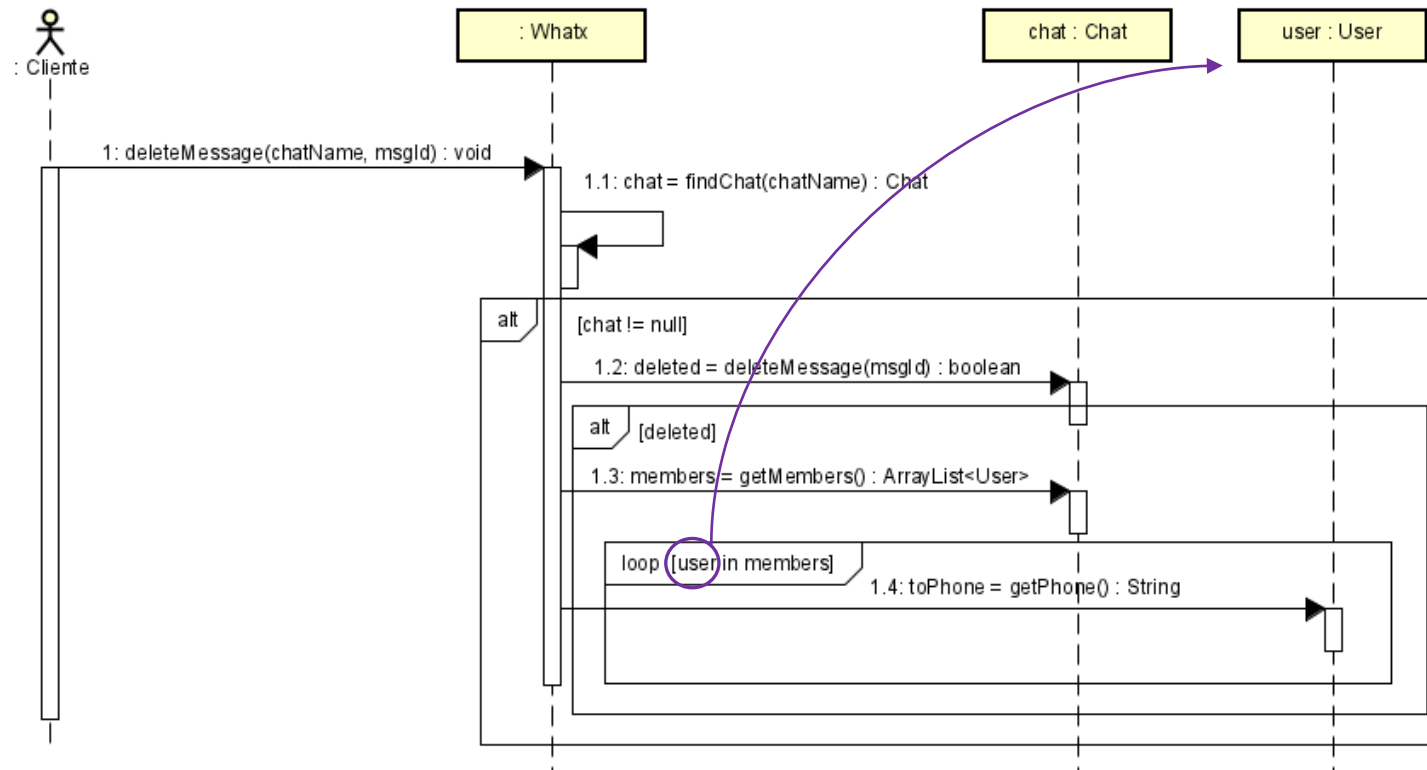
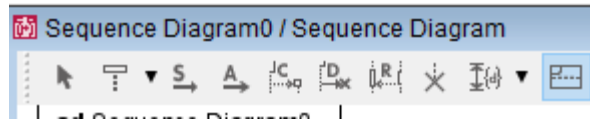
[IF] El recuadro debe abarcar todas las invocaciones que están dentro del condicional

Diagrama de secuencia



Condicionales: Opción 3
[IF / ELSE / SWITCH]

Diagrama de secuencia

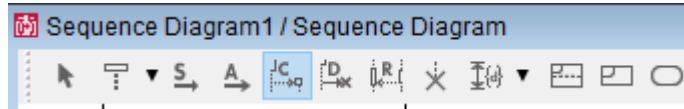


Base	Operand	Hyperlink
Name	<input type="text"/>	
Operator	loop	

Base	Operand	Hyperlink
Name	Guard	
user in members		

Ciclos
[FOR / WHILE / DO WHILE]

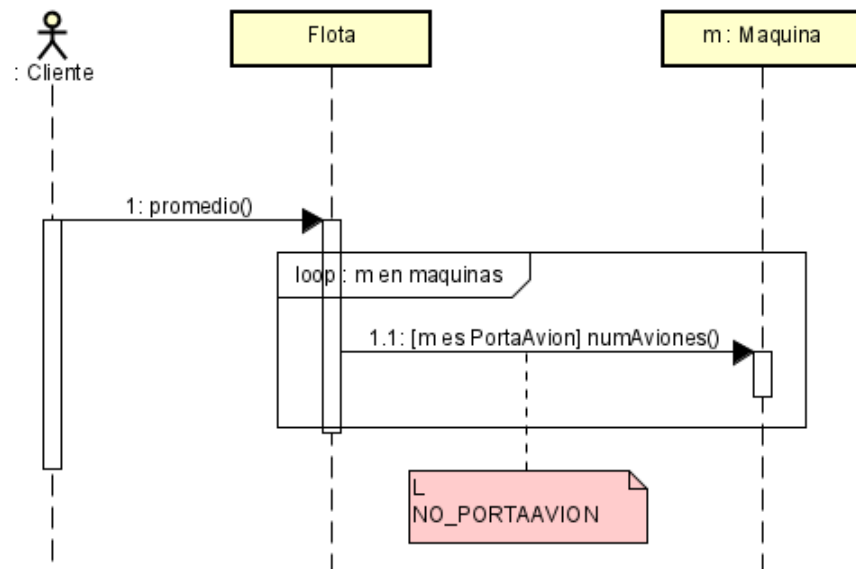
Diagrama de secuencia



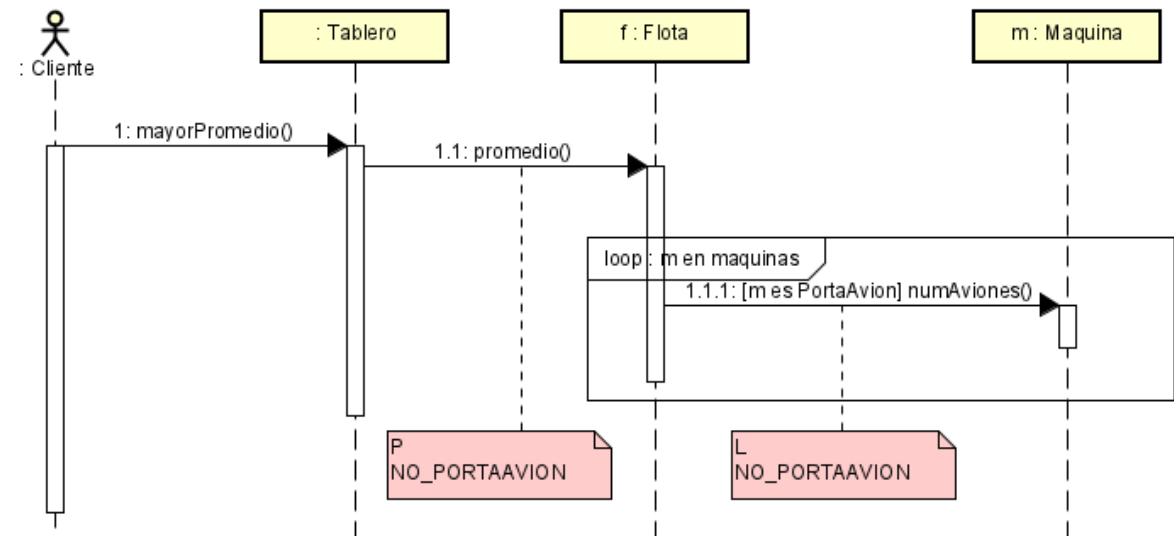
Base	Stereotype	Constraint	Hyperlink
Name	Chat		
Argument	chatName		
Guard			
Return Variable			
Return Type			
Operation	<<Unspecified>>		
Property		New	

Crear Objeto
[new]

Diagrama de secuencia

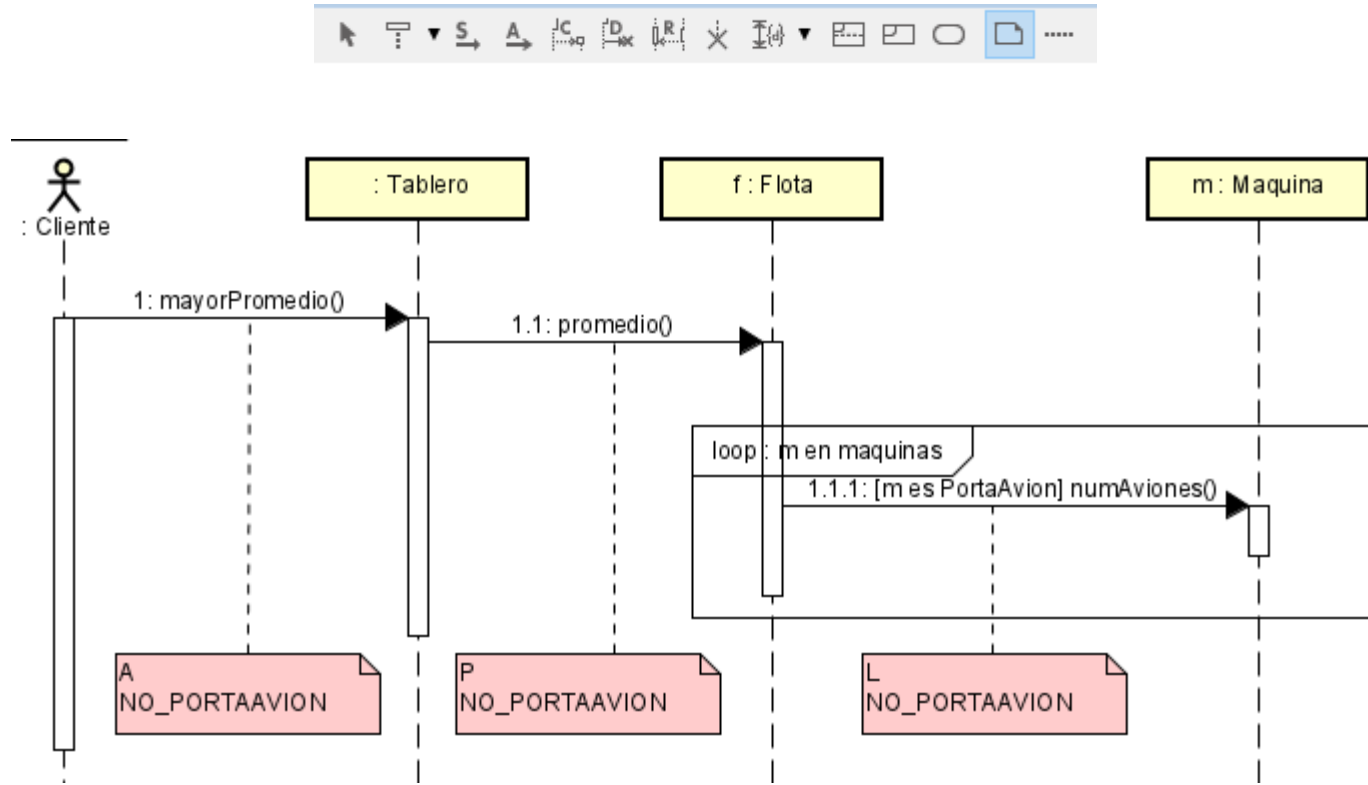


Lanzar excepciones
[new throw]



Propagar excepciones
[throws]

Diagrama de secuencia



Atender excepciones
[try / catch]